

Серия «Библиотека студента»

В. П. Дьяконов

Mathcad 8—12 для студентов

**Москва
СОЛОН-Пресс
2005**

УДК 621.396.218

ББК 32.884.1

Д93

В. П. Дьяконов

Д93 **Mathcad 8—12 для студентов.**

Серия «Библиотека студента». — М.: СОЛОН-Пресс, 2005. — 632 с.: ил.

ISBN 5-98003-212-6

Содержит полное описание самых популярных версий системы Mathcad (от Mathcad 8 до новейшей Mathcad 12), ориентированное на студентов и преподавателей вузов, а также инженеров и научных работников. Особое внимание уделено наглядности описания, визуализации результатов вычислений и примерам применения систем в практике математических, научно-технических и учебных расчетов. В книге описаны многие сотни примеров применения систем класса Mathcad. Для всех пользователей системами Mathcad. Может служить описанием систем, руководством пользователя по ним и самоучителем.

КНИГА — ПОЧТОЙ

Книги издательства «СОЛОН-Пресс» можно заказать наложенным платежом (оплата при получении) по фиксированной цене. Заказ оформляется одним из двух способов:

1. Послать открытку или письмо по адресу: 123242, Москва, а/я 20.

2. Передать заказ по электронной почте на адрес: magazin@solon-r.ru.

Бесплатно высылается каталог издательства по почте.

При оформлении заказа следует правильно и полностью указать адрес, по которому должны быть высланы книги, а также фамилию, имя и отчество получателя. Желательно указать дополнительно свой телефон и адрес электронной почты.

Через Интернет вы можете в любое время получить свежий каталог издательства «СОЛОН-Пресс». Для этого надо послать пустое письмо на робот-автоответчик по адресу: katalog@solon-r.ru.

Получать информацию о новых книгах нашего издательства вы сможете, подписавшись на рассылку новостей по электронной почте. Для этого пошлите письмо по адресу: news@solon-r.ru. В теле письма должно быть написано слово SUBSCRIBE.

По вопросам приобретения обращаться:

ООО «Альянс-книга»

Тел: (095) 258-91-94, 258-91-95, www.abook.ru

ISBN 5-98003-212-6

© Макет и обложка «СОЛОН-Пресс», 2005

© В. П. Дьяконов, 2005

Введение

Система компьютерной математики [1] Mathcad бурно развивается. Недавно появилась новейшая версия системы Mathcad 12. Но в пользовании находится и ряд предшествующих версий этой популярной системы — от Mathcad 8 до Mathcad 11. Нередки случаи, когда пользователь — студент, имеющий дешевый и простенький компьютер, вынужден работать с Mathcad 8, тогда как в учебных классах его университета установлены новейшие Mathcad 11/12. Бывает, впрочем, и обратная ситуация.

Некоторые авторы книг по системам Mathcad наперегонки стараются выпускать книги по новейшим версиям системы Mathcad. Порою не успевает появиться самоучитель или учебный курс по одной версии Mathcad, как тут же издается то же самое по очередной версии. Увы, такова практика издания книг по программным продуктам как у нас, так и за рубежом. Хуже, однако, что некоторые авторы, издав скороспелую книгу по новейшей версии Mathcad, забывают в ней указать, чем же эта книга принципиально отличается от книг других авторов, а то и от ранних книг самого автора. Все чаще для этого из подобных книг просто исключается список литературы. Надо ли говорить, что для учебной и научной литературы такая практика не только не этична, но и мешает читателям оценить место и достоинства той или иной книги. Странно, что подобные книги выпускают некоторые крупные издательства.

В этой связи скажу прямо, что данная книга не претендует на уникальность. Она целиком построена на материале ранее изданных книг автора [2—11] и фактически является сокращением и переработкой его последней крупной энциклопедии по системе Mathcad 2001i/11 [11], выпущенной издательством «СОЛОН-Пресс» в серии «Библиотека профессионала». Эта переработка свелась к следующему:

- в книге впервые описаны все версии систем Mathcad нового поколения — от Mathcad 8 до Mathcad 12;
- в книгу включены подробные данные по основным различиям между разными версиями систем Mathcad;
- книга ориентирована прежде всего на студентов и преподавателей вузов и университетов, поэтому в ней имеется множество практических примеров применения системы Mathcad, в том числе учебных;
- из книги удален неактуальный и малодоступный для большинства читателей материал по пакетам расширения систем Mathcad и электронным книгам;
- книга дополнена новыми материалами по новейшим реализациям Mathcad 11 и, особенно, Mathcad 12;
- в примерах использованы только встроенные в ядро Mathcad функции;
- подавляющее большинство примеров может использоваться в любой версии Mathcad, иное отдельно оговаривается;
- ввиду учебного характера книги файлы примеров не даются, но их можно найти на CD-ROM, прилагаемом к энциклопедии [11].

Все это позволило уменьшить объем и стоимость книги, что немаловажно для студентов и преподавателей университетов и вузов, и выпустить книгу в серии «Для студентов». Читателям, купившим энциклопедию [11], покупать эту книгу не

стоит — проще разобраться с небольшими отличиями Mathcad 12 от Mathcad 2001i/11 и учесть их в своей работе.

Адреса

Отзывы и пожелания о книге можно отправлять по адресу издательства «СОЛОН-Пресс» или по адресу электронной почты автора vpdyak@keytown.com.

С фирмой MathSoft, Inc. вы можете познакомиться, посетив ее сайт www.mathsoft.com. С представителем фирмы в России — корпорацией SoftLine и ее шикарным Интернет-магазином — также можно ознакомиться на ее страничке в Интернете www.softline.ru. Полезно также знакомство с сайтом www.exponenta.ru, на котором имеется много материалов по применению современных систем компьютерной математики, включая системы класса Mathcad, в образовании.

Принятые в книге обозначения

Для облегчения чтения книги в ней используются следующие обозначения.

- Имена элементов интерфейса выделены специальным шрифтом, например: «Перечень шаблонов имеется в подменю Graph (график) меню Insert (вставка)».
- Термины, встречающиеся в тексте в первый раз, как правило, выделены курсивом, например: «*Строка меню* — это строка, открывающая доступ к пунктам меню с различными командами».
- Моноширинным шрифтом обозначены векторы, матрицы, функции, операторы, программные блоки и вообще все, что можно увидеть в окне редактора системы Mathcad. Например: «Для создания ранжированной переменной используется выражение: `Name := Nbegin..Nend`, где Name — имя переменной, Nbegin — ее начальное значение, Nend — конечное значение, .. — символ, указывающий на изменение переменной в заданных пределах».

В тексте вам также будут встречаться специальные врезки, в которых содержатся примечания, советы и предупреждения.

ПРИМЕЧАНИЕ *В примечаниях даются сведения, позволяющие лучше понять тему раздела.*

СОВЕТ *Советы предлагают альтернативный и иногда более эффективный путь решения типичных задач.*

ВНИМАНИЕ *Под этим заголовком идут предупреждения. В них содержится важная информация, обращающая ваше внимание на точное выполнение тех или иных пунктов инструкций с тем, чтобы не нанести вред вашей системе или не потерять данные.*

Глава 1. Первое знакомство с Mathcad

Цель этой главы — дать начальные сведения о системе Mathcad, в частности о ее версиях от Mathcad 8 до Mathcad 12. Начинающему пользователю полезно получить из этой главы общее представление о системах класса Mathcad. А опытного пользователя, возможно имеющего свои наборы документов, созданные в прежних версиях системы, не могут не заинтересовать различия между различными версиями и новые возможности последних версий. Детальные данные об особенностях реализации разных версий системы Mathcad имеются в Приложении 1 справочного раздела книги.

1.1. Общие сведения о системе Mathcad

1.1.1. История создания и место системы Mathcad

Mathcad — это популярная система компьютерной математики [1—23], предназначенная для автоматизации решения массовых математических задач в самых различных областях науки, техники и образования. Название системы происходит от двух слов — MATHematica (математика) и CAD (Computer Aided Design — системы автоматического проектирования, или САПР). Так что вполне правомерно считать Mathcad математической САПР.

Первые версии системы Mathcad 1.0/2.0 работали в среде MS-DOS и предназначались только для численных расчетов [2]. При этом, впервые в разработке систем компьютерной математики, был использован визуально-ориентированный подход к подготовке документов с применением для задания математических формул шаблонов. Начиная с версии Mathcad 3.0 [2] система ориентировалась на операционные системы класса Windows и приобрела средства для выполнения символьных (аналитических) операций.

Сегодня различные версии Mathcad [2—23] являются математически ориентированными универсальными вычислительными системами. Помимо собственно вычислений, как численных, так и аналитических, они позволяют с блеском решать сложные оформительские задачи, которые с трудом даются популярным текстовым редакторам или электронным таблицам. С помощью Mathcad можно, например, готовить статьи, книги, диссертации, научные отчеты, дипломные и курсовые проекты не только с качественными текстами разного стиля, но и с легко осуществляемым набором самых сложных математических формул, изысканным графическим представлением результатов вычислений и многочисленными «живыми» примерами. А применение библиотек и пакетов расширения обеспечивает профессиональную ориентацию Mathcad на любую область науки, техники и образования.

Начиная с версии Mathcad 8.0 [6, 7, 14] система обрела основную массу встроенных в ее ядро функций:

- свыше 80 математических функций;
- 10 функций преобразования;
- 110 статистических функций;
- 18 функций для решения дифференциальных уравнений;

- функции поиска корней уравнений;
- 47 функций для работы с векторами и матрицами;
- 28 функций для работы с файлами;
- 14 функций для работы с выражениями и строками.

Mathcad 2000 [7, 8] обогатил систему 18 новыми функциями для финансово-экономических вычислений. В последующем рост числа функций, к сожалению, приостановился — каждая очередная версия прибавляла от силы несколько функций — особенно преуспела в этом версия Mathcad 11, в ней появились комплексные функции Бесселя и Ханкеля, *sin*-функция и несколько функций для обработки данных. Mathcad 12 добавила к этому функции вычислений 1D- и 2D-корреляции и генерации логарифмически распределенных точек. Была восстановлена функция *until*, бывшая в Mathcad 8, но затем исчезнувшая.

Среди других систем компьютерной математики, таких как Maple [24], Mathematica [25] и MATLAB [26, 27], система Mathcad по-прежнему выделяется своим дружелюбным по отношению к пользователю интерфейсом, а также удобным и чрезвычайно простым в применении математически и визуально ориентированным языком общения с пользователем. Именно это объясняет популярность системы. Число ее только легальных пользователей в мире превысило два миллиона.

К важным достоинствам новых версий Mathcad относятся возможность настройки под любой мало-мальски известный тип печатающих устройств, богатый набор шрифтов, возможность использования всех инструментов Windows, прекрасная графика и современный многооконный интерфейс. В новые версии Mathcad включены самые эффективные средства оформления документов в цвете с гипертекстовыми и гипермедиассылками, открытыми и закрытыми областями. Имеется возможность создания самых разнообразных двумерных, трехмерных и даже анимационных (движущихся) графиков и звукового сопровождения. Будучи интегрированными программными системами, Mathcad объединяют в себе текстовый, формульный и графический редакторы и два ядра (самой системы и ядро символьных операций системы Maple, приобретенной по лицензии у фирмы Waterloo Maple).

Предусмотрена и возможность объединения с другими математическими и графическими системами для решения особо сложных задач. Отсюда и название таких систем — *интегрированные* системы. Впрочем, в решении задач интеграции создатели Mathcad пошли намного дальше — эта система обеспечивает подлинную интеграцию с целым рядом других математических, графических и офисных систем.

Между версиями Mathcad 8 — Mathcad 12 есть определенные, порою достаточно серьезные, отличия. Они представлены в Приложении 1. Это приложение позволяет пользователю в сомнительных случаях проверить, есть ли в его текущей версии Mathcad необходимые возможности и функции. Однако материал данной книги построен так, что для подавляющего большинства примеров на эти отличия можно не обращать внимания. Там, где это не так, отличия отмечаются отдельно.

Особенности новейшей версии Mathcad 12 сводятся к следующему:

- повышена скорость операций — для больших матриц до трех раз;
- введены новые форматы файлов XML (без компрессии и с ней);
- введен новый тип данных NaN (не физические значения) и функция проверки этих данных;

- улучшена работа с размерными переменными и несколько расширен список размерностей;
- введены новые нормированные функции Эйри;
- введены новые функции одномерного и двумерного корреляционного анализа;
- введены функции `logspace` и `logpts` для построения логарифмических графиков;
- введена новая функция `time` для хронометрирования вычислений;
- восстановлена ранее удаленная функция `until`;
- системная переменная `ORIGIN` может использоваться для индексирования строковых переменных;
- удалена функция Кронекера и введен запрет на задание рекуррентных функций пользователя;
- улучшены средства регрессии общего вида, ввода векторных и матричных операций, улучшена работа со строковыми переменными, стало возможным переопределение встроженных функций и переменных и т. д.

1.1.2. Документы и рабочие листы Mathcad

Отличительной чертой всех версий Mathcad являются так называемые *документы*, которые объединяют описание математического алгоритма решения задачи (или ряда задач) с текстовыми комментариями и результатами вычислений, заданными в форме символов, чисел, таблиц или графиков. В оригинале документы системы Mathcad ныне названы английским словом «worksheets». В научном мире длинные листы бумаги с текстами, формулами и графиками шутливо называют рабочими «простынями» — так и переводится слово «worksheets» буквально. Или, более скромно, — рабочими листами. Мы оставим более короткий и привычный нам термин «документы».

Фактически документы Mathcad объединяют программу на визуально-ориентированном языке программирования Mathcad с результатами ее работы и текстовыми и формульными комментариями. Напомним, что визуально-ориентированные языки программирования задают программу не в виде малопонятных кодов, а в виде визуально понятных объектов. Язык программирования Mathcad ориентирован на математические вычисления и потому практически не отличается от обычного языка математических статей, отчетов и книг. Это огромное достоинство системы Mathcad. Оно делает документы Mathcad вполне ясными даже школьникам и студентам младших курсов вузов, имеющим элементарные знания по математике.

Итак, уникальное свойство Mathcad — возможность описания математических алгоритмов в естественной математической форме с применением общепринятой символики для математических знаков, таких, например, как квадратный корень, знак деления в виде горизонтальной черты, знак интеграла и так далее. Это делает документ, видимый на экране дисплея, чрезвычайно похожим на страницы текста математических книг и научных статей.

Такой подход значительно облегчает восприятие математической сущности решаемой задачи и избавляет пользователя от изучения некоторого промежуточного языка программирования (например, Фортрана, Бейсика, Паскаля и т. д.). Можно сказать, что в Mathcad идея решения математических задач без их программирования доведена до высшего совершенства — подавляющее большинство решаемых

задач требуют лишь корректного формульного описания решения и не нуждаются в подготовке программ в общепринятом понимании.

1.1.3. Входной язык и язык реализации Mathcad

Как следует из вышесказанного, общение пользователя с системой Mathcad происходит на уровне так называемого *входного языка*, максимально приближенного к обычному языку описания математических задач. Поэтому решение таких задач не требует программирования в общепринятом смысле — написания программ на некотором промежуточном языке или в машинных кодах.

Вот, к примеру, как выглядит вычисление квадрата переменной x с заданным значением $x = 3$ на популярном языке Бейсик и на Mathcad:

Бейсик	Mathcad	Комментарий
<code>x = LET(3)</code>	<code>x := 3</code>	Переменной x присваивается значение 3
<code>y = x^2</code>	<code>y := x^2</code>	Переменной y присваивается значение x^2
<code>PRINT("y =" ; y)</code>	<code>y = 9</code>	Вывод значения переменной y

Нетрудно заметить, что запись выражений на входном языке Mathcad куда более естественна, чем на Бейсике. К тому же она существенно короче. Эти достоинства проявляются еще сильнее при сложных вычислениях. Тем не менее это не означает, что в системе нет своего языка программирования. В действительности он есть, но это математически ориентированный особый язык программирования сверхвысокого уровня, используемый в основном как входной язык для диалога с системой.

Входной язык Mathcad относится к *интерпретирующему* типу. Это означает, что когда система опознает какой-либо объект, она немедленно исполняет указанные в блоке операции. Объектами системы могут быть формульные, текстовые и графические блоки. При этом формульные блоки могут иметь особые признаки — атрибуты, например, активности, пассивности и оптимизации. Мы их рассмотрим в дальнейшем.

Важно сразу учесть, что Mathcad выполняет действия над блоками в строго определенном порядке — блоки анализируются (оцениваются) слева направо и сверху вниз. Это означает, что блоки нельзя располагать в документе произвольно. Блоки, готовящие какие-либо операции, должны предшествовать блокам, выполняющим эти операции. Исключением являются блоки с глобальным определением (они также будут рассмотрены позже). Их можно располагать в любом месте документа, например в конце.

В подавляющем большинстве расчетных задач входной язык общения с Mathcad позволяет задавать их решение в виде вводимых с помощью операторов и функций математических формул и указывать тип желаемых результатов (таблицы или графики). Специальные приемы предусмотрены лишь для циклического изменения переменных и создания так называемых ранжированных переменных, имеющих набор значений.

Визуально-ориентированный язык общения системы Mathcad надо отличать от *языка реализации* системы, то есть обычного языка программирования высокого уровня, на котором написана система. Языком реализации системы Mathcad является один из самых мощных языков высокого уровня — C++.

По существу, входной язык системы — промежуточное звено между скрытым от пользователя языком документа и языком реализации системы. По мере того как пользователь создает (средствами текстового, формульного и графического редакторов) в окне редактирования объекты (тексты, формулы, таблицы и графики), система сама составляет программу на некотором промежуточном языке связи, которая хранится в оперативной памяти до тех пор, пока не будет сохранена на диске в виде файла с расширением *.mcd*. Однако важно подчеркнуть, что от пользователя не требуется знать языки программирования (реализации и связи), достаточно освоить приближенный к естественному математическому языку входной язык системы.

В новых версиях Mathcad значительно снижены требования и к знанию даже входного языка. Практически все операторы, имеющие вид привычных математических символов, можно выбирать мышью в палитрах математических объектов, а большинство математических функций (например, \sin , \cos , \exp и т. д.) имеют естественную форму задания, например, $\sin(x)$ так и вводится — $\sin(x)$. К тому же есть возможность выбора функций из списка, имеющегося в специальном окне, что резко уменьшает вероятность ошибок при вводе. Этот список выводится с помощью кнопки $f(x)$ на панели инструментов.

В Mathcad эффективно решена проблема сквозной передачи данных от одного объекта к другому, например, от одного математического выражения к другому, от него к таблицам, от таблиц к графикам и т. д. Поэтому изменение в любой формуле или в задании входных данных тут же ведет к пересчету задачи по всей цепи взаимодействия объектов, что, однако, не относится к символьным операциям, реализуемым с помощью команд меню.

1.1.4. Традиционные средства программирования

Лучшее — враг хорошего! Это в полной мере можно сказать о входном языке первых версий Mathcad для Windows. Этот язык был доведен до такого «совершенства», что из него напрочь исчезли многие основополагающие средства программирования (например, циклы, условные выражения, процедуры, обработчики ошибок и т. д.) в привычном для программистов виде. В результате пользователь, привыкший к обычному программированию, испытывал большие неудобства от замены этих средств непривычными средствами системы Mathcad. Тем более что иногда описание задачи в общепринятом на языках программирования виде оказывается предпочтительным и куда более коротким.

Впрочем, рассматриваемые в данной книге версии Mathcad давно избавились и от этого недостатка. В них включены важнейшие средства программирования и предусмотрена возможность задания программных блоков — процедур с общепринятыми операторами программирования. В совокупности с иными средствами входного языка это делает его необычайно гибким, мощным и наглядным.

1.1.5. Средства оптимизации вычислений

Как отмечалось, входной язык системы Mathcad — интерпретирующий. Интерпретаторы работают медленно, поэтому не случайно, что пользователи, работавшие со старыми версиями Mathcad, отмечали медлительность систем, особенно при сложных вычислениях и при построении графиков.

В последние версии Mathcad введена экспертная система *SmartMath*. Эта система старается использовать при численных вычислениях конечные формулы, полу-

ченные в результате символьных (аналитических) преобразований. Часто (хотя и не всегда) это дает значительное ускорение вычислений в сравнении с их реализацией численными методами.

Операция оптимизации вычислений с помощью системы SmartMath вводится специальными атрибутами (знак * после формул) и словами — директивами. Их число в новых версиях Mathcad значительно увеличено и для ввода операторов и директив символьной математики добавлена специальная палитра. Так что система SmartMath превратилась в полноправного члена семьи Mathcad. Наиболее развит этот подход в версиях системы начиная с Mathcad 2001 Premium и Mathcad 2001i, в состав которых введено оптимизирующее расширение *The Expert Solver*, автоматически включающее «на всю катушку» средства SmartMath и превращающее Mathcad в систему искусственного интеллекта. В Mathcad 11/12 эти средства стали встроенными в систему. Можно включать оптимизацию как отдельных выражений, так и документа в целом.

1.1.6. Средства расширения систем Mathcad

Начиная с версии Mathcad PLUS 5.0 в систему введена возможность ее расширения функциями, которые задаются обычными программами на языке C или C++. Однако это не позволяет эффективно и просто решить проблему «усиления» системы. На C или C++ хорошо программируют системные программисты, но они весьма редко разбираются в сути математических задач. Обычные пользователи, освоив возможности Mathcad, не захотят возвращаться к тому, от чего уже ушли — к программированию на достаточно сложном языке. В общем, как в ходячей фразе — «за что боролись, на то и напоролись»!

Как отмечалось, начиная с версии Mathcad PLUS 6.0 у системы появилась весьма изящная возможность записи встроенных в документ программных модулей, реализующих типовые управляющие структуры и записанных в виде обычных программ. Так что теперь Mathcad предоставляет программистам полную возможность для самовыражения. Средством локального расширения системных возможностей являются также функции пользователя.

Однако в наивысшей степени средства расширения системы Mathcad представлены сменными проблемно-ориентированными электронными книгами, библиотеками и пакетами расширения. Они позволяют настроить систему на наиболее эффективное решение задач в любой области науки и техники — в математике, физике и химии, в астрономии, механике, электротехнике и радиотехнике, в биологии и в экономике, в финансах, статистике и т. д.

Электронные книги — это пакеты для решения задач в определенной области науки и техники, ориентированные на типовые средства систем класса Mathcad.

Пакеты расширения — это укрупненные библиотеки, поставляемые с электронными книгами, учитывающими новые операторы и функции, которые пакеты расширения вводят в базовую систему Mathcad. Входящие в них электронные книги нельзя использовать без соответствующих библиотек. Описание ряда пакетов расширения можно найти в [10, 11].

Библиотеки — это комплекты электронных книг и пакетов расширения. Очень часто библиотеки строятся на основе электронных вариантов крупных справочников и энциклопедий по тем или иным отраслям знаний, например по механике, астрономии или электротехнике.

ПРИМЕЧАНИЕ В конце названий пакетов расширения имеются слова «*Extension Pack*», а библиотек — слово «*Library*». Уже сейчас суммарный объем выпущенных электронных книг, пакетов расширения и библиотек достигает многих сотен мегабайт и для их мало-мальски серьезного описания нужны специальные книги (на сей раз обычные).

1.2. Начало работы с Mathcad

1.2.1. Запуск Mathcad

Mathcad обычно используется в среде операционных систем Windows 96/98/ME/NT/2000/XP. Естественно, что более ранние версии Mathcad могут работать с более ранними версиями Windows. Установка (инсталляция) системы Mathcad в наши дни общеизвестна и описана в многочисленных книгах [6—11]. Поэтому будем считать, что система Mathcad той или иной версии уже установлена на вашем ПК. Хотя ниже описана работа с новейшими версиями Mathcad 11 и 12, пользователи более старыми версиями не испытают особых затруднений в знакомстве с ними.

Для запуска Mathcad откройте главное меню Windows, щелкнув на кнопке Start (Пуск) и раскрыв меню Programs (Программы), в котором можно обнаружить подменю с относящимися к Mathcad командами. Это подменю имеет заданное при установке имя — MathSoft Apps (или иное, если вы его задали сами). Команда запуска системы называется Mathcad*, где * обозначает конкретную используемую версию системы. Если вы намерены часто работать с системой Mathcad, имеет смысл разместить ярлык программы на рабочем столе Windows 98/ME/2000/NT (рис. 1.1). Как это делается, читатель хорошо знает.

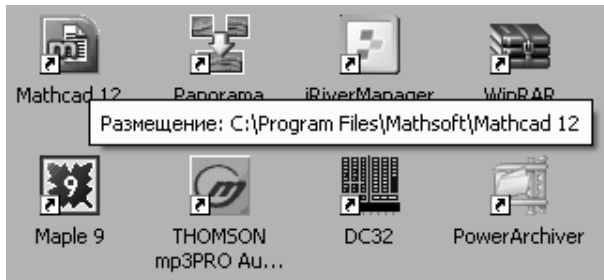


Рис. 1.1. Подготовка к запуску системы Mathcad 12 с помощью ярлыка на рабочем столе Windows

После запуска Mathcad на некоторое время (зависит от быстродействия ПК) появляется заставка, которая вскоре сменяется основным окном системы (рис. 1.2). В окне системы присутствует окно Mathcad Tips с подсказками по работе с системой. Кроме того, видна панель математических палитр, которую можно перенести в область инструментальных панелей, расположенную под меню системы.

О различии интерфейса систем можно судить по рис. 1.3, на котором показано окно системы Mathcad 8. Сравнив его с окном Mathcad 12 (рис. 1.2), можно убедиться в том, что существенного изменения интерфейса не произошло. В Mathcad 12 появился ряд новых кнопок в панелях инструментов, но разобраться в их на-

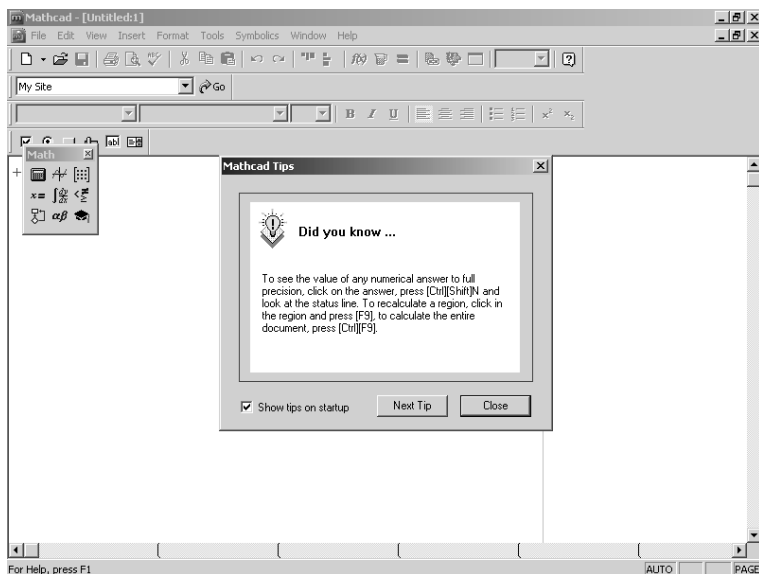


Рис. 1.2. Окно системы Mathcad 12 в среде операционной системы Windows XP

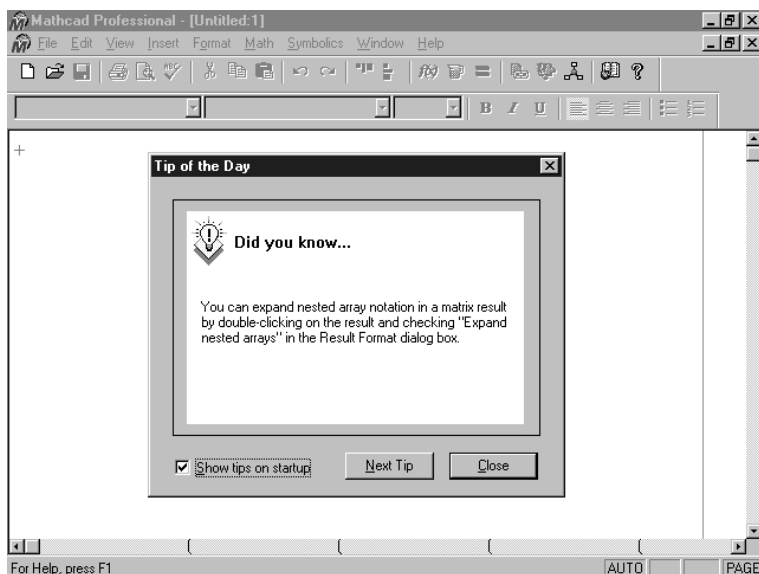


Рис. 1.3. Окно системы Mathcad 8

значении нетрудно. Тем более если учесть, что при наведении курсора мыши на ту или иную кнопку около нее появляется желтый прямоугольник со справкой о назначении данного объекта интерфейса.

В окне систем Mathcad 2000 — Mathcad 11 присутствует окно Центра ресурсов (Resource Center), дающее доступ к учебнику для новых пользователей, средствам обновления (Upgraders), Интернет-сайту фирмы MathSoft, Inc. и средствам коллективной работы над научными проектами. Это окно имеет стандартные средства

управления окнами. В частности, используя кнопку **Заккрыть**, это окно можно закрыть и оно исчезнет с экрана. Обычно в окне системы присутствует и панель палитр математических знаков. С помощью позиции **View** меню можно менять состав инструментальных панелей и подгонять интерфейс систем Mathcad под свой вкус.

1.2.2. Основные возможности систем Mathcad

Приведем примеры работы с системой Mathcad (рис. 1.4), сразу выявляющие ее главные возможности и особенности. Нетрудно заметить, что в этих примерах задана некоторая функция $f(x)$, построен ее график, вычислены корни функции (значения x , при которых $f(x) = 0$), найдено значение производной $f'(x)$ в точке $x = 0$ и, наконец, вычислен определенный интеграл с подынтегральной функцией $f(x)$ и пределами от $a = 0$ до $b = 10$. Таким образом, проведен хотя и не исчерпывающий, но достаточно полный анализ функции $f(x)$.

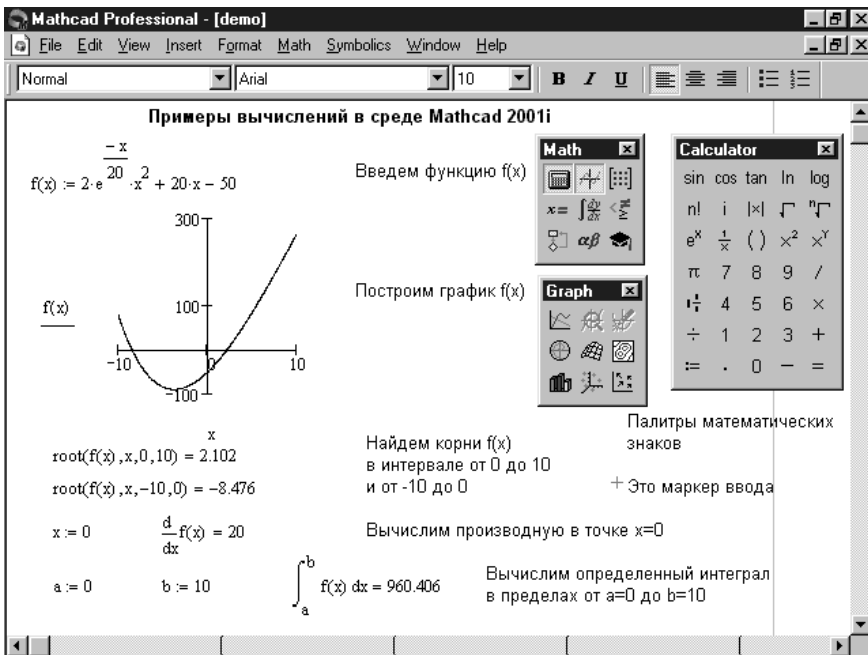


Рис. 1.4. Примеры работы с системой Mathcad 2001i

Рисунок 1.5 показывает этот же документ в среде новейшей версии Mathcad 12. Сравнение рис. 1.5 с 1.4 показывает на полное отсутствие отличий в самом документе — разве что только чуть заменена надпись, в которой уточнена версия системы. Основные отличия видны в интерфейсе, да и те, скорее, вызваны разными наборами выведенных панелей инструментов.

Учитывая сказанное, в большинстве примеров применения систем Mathcad в данной книге приводится только вид документов — рисунки даются в рамке. Подавляющее большинство примеров при этом не зависит от используемой версии системы Mathcad. Однако, если используются средства, характерные для той или иной версии, то документы для них будут непригодны для других версий без определенной доработки. Такие случаи будут оговариваться отдельно.

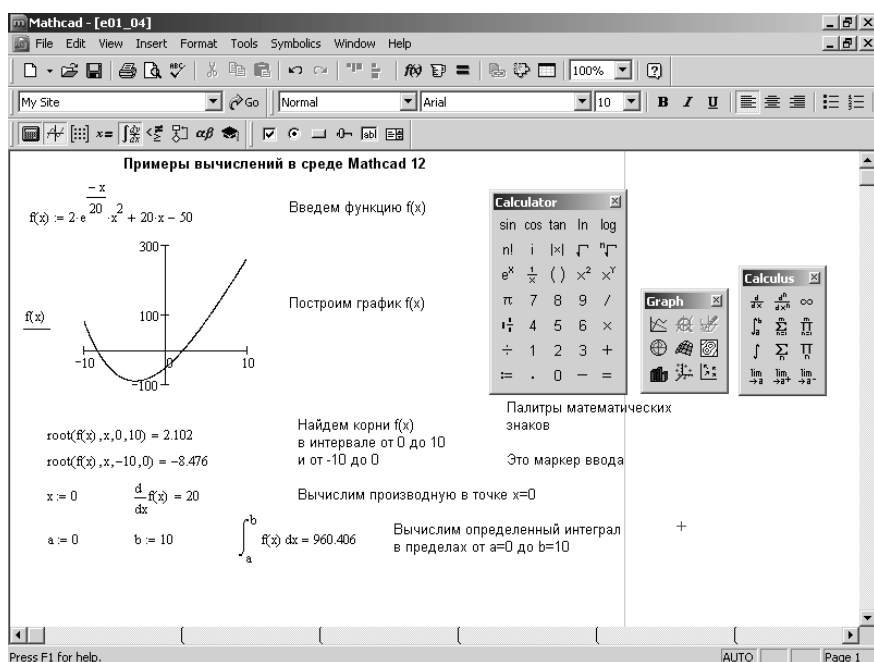


Рис. 1.5. Примеры работы с системой Mathcad 12

Все приведенные на рисунке примеры предельно наглядны и одинаковы для любой версии Mathcad. Лишь при внимательном рассмотрении возникают отдельные вопросы: как введена функция $f(x)$, как конкретно построен ее график, почему вместо записи $x = 0$ стоит «странное» $x := 0$, что это за функция root , как задаются математические символы дифференцирования и интегрирования, откуда в англоязычной программе взялись русскоязычные комментарии? И что еще может Mathcad? Ответы на эти и множество других подобных вопросов и составляют содержание последующих глав этой книги.

1.3. Основы работы с системой Mathcad

1.3.1. К чему сводится работа в среде Mathcad

В простейшем случае работа с системой Mathcad сводится к вводу в окне редактирования заданий на вычисления и установке форматов для их результатов. Для этого используются различные приемы подготовки блоков. Пока мы рассмотрим подготовку блоков двух типов — текстовых и формульных. Заметим, что вычисления выполняются автоматически, поскольку по умолчанию установлен режим автоматических вычислений.

ПРИМЕЧАНИЕ Подробно интерфейс программы мы будем изучать в главе 2, поэтому, если какие-нибудь детали окажутся вам непонятными, не отчаивайтесь. Здесь нам важно получить общее представление о возможностях системы Mathcad, а к конкретным приемам работы мы будем многократно возвращаться по мере изложения материала.

1.3.2. Работа с текстовым редактором

Тексты в математической системе Mathcad лишь на первый взгляд имеют второстепенное значение. На самом деле профессионально сделанные в Mathcad документы прежде всего должны иметь достаточно подробные текстовые комментарии. Если в данной книге и есть отступления от этого правила, то только потому, что такие комментарии уже имеются в тексте книги и нет смысла повторять их в документах Mathcad.

В простейшем случае для ввода текстового комментария достаточно ввести символ " (одна двойная кавычка — не путайте с двумя одиночными кавычками или двумя апострофами). В появившемся прямоугольнике можно начинать вводить текст (рис. 1.6). В текстовом блоке курсор имеет вид красной вертикальной черты и отмечает место ввода. Текст редактируется общепринятыми средствами — перемещением курсора ввода клавишами управления курсором, установкой режима вставки или замещения символов (клавиша Insert), стиранием (клавиши Del и Backspace), выделением, копированием в буфер обмена, вставкой из буфера и т. д.

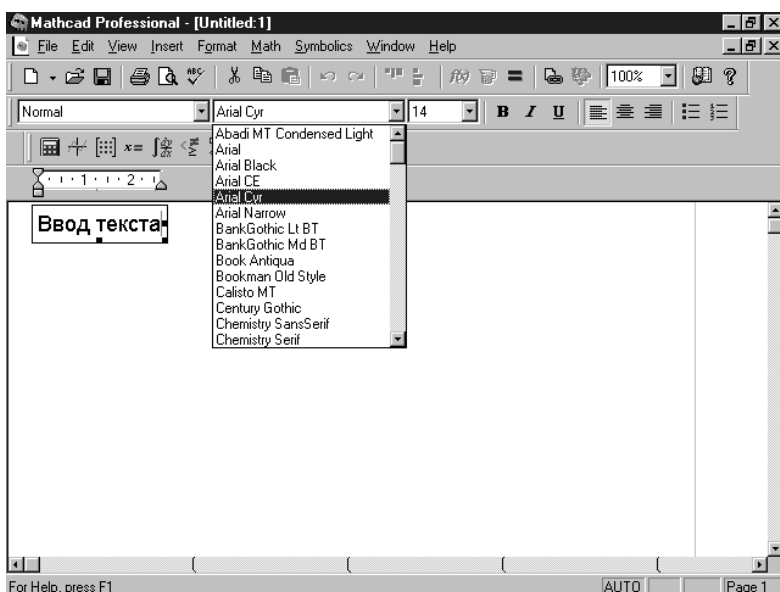


Рис. 1.6. Пример ввода текстового комментария

Удобно осуществлять управление стилем текста с помощью панели форматирования. В текстовом блоке есть также возможность выравнивать текст по левой его границе, правой границе или по центру. А для англоязычных текстов предусмотрен даже орфографический контроль с применением встроенного словаря.

Клавиши перемещения курсора можно использовать с нажатой клавишей Shift, что обеспечивает ускоренное перемещение в текстовом блоке. Например, сочетания клавиш Shift+← и Shift+→ позволяют перемещать курсор ввода не по отдельным символам, а по целым словам.

Пока мы ограничимся простейшими правилами ввода текста.

1. Введите знак двойной кавычки " при английской раскладке клавиатуры — появится прямоугольник с курсором ввода.

2. С помощью раскрывающегося списка выбора шрифта на панели форматирования (см. рис. 1.6) установите нужный шрифт, например кириллический при вводе русскоязычного текста.
3. Начните посимвольно набирать текст, используя типовые средства текстового редактора (клавиши Пробел, Del, Backspace и т. д.).
4. Нажимайте клавишу Enter для перехода на новую строку (если этого не делать, переход на новую строку будет осуществляться автоматически).
5. Для завершения ввода текста отведите указатель мыши в сторону от текстового блока и щелкните левой кнопкой мыши.

Текстовый блок в ходе набора текста расширяется (см. рис. 1.6, на котором этот блок виден в окне документа). Блок имеет маркеры изменения размера в виде маленьких черных прямоугольников, уцепившись за которые блок можно растягивать в том или ином направлении. Это иллюстрирует рис. 1.7, на котором показан растянутый блок. Обратите внимание, что размер символов текстового комментария при растяжении (или сжатии) блока не меняется — меняется только размер блока. В блок большего размера можно ввести больше надписей. Как только длина строки текста достигает правой границы текстового блока, происходит перенос вводимого текста на новую строку внутри блока.

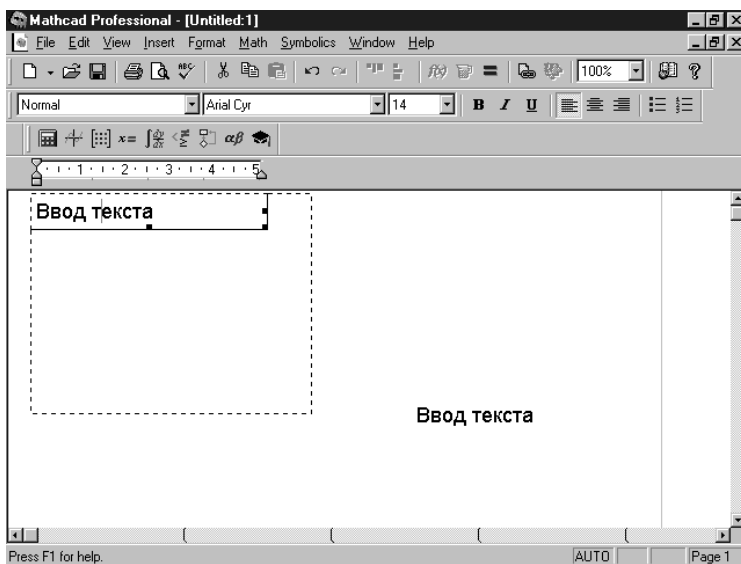


Рис. 1.7. Растяжение текстового блока и его копирование

Выделенные рамкой текстовые блоки можно переносить на другое место, уцепившись за рамку указателем мыши — он при этом превращается в изображение черной ладошки. Если в начале перемещения нажать клавишу Ctrl, то будет выполняться перенос блока с его сохранением на первоначальном месте. Именно так получен показанный на рисунке второй текстовый блок.

Для завершения создания текстового блока достаточно отвести указатель мыши в сторону от блока и щелкнуть левой кнопкой. Можно также нажать одновременно клавиши Ctrl+Shift+Enter.

Для коррекции текста надо подвести указатель мыши к месту коррекции и щелкнуть левой кнопкой мыши. Появится рамка текстового блока, а на месте

указателя мыши — курсор ввода. Теперь можно корректировать текст, дописывать его, вставлять новые символы между старыми (режим Insert), удалять символы и т. д. Словом, можно пользоваться всеми приемами редактирования, хорошо известными пользователям любой программы подготовки текстов. Позже мы увидим, что текстовые комментарии могут форматироваться и даже подвергаться синтаксическому контролю (увы, только для англоязычных текстов).

СОВЕТ *Нередко пользователь начинает набор текстов, забыв установить признак текстового блока (ввести двойную кавычку "). Mathcad воспринимает такой набор как ввод математического выражения. Однако, нажав клавишу Пробел, можно тут же превратить набранный фрагмент в текстовый.*

1.3.3. Работа с формульным редактором

Фактически интегрированная прикладная система Mathcad интегрирует в себе три редактора: формульный, текстовый и графический. Для запуска формульного редактора достаточно установить указатель мыши в любом свободном месте окна редактирования и щелкнуть левой кнопкой. В этом месте окажется курсор ввода в виде маленького красного крестика. Его можно перемещать клавишами перемещения курсора. Курсор ввода не надо путать с указателем мыши, который живет своей «жизнью» и имеет вид жирной наклонной стрелки.

Курсор (маркер) ввода указывает место, с которого можно начинать набор формул — вычислительных блоков. В зависимости от места расположения курсор ввода может менять свою форму, к чему быстро привыкаешь. Так, в области формул он превращается в синий уголок, указывающий направление и место ввода. Его также можно перемещать клавишами перемещения курсора. Для расширения охваченной уголком области (вплоть до полного охвата выражения) можно пользоваться клавишей Пробел.

В системах класса Mathcad 11/12 редактирование математических выражений существенно улучшено и во многом напоминает редактирование текстовых комментариев. Так, можно использовать режим вставки (клавиша Insert, меняющая направление охвата курсором ввода того или иного объекта), режимы стирания (клавиши Backspace и Del) объекта с той или иной стороны относительно вертикальной черты курсора ввода, удаление объекта командой Cut (клавиша F3) и его ввод на указанное место командой Paste (клавиша F4) и др. Редактирование интуитивно понятно, поэтому мы ограничимся рядом простых примеров, учитывая, что лишь достаточный опыт практической работы с системой гарантирует должную эффективность такой работы.

1.3.4. Подготовка и вычисление простых выражений

Рассмотрим пример вычисления отношения суммы чисел 2 и 3 к корню квадратному из числа 5. Вначале введем подряд символы $2 + 3$. Вид формульного блока при этом показан на рис. 1.8. Обратите внимание, что при таком последовательном вводе уголок охватывает последний операнд.

Теперь нам надо ввести знак деления. Однако если сделать это сразу, то данный знак будет относиться не ко всей сумме, а только к последнему операнду — числу 3. Чтобы знак деления относился ко всей сумме, надо выделить все выражение $2 + 3$. Для этого достаточно нажать клавишу Пробел. Результат будет иметь вид, представленный на рис. 1.9.

$$2 + 3$$

Рис. 1.8. Создание формульного блока и ввод в него суммы чисел 2 и 3

$$2 + 3$$

Рис. 1.9. Выделение всей суммы курсором ввода

Теперь можно ввести знак деления, нажав клавишу / (со знаком наклонной черты). Формульный блок приобретет вид, показанный на рис. 1.10. Обратите внимание, что знак деления в виде наклонной черты автоматически приобрел вид длинной горизонтальной черты под суммой, а под ним появилось место для ввода знаменателя дроби в виде черного квадрата, охваченного курсором ввода.

Разумеется, как это принято даже при работе с калькуляторами, можно было бы выделить сумму скобками, записав ее в виде $(2 + 3)$, тогда знак деления относился бы явно к сумме. Однако не следует вводить скобки там, где без них можно легко обойтись.

Следующий этап — ввод знака квадратного корня. Начинаящий пользователь введет такой знак, используя палитру математических знаков для арифметических операций Calculate (о том, как вывести палитры, вы узнаете в главе 2). Для такого ввода достаточно щелкнуть мышью на знаке квадратного корня. Однако более опытный пользователь вспомнит, что многие знаки вводятся прямо нажатием соответствующей клавиши или комбинации клавиш. Например, для ввода квадратного корня достаточно нажать клавишу со знаком обратной косой черты \ (ее вид напоминает черту, с которой мы начинаем писать знак квадратного корня). Теперь формульный блок приобретет вид, представленный на рис. 1.11. Обратите внимание на естественность записи оператора квадратного корня и наличие в нем места для ввода подкоренного выражения.

$$\frac{2 + 3}{\quad}$$

Рис. 1.10. Формульный блок после ввода знака деления

$$\frac{2 + 3}{\sqrt{\quad}}$$

Рис. 1.11. Формульный блок после ввода знака квадратного корня

Следующий этап конструирования выражения — ввод подкоренного выражения в виде числа 5. Для этого достаточно нажать клавишу 5 на цифровой части клавиатуры. Формульный блок предстанет в виде, показанном на рис. 1.12.

$$\frac{2 + 3}{\sqrt{5}}$$

Рис. 1.12. Формульный блок с заданным выражением

Теперь выражение по существу введено полностью, и нам осталось увидеть результат его вычисления. Для этого в конце выражения нужно поставить оператор вывода — знак равенства =. Однако сразу его вводить нельзя, поскольку

$$\frac{2 + 3}{\sqrt{5}}$$

Рис. 1.13. Выделение всего выражения

$$\frac{2 + 3}{\sqrt{5}} = 2.236$$

Рис. 1.14. Формульный блок после введения оператора вывода

маркер вывода установлен на последнем операторе. Надо выделить все выражение. Для этого нажмите клавишу Пробел — будет выделен весь числитель, а затем нажмите клавишу Пробел еще раз — будет выделено все выражение (рис. 1.13).

Осталось последнее — ввести оператор вывода =. Mathcad тут же автоматически отобразит результат вычислений, поскольку по умолчанию установлен режим автоматических вычислений. Вид формульного блока при этом представлен на рис. 1.14.

Mathcad можно легко использовать для различных экспериментов вычислительного характера. Допустим, мы решили вычислить результат, когда подкоренным выражением в знаменателе должно быть число 5 в степени 1.25. Надо ли набирать новое выражение с начала? Конечно же нет. Достаточно модернизировать уже введенное (см. рис. 1.22) выражение. Для этого аккуратно поместите указатель мыши после числа 5 и щелкните левой кнопкой. Вы увидите, что курсор ввода отметит это число (рис. 1.15).

$$\frac{2 + 3}{\sqrt{5}} = 2.236$$

Рис. 1.15. Изменение одного из операндов в формульном блоке

Нетрудно догадаться, что теперь надо ввести знак возведения в степень. Его можно выбрать в палитре арифметических операций или просто нажать клавишу со знаком ^. При этом в формульном блоке появится запись возведения числа 5 в некую степень (рис. 1.16). Место ввода служит для задания нужной степени.

Ясно, что теперь нам надо ввести показатель степени числа 5, то есть число 1.25. Это иллюстрирует рис. 1.17. Обратите внимание на естественность записи числа 5 в степени 1.25.

Итак, выражение создано, но вычисления не выполняются, пока курсор ввода (уголок) выделяет число 1.25. Для вычисления достаточно отвести указатель мыши от формульного блока (например вниз) и щелкнуть левой кнопкой. Сконструированное выражение тут же будет вычислено (рис. 1.18).

Этот пример приведен, дабы показать естественность работы с математическими выражениями. Конечно, ввод слож-

$$\frac{2 + 3}{\sqrt{5^{\quad}}} = \dots$$

Рис. 1.16. Вид формульного блока после ввода оператора возведения в степень

$$\frac{2 + 3}{\sqrt{5^{1.25}}} = \dots$$

Рис. 1.17. Вид формульного блока после ввода показателя степени

$$\frac{2 + 3}{\sqrt{5} \cdot 1.25} = 1.829$$

Рис. 1.18. Вычисление скорректированного выражения

ных выражений не так прост, но, поупражнявшись в таком вводе, можно быстро «набить руку» для ввода даже сложных выражений.

1.3.5. Операции вывода и присваивания

Для вычисления любого выражения достаточно установить после него оператор вывода (знак =). Покажем это на нескольких примерах простых вычислений. Для ввода десятичных чисел в качестве разделителя целой и дробной частей используется точка, а не запятая.

Ввод	На экране дисплея
1.234 * 2.345 =	1.234 · 2.345 = 2.894
1/7 =	$\frac{1}{7}$ = 0.123
cos(0.5) =	cos(0.5) = 0.878
e^2 =	e ² = 7.389

В математике для придания вычислениям общности часто используются переменные в виде некоторых обобщенных обозначений данных определенного типа. Переменные имеют имена (идентификаторы), и для них характерна операция присваивания значений. Начиная с Mathcad 2000 оператор = можно использовать как оператор первого присваивания. Покажем это «на пальцах», то бишь в виде обычного описания.

Ввод	На экране дисплея
a = 2	a := 2
b = 3	b := 3
a + b =	a + b = 5

Если теперь попытаться присвоить переменным a и b новые значения с помощью оператора =, то ничего из этого не выйдет. Как только после имени переменной мы попытаемся поставить знак =, появится старое значение переменной.

Ввод	На экране дисплея
a =	a = 2
b =	b = 3

Чтобы все же присвоить переменным новые значения, придется использовать стандартный оператор присваивания $:=$, который вводится своим первым символом $:$ (двоеточие).

Ввод	На экране дисплея
a : 1	a := 1
b : 1	b := 1
a + b =	a + b = 2

Из этих примеров можно заметить некоторые особенности работы Mathcad при выполнении простых вычислений:

- некоторые комбинированные операторы (например $:=$) вводятся одним символом;
- Mathcad вставляет пробелы до и после арифметических операторов;
- оператор умножения вводится как звездочка, но представляется точкой в середине строки;
- оператор деления вводится как косая черта, но заменяется горизонтальной чертой;
- оператор возведения в степень вводится знаком $^$, но число в степени представляется в обычном виде (степень как верхний индекс);
- по умолчанию десятичные числа имеют представление с тремя знаками после разделительной точки;
- Mathcad понимает наиболее распространенные константы, например e — основание натурального логарифма (убедитесь, что программа распознает также π или i);
- математические выражения могут редактироваться внутри формульного блока с использованием для этого курсора ввода и типовых приемов редактирования.

Таким образом, даже без вмешательства пользователя Mathcad старается придать математическим выражениям обычный вид. На рис. 1.19 представлен документ, в котором выполнены описанные выше простые вычисления.

Рисунок дает представление и о некоторых дополнительных приемах работы с системой Mathcad. Например, как быть, если точность представления десятичных чисел тремя знаками после точки вас явно не устраивает. Для этого надо использовать форматирование чисел (подробно это рассматривается в главе 5). Подведите указатель мыши к числу (на рисунке — это значение e^2) и дважды щелкните левой кнопкой. Появится окно форматирования чисел, показанное на рисунке в правой части окна документа. Введите в поле Number of decimal places (Число цифр после десятичной точки) число 15 вместо 3 и наслаждайтесь созерцанием результата с 15 знаками после десятичной точки.

В Mathcad 2001 введена новая возможность представления результатов вычислений с заданной погрешностью в виде рациональных чисел. Для этого в окне форматирования чисел появился дополнительный вариант формата — Fraction (Дроби). При выборе этого формата Mathcad вместо обычного десятичного числа дает его значение в виде отношения целых чисел.

Mathcad переводит в категорию простых вычислений даже сложные математические расчеты, требующие применения специальных численных методов. Позже

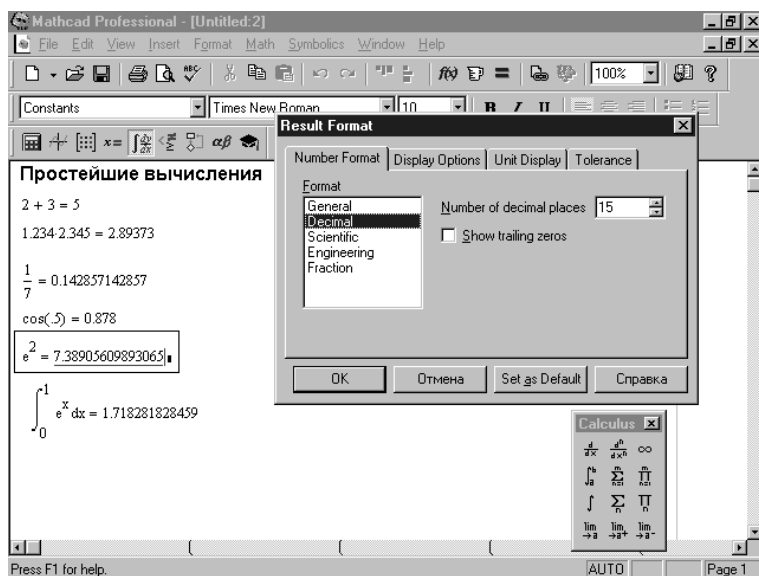


Рис. 1.19. Окно Mathcad с простейшими вычислениями

мы рассмотрим их более подробно, а пока отметим последний пример на рис. 1.19 — вычисление определенного интеграла. Нетерпеливый читатель, вероятно, уже понял, что для этого был использован шаблон интеграла из представленной на рисунке палитры Calculus. Для Mathcad все равно, что вычисляется — будь то привычный школьнику квадратный корень или милый инженеру определенный интеграл. Столь же просто, воспользовавшись шаблоном суммы ряда или произведения ряда, можно вычислить сумму или произведение членов ряда.

Учтите только, что Mathcad ведет численные расчеты с конечной погрешностью. Ее максимальное значение задается значением переменной TOL. По умолчанию TOL=0.001, т. е. гарантируется погрешность не более единицы третьего знака после десятичной точки. Нередко она оказывается намного меньше, но «гарантия есть гарантия». Вы можете присвоить TOL и иное значение, например TOL:=10⁻⁶. Чем меньше значение TOL, тем выше точность вычислений, но больше время, затрачиваемое на их выполнение. Без погрешности выполнение ряда вычислений возможно в символьном режиме вычислений.

1.3.6. Применение шаблонов математических операторов и символов

Подготовка вычислительных блоков облегчается благодаря выводу шаблона при задании того или иного оператора. Для этого в Mathcad служат палитры математических символов и шаблонов операторов и функций. В Приложениях 2—4 справочного раздела книги приведены все операторы, функции и системные переменные систем Mathcad.

Надо полагать, что читатель знаком с применением большинства математических операторов и знает, сколько операндов в них используется. Например, оператор суммы (просто знак «плюс» — +) на самом деле выводит шаблон, в котором места ввода двух операндов идентифицируются черными квадратиками. Так, если

требуется составить выражение $2 + 3$, то в первое место ввода достаточно вставить число 2, а во второе — 3.

Допустим, мы желаем вычислить уже упомянутый определенный интеграл. Для этого вначале надо вывести на экран палитру операторов математического анализа. Щелкните на кнопке с изображением знака интеграла и производной, и палитра появится в окне программы. Затем следует установить курсор в требуемое место экрана (там появится шаблон) и щелкнуть в палитре на значке с изображением знака определенного интеграла.

В составе сложных шаблонов часто встречаются меньшие шаблоны для ввода отдельных данных. Они имеют вид маленьких черных прямоугольников. Мы будем называть их, как и ранее, *местами ввода*. В шаблоне интеграла их четыре: для ввода верхнего предела интегрирования, для ввода нижнего предела интегрирования, для задания подынтегральной функции и для указания имени переменной, по которой идет интегрирование. На рис. 1.20 сверху показана работа с шаблонами некоторых функций и операторов.

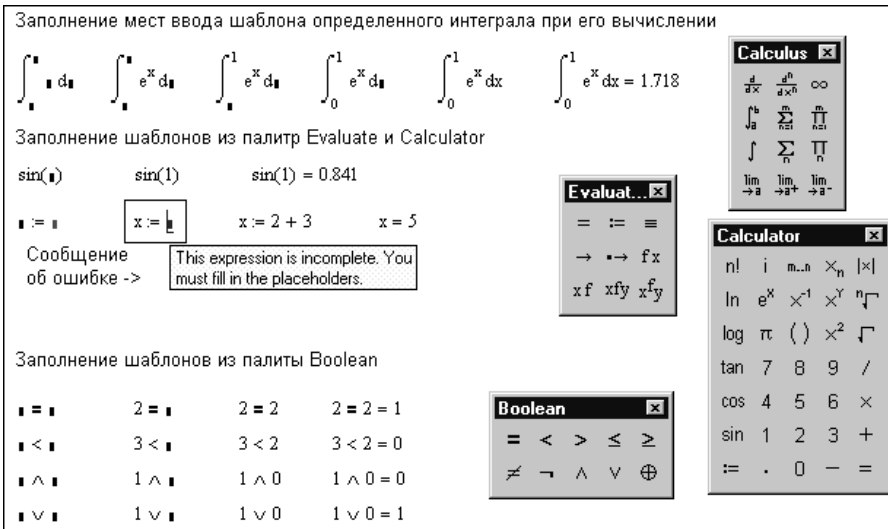


Рис. 1.20. Работа с шаблонами

Аналогичным путем задаются шаблоны для вычисления функций и для применения операторов — в частности, на рис. 1.20 показано применение логических операторов из палитры таких операторов Boolean. Обратите внимание также на вывод всплывающего сообщения об ошибке в центре экрана. Обычно признаком ошибки является окраска неверного выражения в красный цвет. Если в такое выражение ввести курсор мыши, то и появится указанное сообщение об ошибке.

Рассмотрим процесс работы с шаблонами немного подробнее. После вывода шаблона он появляется с пустыми местами ввода. Для ввода данных можно установить указатель мыши на нужный шаблон данных, щелкнуть ее левой кнопкой для фиксации места ввода и ввести данные. Рассмотрим процесс последовательного заполнения (слева направо) мест ввода в шаблоне, например, вычисления определенного интеграла — рис. 1.20, сверху. Заметим, что вычисления выполняются после охвата выражения (интеграла) синим уголком и установки оператора вывода — знака =.

Следует отметить, что шаблоны многих операций вводятся такими символами, которые хотя бы в первом приближении напоминают вид оператора. Например, оператор квадратного корня вводится клавишей $\sqrt{\quad}$, шаблон определенного интеграла — клавишей \int , а клавиша $\#$ используется для ввода знака произведения ряда и т. д. Для ввода знака суммы членов ряда требуется нажатие сразу трех клавиш — $\text{Ctrl}+\text{Shift}+\text{F4}$. Несмотря на архаичность такого ввода, ему нельзя отказать в оперативности, ибо вывод палитр математических знаков и удобное их расположение требуют времени. Правила ввода всех шаблонов даны в Приложении 2 справочной части книги.

1.3.7. Применение встроенных функций

Сложные математические выражения наряду с операторами содержат математические функции. Mathcad имеет множество встроенных элементарных, специальных и статистических функций. Наиболее известные из них — элементарные — могут вводиться прямо их обозначениями, например, $\sin(1)$, $\cos(0.5)$, $\text{asin}(0.5)$, $\sinh(1)$, $\ln(2)$ и т. д. Однако на первом этапе освоения системы многие пользователи путаются в обозначениях функций и не представляют, какие функции есть в системе и как именно их вводить. Особенно это относится к обратным функциям: многие задают встроенную функцию asin Mathcad, как это принято в математике — \arcsin . И получают сообщение об ошибке.

Для облегчения ввода математических функций служит кнопка $f(x)$, которая выводит окно с полным перечнем функций, разбитым на тематические разделы. Выбранная функция может быть перенесена в окно документа щелчком на кнопке внизу окна с перечнем функций. Например, ниже представлена последовательность заполнения шаблона для вычисления функции Бесселя, которую иллюстрирует рис. 1.21.

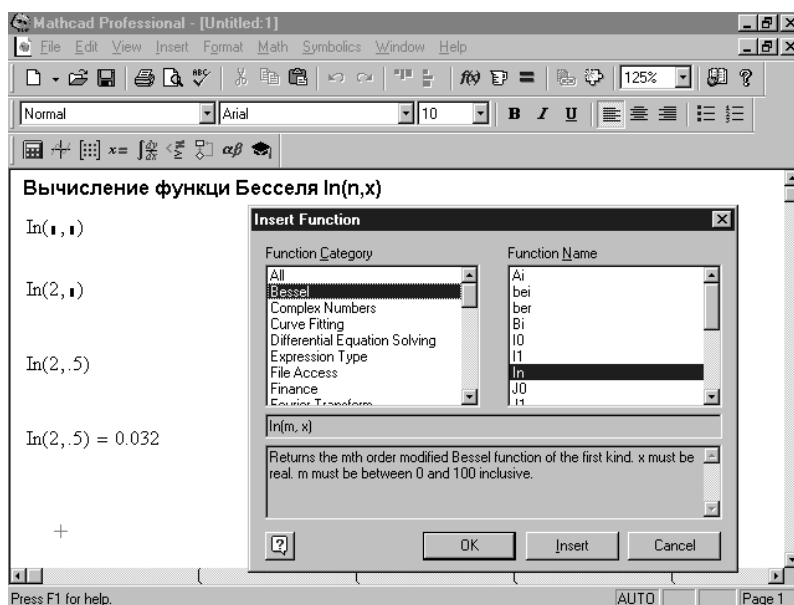


Рис. 1.21. Пример вычисления функции Бесселя с выбором ее из перечня функций

Функции имеют параметры (аргументы), которые записываются в круглых скобках после имени функции. Функции могут иметь один параметр (например, $\sin(x)$ или $\cos(0.5)$), два параметра (например, $\ln(m, x)$) или много параметров. Параметры могут иметь численное значение, быть константой, определенной ранее переменной или математическим выражением, возвращающим численное значение. Функции имеют свойство возвращать результат, поэтому их можно использовать в сложных математических выражениях, например:

$$(2 + 3i) \cdot \sin(3 \cdot e^{-1})$$

Заметим, кстати, что i в этом выражении — мнимая единица (квадратный корень из -1) и что большинство функций может иметь комплексные аргументы и возвращать комплексные значения. Любопытно, что знак умножения в мнимой части комплексных чисел Mathcad опускает, как это и принято при записи в математике.

1.3.8. Применение ранжированных переменных и векторов

В первых примерах мы обходились хорошо известными типами переменных, хранящих значения обычных чисел (целых, вещественных или действительных). Mathcad допускает для таких переменных и комплексные значения, например $2 + 3i$. В отличие от языков программирования Mathcad не требует точного задания типов обычных переменных, например таких, как целочисленные переменные одинарной или двойной точности, вещественные переменные с плавающей точкой и т. д. Тип переменной автоматически определяется присвоенным ей значением.

Однако есть и специальные типы переменных, именуемые ранжированными. Они имеют множественные значения. Например, если записать $n := 1..5$, то переменная n будет представлять целые числа от 1 до 5 с шагом 1, то есть значения 1, 2, 3, 4 и 5. Возможность доступа отдельно к каждому значению отсутствует.

СОВЕТ Для ввода знака $:=$ надо нажать клавишу $:$ (двоеточие), а для ввода знака $..$ (две точки подряд) — клавишу $;$ (точка с запятой).

Если требуется задать ряд чисел с шагом d , то ранжированная переменная записывается следующим образом:

$$X := Xstart, Xstart + d .. Xend$$

Здесь $Xstart$ — начальное значение переменной X , $Xend$ — конечное значение переменной X . Например, $X := 1, 1.25..2$ дает ранжированную переменную X со значениями 1, 1.25, 1.5, 1.75 и 2.

ПРИМЕЧАНИЕ Обратите внимание на то, что второй член в задании ранжированной переменной — это сумма ее начального значения с приращением d . Можно задать приращение отрицательным, но в этом случае значение $Xstart$ должно быть больше, чем $Xend$. В явном виде приращение обычно не задается и фигурирует как разность чисел, разделенных запятыми, например в нашем примере $X := 1, 1.25..2$ приращение есть $1.25 - 1 = 0.25$.

На первый взгляд ранжированная переменная напоминает вектор — одномерный массив индексированных переменных. Например, вектор X из трех элементов 1, 1.25 и 1.5 может быть представлен тремя компонентами X_0 , X_1 и X_2 с указанными

значениями. Индексы могут начинаться с 0 или 1. Эти значения задает специальная системная переменная `ORIGIN`, имеющая по умолчанию нулевое значение. Если переменная — не вектор, то задавать индексы недопустимо и Mathcad будет отмечать это как ошибку, выделяя выражение красным цветом и сопровождая его сообщением об ошибке в виде всплывающей подсказки.

Ниже представлен ряд примеров задания и вывода ранжированных переменных и вектора V :

$i := 0..4$	$V_i := i$	$x := 0,0.25..1$	$y := -1,-.75..0$																							
$i =$	$V =$	$x =$	$y =$																							
<table border="1" style="margin: auto; border-collapse: collapse;"> <tr><td style="padding: 2px 10px;">0</td></tr> <tr><td style="padding: 2px 10px;">1</td></tr> <tr><td style="padding: 2px 10px;">2</td></tr> <tr><td style="padding: 2px 10px;">3</td></tr> <tr><td style="padding: 2px 10px;">4</td></tr> </table>	0	1	2	3	4	<table style="margin: auto;"> <tr> <td style="font-size: 2em; vertical-align: middle;">(</td> <td style="padding: 0 10px;"> <table border="1" style="margin: auto; border-collapse: collapse;"> <tr><td style="padding: 2px 10px;">0</td></tr> <tr><td style="padding: 2px 10px;">1</td></tr> <tr><td style="padding: 2px 10px;">2</td></tr> <tr><td style="padding: 2px 10px;">3</td></tr> <tr><td style="padding: 2px 10px;">4</td></tr> </table> </td> <td style="font-size: 2em; vertical-align: middle;">)</td> </tr> </table>	(<table border="1" style="margin: auto; border-collapse: collapse;"> <tr><td style="padding: 2px 10px;">0</td></tr> <tr><td style="padding: 2px 10px;">1</td></tr> <tr><td style="padding: 2px 10px;">2</td></tr> <tr><td style="padding: 2px 10px;">3</td></tr> <tr><td style="padding: 2px 10px;">4</td></tr> </table>	0	1	2	3	4)	<table border="1" style="margin: auto; border-collapse: collapse;"> <tr><td style="padding: 2px 10px;">0</td></tr> <tr><td style="padding: 2px 10px;">0.25</td></tr> <tr><td style="padding: 2px 10px;">0.5</td></tr> <tr><td style="padding: 2px 10px;">0.75</td></tr> <tr><td style="padding: 2px 10px;">1</td></tr> </table>	0	0.25	0.5	0.75	1	<table border="1" style="margin: auto; border-collapse: collapse;"> <tr><td style="padding: 2px 10px;">-1</td></tr> <tr><td style="padding: 2px 10px;">-0.75</td></tr> <tr><td style="padding: 2px 10px;">-0.5</td></tr> <tr><td style="padding: 2px 10px;">-0.25</td></tr> <tr><td style="padding: 2px 10px;">0</td></tr> </table>	-1	-0.75	-0.5	-0.25	0
0																										
1																										
2																										
3																										
4																										
(<table border="1" style="margin: auto; border-collapse: collapse;"> <tr><td style="padding: 2px 10px;">0</td></tr> <tr><td style="padding: 2px 10px;">1</td></tr> <tr><td style="padding: 2px 10px;">2</td></tr> <tr><td style="padding: 2px 10px;">3</td></tr> <tr><td style="padding: 2px 10px;">4</td></tr> </table>	0	1	2	3	4)																			
0																										
1																										
2																										
3																										
4																										
0																										
0.25																										
0.5																										
0.75																										
1																										
-1																										
-0.75																										
-0.5																										
-0.25																										
0																										

Нетрудно заметить, что таблица значений ранжированной переменной дается под оператором ее вывода, а таблица значений элементов вектора — справа от оператора вывода. Это подчеркивает отличие ранжированной переменной от вектора. К тому же вектор представляется в скругленных скобках, тогда как таблица значений ранжированной переменной прямоугольная. Далее (следующий раздел) вы увидите, что вектор дает доступ к любому своему элементу, тогда как ранжированная переменная такого доступа не обеспечивает — попытка доступа сопровождается сообщением об ошибке.

ПРИМЕЧАНИЕ Ранжированная переменная не предоставляет полноценного средства замены циклов, обычно реализуемых с помощью языков программирования. Для создания циклов и других традиционных программных конструкций Mathcad имеет специальные средства. Они рассмотрены в главе 10, посвященной программированию в среде Mathcad.

1.3.9. Ввод матриц и векторов

Матрицы в виде двумерных массивов широко применяются при решении задач механики, электротехники, радиотехники и других областей науки и техники [28].

Матрицу можно представить как таблицу, имеющую m строк (rows) и n столбцов (columns). Если $m = n$, то матрицу именуют квадратной. Число элементов или размер матрицы есть $m \times n$. Вектор с длиной m рассматривается как одномерная матрица размера $m \times 1$. Для операций с матрицами, включая их ввод, служит палитра матричных операций Matrix (Матрицы). В ней, в частности, имеется шаблон для ввода матриц — он имеет вид удлинненных скругленных скобок с местами для ввода элементов матриц. При вводе шаблона появляется небольшое окно, запрашивающее число строк и столбцов матрицы. Рис. 1.22 иллюстрирует процедуру ввода матрицы и использование простейших векторных и матричных операций.

Элементы матриц являются индексированными переменными и характеризуются двумя индексами — номером элемента в строке и номером элемента в столбце. Например, $M_{i,j}$ означает элемент матрицы, расположенный в строке i и столб-



Рис. 1.22. Примеры простейших операций с матрицами

це j . Матрицы одного размера можно складывать и вычитать. Возможна замена строк на столбцы — операция транспонирования (значок M^T в палитре). Другие важные матричные операции, например обращение матрицы и решение матричных уравнений, мы рассмотрим позже.

Mathcad допускает отсчет индексов с 0 или 1. Это задается соответствующим значением системной переменной $ORIGIN$. По умолчанию $ORIGIN=0$. Для индексации векторов и матриц с 1 введите $ORIGIN:=1$. Если надо вернуться к индикации с 0, задайте $ORIGIN:=0$.

Элементы векторов и матриц можно задавать явно, присваивая индексированной переменной то или иное значение. Аналогично можно получать или выводить значение нужного элемента. Это хорошо иллюстрирует рис. 1.23.

Обратите особое внимание на последний пример на рис. 1.23. Здесь показана возможность создания матрицы внутри матрицы. Центральный элемент матрицы M задан как матрица $M2$, созданная ранее. Но отображается не она, а объект $\{2,2\}$, означающий, что это массив размера 2×2 . Его вызов на индикацию выводит массив в виде матрицы.

1.3.10. Запрет вычисления математических выражений

Математические выражения можно сделать пассивными. Для этого достаточно выделить выражение и выбрать команду *Disable Evaluation* (Запретить выполнение) в контекстном меню, появляющемся при щелчке на выражении правой кнопкой мыши (см. рис. 1.23). Выражение будет помечено значком черного прямоугольника и не будет исполняться ни в автоматическом, ни в ручном режимах вычислений. Такое выражение можно рассматривать просто как сложный комментарий в виде формулы. Для снятия с выражения статуса невычисляемого и неисполняемого достаточно выделить выражение и выбрать команду *Enable Evaluation* (Разрешить выполнение) в контекстном меню.

Позлементный ввод и вывод векторов и матриц

$k := 0..5$ $v_k := k^2$
 $v_1 = 1$ $v_2 = 4$
 $v_3 = 9$ $v_5 = 25$

$$v = \begin{pmatrix} 0 \\ 1 \\ 4 \\ 9 \\ 16 \\ 25 \end{pmatrix}$$

Вектор-столбец и вектор-строка с элементами - квадратами чисел от 0 до 5

$$v^T = (0 \ 1 \ 4 \ 9 \ 16 \ 25)$$

$i := 0..2$ $j := 0..2$
 $M0_{i,j} := 0$

$$M0 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Нулевая матрица

$M1_{i,i} := 1$

$$M1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Единичная матрица

$M2_{0,0} := 1$ $M2_{0,1} := 2$ $M2_{1,0} := 3$ $M2_{1,1} := 4$ $M2 = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$

$M_{1,1} := M2$

$$M = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \{2,2\} & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Матрица в матрице

$M_{1,1} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$

Вызов матрицы из матрицы

Рис. 1.23. Примерыazoleментного создания и вывода векторов и матриц

1.3.11. Прерывание вычислений

При задании сложных вычислений работа системы может быть долгой. Для ее прерывания можно нажать клавишу Esc. Mathcad выведет сообщение о прерывании вычислений и небольшое окно с двумя кнопками: ОК — подтвердить прерывание и Cancel (Отмена) — отменить прерывание. После прерывания можно возобновить работу, нажав клавишу F9 или щелкнув в панели инструментов на кнопке с изображением жирного знака равенства.

1.4. Введение в графику Mathcad

Практика показывает, что, едва освоив простейшие вычисления, пользователи системы Mathcad тут же пытаются строить графики простых функций, например отрезка прямой или синусоиды. Не будем томить изучающих Mathcad и рассмотрим простейшие приемы построения графики (подробно эта тема рассматривается в главе 4).

1.4.1. Построение двумерного графика одной функции

Простые вычисления в новых версиях Mathcad выполняются почти мгновенно, и это вполне естественно. Столь же просто можно построить графики функций самого различного вида. Нередко к ним сводятся нужные результаты вычислений.

Начнем с построения графика функции $\sin(x)^3$. Для этого достаточно выполнить следующие простые действия.

1. Введите функцию, набрав выражение $\sin(x)^3$.
2. В панели математических знаков щелкните на кнопке с изображением графика — на экране появится палитра графиков.
3. В палитре графиков щелкните на кнопке с изображением двумерного графика — на экране появится шаблон графика (рис. 1.24) с уже введенной по оси Y функцией.
4. Введите в место ввода шаблона по оси X имя независимого аргумента — x .
5. Щелкните вне пределов графика левой кнопкой мыши — график построен (рис. 1.25).

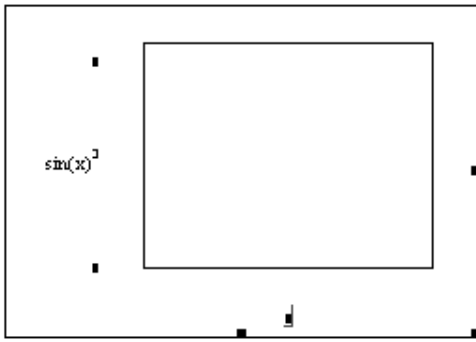


Рис. 1.24. Незаполненный шаблон двумерного графика

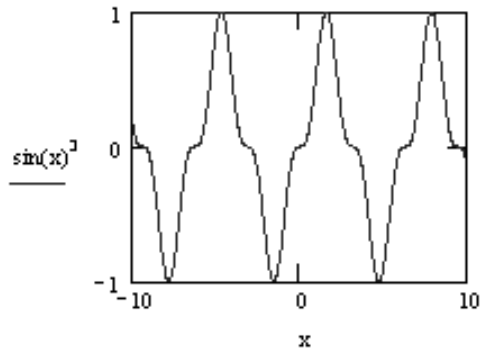


Рис. 1.25. Результат построения графика функции $\sin(x)^3$

Обратите внимание, что в данном случае не требуется даже полного выделения выражения для функции, но курсор ввода должен быть в блоке с этим выражением.

1.4.2. Изменение размеров и перемещение графика

Теперь, предположим, мы хотим увеличить график и немного сместить его. Для этого продумаем следующее.

1. Поместите указатель мыши в область графика и щелкните левой кнопкой мыши — вокруг графика появится рамка из черных линий, обрамляющая блок графика.
2. Подведите указатель мыши к черному квадратику (маркеру изменения размера) в правом нижнем углу рамки, при этом указатель мыши должен превратиться в двухстороннюю диагональную стрелку.
3. Нажав левую кнопку мыши, растяните график по диагонали.
4. Завершив расширение (или сужение) графика, отпустите кнопку мыши.
5. Наведите указатель мыши на любую сторону рамки (кроме черных квадратиков), при этом указатель должен превратиться в черную ладошку.
6. Нажав левую кнопку мыши, передвиньте весь блок графика в желаемое место экрана.
7. Завершив перемещение, отпустите кнопку мыши.

В результате этих действий мы увидим увеличенный и перемещенный в другое место экрана график с обрамляющей его рамкой (рис. 1.26). Убрать рамку можно, отведя указатель мыши в сторону от графика и щелкнув левой кнопкой мыши.

Обратите внимание: когда график находится в рамке, на нем в характерных черных уголках появляются числа, идентифицирующие масштаб графика по оси Y и по оси X. По умолчанию по оси X график строится на отрезке изменения аргумента x от -10 до $+10$. Масштаб по оси Y Mathcad устанавливает автоматически. Изменив эти числа, можно задать свой масштаб графика.

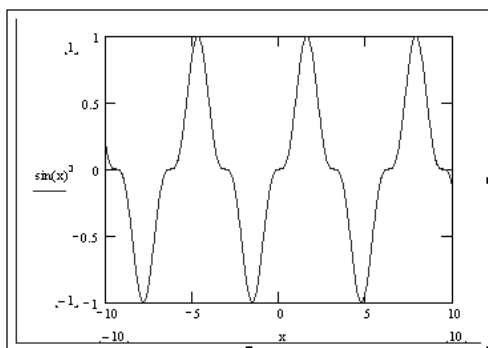


Рис. 1.26. График одной функции после изменения размеров

1.4.3. Построение графиков ряда функций

А теперь рассмотрим, как в полученном графике отобразить еще две функции, например $\sin(x)^2$ и $\cos(x)$. Для этого их надо просто перечислить после первой функции в месте ввода возле оси Y, отделяя выражения функций запятыми. Вот как это делается.

1. Поместите указатель мыши точно в конец выражения $\sin(x)^3$.
2. Щелкните левой кнопкой мыши — появится синий уголок в конце выражения (или перед ним).
3. Клавишами перемещения курсора по горизонтали переместите уголок точно в конец выражения и нажатием клавиши Пробел добейтесь, чтобы уголок охватил все выражение.
4. Введите знак запятой, при этом вы заметите, что первое выражение ушло вверх, а под ним появилось новое место ввода.
5. Введите выражение $\sin(x)^2$.
6. Нажимая клавишу Пробел, добейтесь, чтобы это выражение тоже было полностью охвачено синим уголком.
7. Введите знак запятой, и вы заметите, что два первых выражения ушли вверх, а под ними появилось новое место ввода.
8. Введите в новое место ввода выражение $\cos(x)$.
9. Отведите указатель мыши за пределы графика и щелкните левой кнопкой мыши — появится график с тремя кривыми (рис. 1.27).

Обратите внимание, что Mathcad автоматически отображает каждую кривую своим стилем и своим цветом. И делает это весьма недурно — редко кто из пользователей вмешивается в решение систе-

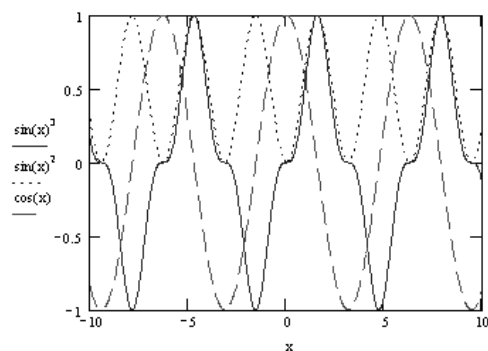


Рис. 1.27. Двумерный график с тремя кривыми

мы. Тем не менее путем форматирования графиков, которое мы рассмотрим в главе 5, вид графика можно менять. К примеру, выделив график, его можно растягивать или сжимать с помощью мыши, уцепившись ее указателем за специальные места на сторонах графика в виде темных маленьких прямоугольников (см. рис. 1.26).

1.4.4. Виды графиков

Об общем характере графиков, которые может строить система Mathcad, говорит рис. 1.28. На нем показана панель вставки графиков с пронумерованными кнопками, а в окне документа представлены соответствующие графики с их наименованиями.

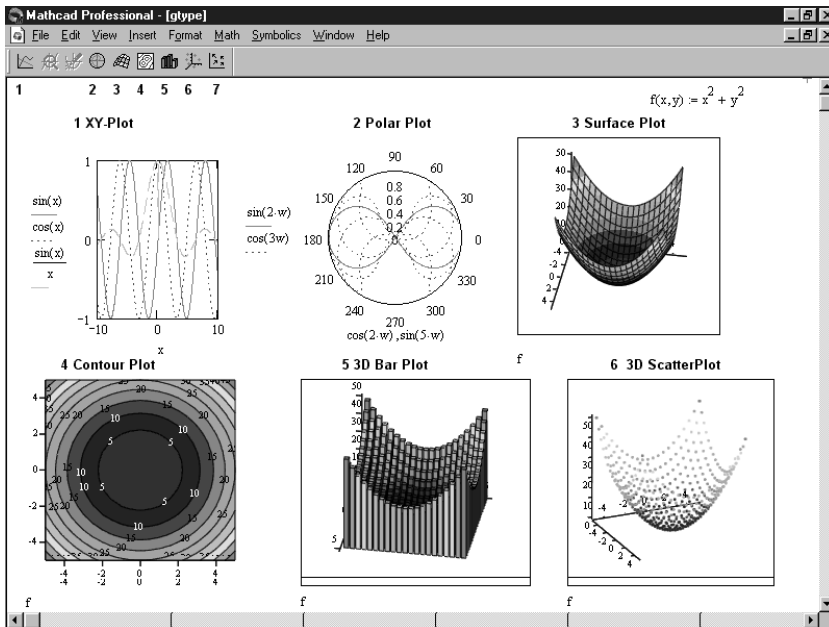


Рис. 1.28. Основные типы графиков системы Mathcad

1. XY-Plot — график функции одной переменной в системе координат Декарта.
2. Polar Plot — график в полярной системе координат.
3. Surface Plot — график поверхности (функции двух переменных).
4. Contour Plot — контурный график (линий равного уровня).
5. 3D Bar Plot — график поверхности в виде трехмерных столбиков.
6. 3D Scatter Plot — график поверхности, построенный точками.

Форматированием приведенных типов графиков можно получить множество их вариантов.

1.4.5. Построение графиков поверхностей

Построение графиков поверхностей (их называют также трехмерными или 3D-графиками) всегда было ахиллесовой пятой систем класса Mathcad. Связано это с тем, что такие графики даже в простейшем случае требуют создания матри-

цы точек (аппликат) зависимости $z(x, y)$, то есть функции двух переменных. Создание такой матрицы нетривиальная задача, что приводило не только к усложнению построений, но и к потере их наглядности. Позже мы рассмотрим построение графиков таким образом.

Однако в новейших версиях Mathcad 2001i/11/12 эта трудность блестяще преодолена. Теперь трехмерный график построить даже проще, чем двумерный.

1. Определите функцию $z(x, y)$ двух переменных x и y . В качестве примера рассмотрим график функции суммы квадратов ($x^2 + y^2$). Тогда определение функции будет выглядеть следующим образом:

$$z(x, y) := x^2 + y^2$$

2. Используя палитру графики, введите шаблон трехмерного графика (рис. 1.29).
3. На единственное место ввода под шаблоном введите имя функции — z .
4. Выведите указатель мыши за пределы графика и щелкните левой кнопкой мыши — будет построен график в виде «проволочного каркаса».
5. Растяните (или сожмите) график и поместите его в нужное место экрана (рис. 1.30).

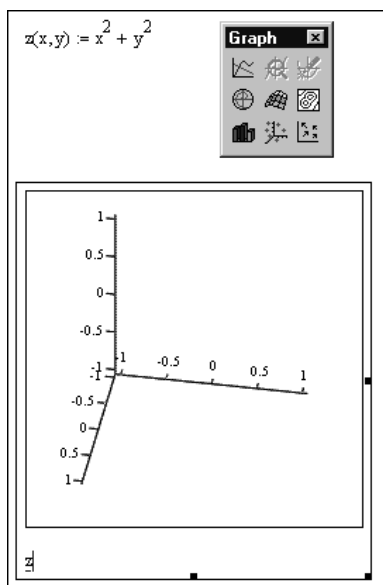


Рис. 1.29. Подготовка к построению графика поверхности

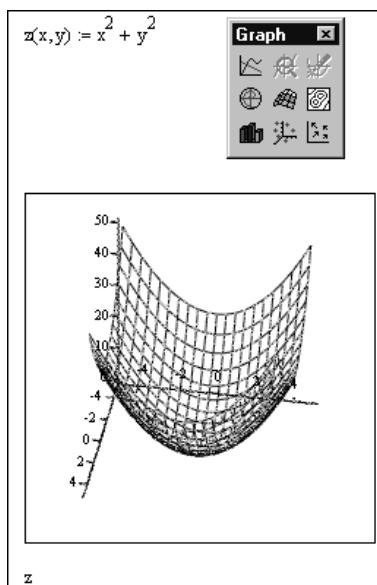


Рис. 1.30. График поверхности в виде «проволочного каркаса»

График в виде «проволочного каркаса» не слишком эстетичен, хотя иногда и полезен, поскольку сквозь каркас видны обычно невидимые детали. Для изменения вида графиков трехмерных поверхностей используется их форматирование, которому посвящена глава 5. Нетерпеливые читатели могут попытаться самостоятельно освоить форматирование — для них по секрету сообщим, что окно форматирования вызывается, если в область графика поместить указатель мыши и дважды щелкнуть левой кнопкой (рис. 1.31).

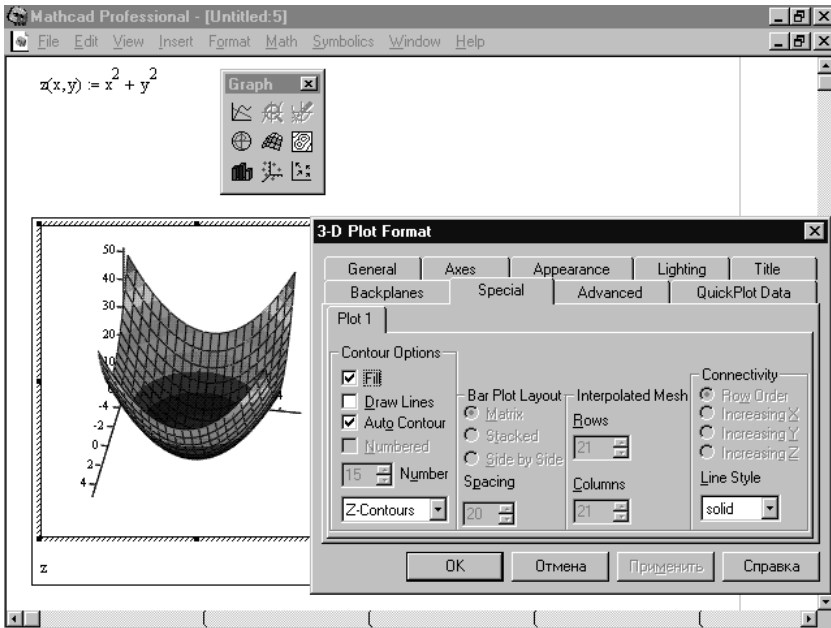


Рис. 1.31. График поверхности с функциональной окраской и окном форматирования

Сравните внимательно графики на рис. 1.30 и 1.31. Вы заметите, что на рис. 1.31 отсутствуют невидимые участки и использована функциональная окраска поверхности. Все это и достигается средствами форматирования графика.

Еще один вариант форматирования построенного нами графика представлен на рис. 1.32. В этом случае строятся контурные линии поверхности, которые образуются при пересечении ее рядом плоскостей, расположенных на одинаковом расстоянии друг от друга. Кроме того, рисунок иллюстрирует возможность растяжения графика поверхности мышью.

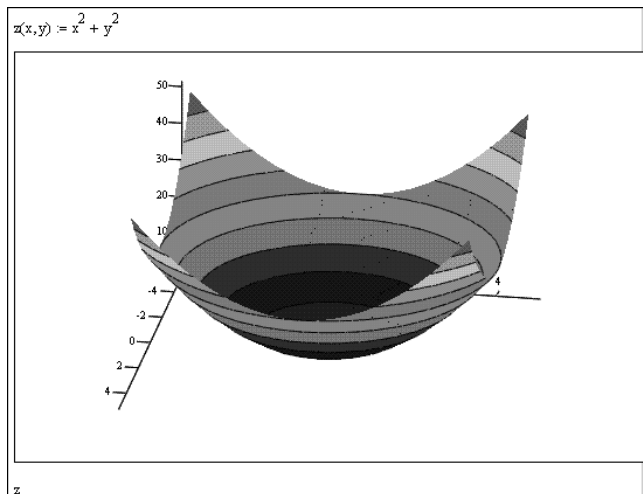


Рис. 1.32. График поверхности с функциональной окраской

1.4.6. Вращение трехмерного графика мышью

А теперь рассмотрим трюк с трехмерной графикой, впервые введенный в систему Mathcad 8 и, естественно, сохраненный в Mathcad 2001i/11/12. Он заключается во вращении в пространстве трехмерной фигуры с помощью мыши.

Для чего это нужно? Дело в том, что вид трехмерных фигур сильно зависит от того, под какими углами относительно осей X, Y и Z фигуру рассматривают. Нередко небольшие пики, впадины или лепестки фигуры просто не видны на фоне высоких пиков или глубоких впадин. Вращение фигуры эквивалентно ее просмотру с разных сторон. Вы можете заглянуть внутрь впадины или посмотреть, что находится за пиком.

Для этого и иных действий надо просто поместить указатель мыши в область графика, нажать левую кнопку мыши и, удерживая ее, начать перемещать мышью в том или ином направлении. Вы тут же увидите, что фигура вместе с осями координат и призмой, в которой она находится, начнет вращаться в ту или иную сторону. Один из таких моментов зафиксирован на рис. 1.33. Сравните его с рис. 1.32, и вам станет ясно, насколько поворот даже такой простой фигуры (пространственной параболы) влияет на вид графика.

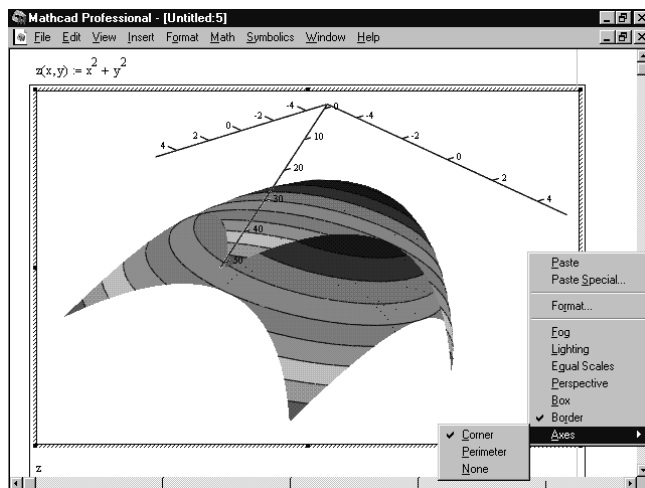


Рис. 1.33. Вид фигуры, показанной на рис. 1.32, после ее поворота мышью

Однако поворотом фигуры в пространстве возможности этого метода не исчерпываются. Если оперировать мышью при нажатой клавише Ctrl, можно удалять объект от наблюдателя или приближать объект к наблюдателю. Если проделать те же действия с нажатой клавишей Shift, то после отпускания левой кнопки можно вообще наблюдать анимированную («живую») картину вращения объекта в любом заданном предварительно направлении. Для остановки вращения надо щелкнуть левой кнопкой мыши.

Конечно, скорость вращения зависит от быстродействия ПК. Компьютеры с новыми процессорами Celeron, Pentium III и Pentium 4 позволяют добиться весьма быстрого и эффектного вращения или перемещения фигуры, тогда как ПК с устаревшими процессорами Celeron и Pentium II делают то же с заметными рывками. Тем не менее и в этом случае можно быстро подобрать наиболее удачный «ракурс» для просмотра фигуры.

Как уже говорилось, при нажатой клавише Ctrl перемещение указателя мыши вверх или вниз ведет к приближению или удалению от нас наблюдаемой картины, то есть к изменению масштаба изображения. Это тоже удобный способ получить изображение наилучшего вида.

На рисунке показано также контекстное меню, появляющееся при щелчке правой кнопкой мыши в области графика. С помощью этого меню можно осуществлять различные операции форматирования графиков, например, смену типа осей, ввод перспективы при отображении графиков и т. д.

1.4.7. Построение на одном трехмерном графике ряда поверхностей

Еще один приятный сюрприз новых версий Mathcad — возможность построения на одном графике ряда поверхностей. Начиная с Mathcad 2000 это делается предельно просто — определите ряд функций двух переменных, описывающих поверхности, и введите через запятую имена этих функций в месте ввода шаблона трехмерного графика. Построим график двух объемных парабол, соприкасающихся вершинами, по следующим их формулам:

$$\begin{aligned} z_1(x, y) &:= x^2 + y^2 \\ z_2(x, y) &:= -(x^2 + y^2) \end{aligned}$$

Как уже упоминалось, единственное место ввода трехмерного графика заполняется следующим образом:

$$z_1, z_2$$

Выполнив ввод этих блоков, отведите курсор мыши в свободную часть документа и щелкните левой клавишей мыши — будет построен график, представленный на рис. 1.34.

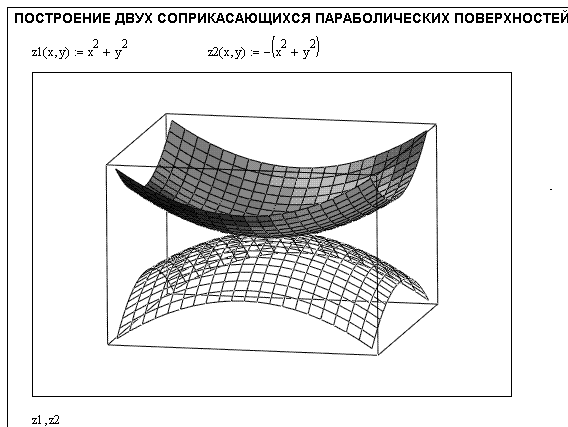


Рис. 1.34. Построение двух объемных парабол, соприкасающихся вершинами

Заглядывая вперед, отметим, что каждую фигуру можно форматировать отдельно, что и видно из рисунка. В данном случае фигуры не пересекаются, а лишь соприкасаются. В случае пересекающихся фигур Mathcad автоматически определяет линии пересечения и обеспечивает реализацию алгоритма удаления невидимых (вследствие пересечения) частей фигур. На рис. 1.35 показан такой случай — для иллюстрации пересечения парабол одна из них поднята по вертикали, а другая опущена. Присмотритесь к записи функций $z_1(x, y)$ и $z_2(x, y)$:

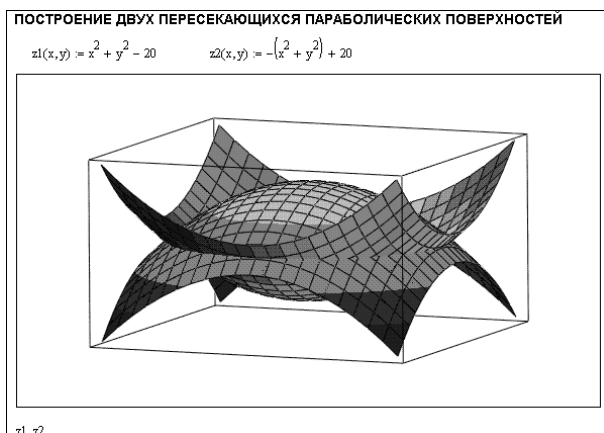


Рис. 1.35. Построение двух пересекающихся объемных парабол

$$z1(x, y) := x^2 + y^2 - 20$$

$$z2(x, y) := -(x^2 + y^2) + 20$$

В дальнейшем мы рассмотрим графические возможности Mathcad более детально, но и приведенные примеры дают достаточное представление о простоте работы с графическим программным процессором этой системы.

1.4.8. Одновременное построение разных типов трехмерных графиков

Графика систем Mathcad позволяет одновременно строить графики разных типов. Для этого достаточно задать выражения для графиков, перечислить их в шаблоне с разделением запятыми и отформатировать графики под нужный их тип. Пример этого представлен на рис. 1.36.

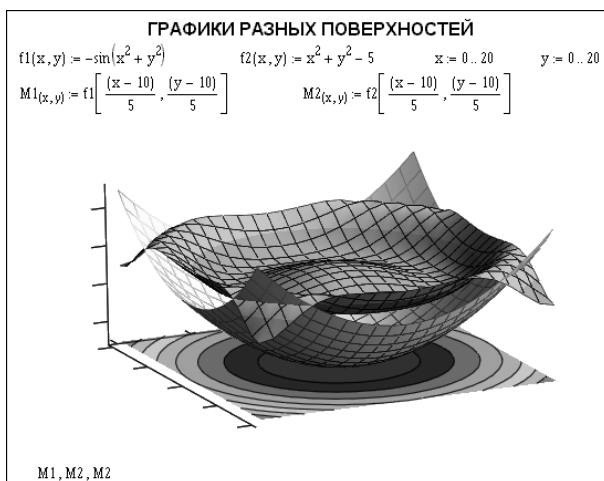


Рис. 1.36. Построение двух пересекающихся поверхностей и контурного графика одной из них

Здесь заданы матрицы аппликат двух поверхностей $M1$ и $M2$. Это соответствует «старому» стилю построения поверхностей. Затем эти матрицы указаны внизу шаблона графиков. Матрица $M2$ указана дважды — первый раз она отформатирована под построение поверхности, а второй раз под построение контурного графика поверхности. Так именуют график линий пересечения поверхности рядом равноудаленных друг от друга плоскостей (график линий равного уровня). Для параболы такой график выглядит как концентрические окружности, расположенные на плоскости проекции фигуры. Графики подобного комбинированного типа отличаются повышенной наглядностью.

1.4.9. Трехмерные гистограммы и визуализация матриц

Графики типа 3D Bar Plot могут использоваться для построения трехмерных гистограмм (столбиковых графиков) и визуализации содержимого матриц. Пример этого представлен на рис. 1.37.

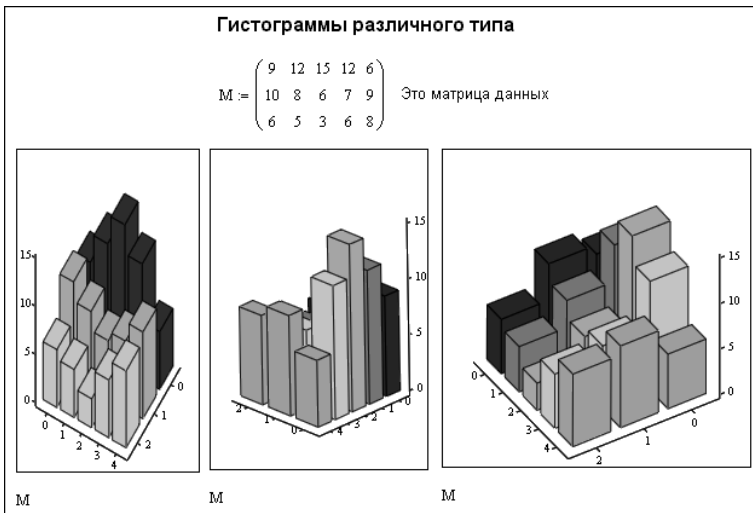


Рис. 1.37. Построение трехмерных гистограмм по данным матрицы M

Вращая график, можно менять ориентацию набора столбиков такой диаграммы. При форматировании можно также менять цветовую гамму окраски столбиков.

1.4.10. Двумерные графики с двумя осями Y в Mathcad 12

К новинкам типа «приятные мелочи» в Mathcad 12 относится построение двумерных графиков с двумя осями Y . Такие графики дают следующие преимущества:

- при построении двух графиков они проще распознаются, поскольку их функции указываются раздельно слева и справа от области графика;
- при построении графиков множества функций можно уменьшить вдвое число их имен и легенд;
- удобно совмещать графики при разных масштабах по вертикальной оси.

Для построения графиков с двумя осями Y достаточно вывести обычный шаблон двумерного графика, ввести в него маркер мыши и двойным щелчком мыши вызвать окно форматирования — рис. 1.38.

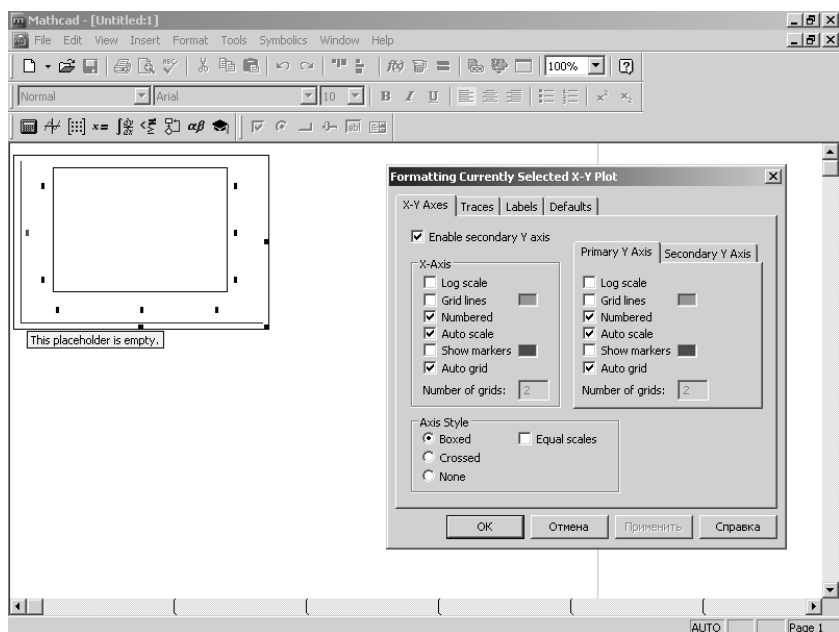


Рис. 1.38. Подготовка к созданию графика с двумя осями Y

На вкладке X-Y Axes можно узреть некоторые различия в сравнении с видом этой вкладки в более ранних версиях Mathcad. Так, на ней присутствует новая опция Enabled secondary Y axis. По умолчанию эта опция отключена и можно строить обычные графики с одной осью Y, расположенной слева. Но если включить эту опцию, то в шаблоне двумерной графики появятся новые места ввода в правой части (шаблон графика показан на рис. 1.38 в левой части окна документа).

Установив опцию, можно ввести в левую и правую части графика отображаемые на нем функции. На рис. 1.39 показано задание в левой части графика функции $\sin(x)/x$, а в правой части функций x^3 и x^2 . В результате все три графика функций строятся в окне графика. При этом для каждого (левого и правого) графиков устанавливается свой масштаб по умолчанию и свое расположение координатных осей.

На вкладке XY Axis есть еще две приятные мелочи — прямоугольники против опций построения сетки Grid и маркера Show Marker с указанием их текущего цвета (по умолчанию он, соответственно, зеленый и красный). Щелчок мыши по прямоугольнику открывает панель смены цвета, которая видна на рис. 1.39 в правом нижнем углу окна документа. С помощью этой панели (щелчком мыши по прямоугольнику с нужным цветом) можно сменить цвет линий масштабной сетки и маркера.

На рис. 1.40 показан окончательно построенный и расширенный график с двумя осями Y. В конце графика виден крестообразный маркер. Обратите внимание на то, что некоторые шероховатости в построении таких графиков есть — в частности, не всегда удачна оцифровка осей.

С помощью панели форматирования (рис. 1.38 и 1.39) можно выполнить форматирование графика, например изменить цвет линий, их стиль и толщину, добавить в график обозначения кривых — легенды, титульную надпись и т. д. Подробно операции с графиками и их форматирование описаны в главах 4 и 5.

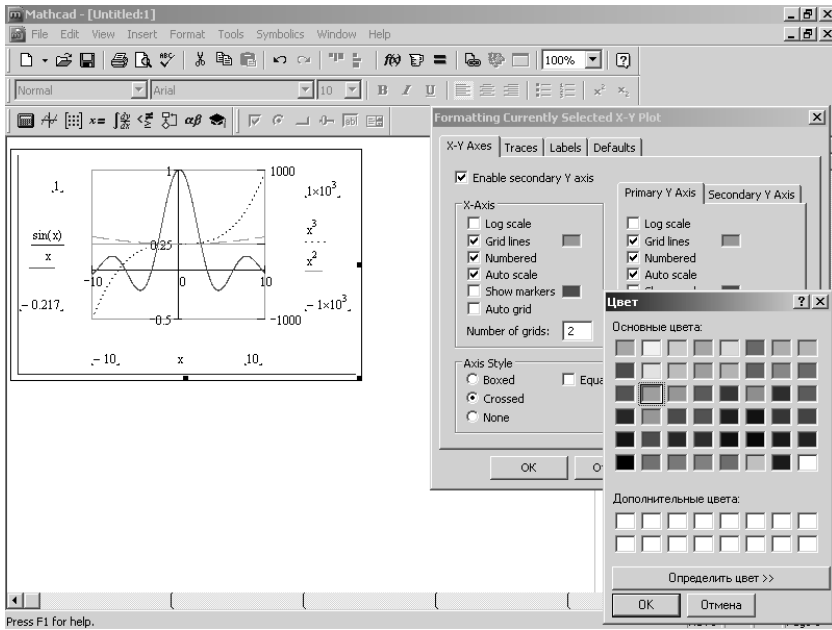


Рис. 1.39. Построение графика трех функций с двумя осями Y

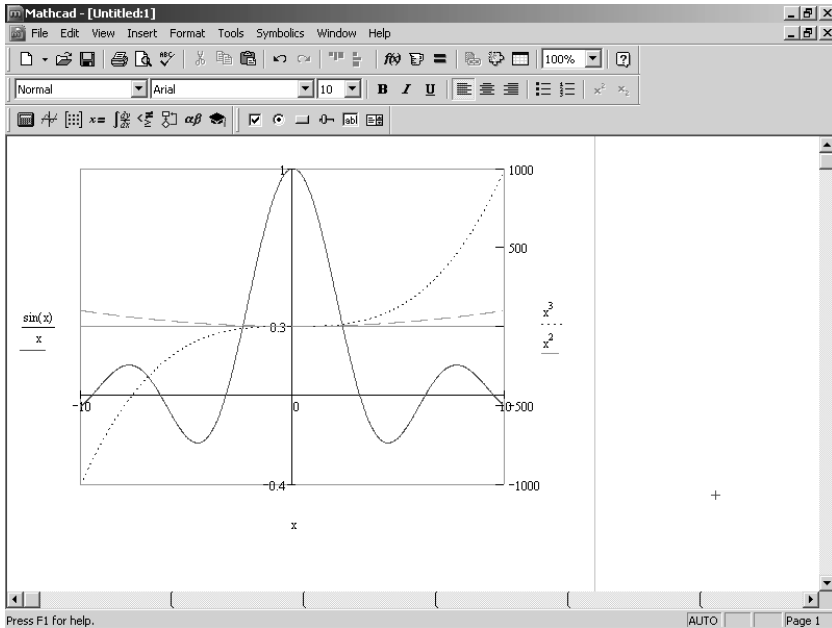


Рис. 1.40. Окончательно построенный график трех функций с двумя осями Y

Глава 2. Интерфейс пользователя

В этой главе мы продолжим знакомство с системами компьютерной математики Mathcad, начатое в предыдущей главе, однако в несколько ином ключе. Основная часть главы 1 была посвящена обзору общих возможностей, здесь же мы остановимся на углубленном изучении и применении интерфейса пользователя программы, то есть на средствах, посредством которых пользователям удастся реализовать эти и многие другие возможности системы Mathcad. При этом, учитывая множество мелких отличий в интерфейсе трех последних версий системы Mathcad 2001i/11/12, особое внимание уделим этим отличиям.

2.1. Работа с окном документа

2.1.1. Окна документов систем Mathcad 2001i/11/12

Интерфейс системы Mathcad во многом напоминает интерфейс популярных программ из пакета Microsoft Office [42—44]. Однако было бы серьезной ошибкой отождествлять его с интерфейсом программ Microsoft Office, например с общеизвестным текстовым редактором Word. На самом деле интерфейс систем Mathcad содержит массу тонких моментов, знание которых и определяет профессиональное применение этих систем.

Сразу после запуска система Mathcad 2001i готова к созданию документа. Первая же кнопка (с изображением чистого листка бумаги) на панели инструментов позволяет создать новый документ. Соответствующее ему окно редактирования получает название *Untitled: N*, где *N* — порядковый номер документа. Вначале окно редактирования очищено от данных (рис. 2.1).

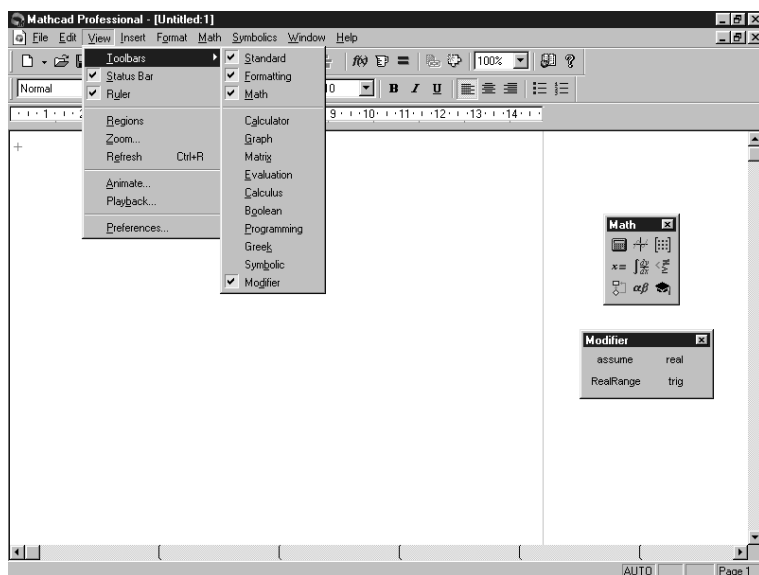


Рис. 2.1. Окно Mathcad 2001i сразу после запуска

На рис. 2.1 показана открытая позиция View меню, которая задает вывод или скрытие основных элементов интерфейса. Для вывода какого-либо элемента интерфейса надо мышью установить знак птички напротив его имени, а для скрытия — снять этот знак.

Окно систем Mathcad 11/12 представлено на рис. 2.2. С первого же взгляда бросается в глаза сходство интерфейса этой реализации с интерфейсом предшествующей версии. Но есть и отличия. Пожалуй, главным видимым отличием является наличие в меню новой позиции Tools (Инструменты) вместо позиции Math. Кроме того, в позиции View (на рис. 2.2 она также открыта) в списке панелей инструментов (Toolbars) присутствуют две новые панели — Controls (панель инструментов контроля) и Resources (панель Центра ресурсов). Обе эти панели показаны на рис. 2.2 в правой части окна под панелью палитр математических объектов.

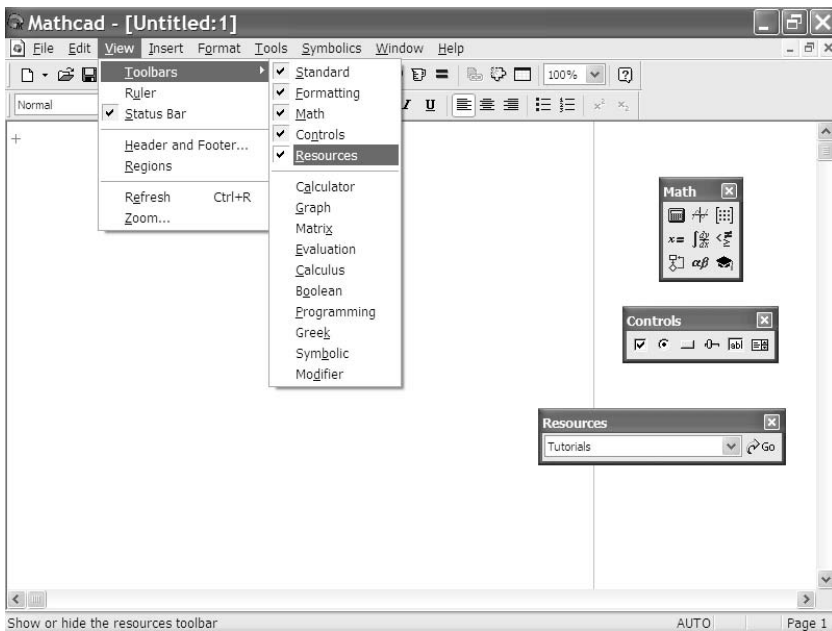


Рис. 2.2. Окно Mathcad 11/12 сразу после запуска

Ниже перечислены некоторые характерные элементы интерфейса окна любой из рассматриваемых версий Mathcad, которые можно увидеть на рисунке:

- *строка заголовка* — строка с именем системы и текущего документа, а также с кнопками управления окном системы;
- *строка меню* — строка, открывающая доступ к пунктам меню с различными командами;
- *панель инструментов* — панель с кнопками (значками), обеспечивающими быстрое исполнение наиболее важных команд при работе с системой;
- *панель форматирования* — панель с кнопками (значками), обеспечивающими быстрое форматирование текстовых и формульных блоков в документах;
- *панель вывода палитр математических знаков* — панель с кнопками (значками), выводящими палитры специальных математических знаков и греческих букв;

- *строка состояния* — строка контекстной справки по работе с системой, отображающая текущее состояние последней;
- *полосы прокрутки* — элементы управления, расположенные вдоль нижней и правой границ текущего окна и предназначенные для прокрутки изображения на экране по горизонтали и/или вертикали;
- *курсор ввода* — курсор (маркер) в виде крестика (красного на цветном дисплее), намечающий место ввода блока документа;
- *линия раздела страниц* — вертикальная линия, отделяющая текущую страницу от соседней справа;
- *всплывающая подсказка* — небольшое текстовое поле (желтое на цветном дисплее), появляющееся при наведении указателя мыши на многие (но не все) элементы интерфейса и блоки в окне редактирования (на рисунке всплывающая подсказка относится к кнопке вывода палитры символьных операций);
- *координатная линейка* — линейка с делениями, позволяющая (если это нужно) точно располагать блоки по горизонтали.

СОВЕТ *Отдельные детали интерфейса можно скрывать или выводить на экран с помощью команд меню View (Вид). Это полезно для освобождения полезной площади экрана (см. ниже раздел «Управление видом документа»).*

2.1.2. Курсор ввода и линия раздела страниц

Внимательно присмотритесь к окну редактирования (см. рис. 2.1, 2.2). Даже когда окно очищено, оно оказывается не совсем пустым. Обычно на нем можно увидеть два важных объекта: курсор (маркер) ввода в виде крестика (красного на цветном дисплее) и вертикальную линию, отделяющую текущую страницу от соседней справа. Положение линии определяется заданными по умолчанию параметрами страницы документа. А вот курсор ввода устанавливается мышью. Обычно изначально он находится в левом верхнем углу окна документа. Для его установки в другое место подведите туда указатель мыши и щелкните левой кнопкой (то есть быстро нажмите и отпустите ее). Курсор ввода переместится в новое место. Этот курсор намечает места ввода блоков текущего документа.

ПРИМЕЧАНИЕ *В Mathcad 2001i/11/12 убрать вертикальную линию раздела страниц можно, если в окне настройки параметров страницы задать альбомную ориентацию страниц, установив переключатель Landscape (Альбомная).*

2.1.3. Строка заголовка

Строка заголовка находится в верхней части окна Mathcad. Она отображает название загруженного или вводимого с клавиатуры документа. В левой части строки имеется стандартная кнопка управления окном, а в правой части — три маленькие кнопки для свертывания окна, развертывания его во весь экран и закрытия. Уцепившись указателем мыши за строку заголовка и удерживая левую кнопку мыши, можно перемещать окно (если оно не раскрыто на весь экран) по рабочему столу Windows.

Если используется русифицированная версия Windows, то щелчок на левой кнопке строки заголовка выведет на экран меню с русскими названиями команд (см. ниже раздел «Управление окнами»). Пользователям англоязычной версии

Mathcad не стоит этому удивляться. Дело в том, что эти кнопки типичны для всех Windows-приложений. Они относятся не к Mathcad, а к операционной системе Windows.

ПРИМЕЧАНИЕ В дальнейшем предполагается, что система Mathcad установлена в локализованной (русифицированной) версии Windows, поэтому не удивляйтесь, что некоторые меню и окна диалога, показанные на рисунках, представляют собой причудливую смесь русских и английских названий.

2.1.4. Строка и состав меню

В строке меню системы Mathcad 2001i (рис. 2.1) представлены следующие пункты меню:

- File (Файл) — работа с файлами, Интернетом и электронной почтой;
- Edit (Правка) — редактирование документов;
- View (Вид) — изменение способов представления документа и скрывание/отображение элементов интерфейса;
- Insert (Вставка) — вставка объектов и их шаблонов (включая графику);
- Format (Формат) — изменение формата объектов;
- Math (Математика) — управление процессом вычислений;
- Graphics (Графика) — работа с графическим редактором;
- Symbolics (Символьные вычисления) — выбор операций символьного процессора;
- Window (Окно) — управление окнами системы;
- Help (Справка) — работа со справочной базой данных о системе, центром ресурсов и электронными книгами.

В меню Mathcad 11/12 исчезла позиция Math и вместо нее появилась новая позиция:

- Tools (Инструменты) — управление процессом вычислений.

Меню Mathcad — контекстные. Это значит, что число позиций в них и их назначение зависят от состояния системы. Указанные выше пункты меню характерны для рабочего состояния, когда идет редактирование документа. На рис. 2.3 показан вид части окна с видом меню, который оно имеет при всех закрытых окнах. Нетрудно заметить, что при этом осталось только три позиции меню File, View и Help (в Mathcad 11 к ним добавляется еще и позиция Tools).

Для активизации строки меню без применения мыши достаточно нажать и отпустить клавишу Alt. После этого, нажимая клавиши перемещения курсора → и ←, можно перемещать световое выделение по позициям меню. Клавиши ↓ и ↑ позволяют перемещаться по позициям меню. Выбрав нужную позицию, для ее активизации достаточно нажать клавишу Enter.

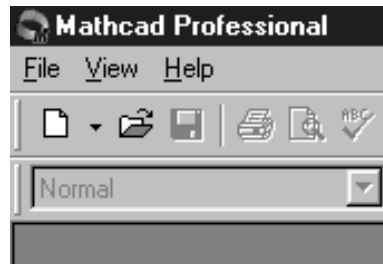


Рис. 2.3. Фрагмент окна Mathcad 2001i сразу после запуска

Еще один быстрый путь открытия нужного меню — нажатие клавиши **Alt** и одновременно клавиши с буквой, которая подчеркнута в названии нужного пункта меню. Для всех пунктов строки меню это начальная буква их имени. Важно не путать регистры смены языков и пользоваться только латинскими буквами.

В раскрытом меню обычно представлен список доступных и недоступных в данный момент команд. Доступные команды выведены четким шрифтом, а недоступные — шрифтом с характерным затенением, позволяющим все же прочесть их названия. Перемещение в меню и выбор нужной команды производится аналогично тому, как это описано для строки меню, с той лишь разницей, что при управлении с клавиатуры используются клавиши перемещения курсора по вертикали.

2.1.5. Строка состояния

В нижней части окна системы (рис. 2.1 и 2.2) имеется строка состояния. Она является элементом контекстной оперативной справки по работе с системой, поскольку отображает текущее состояние последней (например, режим вычисления и номер страницы документа). Если система не выполняет действий, то в этой панели будет надпись **Press F1 for help** (Нажмите клавишу **F1** для получения справки). Если это сделать, то будет выведено окно справочной системы.

2.1.6. Полосы прокрутки

Полосы (линейки) прокрутки вдоль нижней и правой границ текущего окна предназначены для прокрутки изображения на экране по горизонтали и/или вертикали. Для этого достаточно установить указатель мыши на кнопку со стрелкой, указывающей направление перемещения и нажать левую кнопку мыши. При этом будет обеспечена плавная, но медленная прокрутка изображения в окне. Можно прокручивать изображение намного быстрее, поместив указатель мыши внутрь полосы и также нажав кнопку мыши. При этом длина всей полосы соответствует (в некотором масштабе) полной длине документа, так что мышью можно сразу приблизительно указать на ту часть документа, которую требуется вывести в окно. Еще один способ прокрутки обеспечивает бегунок (квадратик на полосе, расположенный между стрелками), уцепившись за который можно прокручивать документ с максимальной скоростью.

2.1.7. Всплывающие подсказки

Mathcad имеет тщательно выполненную систему оперативной справки. Одним из ее элементов являются всплывающие подсказки — небольшие текстовые поля, появляющиеся при наведении указателя мыши на многие (но не все) элементы интерфейса и блоки в окне редактирования. Подсказки появляются в прямоугольнике желтого цвета (см. рис. 2.4, на котором всплывающая подсказка относится к кнопке вывода палитры вычислений Calculator Toolbar).

Функционирование всплывающих подсказок обеспечивается не системой Mathcad самой по себе, а операционной системой Windows. Поэтому текст подсказки может быть на русском языке,

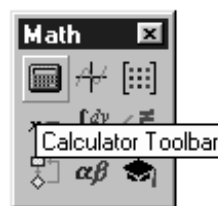


Рис. 2.4. Панель математических палитр с всплывающей подсказкой

если он относится к типовой детали интерфейса русифицированной версии Windows. Для деталей интерфейса Mathcad он будет, естественно, на английском языке. «Пройдясь» указателем мыши по всем кнопкам любой панели, можно ознакомиться с их названиями.

2.1.8. Контекстное меню правой клавиши мыши

Полезно помнить, что щелчок правой кнопкой мыши в окне Mathcad вызывает появление контекстного меню. Команды этого меню могут сильно отличаться в зависимости от того, в каком режиме работает система и в каком месте установлен указатель мыши перед щелчком. Например, контекстное меню, показанное на рис. 2.5, относится к математическому выражению в текущем документе. Заметим, что контекстное меню характерно для всех приложений Windows. Набор команд контекстного меню самый разный — в зависимости от того, к какому объекту это меню относится.

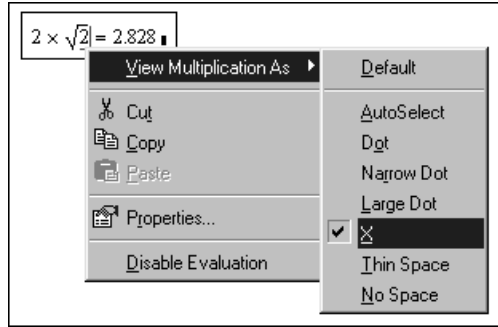


Рис. 2.5. Контекстное меню правой клавиши мыши для выражения

2.1.9. Представление операции умножения

При записи выражений и формул часто используется знак умножения. В литературе, в том числе математической, нет единых правил записи этого знака. Он может быть представлен точкой, пробелом, знаком \times и т. д. Контекстное меню правой клавиши мыши для выражения (рис. 2.5) позволяет установить разную форму оператора умножения. Следует учитывать, что эта форма относится только к этому выражению. По умолчанию оператор умножения задается как точка.

2.2. Работа с панелью инструментов

2.2.1. Обзор панели инструментов

Третью строку окна системы занимает панель инструментов (Toolbox). Она содержит несколько групп кнопок управления, каждая из которых дублирует наиболее важные команды меню. Глядя на эти кнопки, можно легко уяснить их функции. Стоит только остановить указатель мыши на любой из них, как появится всплывающая подсказка с именем этой кнопки.

В начале каждой из трех описанных выше панелей (панели инструментов, панели форматирования и панели вывода палитр математических знаков) имеется характерная вешка перемещения в виде выпуклой вертикальной черты. Уцепившись за нее указателем мыши и удерживая левую кнопку, можно перетащить ту или иную панель в любое место окна редактирования или «прилепить» ее к любой стороне окна. Это иллюстрирует рис. 2.6, на котором панель вывода палитр математических знаков «приклеена» к левой стороне окна редактирования, а панели инструментов и форматирования выведены в центр окна редактирования.

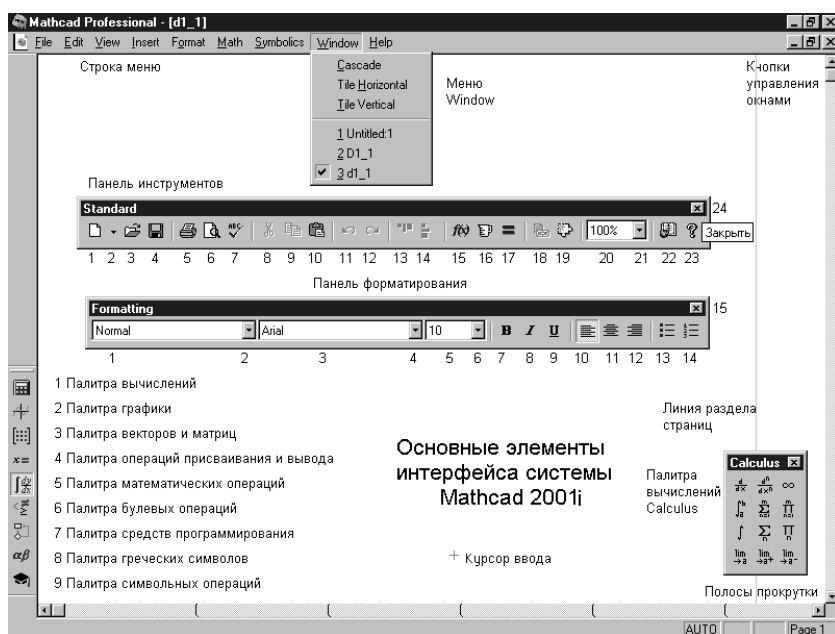


Рис. 2.6. Основные элементы интерфейса Mathcad 2001i

Панель инструментов можно переместить в любое место экрана, сделав «плавающей» (на рисунке плавающими являются панели инструментов и форматирования). Плавающую панель можно закрыть с помощью кнопки закрытия (всплывающая подсказка на рисунке относится именно к кнопке закрытия панели инструментов). Для прикрепления панели к любой стороне окна документа достаточно мышью подвести панель к этой стороне и наблюдать «растекание» панели по этой стороне. Для открепления панели надо дважды щелкнуть мышью на ее вешке. А для прикрепления плавающей панели достаточно дважды щелкнуть мышью на ее титульной строке.

Большинство пользователей вполне удовлетворены тем набором кнопок, которые есть на панелях инструментов. Но при желании можно воспользоваться средствами Windows по модификации панелей инструментов. Для этого достаточно установить курсор мыши на панели и щелкнуть правой клавишей мыши. В появившемся контекстном меню будет команда Customize (настроить). Ее исполнение выведет окно настройки панели, с помощью которого можно изменить состав средств панели.

Таким образом, интерфейс системы можно модифицировать, подстраивая его под свой вкус. Рассмотрим назначение кнопок на панели инструментов.

2.2.2. Файловые операции с панели инструментов

Документы системы Mathcad являются файлами, то есть именованными блоками хранения информации, содержащимися на магнитных дисках. Файлы можно создавать, загружать (открывать), записывать и распечатывать на принтере. Соответственно, файловые операции представлены на панели инструментов первой группой из четырех кнопок (см. рис. 2.6, на котором номера кнопок соответствуют их позициям на панели инструментов):

1. New (Создать) — создание нового документа с очисткой окна редактирования.
2. New (Создать) — открытие списка стилей создаваемого документа.
3. Open (Открыть) — загрузка ранее созданного документа путем выбора его в списке диалогового окна открытия файла.
4. Save (Сохранить) — запись текущего документа с его текущим именем.

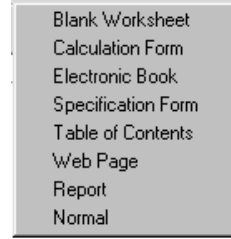


Рис. 2.7. Список стилей документов Mathcad 2001i

Кнопка 1 New по существу загружает шаблон пустого документа со стилем Normal (Обычный). Список 2 New позволяет выбрать стиль документа (рис. 2.7). Аналогичный список для Mathcad 11/12 показан на рис. 2.8.

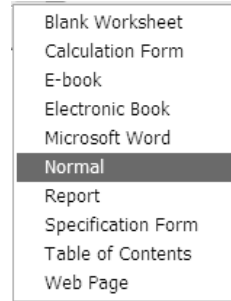


Рис. 2.8. Список стилей документов Mathcad 11/12

Кнопка Open открывает стандартное окно открытия (загрузки) файла (рис. 2.9). С помощью раскрывающегося списка Look In (Папка) в окне загрузки можно «путешествовать» по файловой системе компьютера, добираясь до самых удаленных ее уголков.

Выбрав нужный файл, для его загрузки достаточно щелкнуть на кнопке Open (Открыть). На рис. 2.9 виден открытый список типов файлов, которые можно загружать из окна Open.

Окно открытия файлов является стандартным окном Windows. Поэтому его вид зависит от конкретной версии этой операционной системы. Так, на рис. 2.10 представлено окно открытия файла системы Mathcad 11, работающей в среде операционной системы Windows XP.

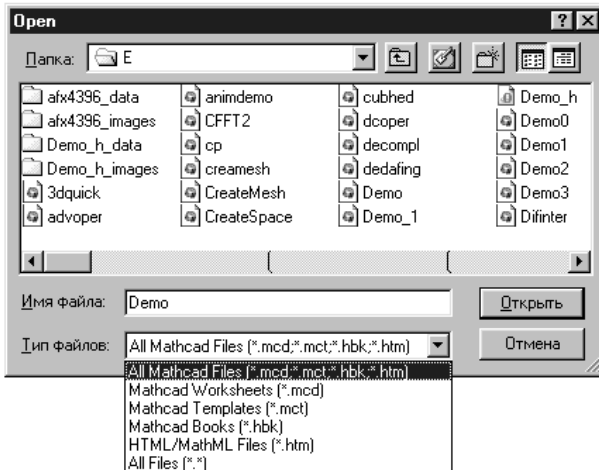


Рис. 2.9. Окно открытия файла Mathcad 2001i в среде Windows 98

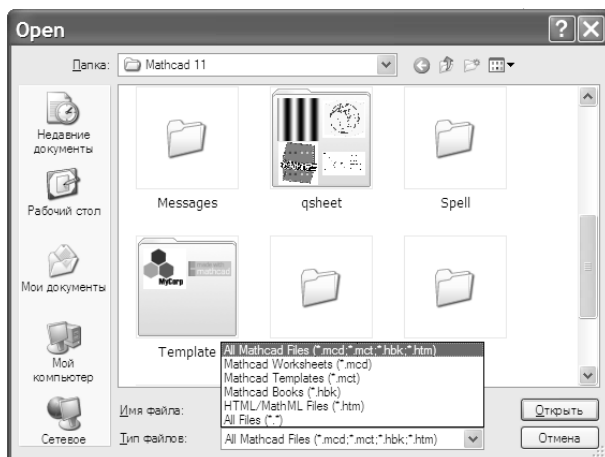


Рис. 2.10. Окно открытия файла Mathcad 11 в среде Windows XP

Как нетрудно заметить, это окно для приложений Windows XP существенно улучшено. В частности, его компоненты имеют более наглядный вид, а папки, содержащие графические файлы, представлены несколькими картинками из этих файлов. Однако принципиальные возможности окон, показанных на рис. 2.9 и 2.10, одинаковы.

Щелчок на кнопке Save внешне никак не проявляется. Происходит запись открытого документа со всеми внесенными вами изменениями под его текущим именем.

2.2.3. Печать и просмотр документов

Группа печати и контроля документов представлена тремя кнопками (см. рис. 2.6):

5. Print (Печать) — распечатка документа на принтере.
6. Print Preview (Просмотр) — предварительный просмотр документа.
7. Check Spelling (Проверка) — проверка орфографии в документе.

Назначение кнопок довольно очевидно, а операции печати и предварительного просмотра документов мы рассмотрим в следующей главе более подробно. Отметим лишь, что проверка орфографии действует только для англоязычных документов. Так что для русскоязычных пользователей эта кнопка по существу бесполезна.

2.2.4. Операции редактирования в панели инструментов

Во время подготовки документов их приходится видоизменять и дополнять — редактировать. Следующие три кнопки служат для выполнения наиболее типовых операций редактирования документов:

8. Cut (Вырезать) — перенос выделенной части документа в буфер обмена (clipboard) с очисткой этой части документа.
9. Copy (Копировать) — перенос выделенной части документа в буфер обмена с сохранением выделенной части документа.

10. Paste (Вставить) — перенос содержимого буфера обмена в окно редактирования на место, в котором находится курсор ввода.

Эти кнопки работают через буфер обмена Windows (clipboard), возможности и назначение которого хорошо известны пользователям Windows. Он предназначен для временного хранения данных и их переноса из одной части документа в другую либо для организации обмена данными между различными приложениями. Команда Cut (дублируется клавишей F3) вырезает выделенную часть документа (в виде отдельного объекта или группы объектов) и помещает ее в буфер. При этом эта часть документа из него удаляется. Команда Copy копирует выделенную часть документа в буфер и сохраняет ее в документе. А команда Paste (дублируется клавишей F4) помещает содержимое буфера по месту расположения курсора ввода.

Для выполнения операций редактирования предназначены еще две кнопки:

11. Undo (Отменить) — отмена предшествующей операции редактирования.
12. Redo (Повторить) — повторение ранее отмененной операции.

Полноценная работа с этими операциями реализована только в системах Mathcad 11/12. В предшествующих версиях Mathcad возможности этих операций были ограничены.

2.2.5. Управление размещением блоков

Здесь уместно отметить, что документы состоят из различных блоков: текстовых, формульных, графических и т. д. Блоки просматриваются системой, интерпретируются и исполняются. Просмотр идет слева направо и сверху вниз. Две операции размещения блоков представлены кнопками следующей группы (см. рис. 2.6):

13. Align Across (Выводит по горизонтали) — блоки выравниваются по горизонтали.
14. Align Down (Выводит вниз) — блоки выравниваются по вертикали, располагаясь сверху вниз.

Значки на этих кнопках изображают блоки и указанные варианты их размещения. Определенное размещение блоков в некоторых случаях имеет значение. Например, нередко результат операции желательно разместить в одной строке с заданием на операцию. Размещение блоков по вертикали характерно для больших документов.

2.2.6. Операции с выражениями

Формульные блоки часто являются вычисляемыми выражениями или выражениями, входящими в состав заданных пользователем новых функций. Для работы с выражениями служат следующие кнопки:

15. Insert Function (Вставить функцию) — вставка функции из списка, появляющегося в диалоговом окне.
16. Insert Unit (Вставить единицу) — вставка размерных единиц.
17. Calculate (Вычислить) — вычисление выделенного выражения.

Mathcad имеет множество встроенных функций — от элементарных до сложных статистических и специальных математических функций. Синтаксис их записи порой легко забывается, поэтому возможность вставки функции с помощью

кнопки Insert Function очень удобна. Другая кнопка, Insert Unit, позволяет вставить нужную размерную величину.

Если документы большие, то при их изменении не всегда выгодно запускать вычисления с самого начала. Кнопка Calculate позволяет инициировать вычисления только для выделенных блоков, что может сократить время вычислений.

2.2.7. Доступ к дополнительным возможностям Mathcad

Начиная с версии Mathcad 7.0 на панели инструментов появились следующие кнопки:

18. Insert Hyperlink (Вставка гиперссылки) — обеспечивает создание гиперссылки.

19. Component Wizard (Мастер компонентов) — открывает окно мастера, дающего удобный доступ ко всем компонентам системы.

Для оперативного изменения масштаба отображения символов в текущем документе на панели инструментов имеется раскрывающийся список Zoom (Масштаб). В поле раскрывающегося списка (см. элементы панели инструментов с номерами 20 и 21 на рис. 2.6) отображается значение выбранного масштаба, а кнопка с направленной вниз стрелкой раскрывает список доступных значений.

Масштаб задается в процентах — за 100 % принят масштаб по умолчанию. Задав масштаб 200 %, можно увеличить масштаб отображения документа вдвое, а задав масштаб в 50 %, этот масштаб можно уменьшить вдвое (см. также подраздел «Изменение масштаба документа» в разделе «Управление видом документа»).

2.2.8. Вставка переменной-таблицы в Mathcad 11/12

В системе Mathcad 11/12 в панели инструментов имеется новая кнопка Insert Table для вставки таблиц. Она вызывает появление в окне документа шаблона для переменной типа таблицы и самой таблицы (рис. 2.11).

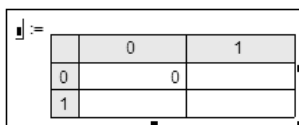


Рис. 2.11. Шаблон для переменной-таблицы Mathcad 11/12

2.2.9. Управление ресурсами Mathcad 8 — Mathcad 2001i

Заключительная группа из двух кнопок (см. рис. 2.6) позволяет обратиться к центру ресурсов и встроенной справочной базе данных системы:

22. Resource Center (Центр ресурсов) — обеспечивает доступ к центру ресурсов (в Mathcad 11/12 отсутствует).

23. Help (Справка) — обеспечивает доступ к ресурсам справочной базы данных системы.

Для закрытия панели инструментов и ее удаления с экрана служит кнопка Close (Закрыть) под номером 24 в правой верхней части панели.

Указанных инструментов обычно вполне достаточно для работы с документами без обращения к командам меню. Тем не менее меню имеет значительно большие возможности. В дальнейшем мы рассмотрим все команды меню.

2.3. Работа с панелью форматирования

2.3.1. Обзор средств панели форматирования

Панель форматирования содержит типовые средства управления шрифтами: набор стилей и размеров шрифта, три кнопки для изменения начертания шрифта (полужирный, наклонный и подчеркнутый), а также три кнопки для выравнивания текста в абзацах (по левому краю, по центру и по правому краю). Все эти средства позволяют форматировать документы, придавая им нужный пользователю вид.

2.3.2. Кнопки переключения стилей документов

Для выбора стиля отображения текстовых блоков (см. также подраздел «Задание стиля текстовых комментариев» раздела «Форматирование текста» в главе 5) служит раскрывающийся список с кнопкой активизации (см. рис. 2.6):

1. Style (Стиль) — текущий стиль отображения текстовых блоков.
2. Style (Стиль) — открытие списка выбора стиля.

2.3.3. Выбор шрифтов для документа

Для выбора шрифтов и их размеров служит следующая группа кнопок панели форматирования (см. рис. 2.6):

3. Font (Шрифт) — текущий шрифт символов.
4. Font (Шрифт) — открытие списка выбора шрифта.
5. Font Size (Размер шрифта) — текущий размер символов.
6. Font Size (размер шрифта) — открытие списка выбора размеров шрифта.

Для изменения начертания символов (см. также подраздел «Изменение параметров шрифта» раздела «Форматирование текста» в главе 5) предназначены три кнопки (см. рис. 2.3):

7. Bold (Полужирный) — полужирное начертание.
8. Italic (Курсив) — наклонное (курсивное) начертание.
9. Underline (Подчеркнутый) — подчеркнутое начертание.

Возможна произвольная комбинация начертания символов. Например, Bold и Italic дают полужирные наклонные символы.

2.3.4. Выравнивание текста

Три следующие кнопки (см. рис. 2.6) задают режим выравнивания текста (см. также подраздел «Форматирование абзацев» раздела «Форматирование текста» в главе 5):

10. Align Left (По левому краю) — выравнивание текста по левой границе.
11. Align Center (По центру) — выравнивание текста по центру.
12. Align Right (По правому краю) — выравнивание текста по правой границе.

2.3.5. Создание маркированных и нумерованных списков

Две последние кнопки (см. рис. 2.6) предназначены для создания списков (см. также подраздел «Форматирование абзацев» раздела «Форматирование текста» в главе 5):

13. Bullets (Маркированный) — создание маркированного списка.

14. Numbering (Нумерованный) — создание нумерованного списка.

Для закрытия панели форматирования и ее удаления с экрана служит кнопка Close (Закрыть) под номером 15 в правой верхней части панели.

2.3.6. Кнопки задания индексов Mathcad 11/12

В системе Mathcad 11/12 в конце панели форматирования размещены еще две кнопки — для задания надстрочного индекса x^2 и подстрочного индекса x_2 . Их применение заметно облегчает ввод индексированных переменных для массивов, матриц и векторов.

Все кнопки панели форматирования хорошо известны каждому пользователю текстового процессора Word.

2.4. Применение палитр математических знаков

2.4.1. Вывод панели палитр математических знаков

Вместо раскрывающихся перемещаемых палитр математических знаков, применяемых в ранних версиях Mathcad 3.0/4.0 для Windows 3.1/3.11, начиная с седьмой версии Mathcad используются более удобные перемещаемые палитры. Они служат для вывода заготовок (шаблонов) математических операторов (цифр, знаков арифметических операций, матриц, интегралов, производных и так далее), функций системы и отдельных символов, например греческих букв.

Для вывода шаблона того или иного объекта с помощью палитры нужно выполнить описанную ниже процедуру.

1. Выведите нужную палитру на экран.
2. Поместите курсор ввода в нужное место и зафиксируйте его щелчком левой кнопки мыши.
3. Щелкните левой кнопкой мыши на нужном объекте или символе в палитре. Объект (символ) появится в окне редактирования на месте курсора ввода.
4. Уберите палитру (если нужно), щелкнув на кнопке закрытия в конце строки заголовка палитры.

2.4.2. Применение палитр математических знаков

Кнопки вывода палитр занимают пятую сверху панель окна системы. Назначение этих кнопок показано на рис. 2.6, на котором панель вывода палитр перемещена в левую часть экрана, а в правой части окна Mathcad видна открытая палитра Calculus (Вычисления). Палитры появляются в окне редактирования документов при щелчке на соответствующих кнопках панели вывода палитр.

На рис. 2.12 представлены панель для вывода математических знаков, расположенная под панелью форматирования, и все палитры. С их помощью можно вводить в документы практически все известные математические символы и операторы. К сожалению, их так много, что вывод всех палитр (как это показано на рисунке) обычно бесполезен, так как в окне редактирования не остается места для подготовки документов. Поэтому рекомендуется неиспользуемые в данное время палитры закрывать.

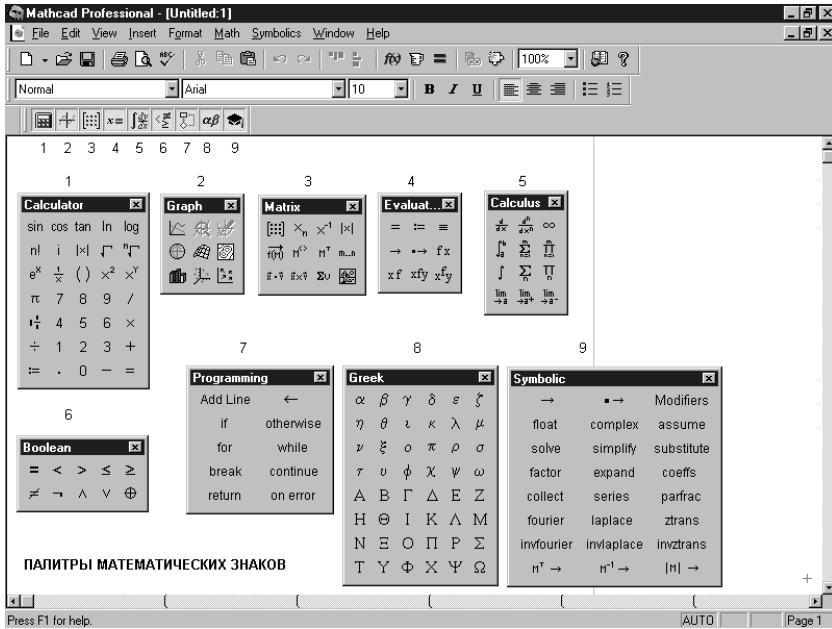


Рис. 2.12. Окно Mathcad 2001i со всеми палитрами математических знаков

Палитры можно располагать в удобном месте окна редактирования, причем пользоваться сразу несколькими (в раскрывающихся палитрах ранних версий Mathcad можно было открыть только одну, при открытии следующей предыдущая закрывалась).

С помощью общей панели (или «панели палитр») можно вывести все палитры сразу или только нужные для работы. Для установки с их помощью необходимого шаблона (объекта) достаточно поместить курсор ввода в желаемое место окна редактирования (красный крестик на цветном дисплее) и затем щелкнуть на значке нужного шаблона.

Применение палитр для вывода шаблонов математических знаков очень удобно, поскольку не надо запоминать разнообразные сочетания клавиш, используемые для ввода специальных математических символов. Впрочем, и эта возможность сохранена, так что привыкшие к работе с клавиатурой пользователи, имевшие дело с более ранними версиями системы Mathcad (в том числе для MS-DOS), могут воспользоваться навыками виртуозного набора формул и при работе с новой версией системы.

Любую палитру можно переместить в удобное место экрана, уцепившись за строку заголовка указателем мыши. Перемещая палитру, левую кнопку мыши нужно держать нажатой. В правом верхнем углу каждой палитры имеется единственная маленькая кнопка с жирным наклонным крестиком, служащая для удаления палитры с экрана, как только она становится ненужной.

Большинство кнопок на палитрах выводят общепринятые и специальные математические знаки и операторы, помещая их шаблоны в месте расположения курсора в документе. Позже, по мере изложения материала мы будем последовательно рассматривать эти знаки, в том числе, на первый взгляд, не вполне очевидные. Таких очень мало, и большинство знаков в палитрах вполне традиционны для математической литературы.

***СОВЕТ** Несмотря на удобство палитр, не стоит привыкать делать все операции с их помощью, поскольку для вывода палитры, ее установки в удобное место и закрытия нужно выполнить ряд операций. Зачастую вводить математические операторы, например знаки арифметических операций, куда быстрее и удобнее с помощью клавиатуры.*

2.5. Управление видом интерфейса

2.5.1. Обзор позиции View меню системы Mathcad 2001i

Для отображения/скрытия элементов интерфейса и для изменения вида документа предназначены команды меню View (Вид), которые перечислены ниже (рис. 2.1):

- Toolbars (Панели инструментов) — подменю со списком панелей и палитр, которые можно показать или скрыть;
- Status Bar (Строка состояния) — отображение или скрытие строки состояния внизу окна документа;
- Ruler (Линейка) — отображение или скрытие мерной линейки;
- Regions (Области) — управление видом блоков документа;
- Zoom (Масштаб) — изменение масштаба документа;
- Refresh (Обновить) — обновление экрана;
- Animate (Анимация) — анимация графиков;
- Playback (Воспроизвести) — вызов проигрывателя;
- Preferences... (Предпочтения) — настройка интерфейса в соответствии с пожеланиями пользователя.

В следующих подразделах этого раздела перечисленные выше команды будут рассмотрены более подробно. При этом команды Animate и Playback, как относящиеся к графике, мы рассмотрим в главе 4.

2.5.2. Обзор позиции View меню Mathcad 11/12

Набор позиции View меню новейших версий Mathcad 11/12 претерпел некоторые изменения. В нем теперь представлены следующие команды:

- Toolbars (Панели инструментов) — подменю со списком панелей и палитр, которые можно показать или скрыть;
- Ruler (Линейка) — отображение или скрытие мерной линейки;
- Status Bar (Строка состояния) — отображение или скрытие строки состояния внизу окна документа;
- Header and Footer... — вывод окна для установки верхнего и нижнего колонтитулов;
- Regions (Области) — управление видом блоков документа;
- Zoom... (Масштаб) — изменение масштаба документа;
- Refresh (Обновить) — обновление экрана.

Окно установки колонтитулов, показанное на рис. 2.13, имеет три панели для задания надписей в колонтитулы слева, по центру и справа колонтитула. Напомним, что колонтитулы это строки сверху и снизу документов. Оба колонтитула

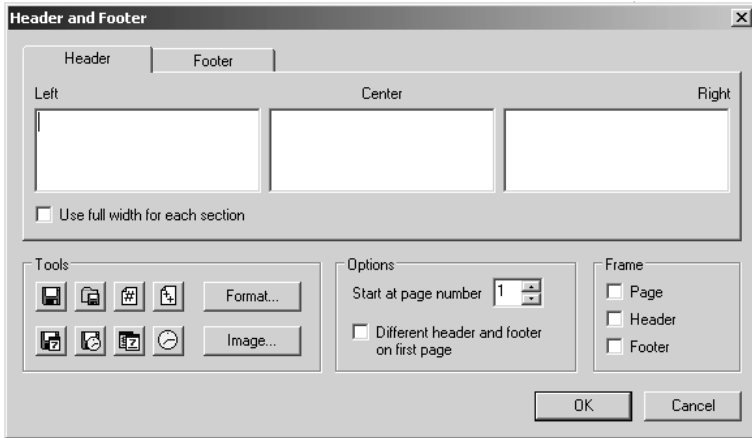


Рис. 2.13. Окно Mathcad 11/12 установки колонтитулов

(верхний Header и нижний Footer) одинаковы. Соответственно, одинаковы и вкладки окна колонтитулов.

Нетрудно заметить, что из этой позиции меню исчезли средства анимации — они перешли в новую позицию Tool меню. Эта участь постигла и команду Preferences....

2.5.3. Отображение и скрытие элементов интерфейса

Как уже отмечалось, пользовательский интерфейс новых версий Mathcad содержит такие элементы, как панель инструментов, панель форматирования и набор палитр для ввода математических символов. Наряду с полезными функциями (облегчение работы с системой) эти элементы имеют и недостаток — они заметно сужают полезную область рабочего экрана, уменьшая ее примерно на 3—4 строки.

Существуют три команды в меню View (Вид), позволяющие управлять выводом дополнительных элементов пользовательского интерфейса:

- Toolbars (Панели инструментов) — отображение или скрытие стандартной панели инструментов (Standard), панели форматирования (Formatting), панели вывода палитр математических спецзнаков (Math), всех палитр математических знаков;
- Status Bar (Строка состояния) — отображение или скрытие строки состояния внизу окна документа;
- Ruler (Линейка) — отображение или скрытие мерной линейки.

Назначение приведенных выше команд достаточно очевидно, и нет смысла пояснять их более подробно. Нужно лишь отметить, что для отображения соответствующей панели достаточно против ее названия в меню View установить галочку (щелчком мыши), а для скрытия — снять (тоже щелчком мыши).

2.5.4. Обновление экрана

При манипуляциях с объектами (например, при изменении их размеров и выделении) на экране могут оставаться нежелательные искажения, остатки графических объектов и линий выделения. Для их устранения следует использовать команду Refresh (Обновить) меню View (Вид). Эта команда никак не влияет на объекты

документа, она лишь уничтожает дефекты прорисовки, если они возникают при работе с документом. Команда полезна также в тех случаях, когда часть текстового блока не видна из-за наложения на него другого блока — формульного или графического. Ее исполнение ведет к прорисовке невидимой части текстового блока.

2.5.5. Задание предпочтений в Mathcad 2001i

Команда Preferences (Предпочтения) меню View (Вид) в Mathcad 2001i позволяет пользователю настроить некоторые элементы интерфейса программы в соответствии со своим вкусом. При выборе этой команды открывается диалоговое окно с четырьмя вкладками. На рис. 2.14 показана вкладка General (Общие).



Рис. 2.14. Общие параметры предпочтений в Mathcad 2001i

В группе Startup Options (Параметры запуска) доступны следующие флажки:

- Show Tip of the Day at startup (Показывать совет дня при запуске) — при установке этого флажка система запускается с открытым окном Tip of the Day (Совет дня) поверх окна Mathcad;
- Open Resource Center at startup (Открывать окно Центра ресурсов при запуске) — при установке этого флажка система запускается с открытым окном центра ресурсов поверх окна Mathcad (см. также раздел «Центр информационных ресурсов и система помощи»).

В группе Keyboard Options (Параметры клавиатуры) доступны следующие флажки:

- Standard Windows shortcut keys (Стандартные клавиши быстрого доступа) — при установке этого флажка в Mathcad можно использовать привычные пользователям Windows сочетания клавиш;
- Context-sensitive equals sign (Контекстное использование знака равенства) — при установке этого флажка попытка ввода знака равенства (=) следом за неопределенной переменной автоматически приведет к его замене знаком присваивания (:=); при снятом флажке подобная попытка приведет к появлению сообщения об ошибке.

На вкладке General имеется также раскрывающийся список диалектов английского языка, которые можно использовать при проверке орфографии в докумен-

тах Mathcad. К сожалению, такая проверка для русского языка не предусмотрена. Имеется также установка папки, в которую должны записываться по умолчанию файлы документов. Кнопка Browse... открывает типовое окно файловой системы, в котором можно наметить нужную папку.

На вкладке Internet имеются следующие поля, которые предназначены для настройки параметров прокси-сервера следующих хорошо известных служб Интернета:

- HTTP — обычная работа в Интернете по протоколу HTTP;
- FTP — работа с файловыми серверами;
- GOPHER — работа с устаревшими файловыми серверами GOPHER.

Необходимые данные можно узнать у вашего поставщика доступа в Интернет или системного администратора. Незаполненные поля означают, что прокси-сервер не используется.

ПРИМЕЧАНИЕ Прокси-сервер, именуемый также сервером приоритетов, это специальная программа у провайдера (поставщика доступа) услуг Интернета, которая принимает запрашиваемые пользователем данные и запоминает их на сервере провайдера. Если другой клиент данного провайдера пользуется этими данными, то он получает их уже не по магистральным каналам Интернета, а прямо с сервера провайдера, что намного быстрее.

На вкладке Script Security имеются установки степени защиты информации при работе в Интернете. Возможны три уровня защиты: высокий High Security, средний Medium Security и низкий Low Security.

Последняя вкладка HTML Output устанавливает опции записи в формате HTML (рис. 2.15). Здесь задаются форматы записи различных компонентов HTML-страниц, в частности рисунков. При этом для рисунков формата JPEG предусмотрена установка степени компрессии файлов.

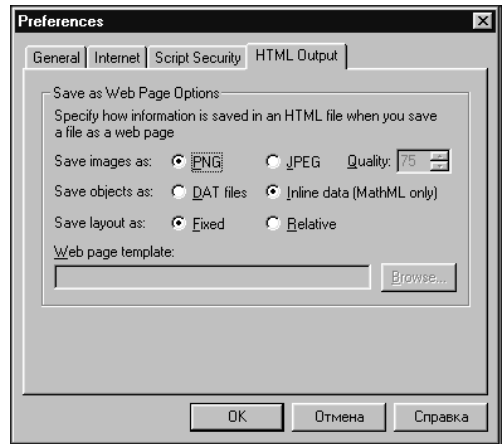


Рис. 2.15. Установка опций выхода HTML в Mathcad 2001i

2.5.6. Задание предпочтений в Mathcad 11/12

Для систем Mathcad 11/12 команда Preferences из меню удалена и перемещена в подменю новой позиции Tools меню. При исполнении этой команды появляется окно, показанное на рис. 2.16. Нетрудно заметить, что наряду с вкладками, имеющимися в окне предпочтений рис. 2.14, появились новые вкладки:

File Locations (Расположение файлов) — установка пути к файлам документов;
 Language (Язык) — установка языка меню (пока, увы, только английский), языка словарей для проверки орфографии текстов (больше десятка языков) и диалекта (английского и американского);

Save (Запись) — установка формата записи файлов (обычный Worksheet, XHTML и XHTML с компрессией);

Warning (Внимание) — установка выводов сообщений (на рис. 2.16 эта вкладка открыта).

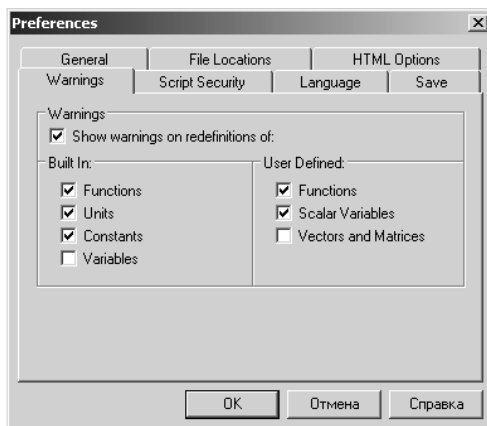


Рис. 2.16. Окно предпочтений Mathcad 11/12 с открытой вкладкой Warning

Установки в новых вкладках довольно просты, и читатель легко справится с ними, если это ему нужно — как правило, необходимость в этом встречается редко.

2.6. Управление окнами

2.6.1. Обзор позиции Window меню

Как и большинство программ, работающих в среде Windows, система Mathcad унаследовала удобный многооконный пользовательский интерфейс. Мы уже видели, что после загрузки системы появляется текущее окно, содержащее средства управления, позволяющие сворачивать окно в значок (кнопку), разворачивать его до размера полного экрана или восстанавливать его размер. Окна появляются при выборе соответствующих команд меню, а также при выводе сообщений об ошибках. Традиционны для Windows и средства перемещения окон и плавного изменения их размеров.

Система Mathcad позволяет одновременно работать со многими задачами. Под каждую задачу она отводит отдельное окно. Одно из них обычно является текущим или активным. Состояние других документов «замораживается», но их можно активизировать в любое время без новой загрузки и без ввода данных (для этого достаточно установить указатель мыши в любом месте окна документа и щелкнуть левой кнопкой).

Основные команды для работы с окнами сосредоточены в меню Window (Окно), показанном на рис. 2.17.

Ниже перечислены команды меню Window:

- Cascade (Каскадом) — расположить окна документов так, чтобы были видны заголовки всех окон;

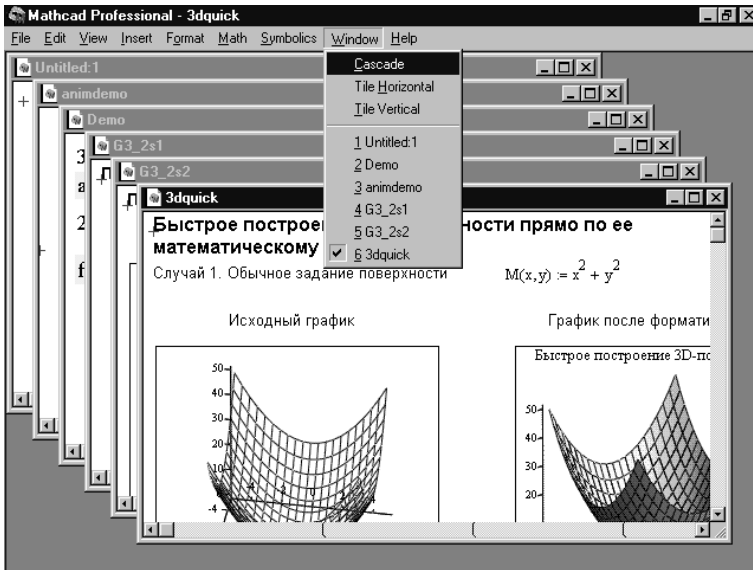


Рис. 2.17. Вид экрана Mathcad с раскрытым меню Window

- Tile Horizontal (По горизонтали) — расположить окна документов без перекрытия рядом друг с другом (горизонтально);
- Tile Vertical (По вертикали) — расположить окна документов без перекрытия друг под другом (вертикально).

Кроме этих команд в меню Window имеется список открытых окон. Он позволяет быстро сделать активным требуемое окно. Если окно загруженного документа свернуто, то оно представляется значком (кнопкой) в нижней части окна Mathcad.

Рассмотрим действие основных команд управления окнами. Они позволяют пользователю подстроить графический интерфейс системы под свой вкус.

2.6.2. Каскадное расположение окон

Команда Cascade (Каскадом) меню Window (Окно) задает каскадное расположение окон (см. рис. 2.17). При этом окна располагаются последовательно, частично перекрывая друг друга. Обычно видны лишь строки заголовков (имя, указанное в строке заголовка, является именем загруженного документа).

Верхнее окно можно свернуть в кнопку, щелкнув на кнопке сворачивания окна. При этом следующее за ним окно выходит на передний план. Щелчок на кнопке восстановления разворачивает окно до размера полного экрана.

Каскадное расположение окон удобно для их переключения. Оно позволяет видеть имена сразу всех файлов, документы которых загружены в окна, и дает достаточно места для обзора активного окна. Однако при таком расположении окон затруднено перетаскивание объектов из одного окна в другое.

2.6.3. Расположение окон по горизонтали

Команда Tile Horizontal (По горизонтали) меню Window (Окно) выводит на экран сразу все окна, причем они имеют одинаковый размер, вытягиваются по горизонтали и расположены друг под другом, не перекрывая друг друга. Размер каж-

дого окна тем меньше, чем больше окон выведено. В каждом окне видна лишь часть загруженного в него документа (рис. 2.18). В нижней части рисунка показана кнопка свернутого окна.

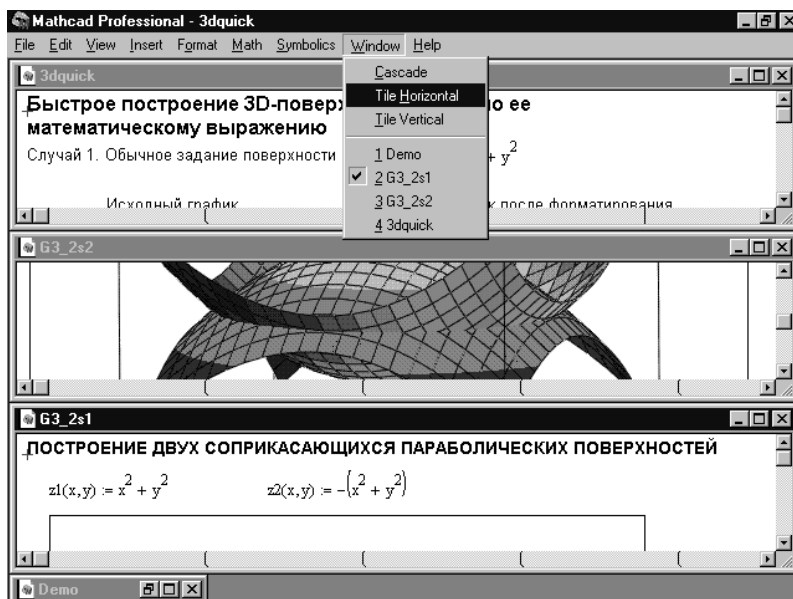


Рис. 2.18. Расположение окон по горизонтали

Равномерное расположение окон может оказаться наиболее удобным, если требуется скопировать блок из одного окна в другое. Однако здесь есть некоторая трудность: в уменьшенном по размеру окне видны не все блоки, поэтому не следует увлекаться одновременным выводом большого числа окон. Практика показывает, что чаще всего достаточно работать с двумя-тремя окнами. И, наконец, надо помнить, что в каждом окне возможна индивидуальная прокрутка, поэтому несложно найти нужный объект, даже если он вначале не попал в видимую область окна.

2.6.4. Расположение окон по вертикали

Команда Tile Vertical (По вертикали) меню Window (Окно) выводит на экран сразу все окна, причем они имеют одинаковый размер, вытягиваются по вертикали и расположены рядом друг с другом, не перекрываясь. Размер каждого окна также тем меньше, чем больше окон выведено. В каждом окне видна лишь часть загруженного в него документа (рис. 2.19).

Строго горизонтальное или вертикальное расположение окон получается, если окон немного, например два или три. При большем числе окон деление экрана происходит как по горизонтали, так и по вертикали (рис. 2.20). При этом расположение окон оказывается одинаковым вне зависимости от выбранной команды (Tile Vertical или Tile Horizontal).

Выбор оптимального расположения окон зависит от характера работы с Mathcad. Система позволяет перетаскивать мышью объекты из одного окна в другое. Для такого перетаскивания достаточно выделить объект (или группу объектов) и, удерживая нажатой левую кнопку мыши, перетащить объект (или группу объек-

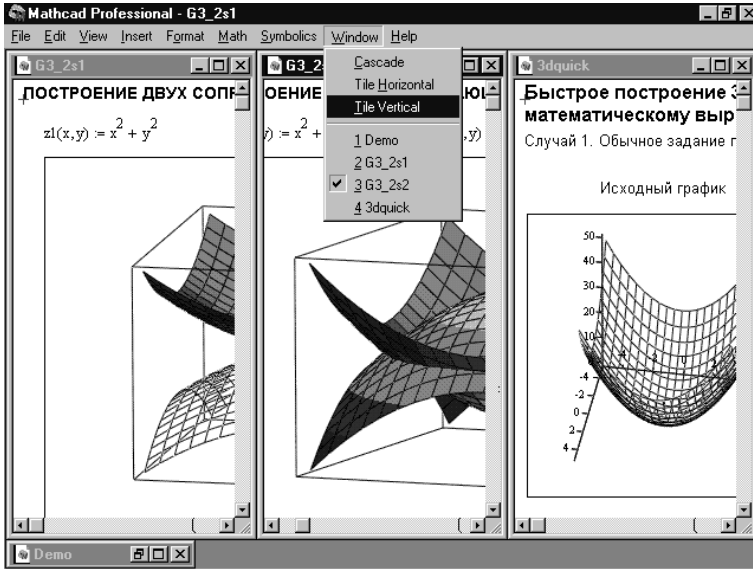
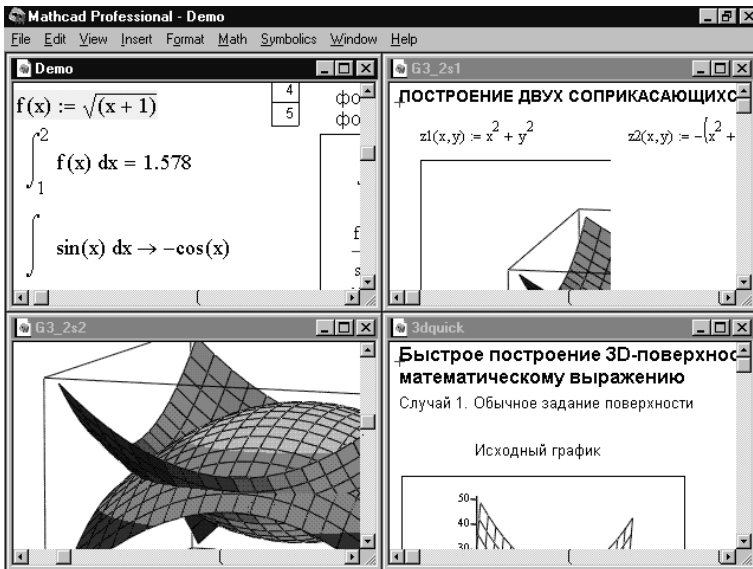


Рис. 2.19. Расположение окон по вертикали

Рис. 2.20. Расположение большого числа окон после выбора команды *Tile Vertical*

тов) в другое окно. Для этой операции, облегчающей компоновку сложных документов, удобное расположение окон имеет важное значение.

2.6.5. Системное меню окна

В каждом окне документа Mathcad, как и в главном окне Mathcad, имеется характерная для приложений под Windows кнопка вызова системного меню (в левом

углу строки заголовка). Если щелкнуть на этой кнопке, раскроется системное меню, содержащее следующие команды (рис. 2.21):

- Restore (Восстановить) — раскрыть окно документа, если оно свернуто в кнопку, или уменьшить его размер, если оно развернуто во весь экран;
- Move (Переместить) — переместить окно документа;
- Size (Размер) — изменить размер окна документа;
- Minimize (Свернуть) — свернуть окно в кнопку;
- Maximize (Развернуть) — развернуть окно во весь экран;
- Close (Закрыть) — закрыть окно и закончить работу с документом;
- Next (Следующее) — перейти к следующему окну.

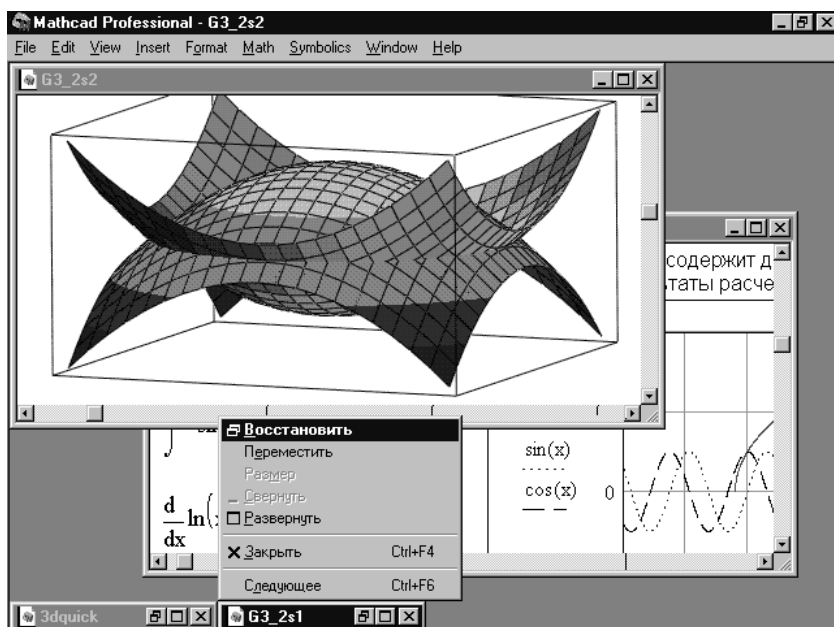


Рис. 2.21. Системное меню окна документа

Эти команды характерны почти для всех прикладных программ, работающих в среде Windows, и на них не стоит останавливаться. В локализованной версии Windows имена команд, естественно, даны на русском языке.

Системное меню главного окна Mathcad отличается от показанного на рисунке системного меню окна документа только отсутствием последней команды, предназначенной для перемещения между окнами открытых документов.

Если открытых окон оказалось слишком много (ограничение в виде 8 окон в новой версии снято), то система откажется от ввода дополнительных документов и сообщит об этом в информационном окне. Для продолжения работы нужно закрыть часть ставших ненужными окон. Для этого используется команда Close (Закрыть) системного меню каждого окна. Вообще говоря, держать открытыми слишком много окон не рекомендуется в любом случае, так как это загромождает экран и уменьшает ресурсы компьютера (прежде всего объем доступной памяти).

2.7. Работа с Центром информационных ресурсов и справкой

2.7.1. Центр информационных ресурсов Mathcad 2001i

В Mathcad 2001i и в более ранних версиях команда Resource Center (Центр ресурсов) в позиции Help (Справка) меню, а также одноименная кнопка на панели инструментов открывают доступ к новому средству системы Mathcad — центру информационных ресурсов (см. рис. 2.22). Центр информационных ресурсов — по существу мощная база данных, объединяющая в себе встроенные в систему электронные книги, обучающую систему, справочную систему, многочисленные примеры применения — «шпаргалки», средства общения с фирмой-разработчиком системы. Кроме того, центр информационных ресурсов предоставляет выход в Интернет, средства регистрации и доступа к Web-библиотеке Mathcad и т. д.

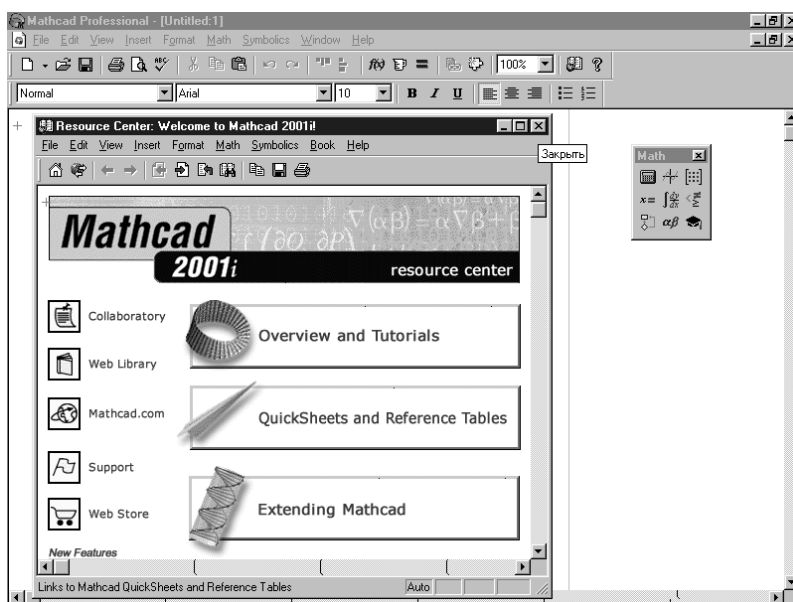


Рис. 2.22. Окно центра ресурсов в окне документа Mathcad 2001i

При выборе команды Resource Center открывается главное окно центра ресурсов с перечнем (в виде красочных значков) входящих в него средств. Это окно было показано на рис. 2.22. Обычно оно выводится при каждом очередном запуске. В левой части окна расположена группа небольших кнопок (значков), дающих прямой доступ в Интернет. Кроме того, имеются три большие кнопки, открывающие доступ к следующим основным разделам центра:

- Overview and Tutorials — данные о новых возможностях Mathcad и доступ к электронному учебнику;
- QuickSheets and Reference Tables — «быстрые шпаргалки» (примеры) и справочные таблицы;
- Extending Mathcad — данные о расширениях системы (электронных книгах, библиотеках и пакетах расширения).

Доступ к тому или иному средству центра ресурсов осуществляется щелчком на соответствующей кнопке. Окно центра имеет несколько расширенную панель инструментов. Кроме кнопок перемещения по страницам есть кнопки перемещения вперед и назад по разделам центра, а также кнопка поиска (Search). Еще есть кнопка для записи раздела справочной системы на диск, а также кнопка вывода списка просмотренных разделов.

2.7.2. Центр информационных ресурсов Mathcad 11/12

Запуск Центра ресурсов в Mathcad 11/12 осуществляется с панели Resources выбором из списка позиции Tutorial и нажатием кнопки Go. Это выводит окно самоучителя, в панели инструментов которого есть кнопка Home. Ее активизация выводит окно Центра ресурсов, показанное на рис. 2.23 для новейшей версии Mathcad 12 (в Mathcad 11 есть только три раздела этого окна — tutorial, quicksheet и reference tables).

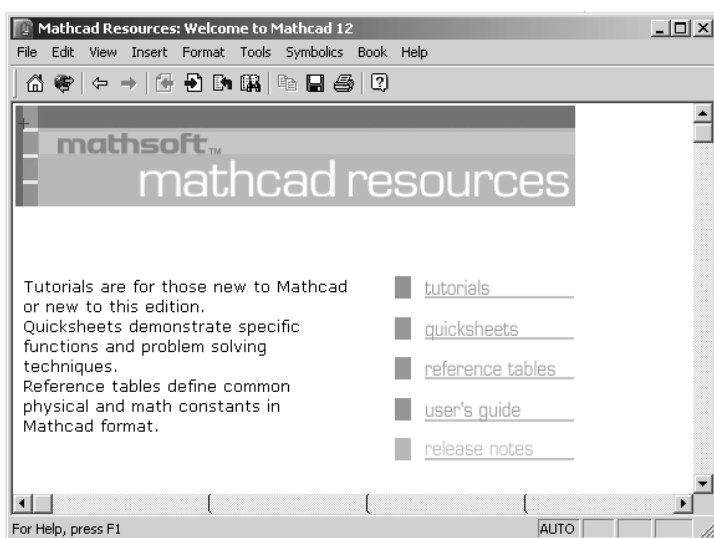


Рис. 2.23. Окно Центра ресурсов Mathcad 12

Помимо англоязычного учебника Tutorial, быстрых «шпаргалок» (примеров) QuitSheet и справочных таблиц Reference Tables в списке ресурсов панели Resources находятся все установленные в Mathcad 11/12 электронные книги. К сожалению, в имеющихся в продаже версиях Mathcad 11/12 не вполне ясного происхождения электронные книги не работоспособны из-за отсутствия или невозможности установки относящихся к ним пакетов расширения. В связи с этим в данной книге описание электронных книг и пакетов расширения по ним отсутствует. Подробное описание их можно найти в энциклопедии по системе Mathcad [11].

2.7.3. Организация справочной системы Mathcad

Электронная справка системы Mathcad это в сущности мощная база данных по всем возможностям системы, построенная по принципу создания гипертекстовых документов. Справка построена на основе HTML-документов и использует для своей работы Интернет-браузер Microsoft Internet Explorer.

Электронная справка имеет очевидные достоинства:

- большой (и почти неограниченный) объем информации;
- быстрый поиск информации;
- прекрасное оформление (в том числе цветное);
- возможность немедленного исполнения примеров и их переноса в документы;
- естественное совмещение с системой.

Помимо достоинств электронная справочная система имеет и существенные недостатки:

- нуждается в компьютере для работы с ней;
- неудобна для начального знакомства с системой;
- заслоняет документ, с которым работает пользователь;
- каждая следующая страница справки закрывает предыдущую;
- написана на языке оригинала (в нашем случае на английском).

Последний недостаток особенно важен для русскоязычных пользователей. Многие считают, что это перечеркивает все достоинства электронной справочной системы и делает ее по существу бесполезной. Однако опытные пользователи, даже плохо владеющие английским языком, время от времени вынуждены обращаться к электронной справочной системе, поскольку только она может дать ответы на ряд конкретных вопросов, которые не попадают в обычные книги вследствие ограниченности их объема (эта книга не является исключением).

2.7.4. Меню справки Mathcad 2001i

Команды для работы со справочной системой сосредоточены в меню Help (Справка), которое в открытом виде показано на рис. 2.24.

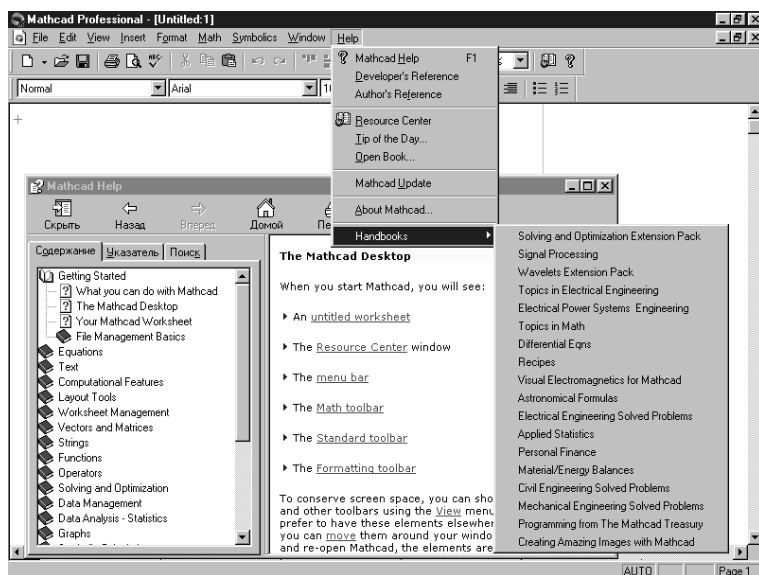


Рис. 2.24. Окно системы Mathcad с раскрытым меню Help и окном справки

Как видно из рисунка, меню Help содержит следующие команды:

- Mathcad Help (Справка по Mathcad) — вызов окна справки по системе Mathcad (см. раздел «Справка по Mathcad»);
- Developer's Reference (Информация для разработчиков) — вызов окна с информацией о средствах интерфейса пользователя и объектной связи между компонентами системы Mathcad (это окно по своему виду аналогично окну справки и содержит разнообразные данные, полезные тем пользователям, которые ведут разработку комплексного программного обеспечения, включающего в себя и систему Mathcad);
- Autor's Reference (Информация для авторов) — вызов окна с информацией для авторов электронных книг (это окно также аналогично окну справки и содержит разделы, полезные для авторов профессионально выполняемых электронных книг);
- Resource Center (Центр ресурсов) — вызов окна центра информационных ресурсов (см. разделы «Центр ресурсов» и «Работа в Интернете»);
- Tip of the Day (Совет дня) — вызов окна оперативной подсказки (см. подраздел «Совет дня» раздела «Установка и запуск системы Mathcad 2001» в главе 1);
- Open Book (Открыть книгу) — вызов окна открытия справочных книг и расширений системы (см. раздел «Электронные справочные книги»);
- Mathcad Update (Обновление Mathcad) — доступ к обновлению через Интернет, если таковое возможно для используемой версии системы (см. раздел «Обновление Mathcad»);
- About Mathcad (О Mathcad) — вызов окна с краткой информацией о системе Mathcad и ее производителе — фирме MathSoft Inc.;
- Handbooks (Справочные книги) — подмену со списком установленных справочных книг — расширений системы (см. раздел «Электронные справочные книги»).

2.7.5. Работа со справкой Mathcad 2001i

Для вызова окна справки по Mathcad можно:

- выбрать команду Mathcad Help (Справка по Mathcad) в меню Help (Справка);
- нажать клавишу F1;
- щелкнуть на кнопке Help (Справка) панели инструментов (кнопка со значком ?).

После вызова справочной системы появляется окно, показанное на рис. 2.24. Следует отметить, что справочная система по существу является самостоятельным приложением и ее можно запускать независимо от самой системы Mathcad.

Как видно из рисунка, окно справочной системы имеет три вкладки:

- Context (Содержание) — оглавление в виде древовидной структуры;
- Index (Индекс) — предметный указатель;
- Search (Поиск) — система поиска в базе данных справочной системы.

Mathcad обеспечивает контекстно-зависимое действие клавиши F1 — она открывает то окно, которое необходимо по контексту. Например, если курсор ввода находится на имени какой-либо функции, то нажатие клавиши F1 обеспечивает вывод окна справочной системы с информацией именно об этой функции. Такая возможность делает работу со справочной системой удобной и эффективной.

Как отмечалось, справочная система Mathcad базируется на широком применении гипертекстовых ссылок, а работа с ней напоминает работу в Интернете. При активизации ссылки открывается связанная с ней часть справочной страницы. При этом ссылка, которая использовалась ранее, меняет цвет.

Окна справочной системы имеют свою панель инструментов. На этой панели имеются кнопки Forward (Вперед) и Back (Назад) для перемещения по страницам справочной системы, а также кнопка Home (Домой) для возврата к начальной странице. Кнопка Hide/Show (Скрыть/Показать) меняет размер окна справочной системы, а кнопка Print (Печать) позволяет распечатать текущую страницу справки. Кнопка Options (Параметры) открывает подменю, в котором имеются уже отмеченные команды и некоторые другие.

Содержание

По умолчанию окно справки открывается на вкладке Context (Содержание). На ней в виде древовидной структуры представлен перечень тем. Рис. 2.24 иллюстрирует процесс получения справки, относящейся к рабочему столу Mathcad. С помощью гиперссылок можно перейти к другим страницам справочной системы, связанным по теме с текущей страницей.

Индекс

На рис. 2.25 показано окно справочной системы с открытой вкладкой Index (Указатель). Для поиска в индексном каталоге требуемое слово нужно ввести в поле Type in the keyword to find (Введите искомое слово). По мере ввода в списке ниже будут появляться наиболее близкие по написанию слова. Чтобы открыть страницу справки, в индексный каталог которой включено искомое слово, это слово в списке надо выделить и щелкнуть на кнопке Display (Показать). Можно

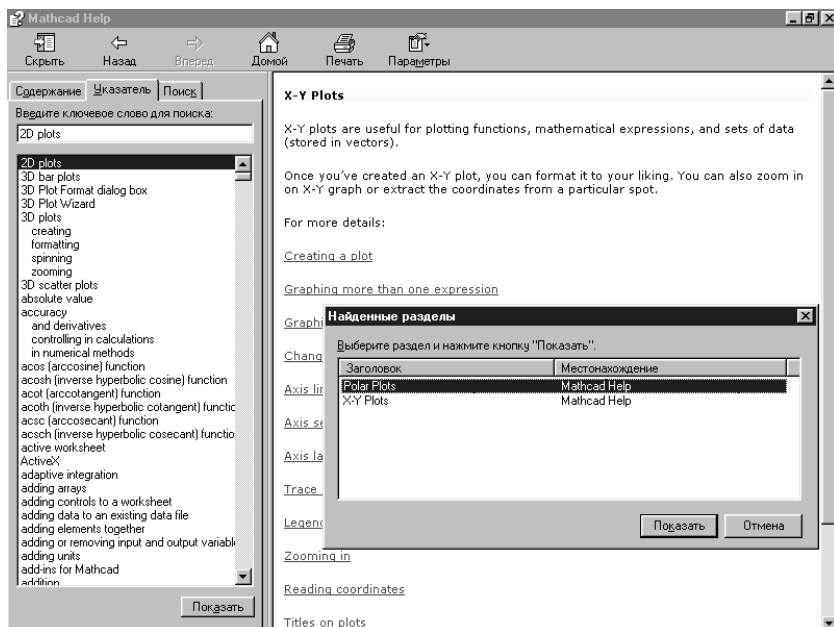


Рис. 2.25. Работа с индексным указателем справочной системы

также ввести курсор мыши в поле ввода и нажать клавишу Enter. Появляется окно со списком найденных разделов, показанное на рис. 2.25.

На страницах справки помимо гипертекстовых ссылок могут встречаться кнопки Related Topics (Связанные или Найденные разделы), активизация которых вызывает появление окна с названиями близких по смыслу тем. Одно из таких окон показано на переднем плане рисунка. Для открытия соответствующего раздела его название нужно выделить и щелкнуть на кнопке Display (Показать).

Поиск

Вкладка Search (Поиск) окна справки позволяет создать профессиональную справочную базу данных по системе Mathcad. Эта база данных дает возможность найти все разделы справочной системы, в которых упоминается заданное слово или хотя бы его начальная часть. Поскольку для базы данных нужна дополнительная память на диске, при установке Mathcad эта база данных обычно не создается. Для создания справочной базы данных надо ввести искомое слово в поле Type in the keyword to find (Введите искомое слово) и щелкнуть на кнопке List Topics (Перечислить разделы). После создания базы данных ее разделы будут представлены в списке ниже. Чтобы открыть соответствующий раздел, его название надо выделить и щелкнуть на кнопке Display (Показать).

Окно справочной системы с открытой вкладкой Search показано на рис. 2.26. В отличие от поиска по индексному каталогу, в данном случае отыскиваются все разделы справочной базы данных, в которых встречается указанное для поиска слово. На рис. 2.26 показано еще одно найденных разделов, которое появляется при активизации кнопки Related Topics.

Таким образом, система общего поиска предназначена для глубокого знакомства с тем или иным понятием или объектом системы.

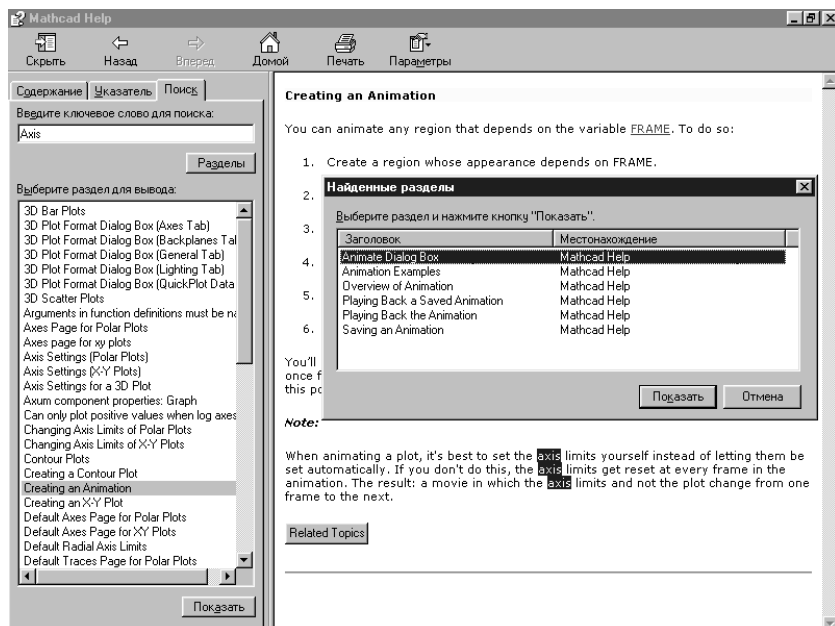


Рис. 2.26. Окно справки при общем поиске

2.7.6. Работа с электронными учебниками

При щелчке в окне центра ресурсов (см. рис. 2.22) на кнопке Overview and Tutorials (Обзор и электронные учебники) открывается окно с оглавлением данного раздела, представленное на рис. 2.27.

Для просмотра обзора по Mathcad 2001i (первая гиперссылка) или какого-либо электронного учебника достаточно активизировать гипертекстовую ссылку с его наименованием. В качестве примера на рис. 2.28 представлено начало раздела электронного учебника, посвященного основным проблемам решения дифференциальных уравнений (описывается метод Эйлера).

Вопреки существующему мнению любой электронный учебник Mathcad вовсе не предназначен для начинающих пользователей. Он содержит достаточно глубокие и тонкие сведения о системе Mathcad, алгоритмах и приемах вычислений. Правильнее сказать, что пользователю, далекому от математики, знакомиться с таким учебником не имеет смысла.

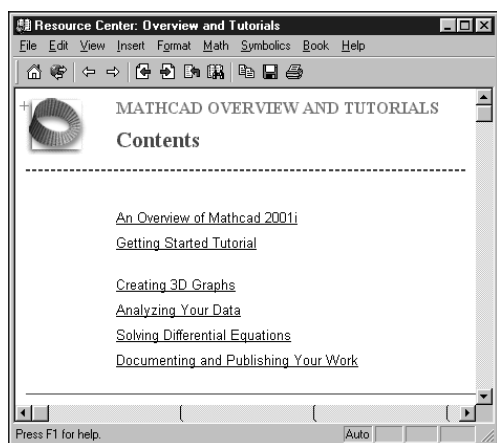


Рис. 2.27. Оглавление раздела Overview and Tutorials

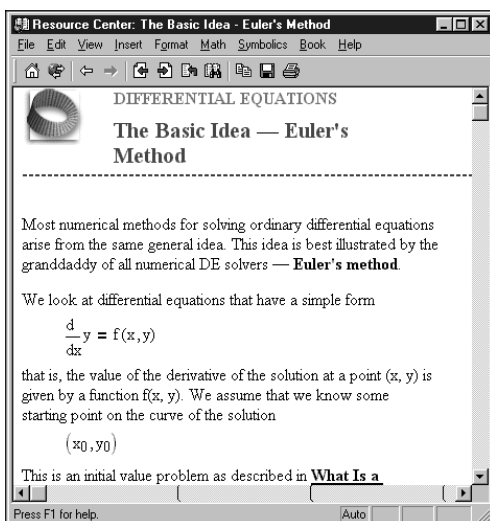


Рис. 2.28. Один из электронных учебников по системе Mathcad

2.7.7. «Быстрые шпаргалки» и справочные таблицы

Разработчики системы Mathcad (а это известные математические школы Запада) хорошо понимают ценность обучения на конкретных примерах. Как сказал И. М. Гельфанд: «Теории приходят и уходят, а примеры остаются». Поэтому в систему Mathcad введены так называемые «быстрые шпаргалки» (quicksheets) — простые примеры без длинных описаний, решающие обширный круг математических и научно-технических задач. Для доступа к этим примерам нужно в окне центра ресурсов (см. рис. 1.2) щелкнуть на кнопке Quicksheets and Reference Tables («Быстрые шпаргалки» и справочные таблицы), а затем в открывшемся окне — на ссылке Quicksheets. Откроется окно со списком разделов «быстрых шпаргалок» (рис. 2.29).

Теперь достаточно щелкнуть на ссылке нужного примера, и к вашим услугам появится документ с соответствующей «шпаргалкой». Окна со «шпаргалками» имеют обычный для Windows-приложений вид. На рис. 2.30 показано окно одной из «шпаргалок», посвященной построению графиков функции и ее производной.

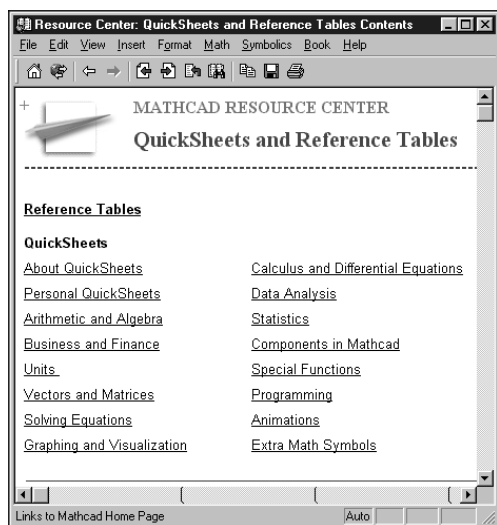


Рис. 2.29. Окно с перечнем разделов «быстрых шпаргалок»

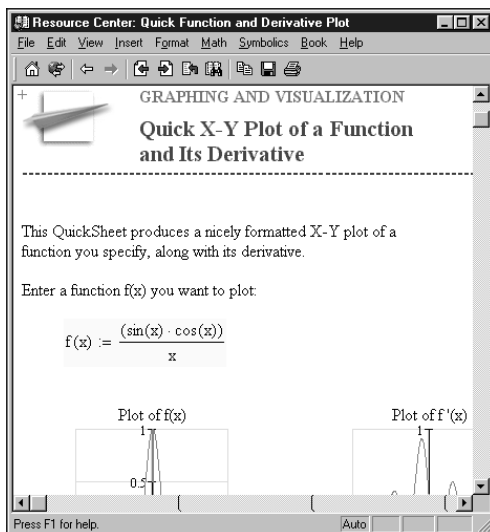


Рис. 2.30. Окно одной из «шпаргалок»

Надо отметить, что «шпаргалки» содержат сотни интересных (как простейших, так и сложных) примеров на применение системы Mathcad в практике математических и научно-технических расчетов. Разумеется, описание «быстрых шпаргалок» дано на английском языке.

Приведенные в данной книге многочисленные примеры в основном оригинальны и не повторяют примеров из «быстрых шпаргалок», поэтому совместно они могут существенно расширить кругозор пользователя и лучше подготовить его к профессиональному применению системы Mathcad. Но некоторые наиболее интересные примеры из справки в книге приведены с некоторой непринципиальной доработкой, в частности переводом комментариев (где это нужно) на русский язык.

Большой интерес даже у русскоязычных пользователей вызывают справочные таблицы (reference tables). Они заменяют обычные таблицы физико-химических и иных свойств различных веществ. Для доступа к этим примерам нужно в окне центра ресурсов (см. рис. 1.2) щелкнуть на кнопке Quicksheets and Reference Tables («Быстрые шпаргалки» и справочные таблицы), а затем в открывшемся окне — на ссылке Reference Tables. Появится окно с перечнем таблиц, показанное на рис. 2.31.

Далее достаточно щелкнуть на ссылке нужной таблицы. В качестве примера на рис. 2.32 показана таблица (точнее, только ее начало) со значениями (символьными) неопределенных интегралов.

Всего в этом разделе имеется 42 обширные таблицы с данными о различных химических элементах (рис. 2.33), их физических свойствах, сведения о математических константах, таблицы производных и интегралов, формулы по механике, электротехнике, радиотехнике и т. д. Фактически с системой Mathcad поставляется обширный справочник по этим разделам, постоянно находящийся «под рукой». Наряду со справочной информацией приводятся «живые» примеры применения этой информации в расчетах. Их всегда можно подстроить под свои исходные данные и тут же получить нужные результаты.

Тексты, формулы, графики (словом, любые объекты) из справочной системы можно легко перетаскивать в документы Mathcad, что облегчает пользователю со-

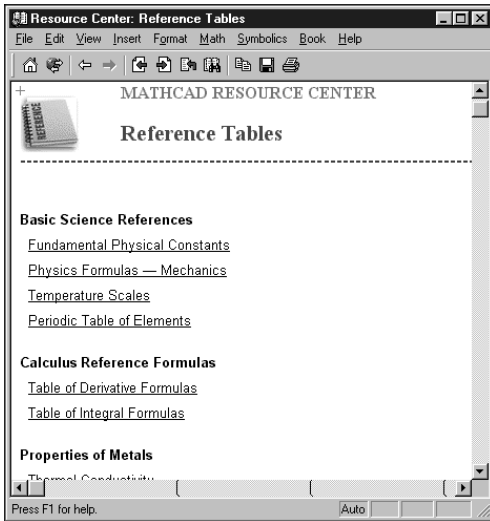


Рис. 2.31. Окно с перечнем справочных таблиц

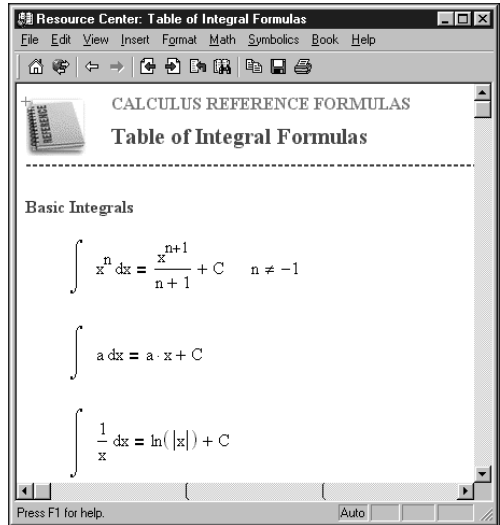


Рис. 2.32. Таблица неопределенных интегралов

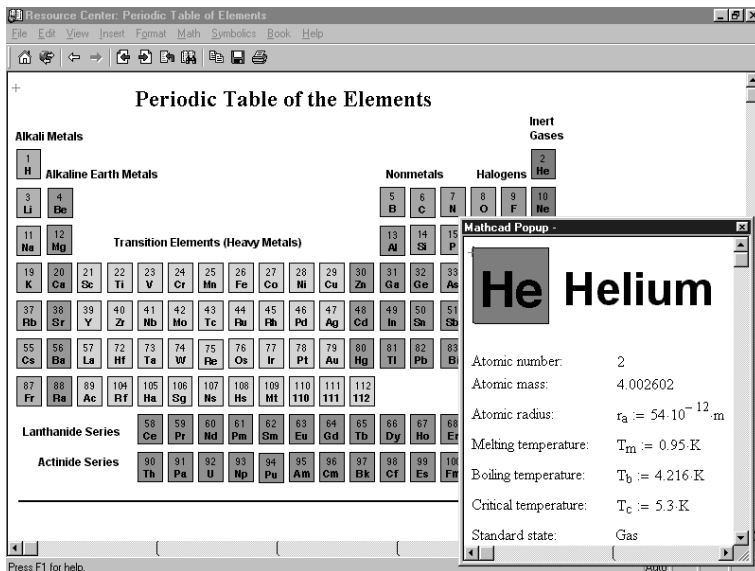


Рис. 2.33. Периодическая таблица химических элементов

ставление своих документов и обучающих программ. Можно также копировать в буфер обмена целые разделы, а затем размещать их в своих документах. На рис. 2.34 показан пример копирования схемы RC-генератора с расчетом частоты его колебаний в окно документа (фрагмент его показан слева).

Справедливости ради надо отметить, что многие справочные таблицы недостаточно полны. Например, таблицы производных и интегралов насчитывают десятки формул, тогда как обычные справочники, как правило, содержат тысячи формул для вычисления производных и интегралов. Порой складывается впечатление,

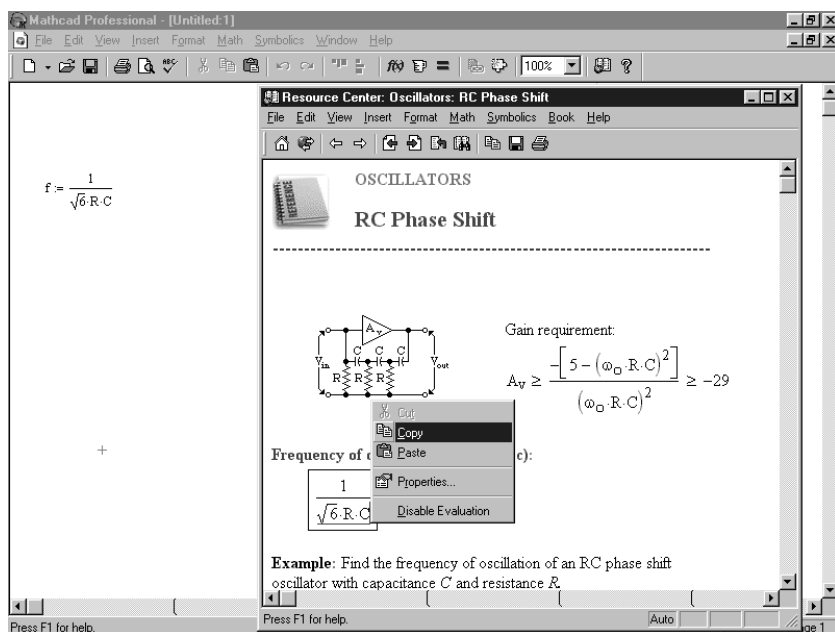


Рис. 2.34. Пример копирования формул для расчета частоты RC-генератора в окно документа

что та или иная таблица включена в справочную базу данных из «идейных» соображений (должна же быть хоть какая-то), а вовсе не для предоставления полной информации. Надо, однако, помнить, что увеличение числа и информационного наполнения таблиц неизбежно повлекло бы за собой резкий рост объема памяти, занимаемой системой на жестком диске.

2.7.8. Работа с электронными книгами и расширениями Mathcad

Mathcad задумана как расширяемая система, которая может подстраиваться под решение самых различных задач пользователей. Расширение достигается следующими средствами:

- обычными электронными книгами, которые используют только средства базовой системы Mathcad и функции пользователя;
- пакетами расширения (Extension Pack), вводящими в систему новые встроенные операторы и функции (последние появляются в списке встроенных функций);
- библиотеками, написанными на языке C и расширяющими или изменяющими возможности ядра системы Mathcad;
- Patch-файлами модернизации системы.

Между этими средствами есть существенная разница, обычно ускользающая от рядового пользователя. Как уже отмечалось, в стандартных электронных книгах используются только встроенные операторы и функции системы Mathcad. Такие книги можно рассматривать просто как тематические сборники примеров, пригодные для работы в любой подходящей версии Mathcad.

Пакеты расширения серьезно модернизируют систему, вводя в нее от 20—30 до 150 и более новых операторов и функций, которые приобретают статус встроен-

ных. Такие пакеты сопровождаются электронной книгой, но *специализированной*, поскольку в ней широко используются новые операторы и функции, не входящие в стандартные поставки (версии) системы Mathcad. Специализированные электронные книги неработоспособны без полной установки соответствующего пакета расширения.

Библиотеки, готовящиеся на языке С, способны коренным образом изменить назначение системы, например, превратив ее из универсальной в узкоспециализированную систему для физических или радиотехнических расчетов. При всей кажущейся перспективности таких превращений на деле они чреваты лишь большими осложнениями. Готовящиеся в измененных системах документы уже невозможно просматривать в обычных вариантах поставки Mathcad (профессиональной, стандартной и студенческой). Вот почему для новых версий Mathcad поставка библиотек на языке С прекращена (а может быть, просто не начиналась).

Особый вид расширения — так называемые Patch-файлы. Они время от времени выставляются на сайте фирмы Mathsoft и встречаются на компакт-дисках с системами класса Mathcad. Это исполняемые файлы, автоматически модернизирующие систему — вплоть до изменения функций ядра.

Электронными книгами в системе Mathcad называются взаимосвязанные наборы объединенных в тематические разделы документов, относящиеся к самым различным сферам применения системы. Документы вызываются гипертекстовыми и гипермедиа-ссылками друг из друга. Количество взаимосвязанных документов в одной книге может быть большим, до нескольких сотен. В конечном счете, оно определяется объемом дисковой памяти компьютера.

Электронной книгой, как и обычной, можно пользоваться для получения нужной информации и освоения приемов работы с Mathcad. Конечно, для этого необходим компьютер с установленной на нем системой Mathcad. Преимущества электронных книг перед обычными весьма существенны:

- оформление материала в электронных книгах значительно улучшено по сравнению с обычными книгами (цветная графика и текст, различные приемы выделения математических выражений и т. д.);
- поиск информации в электронных книгах облегчен гипертекстовыми ссылками;
- электронные книги — действующие (работающие) объекты, то есть вы можете изменить любые данные в них и тут же получить новые результаты;
- электронные книги допускают модификацию их пользователем (например, замену текстов, ввод новых комментариев и т. д.);
- статьи из электронных книг можно распечатать на принтере;
- электронные книги занимают мало места — на одном компакт-диске можно разместить информацию, эквивалентную той, которая хранится в десятках обычных многостраничных книг;
- электронные книги могут посылаться по электронной почте и через Интернет.

Разумеется, электронные книги Mathcad имеют и недостатки. Один из них заключается в том, что почти все они пока англоязычные (есть книги на других языках, но не на русском). Компьютер с электронной книгой не захватишь на дачу, хотя миниатюрные компьютеры класса notebook эту проблему решают.

Фирма MathSoft выпускает десятки книг по самым различным областям применения системы Mathcad:

- математические вычисления общего характера;

- реализация численных методов;
- расчеты в физике, механике, оптике, электро- и радиотехнике;
- статистические и экономические вычисления и т. д.

После установки (ее процедура ничем не отличается от установки любых приложений в среде Windows) указанные книги пополняют набор книг, входящих в систему, и с ними можно работать так, как будто они являются встроенными. Это обеспечивает обширные возможности расширения системы и ее адаптации к интересующим пользователя задачам и проблемам.

К сожалению, с точки зрения наших пользователей, у большинства книг есть уже упомянутый существенный недостаток — они подготовлены на английском языке. Учитывая, что в книгах используются многие специальные термины, перевод и понимание их достаточно сложны. Да и цена высока — от 50 до 100 долларов. Однако, начиная примерно с 2000 года фирма MathSoft выставила на своем сайте с десяток бесплатных электронных книг, что открывает возможности пользования ими. Эти книги можно найти на Интернет-сайте www.exponenta.ru. Ныне их число превышает четыре десятка.

Вместе с Mathcad поставляется всего одна встроенная электронная книга с информацией о расширениях Mathcad. Она посвящена описанию доступных пакетов расширения системы Mathcad и содержит множество примеров их применения. Это единственная встроенная книга, поставляемая с системой Mathcad (остальные электронные книги требуется устанавливать). Она вызывается из окна центра ресурсов (см. рис. 1.2) щелчком на кнопке Extending Mathcad (Расширения Mathcad).

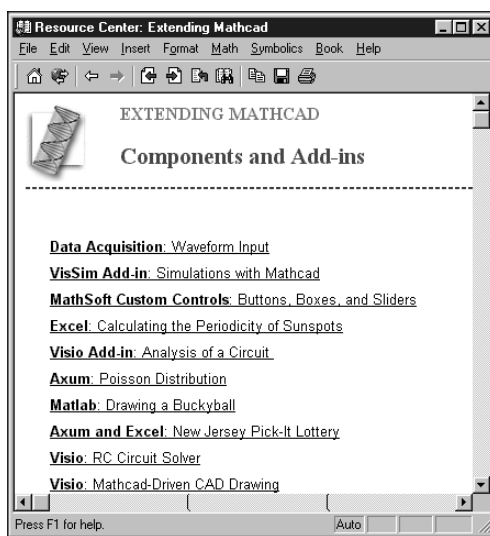


Рис. 2.35. Титульная страница электронной книги о расширениях Mathcad

Титульная страница книги показана на рис. 2.35.

Титульная страница начинается с оглавления книги, реализованного в виде гипертекстовых ссылок. Достаточно активизировать любую из них, и соответствующий документ появится для просмотра. В качестве примера на рис. 2.36 представлен фрагмент

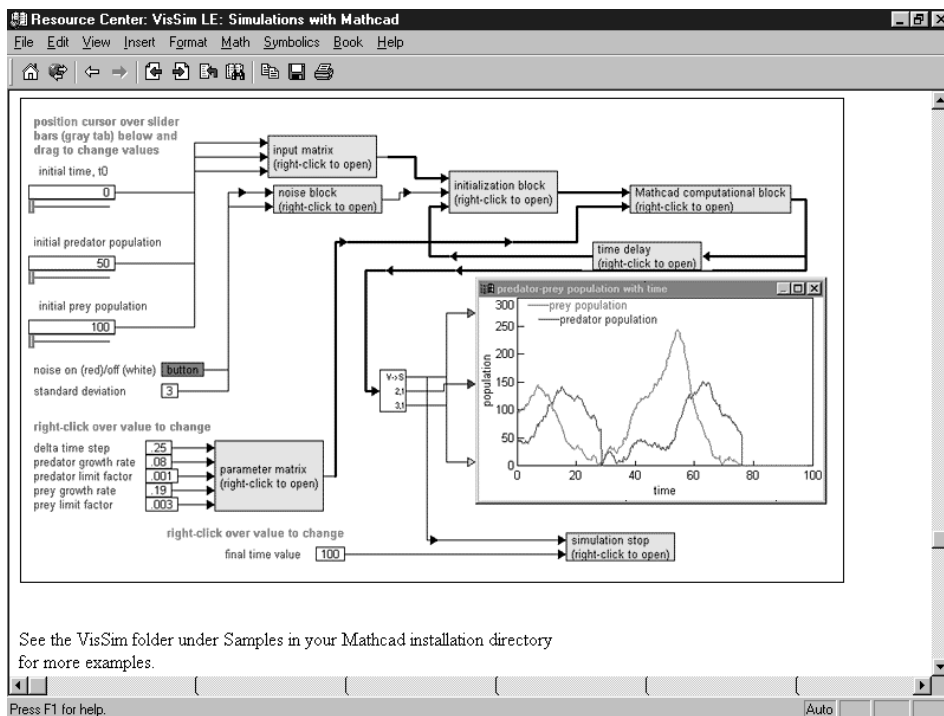


Рис. 2.36. Страница электронной книги про работу Mathcad с системой имитационного моделирования VisSim

мент документа, посвященный описанию применения пакета VisSim совместно с Mathcad для имитационного математического моделирования биологической системы.

Как уже говорилось, большинство электронных книг (см., например, рис. 2.23) не поставляются с самой системой Mathcad. Их нужно приобрести или скачать из Интернета, после чего проинсталлировать. Только после этого список загруженных электронных книг и расширений системы Mathcad появляется при активизации списка команды Handbooks в позиции Help меню и эти книги и входящие в них объекты (определения, формулы, программы и т. д.) можно использовать.

2.7.9. Работа с информационными ресурсами Mathcad 11/12

Новые версии Mathcad 11/12 имеют практически такую же справочную систему, как и Mathcad 2001i, разумеется, учитывающую новые средства, которые в нее введены. Окно справочной системы Mathcad 11 показано на рис. 2.37. Хотя внешне при работе Mathcad 11 в среде Windows XP это окно выглядит несколько иначе и краше, чем окно справки Mathcad 2001i, по своей сути окна ничем не отличаются.

Довольно необычным выглядит появление в окне справки Mathcad 11 русскоязычных надписей для названий кнопок панели инструментов и вкладок. Однако это говорит лишь об одном — при построении интерфейса пользователя использованы элементы интерфейса русскоязычной операционной системы Windows XP и браузера Internet Explorer 6.0. В справке Mathcad 12 этот казус уже устранен.

Более важным является введение в состав Mathcad 11/12 еще одного средства — «бегающей подсказки». Это средство реализовано в виде вопросительного знака со стрелкой ? , который можно «снять» с позиции What's This подменю

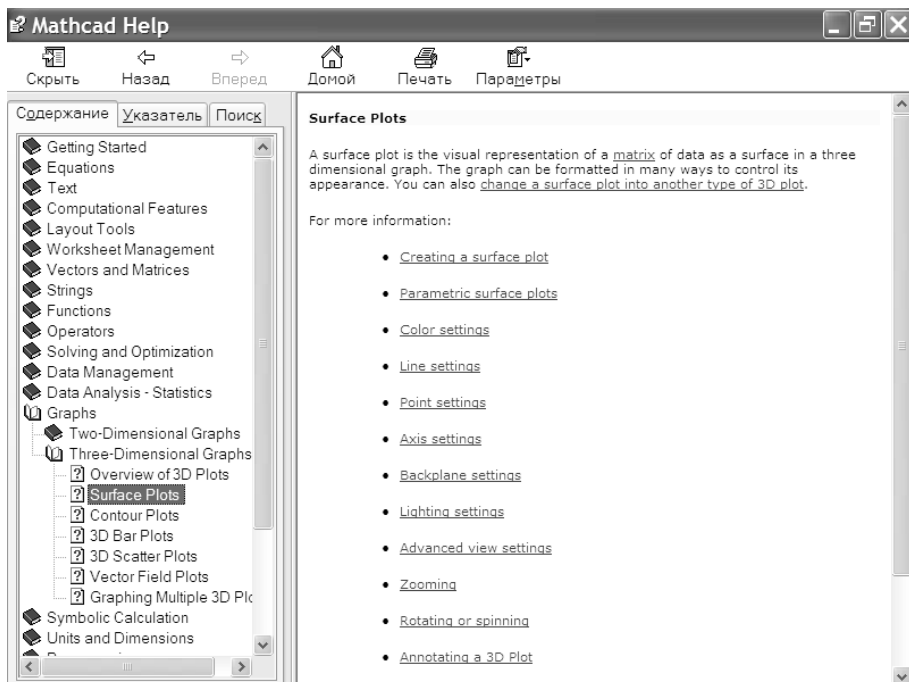


Рис. 2.37. Окно справочной системы Mathcad 11

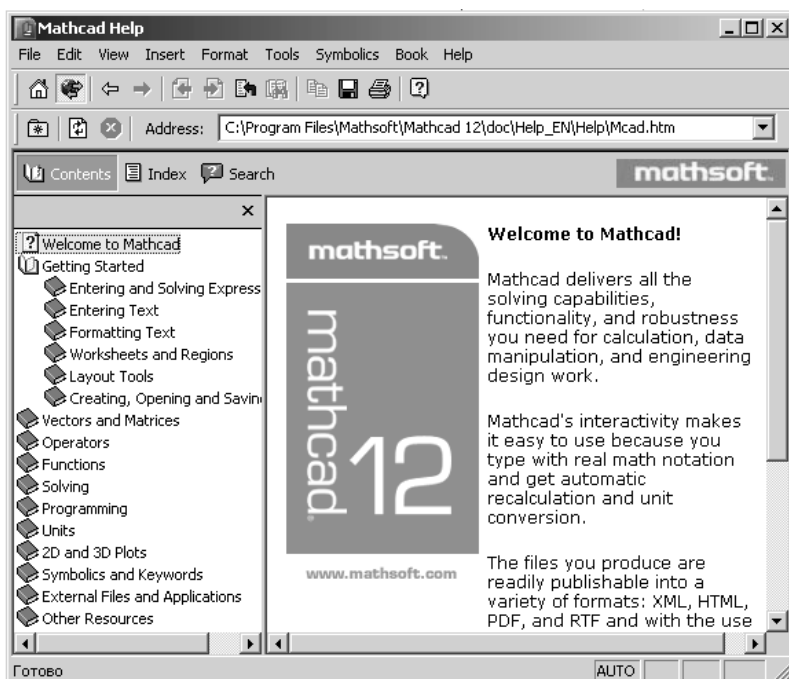


Рис. 2.38. Окно справочной системы Mathcad 12

Help. Наведя этот знак мышью на какой-то объект интерфейса и нажав левую клавишу мыши, можно увидеть появление окна справки с данными об этом объекте.

2.7.10. Оперативная справка по объектам документов

В последние версии системы Mathcad введена очень удобная оперативная справка по объектам документов системы, например выражениям, операторам и функциям. Стоит разместить в них маркер ввода и нажать клавишу F1, как будет тут же выведено окно справки по данному объекту.

2.8. Некоторые особенности интерфейса Mathcad 12

2.8.1. Окно записи файлов в новых форматах

Одно из самых заметных отличий новейшей версии Mathcad 12 — изменение форматов записи файлов документов. Если для предшествующих версий основным был формат .mcd, то Mathcad 12 по умолчанию записывает файлы в новом формате .xmcd. Есть вариант этого формата .xmcdz, дающий компрессию файлов. Эти форматы построены на основе формата XML, идущего на смену формату

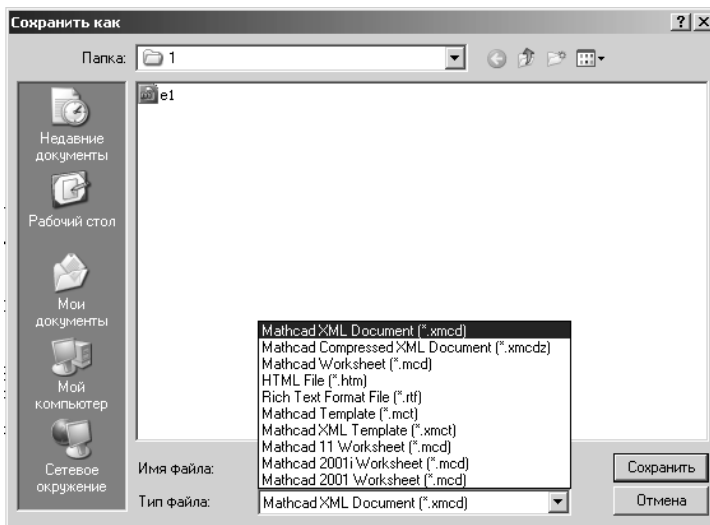


Рис. 2.39. Окно записи файлов документов системы Mathcad 12

MathML. Новые форматы обеспечивают меньшие искажения документов Mathcad, содержащих сложные формульные и графические объекты. Окно записи файлов Save As с новыми форматами файлов представлено на рис. 2.39.

В окне рис. 2.39 список типов файлов представлен в открытом состоянии. Из него видно, что наряду с новыми форматами файлов поддерживается запись (и, соответственно, считывание) и файлов форматов предшествующих версий систем класса Mathcad. В их числе файлы форматов .rtf (Rich) и .htm (формат Интернет-страниц). Заметим, что для записи файлов в последнем формате есть и отдельная команда Save as Web page... в позиции File меню.

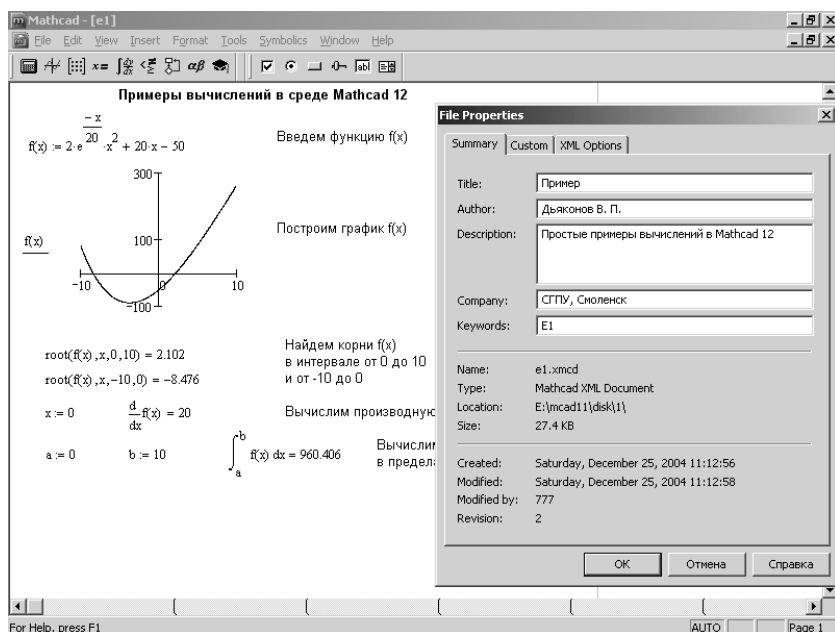


Рис. 2.40. Окно свойств файлов документов Mathcad 12 с открытой вкладкой Summary

2.8.2. Окно свойств файлов

В Mathcad 12 в позиции File меню имеется команда Properties..., выводящая окно свойств файла текущего документа. На документе рис. 2.40 в правой части показано это окно с открытой и заполненной вкладкой Summary. Окно вначале выводится пустым.

Другая вкладка Custom позволяет при необходимости задать установки пользователя, например значения переменных, которые могут использоваться при запуске документа на исполнение. Третья вкладка XML Options служит для контроля и изменения опций, относящихся к XML-файлам. Здесь можно установить формат рисунков, их включение в текст и качество рисунков (для формата JPG), а также задать опцию записи в файл результатов вычислений.

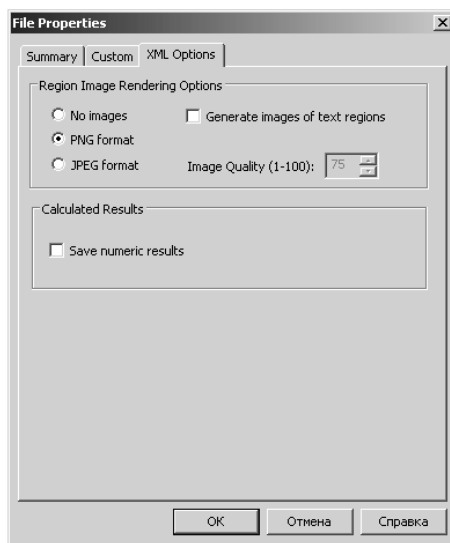


Рис. 2.41. Окно свойств файлов документов Mathcad 12 с открытой вкладкой XML Options

2.8.3. Отправка документов по электронной почте

В Mathcad 12, как и в предшествующих версиях системы, возможна быстрая передача файлов документов по электронной почте (Email). Для этого служит бра-

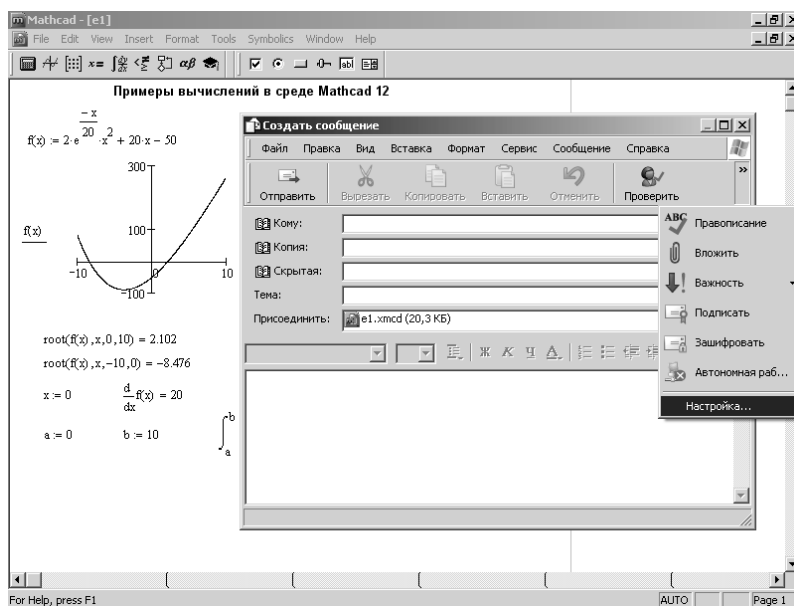


Рис. 2.42. Подготовка к отправке документа Mathcad 12 по электронной почте

узер текущей операционной системы. Например, в Windows XP это Microsoft Internet Explorer 6.0 [43, 44]. Рисунок 2.42 показывает подготовку к отправке документа Mathcad при исполнении команды Send... в позиции File меню.

После исполнения указанной команды появляется окно почтового клиента Outlook Express, показанное на рис. 2.42 в правой части окна документа. Файл текущего документа задается как прикрепленный файл. Пользователю остается указать адрес, куда отправляется сообщение, указать тему, подготовить само сообщение и нажать кнопку Отправить в панели инструментов. Соединение с удаленным компьютером провайдера может быть осуществлено заранее или начнется автоматически после нажатия кнопки Отправить.

2.8.4. Настройка состава кнопок панелей инструментов

Как у многих приложений операционных систем Windows, в Mathcad 12 есть возможность управления составом кнопок панелей инструментов. Для этого надо вывести курсор мыши в свободное от кнопок пространство той или иной панели

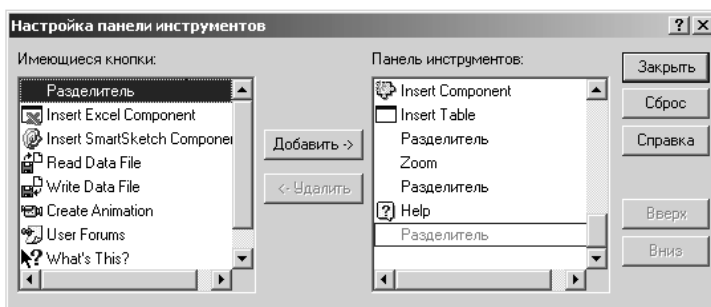


Рис. 2.43. Окно настройки панели инструментов Toolbar

инструментов и нажать правую клавишу мыши. В контекстном меню ее надо выбрать команду *Customize...* Ее исполнение, например для панели *Toolbar*, приводит к появлению окна, показанного на рис. 2.43.

В левой части окна представлены объекты (в основном кнопки), которые можно переместить в соответствующую панель инструментов, а в правой части объекты, которые в ней уже есть. Операции перемещения объектов или их удаления с помощью кнопок *Добавить->* и *<-Удалить* вполне очевидны. Знать эту возможность для студентов не обязательно, а порой просто нежелательно.

2.8.5. Выбор инструментов контроля

В Mathcad 11/12 введены новые средства контроля, которые можно вставлять в документы. Это различные кнопки, «ящики» и др., служащие для повышения интерактивности при работе с документами. Например, можно ввести линейный ре-

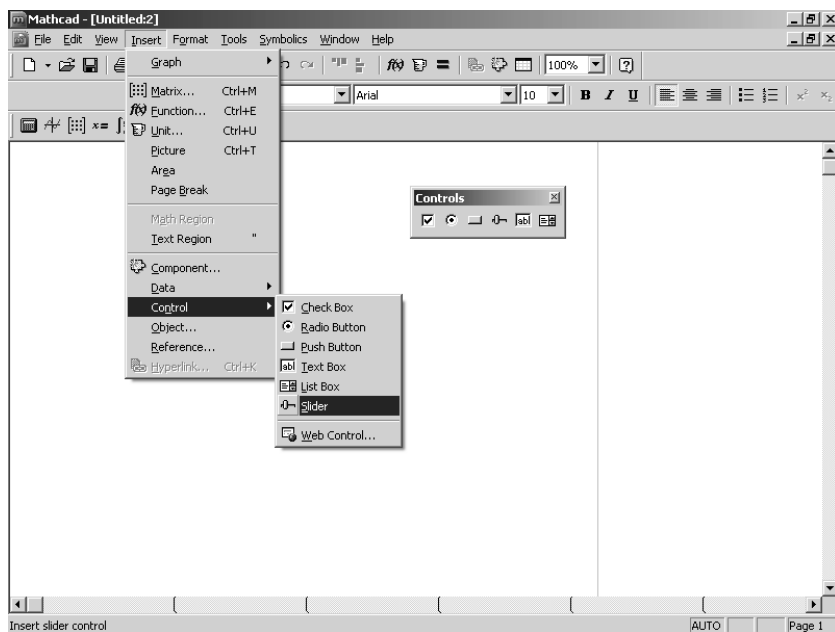


Рис. 2.44. Средства вставки элементов контроля

гулятор для изменения значения некоторой переменной. Для включения этих средств в документы служит панель инструментов *Controls* и (или) подменю с тем же именем в позиции вставки *Insert*. Эти средства и представлены на рис. 2.44.

Глава 3. Подготовка документов

Характерной особенностью документов в системе Mathcad является сочетание текстовых комментариев, формульных блоков и иных средств вычислений с результатами последних, представленными в числовой, табличной или графической форме. Именно такое сочетание (в отличие от малопонятных программ на обычных языках программирования) обеспечивает высокую степень *визуализации* всего процесса вычислений, его наглядность и порою кажущуюся простоту. При этом на первое место в процессе создания математического документа выдвигается корректная подготовка и исполнение алгоритмом выполняемых вычислений с учетом всех возможностей системы Mathcad. В этой главе мы рассмотрим основные формальные аспекты подготовки документов в системе Mathcad, в частности работу с файлами, редактирование и печать, средства вычислений и т. д. Описанию алгоритмов решения задач в среде Mathcad и их реализаций посвящены последующие главы.

3.1. Операции с файлами документов

3.1.1. Типы файлов в Mathcad

Документы и данные вычислений в Mathcad 2001 хранятся на магнитных дисках в виде файлов. *Файлом* называют имеющую имя упорядоченную совокупность данных или кодов программ, размещенную на том или ином носителе, обычно на жестком, гибком или компакт-диске.

Файлы в Mathcad могут иметь различную внутреннюю структуру, именуемую *форматом*. Он указывается после имени файлов в виде расширения, записываемого через точку. Файлы документов в Mathcad имеют расширение .mcd. Такие файлы легко прочитать и модифицировать в любом текстовом редакторе. Однако интерпретация файлов понятна лишь пользователям, хорошо знакомым с программированием.

Файлы документов содержат полный текст программы, выводившей документ в окно редактирования, с указаниями координат расположения блоков, фактического содержания и характера выполняемых операций, форматов представления информации и т. д. Таким образом, файл является, по сути, программой, записанной на внутреннем (промежуточном) языке программирования системы.

Файлы могут содержать результаты вычислений, а также графические объекты. Предусмотрена возможность записи документов и в особом формате RTF, созданном для хранения сложных многокомпонентных данных (содержащих тексты и графики с сохранением, по возможности, их правильного взаимного расположения).

Важно отметить, что даже при сохранении документов со сложными рисунками используется запись не их растровой (битовой) копии, а именно *программы вывода* документа. Поэтому файлы с расширением .mcd невелики по размеру и их легко передавать современными средствами телекоммуникаций, например через Интернет. Для записи они не требуют большого пространства на дисках. Впрочем, для передачи через Интернет Mathcad 2001i/11 позволяет записывать файлы в особом гипертекстовом формате HTML (с расширением .htm). Поддерживается также формат шаблонов документов Template (.mct) и форматы Mathcad версий 6, 7, 8 и 2000 (.mcd).

3.1.2. Меню файловых операций File

Основные команды для работы с файлами документов собраны в меню File (Файл), показанном на рис. 3.1.

Команды в меню File разбиты на группы. В первую группу входят следующие команды:

- New (Создать) — открыть окно для создания нового документа;
- Open (Открыть) — открыть окно для выбора и загрузки существующего документа;
- Close (Закреть) — закрыть документ.

Вторая группа команд служит для сохранения документов:

- Save (Сохранить) — сохранить на диске текущий документ под прежним именем;
- Save As (Сохранить как) — сохранить на диске текущий документ под новым именем.

Третья «группа» команд в Mathcad 2001 содержит всего одну команду Send (Отправить), предназначенную для отправки документа по электронной почте. В прежних версиях Mathcad эта команда открывала доступ к подменю, команды из которого, связанные с Интернетом, перекочевали в центр ресурсов.

Четвертая группа команд служит для подготовки к печати и печати документов:

- Page Setup (Параметры страницы) — открытие диалогового окна для установки параметров страницы;
- Print Preview (Просмотр) — предварительный просмотр документа перед печатью;
- Print (Печать) — печать документа.

Пятая группа представлена командой Exit (Выход), которая позволяет выйти из среды Mathcad. Перед этой командой имеется перечень файлов, открывавшихся последними. Этот перечень позволяет загрузить нужный файл без предварительного поиска в файловой системе компьютера.

В меню File (как и других меню) помимо имени команды указаны изображение соответствующей кнопки на панели инструментов (если она есть) и комбинация клавиш (называемых иногда «быстрыми» или «горячими»), которая вводит соответствующую команду без помощи меню. Опытные пользователи часто пользуются «горячими» клавишами.

В новых версиях Mathcad состав меню File сохранился, но добавилась новая команда — Save as Web Page..., которая позволяет записать файл документа в формате HTML, применяемом для создания Web-страниц.

Вернемся к широко используемым командам первой группы и рассмотрим их более подробно.

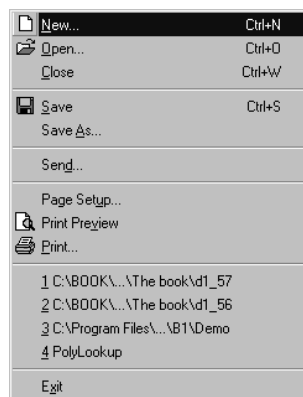


Рис. 3.1.
Раскрытое меню File

3.1.3. Создание нового документа

Кнопка New (Создать) с изображением чистого листа на панели инструментов или команда New в меню File (Файл) служат для создания нового документа. Щелчок на кнопке New ведет к открытию пустого окна и заданию стиля докумен-

та Normal (Обычный). А вот команда New меню File выводит окно стилей будущего документа (рис. 3.2).

Каждый стиль документа задается своим шаблоном. Самым простым шаблоном является шаблон Normal — он соответствует документу с изначально чистым листом. Пользователь, однако, может создавать свои шаблоны. Для этого создается документ в нужном стиле пользователя и записывается в виде шаблона — файла с расширением .mst. Обычно шаблоны хранятся в папке Template.

Начиная с Mathcad 2001i названия стилей стали вполне понятными и не требуют особых пояснений. Появился и новый стиль Web Page (Web-страница), позволяющий записывать документы Mathcad в виде полноценных Web-страниц в формате HTML с элементами нового формата MathML для передачи по Интернету сложных математических формул в их естественном виде.

После выбора стиля будет создан новый документ с именем Untitled: N, где N — порядковый номер документа. В качестве примера на рис. 3.3 показан слегка измененный шаблон для документов в стиле Electronic book (Электронная книга).

После создания шаблона документа можно начинать вводить его содержимое с клавиатуры с применением (при необходимости) палитр математических знаков и функций. Примеры этого мы уже рассматривали. Обратите внимание, что шаблон Electronic book (и другие шаблоны, кроме Normal) добавляют в документ различные текстовые блоки в виде заголовков разделов, поясняющих надписей и др.

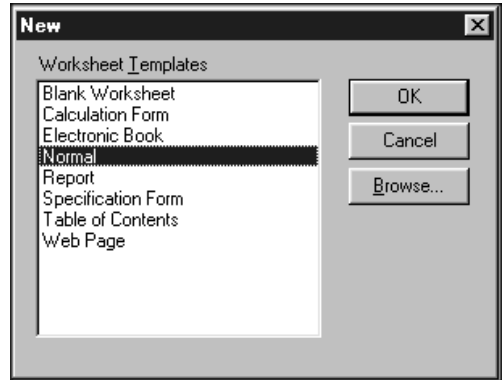


Рис. 3.2. Окно выбора стиля документа

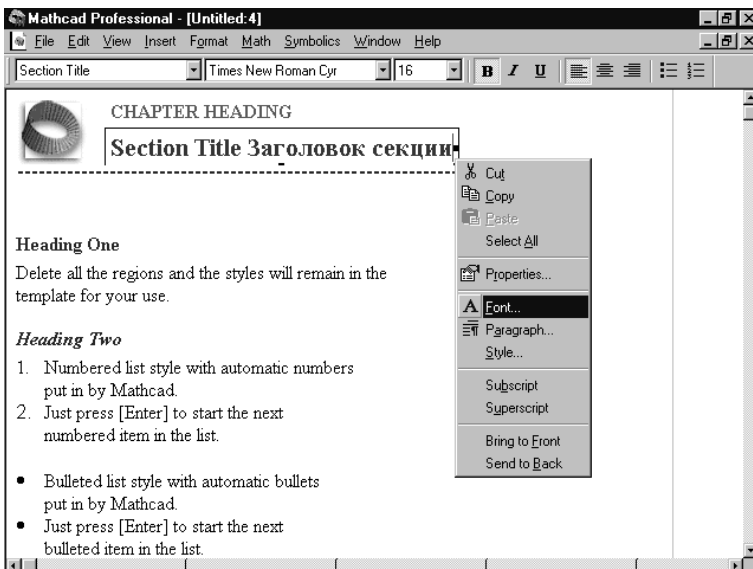


Рис. 3.3. Документ в стиле электронной книги

Они задаются на английском языке, но, как видно из рисунка, их легко сделать русскоязычными.

При открытии документа с пустым окном редактирования (в стиле Normal) курсор ввода (маленький крестик, красный при работе с цветным дисплеем) устанавливается в верхний левый угол окна с небольшим отступом от краев. Становятся доступными все операции ввода объектов документа и их редактирования. Текущий документ приобретает имя Untitled: (в переводе «безымянный») и после этого имени в верхней части окна указывается номер документа.

Поскольку Mathcad — многооконная система, возможно существование нескольких окон с именем Untitled: N, но с разными номерами N.

Если в окне выбора стиля документа щелкнуть на кнопке Browse (Обзор), то можно перейти к загрузке ранее созданного файла шаблона. При этом открывается окно, дающее доступ к файловой системе ПК (рис. 3.4). С его помощью можно открыть любую папку и загрузить из нее нужный файл. Для этого нужно выделить файл и щелкнуть на кнопке Open (Открыть).

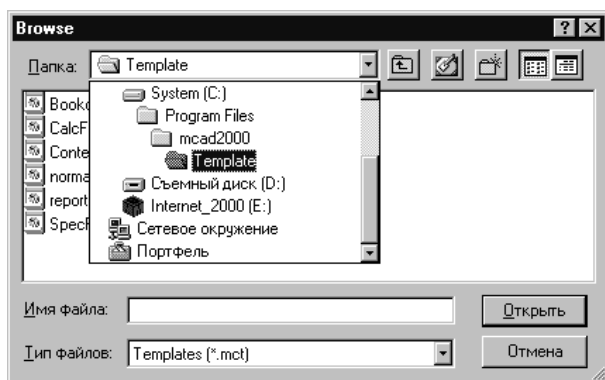


Рис. 3.4. Окно Browse для загрузки файла и доступа к папкам файловой системы

Окно показано с раскрытым списком Look In (Папка), предназначенным для выбора дисков и папок файловой системы ПК. Детали интерфейса этого окна (и подобных ему) будут описаны ниже.

3.1.4. Открытие существующего документа

Команда Open (Открыть) меню File (Файл) служит для загрузки документа из файла. При выборе этой команды появляется диалоговое окно выбора файла, которое по виду аналогично окну Browse (Обзор), представленному на рис. 3.4. Тот же эффект дает щелчок в панели инструментов системы на кнопке с изображением открывающейся папки.

В этом окне с помощью раскрывающегося списка Look In (Папка) и вертикальной полосы прокрутки можно «пройтись» по всем дискам, папкам и файлам текущей папки. Чтобы выбрать нужный файл для загрузки, его требуется выделить мышью. Выбор завершается двойным щелчком на имени файла, нажатием клавиши Enter или щелчком на кнопке Open (Открыть). Любое из этих действий приводит к загрузке документа в текущее окно системы. Кнопка Cancel (Отмена) или клавиша Esc позволяет отказаться от загрузки.

Помимо кнопок Open и Cancel в окне Open (как и в окне Browse) имеется еще пять кнопок. Перечислим их слева направо:

- Up One Level (На один уровень вверх);
- Desktop (Обзор рабочего стола);
- Create New (Создание новой папки);
- List (Список);
- Details (Таблица).

Кнопка Up One Level (На один уровень вверх) позволяет отобразить содержимое более «старшей» в иерархии папки, кнопка Desktop (Обзор рабочего стола) открывает папку рабочего стола Windows, а кнопка Create New (Создание новой папки) позволяет создать новую папку внутри текущей.

Две последние кнопки устанавливают режим отображения файлов и папок в окне — в форме списка или в форме таблицы. Табличная форма менее компактна, но дает больше данных о каждом файле или папке.

После загрузки файла с документом его текст появляется в текущем окне и по умолчанию начинается обработка документа в режиме автоматических вычислений. Можно отключить этот режим, тогда загруженный документ не будет обрабатываться до выбора соответствующей команды.

Для вызова одного из ранее использовавшихся документов достаточно щелкнуть мышью на его имени в списке, находящемся в меню File (Файл) над командой Exit (Выход).

При работе с несколькими документами возможна ситуация, когда будет открыто максимально допустимое число окон. Тогда загрузка нового файла становится невозможной и система выводит сообщение с предложением закрыть окна.

3.1.5. Запись документа на диск без переименования

Для записи файла документа с текущим именем на диск используйте команду Save (Сохранить) меню File (Файл) или кнопку на панели инструментов с изображением гибкого магнитного диска. Эта команда записывает файл на диск под его текущим именем и с учетом всех изменений, которые вносились в документ в ходе его редактирования, доработки или дополнения. Такая операция удобна, если вы совершенствуете ранее созданный и имеющий заданное вами имя документ. Она дает быстрый способ записи файла на диск без поиска его места в файловой структуре и без изменения имени.

ВНИМАНИЕ *Следует с осторожностью пользоваться командой Save в том случае, когда вы модернизируете какой-либо документ, но желаете при этом сохранить оригинал, тем более что эта команда выполняется без выдачи каких-либо предупреждающих сообщений. В случае применения команды Save исходный файл-оригинал будет заменен новым. Чтобы этого не произошло, следует пользоваться командой Save As (Сохранить как), описанной в следующем подразделе.*

СОВЕТ *Командой Save рекомендуется пользоваться периодически при подготовке сложных документов. Это позволяет сохранить все изменения в документе и избежать потери документа при ошибочных действиях, например, при случайном выключении компьютера до записи файла.*

3.1.6. Запись документа на диск с переименованием

Команда Save As (Сохранить как) отличается от команды Save (Сохранить) тем, что файл записывается на диск не сразу. Вместо этого открывается диалоговое окно Save (Сохранение), в котором можно ввести новое имя файла и/или выбрать для него новую папку (рис. 3.5). Тем самым можно записать доработанный документ под новым именем, сохранив оригинал.

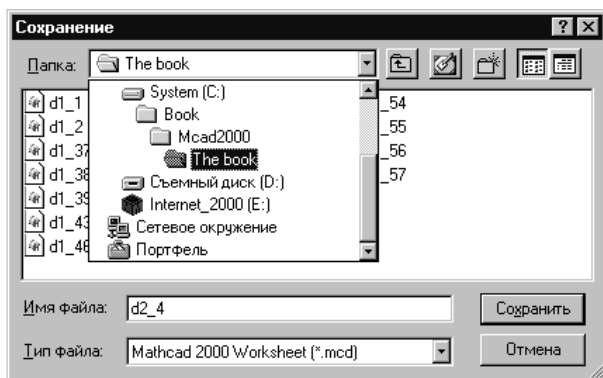


Рис. 3.5. Диалоговое окно для записи файла с указанием его имени

Имя файла может быть коротким, без указания диска и папки. Тогда файл будет помещен в текущую папку. Однако можно указать и полное имя файла, например: d:\wmcad\book\demo.mcd. Тогда файл demo.mcd будет записан на диск d: в папку book, вложенную в папку wmcad.

3.1.7. Закрытие документа

Команда Close (Закреть) меню File (Файл) закрывает окно с текущим документом, при этом система переходит к работе с окном предшествующего документа (либо к пустому окну, если другие открытые документы отсутствуют). Если открыто максимально допустимое число окон, попытка загрузки еще одного документа станет невозможной (о чем будет выведено соответствующее сообщение). Не рекомендуется использовать более десятка окон, поскольку это может вызвать перерасход памяти и сбой в работе.

Если какой-либо из документов в закрываемых окнах подвергался модификации, то система выдаст запрос о необходимости сохранения изменений (рис. 3.6). Если документ не модифицировался, соответствующее окно просто закрывается. Закреть окно также можно, используя кнопку или команду меню управления окном.

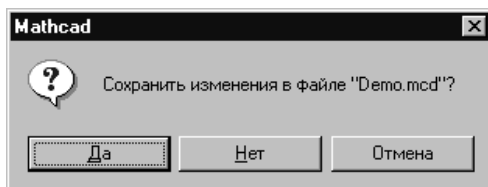


Рис. 3.6. Окно с запросом о сохранении изменения в файле

3.1.8. Отправка документа по электронной почте

Документы можно отправить по электронной почте, если ваш ПК имеет модем и подключен к службе электронной почты или Интернету. Для этого, при наличии активного окна с документом, можно воспользоваться командой **Send** (Отправить) меню **File** (Файл). При работе в среде операционной системы Windows это приведет к запуску программы электронной почты по умолчанию и созданию нового сообщения. Текущий файл системы Mathcad вкладывается в сообщение, а вам остается задать адрес, по которому отправляется сообщение, адрес, по которому направляется копия сообщения, тему сообщения и его текст. Чаще всего отправка сообщения осуществляется с помощью программы электронной почты Outlook Express [43, 44].

3.1.9. Выход из системы

Команда **Exit** (Выход) меню **File** (Файл) используется для окончания работы с системой. Перед этой операцией необходимо побеспокоиться, чтобы все загруженные ранее документы, подвергавшиеся редактированию и модификации, были сохранены на диске с помощью команды **Save As** (Сохранить как) или **Save** (Сохранить). Тогда при выборе команды **Exit** все окна документов будут последовательно закрыты без каких бы то ни было запросов. Если пользователь забыл записать какой-либо документ на диск, система сообщит об этом (соответствующий запрос показан на рис. 3.6). Нужно щелкнуть на кнопке **Yes** (Да), если документ требуется сохранить, в противном случае — на кнопке **No** (Нет).

3.2. Печать и просмотр документов

Печать документов — одна из типовых возможностей системы Mathcad. В общем случае она предполагает установку параметров печатаемой страницы, ее предварительный просмотр и затем уже печать на принтере выбранного типа с возможной коррекцией параметров печати. Перед печатью полезно просмотреть документ как в его окне, так и в специальном окне предварительного просмотра.

3.2.1. Установка параметров страницы

Команда **Page Setup** (Параметры страницы) меню **File** (Файл) служит для установки параметров печатаемых страниц. При ее выборе появляется диалоговое окно, показанное на рис. 3.7.

В разделе **Margins** (Поля) задается ширина отступа (поля) со всех сторон бумажного листа: **Left** (Слева), **Right** (Справа), **Top** (Сверху) и **Bottom** (Снизу). Отступы устанавливаются в миллиметрах. Кнопка **Cancel** (Отмена) отменяет операцию, а кнопка **OK** фиксирует введенные данные.

В разделе **Paper** (Бумага) окна можно задать параметры листа бумаги, на котором будет производиться печать. Раздел **Orientation** (Ориентация) служит для установки ориентации печати: обычной, книжной (**Portrait**) и вдоль листа, альбомной (**Landscape**). При этом макет будущей страницы можно видеть в верхней части окна.

Команда **Page Setup** влияет и на вид страницы документа в окне редактирования. В частности, она задает положение линий раздела страниц. Например, если нежелательно видеть длинную вертикальную черту раздела страниц, то достаточно

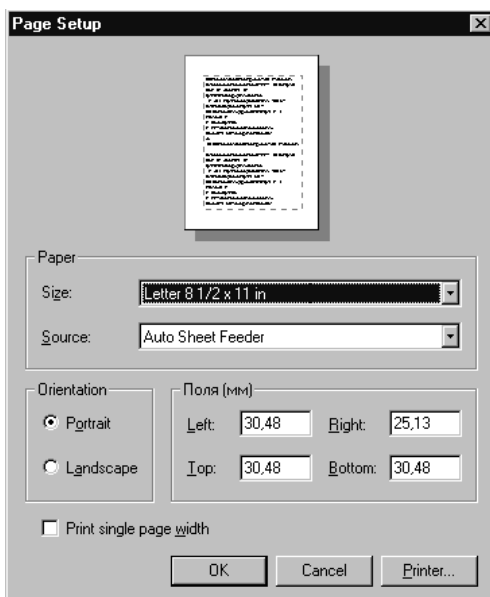


Рис. 3.7. Диалоговое окно установки параметров печатаемой страницы

установить переключатель Landscape (Альбомная). Это сделано для большинства документов данной книги, представленных рисунками или копиями экрана.

3.2.2. Выбор принтера

Из окна установки параметров страницы можно перейти к установкам параметров принтера, щелкнув на кнопке Printer (Принтер). При этом появится окно Print Setup (Макет страницы), показанное на рис. 3.8, с раскрывающимся списком для выбора принтера (такой выбор возможен, если в системе установлено два или более принтера). Один из них используется по умолчанию, его название указано в поле раскрывающегося списка.

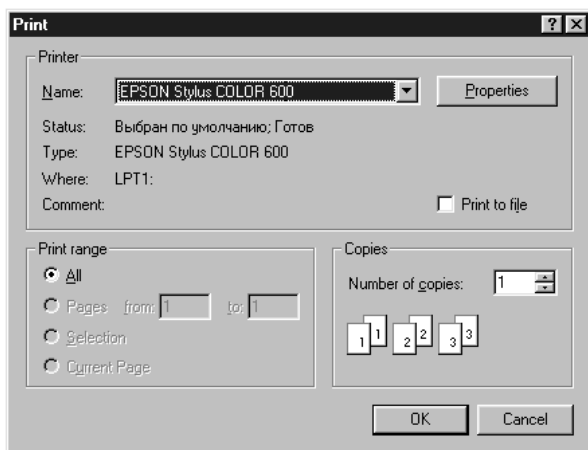


Рис. 3.8. Выбор принтера

Для установки параметров принтера достаточно щелкнуть на кнопке Properties (Свойства). Появится окно свойств принтера, вид которого зависит от типа установленного принтера.

3.2.3. Предварительный просмотр

Даже одна страница большого документа не видна полностью на экране дисплея. Поэтому перед печатью полезно просмотреть расположение элементов документа на странице. Для этого служит команда Print Preview (Просмотр) меню File (Файл), которая выводит контрольное окно с изображением будущей напечатанной страницы или двух страниц (рис. 3.9).

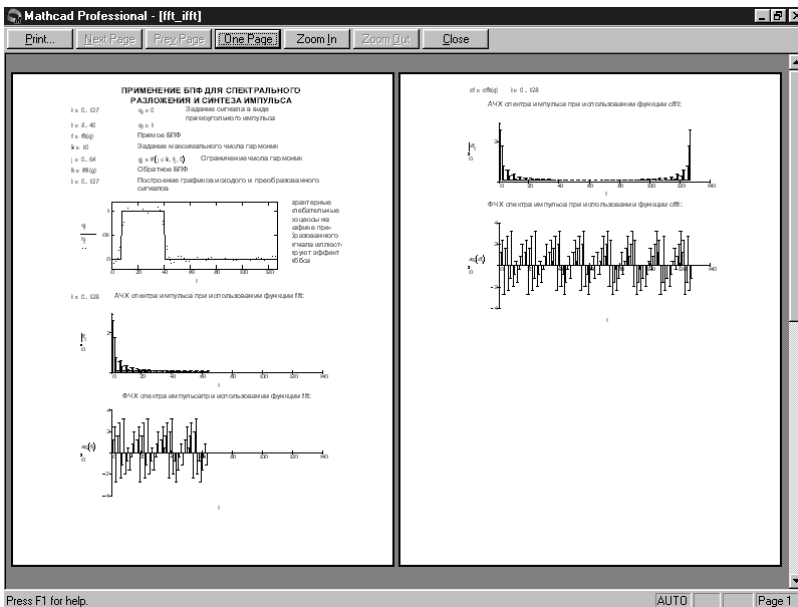


Рис. 3.9. Предварительный просмотр страницы

Вверху окна видны кнопки управления режимом просмотра страниц. Страницу можно рассматривать в обычном и в увеличенном виде. Для просмотра в увеличенном виде надо щелкнуть на кнопке Zoom In (Увеличить), а для возврата к обычному размеру — на кнопке Zoom Out (Уменьшить).

Система предварительного просмотра позволяет просматривать ряд страниц документа, расположенных в разных окнах. Для перемещения от страницы к странице используются кнопки Next Page (Следующая) и Prev Page (Предыдущая). Для одновременного просмотра двух смежных страниц служит кнопка Two Page (Две страницы). Эти кнопки доступны только в том случае, когда документ состоит из двух или более страниц.

3.2.4. Печать документов

Команда Print (Печать) меню File (Файл) инициирует печать документа. Нередко эта команда используется как основная, если существующие параметры страницы и принтера устраивают пользователя. Команда Print дублируется кнопкой с изображением принтера на панели инструментов.

При выборе команды Print появляется диалоговое окно, показанное на рис. 3.10 справа. Это окно, так же как и рассмотренное ранее окно выбора принтера (см. рис. 3.8), не принадлежит системе Mathcad — используется стандартное окно печати Windows.

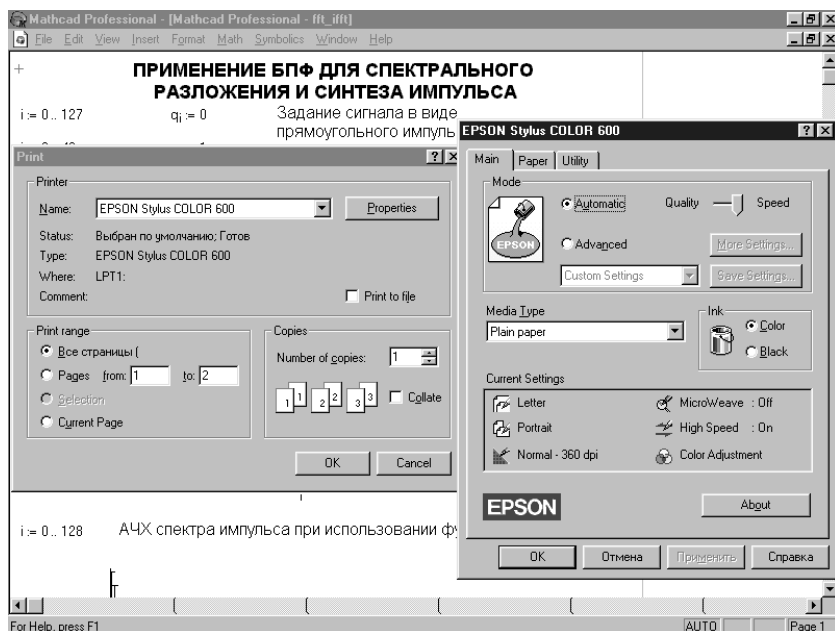


Рис. 3.10. Окно печати (слева) и окно установки свойств печати (справа)

При печати, как правило, сохраняется вид исходного документа. Разумеется, если он многостраничный, то печатаются отдельные страницы (на экране они разделены характерными пунктирными линиями). Современные принтеры (особенно лазерные и струйные) имеют более высокое разрешение при печати, чем разрешение дисплея, поэтому качество отпечатанного документа у них выше, чем видимое на экране.

Документы Mathcad могут иметь удвоенную (в страницах) ширину и значительную длину. Тогда вначале печатаются все страницы левой половины широкого документа, а затем уже страницы правой половины, даже если они пустые (разумеется, при этом принтер вытолкнет пустой лист). Склеив страницы, вы получите полную распечатку документа.

ПРИМЕЧАНИЕ Вид окон управления печатью зависит от применяемого принтера, а точнее от драйвера печати, который устанавливается в зависимости от выбора принтера.

3.2.5. Выделение и показ всех блоков

Команда Select All (Выделить все) меню Edit (Правка) выделяет все объекты пунктирными прямоугольниками. Это позволяет не только видеть взаимное расположение блоков, но и оценить степень их перекрытия. Как правило, оно нежелательно.

При переносе объектов или при их редактировании возможно частичное и даже полное наложение блоков одних объектов на другие, при этом части некоторых объектов могут оказаться скрытыми. Выделение всех объектов полезно для контроля такой ситуации.

Команда Regions (Области) меню View (Вид) позволяет показать все блоки и обеспечивает закраску промежутков между ними серым цветом. Сами блоки при этом приобретают белый фон (рис. 3.11).

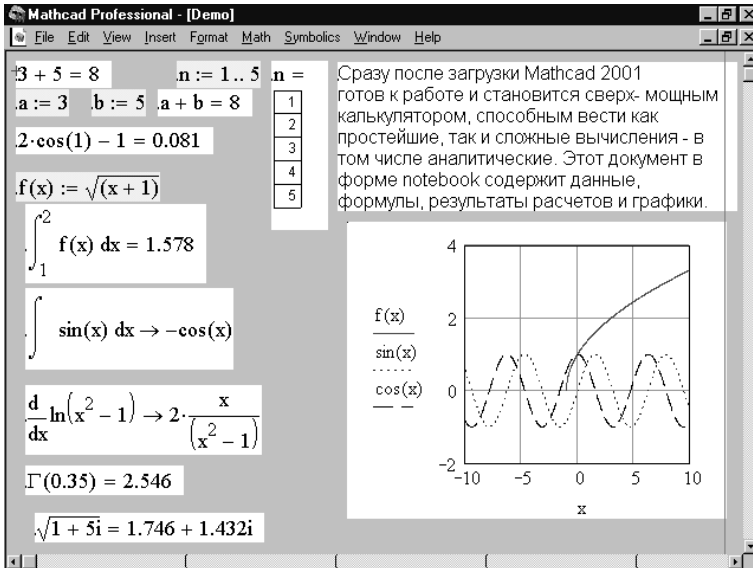


Рис. 3.11. Вид документа с выделенными блоками

Выделение блоков особенно полезно для оценки их взаимного расположения и наложения. Заглядывая вперед (см. подраздел «Разделение областей документов» в разделе «Форматирование электронных документов» главы 5), отметим, что команда Separate Regions (Разделение областей) меню Format (Формат) позволяет автоматически раздвинуть блоки для устранения наложения их друг на друга.

3.2.6. Изменение масштаба документа

Иногда желательно изменить масштаб отображения документа. Для этого служит команда Zoom (Масштаб) меню View (Вид). Она позволяет вывести окно с переключателями доступных масштабов (рис. 3.12).

Для выбора масштаба достаточно установить требуемый переключатель. На рис. 3.13 представлен документ, показанный на рис. 3.12, но в уменьшенном масштабе (75 %).

Пределы изменения масштаба обычно ставит разрешающая способность дисплея. При уменьшении масштаба до 50 % на экране можно видеть две страницы документа. Отметим, что масштаб можно менять и иначе — выбором разрешения дисплея. Однако для работы с высоким разрешением желательно применять дисплеи с увеличенным размером экрана по диагонали (17, 19 или 21 дюйм), которые дороже, чем распространенные дисплеи с размером экрана по диагонали в 14 или 15 дюймов.

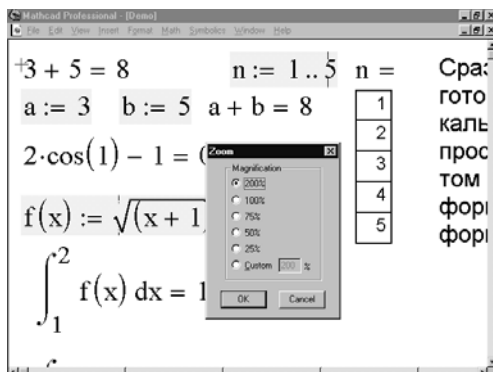


Рис. 3.12. Вид документа в увеличенном масштабе (200 %) и окно масштабирования

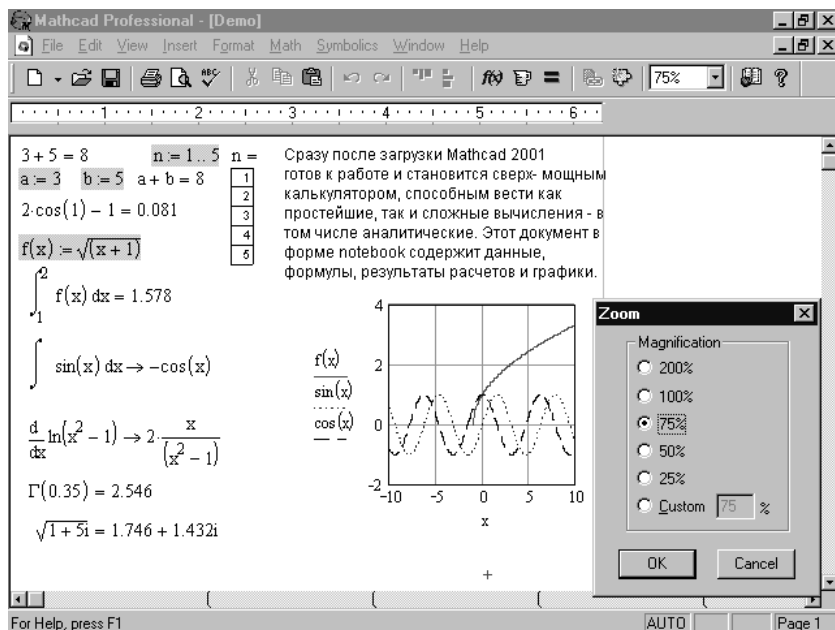


Рис. 3.13. Вид документа в уменьшенном масштабе (75 %) и окно масштабирования

3.3. Ввод с помощью клавиатуры

Редактирование документов возможно как с применением мыши, так и с помощью клавиатуры. В любом случае прежде всего полезно знать возможности графического курсора.

3.3.1. Формы курсора ввода

При перемещении по документу графический курсор может принимать одну из трех описанных ниже форм.

- Крестообразный красный курсор служит для указания места ввода новых блоков (текстовых, формульных или графических). Курсор имеет такой вид только вне пространства блоков, то есть на пустом месте экрана, и может перемещаться клавишами управления курсором или устанавливаться мышью (щелчком).
- Курсор в виде красной вертикальной черты (курсор ввода) служит для указания на отдельные элементы блоков и обычно используется для ввода данных и заполнения шаблонов. В текстовых блоках позволяет указать места вставки или удаления отдельных символов.
- Курсор в виде синих уголков разного размера (┌ или ┐) позволяет выделять отдельные части выражения или выражение целиком. Вид курсора зависит от направления ввода. Нажатие клавиши Ins или клавиш перемещения курсора по горизонтали меняет направление ввода.

3.3.2. Выделение объектов с помощью клавиш

В целом для выделения используются указанные ниже клавиши:

- ↑ — превращает курсор в выделяющий уголок и расширяет его;
- ↓ — сужает выделяющий уголок;
- → — перемещает курсор и меняет вид уголка;
- ← — перемещает курсор и меняет вид уголка;
- Shift+↑ — выводит курсор из выражения вверх в свободное поле, делая его крестообразным;
- Shift+↓ — выводит курсор из выражения вниз в свободное поле;
- Shift+→ — выводит курсор из выражения вправо в свободное поле;
- Shift+← — выводит курсор из выражения влево в свободное поле;
- Пробел — заключает в рамку операнд, действуя как несколько нажатий клавиши ↑, и выводит курсор из выделенного выражения;
- Ins — меняет направление охвата выделяемого объекта.

3.3.3. Применение управляющих клавиш

По мере освоения системы все большую помощь в редактировании оказывает знание функций управляющих клавиш и их комбинаций. Ниже дан перечень осуществляемых ими операций:

- Tab — в тексте перемещает курсор на начало следующего слова, в выражении управляет выделением частей блока (в частности, выделяя выражения в скобках);
- Shift+Tab — в тексте перемещает курсор в начало предыдущего слова, в выражении управляет выделением частей блока;
- PgUp — перемещает курсор и вызывает прокрутку на пять строк вверх;
- PgDn — перемещает курсор и вызывает прокрутку на пять строк вниз;
- Ctrl+PgUp — вызывает прокрутку на одно окно вверх;
- Ctrl+PgDn — вызывает прокрутку на одно окно вниз;
- Home — устанавливает курсор в начало предыдущего блока;
- Ctrl+Home — вызывает прокрутку с установкой курсора в начало документа;
- Ctrl+End — вызывает прокрутку с установкой курсора в конец документа.

Действия некоторых клавиш, связанные с выделением, неоднозначны, и для их интуитивного применения требуется определенный опыт работы. При редактировании текстовых блоков особых трудностей не возникает — оно выполняется так же, как в большинстве текстовых редакторов, и будет подробнее рассмотрено в дальнейшем.

Математические выражения не столько набираются, сколько конструируются. При этом учитывается определенная структура выражений и иерархия выполняемых операций. Конструирование выражений облегчается средствами выделения отдельных фрагментов выражений. Полезно помнить, что все, попадающее в ходе выделения в уголки курсора, становится операндом для следующего вводимого оператора. Операторы возведения в степень, извлечения корня и деления являются «цепкими» операторами. После их ввода все, что набирается следом, становится соответственно показателем степени, подкоренным выражением или знаменателем дроби. Для выхода из операнда обычно приходится дважды нажать клавишу Пробел.

Как правило, удобно начинать выделение мышью, поскольку она позволяет сразу и точно указать место выделения, с последующим расширением или сужением зоны выделения с помощью управляющих клавиш. При определенном навыке все выделения можно выполнять при помощи мыши, расширяя или сужая область выделения в нужном направлении. Выделять можно как целые математические выражения, так и их части.

Нажатие клавиши `Ins` меняет направление ввода и положение курсора, что позволяет оперативно управлять вводом фрагментов формул. Например, если курсор имеет вид уголка, направленного влево (\lrcorner), то ввод будет идти вправо (курсор как бы поддерживает вводимое выражение). Пусть, например, требуется набрать выражение $(a + b)$ и затем $\sin(a + b)$, то есть сделать ранее набранное выражение аргументом функции. Для этого последовательно нажимайте клавиши $(a + b)$ — выражение $(a + b)$ появится на экране и окажется охваченным уголком, направленным влево (\lrcorner). Нажмите клавишу `Ins` — направление уголка станет противоположным (\llcorner) и он переместится в начало выражения. Теперь введите имя функции `sin` — на экране появится выражение $\sin(a + b)$.

Для замены цифры или буквы в выражении поместите указатель мыши после цифры или буквы и щелкните левой кнопкой. В месте щелчка появится курсор, имеющий вид уголка, направленного влево (\lrcorner). Для стирания цифры или буквы нажмите клавишу `BackSpace`. Если курсор находится слева, а направлен вправо (\llcorner), для стирания используйте клавишу `Del`. Пользуйтесь клавишей `F3`, чтобы стереть выделенное рамкой выражение с сохранением его в буфере обмена, и `F4` для вставки выражения из буфера по месту текущего положения курсора.

ПРИМЕЧАНИЕ *Полезно знать, что всякое заключенное в рамку выражение становится операндом следующего вводимого оператора.*

Несмотря на эти и другие тонкости конструирования математических выражений, вряд ли стоит запоминать все правила манипулирования клавишами или мышью. Конструирование выражений чаще всего осуществляется интуитивно, просто нужно некоторое время для экспериментов. После набора первых же нескольких достаточно сложных формул пользователь подсознательно обучится правилам ввода.

3.3.4. Применение клавиш редактирования

Редактирование фрагментов документа трудно отделить от некоторых других операций, поэтому ниже приведено назначение ряда клавиш и их сочетаний, используемых при комплексной работе с документами:

- Shift+F5 — замена одной подстроки другой подстрокой;
- Enter или Ctrl+F9 — вставка пустой строки;
- BackSpace или Ctrl+F10 — удаление пустой строки;
- Alt+BackSpace — отмена последней операции редактирования;
- Ctrl+Ins — копирование (команда Copy) выделенного объекта в буфер;
- Shift+Del — перемещение (команда Cut) выделенного объекта в буфер;
- Shift+Ins — вставка объекта из буфера в окно;
- Ins — включение режима вставки.

3.3.5. Клавишное управления документами и окнами

Для управления документами и окнами приложений используются следующие клавиши:

- Ctrl+Esc — открытие главного меню Windows;
- Alt+Tab — переключение между запущенными приложениями;
- Ctrl+F4 — закрытие документа;
- Ctrl+F6 — активизация следующего окна;
- Ctrl+P — печать документа;
- Alt+F4 — завершение работы с системой (команда Exit);
- Ctrl+R — восстановление изображения на экране;
- F1 — открытие окна справочной системы (команда Help);
- F3 — удаление выделенного объекта с переносом его в буфер обмена;
- F4 — вывод объекта из буфера в текущее окно;
- F5 — вызов окна загрузки файлов документов;
- F6 — запись файла текущего документа на диск;
- F7 — подготовка к созданию нового документа;
- F9 — запуск вычислений;
- Shift+F1 — вызов контекстной справки.

Последняя операция требует некоторых пояснений. При ее исполнении возле указателя мыши (стрелки) появляется жирный знак вопроса. Если щелкнуть таким указателем на каком-либо объекте или его части (например, на имени функции или операторе), будет выведено окно с сообщением о назначении и правилах использования указанного объекта. Для отмены этого режима достаточно нажать клавишу Esc.

3.3.6. Клавиши для создания объектов

Для создания объектов служат следующие клавиши:

- @ — создание шаблона двумерного графика;
- Ctrl+@ — создание шаблона трехмерного графика;
- Ctrl+% — создание шаблона импортируемого рисунка;
- Ctrl+G — замена латинской буквы на греческую;

- Ctrl+V — задание шаблона вектора или матрицы;
- Ctrl+P — ввод греческой буквы «пи»;
- Ctrl+Z — ввод математического символа бесконечности.

Ряд других комбинаций клавиш для ввода математических и логических операторов будет рассмотрен одновременно с их описанием в следующих главах. Их можно найти и в Приложении 2 справочной части книги.

3.4. Редактирование

Под редактированием документов подразумевается изменение ранее введенного, например с клавиатуры или с дисковых накопителей, документа и его пополнение новыми объектами.

3.4.1. Применение команд позиции Edit меню

Большая часть операций редактирования производится непосредственно с отображением объектов в текущем окне редактора. Разумеется, редактирование возможно только после ввода в текущее окно редактора одного или нескольких шаблонов объектов. В противном случае часть операций редактирования осуществить будет нельзя и в меню Edit (Правка) соответствующие команды окажутся недоступными.

Ввод объектов (математических выражений, текстовых комментариев, графиков и др.) в текущее окно редактора производится по-разному. Он будет неоднократно описан по мере изложения последующего материала книги. Оставив пока описание некоторых других деталей ввода (например, выделение частей объектов, перемещение курсора по объектам и др.), рассмотрим команды меню Edit (рис. 3.14).

Как видно из рисунка, в меню Edit представлены следующие команды:

- Undo (Отменить) — отменить последнюю операцию редактирования;
- Redo (Вернуть) — повторить последнюю отмененную операцию редактирования;
- Cut (Вырезать) — переместить выделенный фрагмент в буфер обмена;
- Copy (Копировать) — скопировать выделенный фрагмент в буфер обмена;
- Paste (Вставить) — вставить содержимое буфера обмена в документ;
- Paste Special (Специальная вставка) — вставить в документ объект в том или ином формате, например в формате Bitmap (формат растровых рисунков);
- Delete (Удалить) — удалить выделенные объекты;
- Select All (Выделить все) — выделить все объекты документа;
- Find (Найти) — найти заданную текстовую или математическую строку;
- Replace (Заменить) — найти и заменить математическую или текстовую строку;

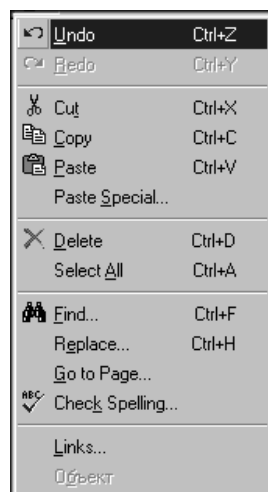


Рис. 3.14.
Раскрытое меню Edit

- Go to Page (Перейти к странице) — расположить начало указанной страницы в верхней части окна Mathcad;
- Check Spelling (Проверка орфографии) — проверка орфографии (для англоязычных документов);
- Links (Связи) — связывание документов;
- Object (Объект) — редактирование объекта, внедренного в документ.

Здесь прежде всего надо отметить очередной казус совместной работы англоязычной системы Mathcad с русскоязычной (локализованной) операционной системой Windows — последняя команда меню Edit (см. рис. 3.16) записана на исконно русском языке!

Позиция Edit меню Mathcad 11 отличается от приведенной на рис. 3.16 только отсутствием позиции Check Spelling. Она перенесена в новое подменю Tools.

3.4.2. Основные приемы редактирования

Прежде чем вплотную заняться командами редактирования, остановимся на специфических приемах редактирования уже введенных выражений в формульных блоках. Основные правила редактирования выражений в Mathcad следующие:

- включение/выключение режима вставки — достигается нажатием клавиши Ins;
- изменение места ввода — достигается нажатием клавиши Ins или клавиш перемещения курсора ← и →;
- перемещение курсора ввода — достигается нажатием клавиш ← и → после задания изменения направления ввода;
- расширение курсора ввода — достигается нажатием клавиши Пробел;
- вставка оператора — достигается установкой курсора между операндами (например, если a и b — операнды и между ними надо вставить оператор +, -, * или /) и вводом символа оператора (курсор ввода при этом имеет вид ⊥);
- вставка оператора после части выражения — достигается выделением курсором ввода части выражения с последующим вводом операнда;
- вставка квадратного корня — достигается выделением курсором ввода подкоренного выражения и затем вводом знака $\sqrt{\quad}$ или знака $\sqrt{\quad}$ на цифровой части клавиатуры (не у всех видов клавиатур);
- вставка унарного минуса — достигается выделением фрагмента выражения, перед которым ставится знак минус, и нажатием клавиши Ins, а затем клавиши - (минус);
- удаление оператора — достигается установкой курсора ввода после оператора и нажатием клавиши Backspace;
- вставка фрагмента выражения в скобки — достигается охватом выражения курсором ввода и набором символа одиночной кавычки ' (не путайте с апострофом);
- освобождение выражения от скобок — достигается установкой курсора ввода справа от открывающей скобки и нажатием клавиши Del или установкой курсора ввода слева от закрывающей скобки и нажатием клавиши Backspace;
- освобождение выражения от прямых скобок (оператора абсолютного значения) — достигается установкой курсора ввода справа от первого знака | и

нажатием клавиши Backspace или установкой курсора ввода слева от второго знака | и нажатием клавиши Del;

- задание выражения в качестве аргумента функции — достигается охватом курсором ввода выражения, вводом скобок, расширением курсора для охвата всего выражения (со скобками) и вводом имени функции — например, с применением кнопки $f(x)$ на панели инструментов;
- отказ от неверно проделанной операции — достигается выбором команды Undo (Отменить) в меню Edit (Правка) или щелчком на кнопке Undo в панели инструментов;
- повторение отмененной операции — достигается выбором команды Redo (Вернуть) в меню Edit или щелчком на кнопке Redo в панели инструментов;
- выход из формульного блока — достигается нажатием клавиши \uparrow , \downarrow или Enter.

СОВЕТ Это множество правил не стоит запоминать наизусть, поскольку большинство из них интуитивно понятно. Однако мы настоятельно рекомендуем читателю поэкспериментировать со всеми правилами на примере любых вызываемых, добиваясь автоматизма в навыках редактирования. Опыт показывает, что это лучше сделать в самом начале знакомства с системой Mathcad. Ибо, если сразу не научиться правильно редактировать математические выражения, переучиваться куда сложнее. При этом вы будете поневоле терять массу времени.

3.4.3. Отмена и повторение операций редактирования

Команда Undo (Отменить) меню Edit (Правка) служит для отмены последней операции редактирования. Обычно она применяется, если в ходе редактирования установлено, что введен неправильный знак, неверное имя или значение переменной. Тогда следует отменить последнюю операцию редактирования, что и реализует команда Undo.

Надо отметить, что команда Undo применима только к тем блокам, в которых находится курсор и начаты операции редактирования. В других случаях эта команда недоступна. При выборе команды Undo восстанавливается то содержимое блока, которое было в нем до выполнения операции редактирования. Например, если в выражении $x := \exp(\sin(1))$ вы решили исправить функцию \sin на функцию \cos , но после удаления имени функции \sin передумали, достаточно выбрать команду Undo, и функция \sin окажется на прежнем месте. Действие этой команды распространяется только на одну последнюю операцию.

Бывает, однако, что, отменив последнюю операцию редактирования, пользователь решает ввести ее снова. Для этого служит команда Redo (Вернуть).

3.4.4. Выделение объектов и перенос их в буфер обмена

Многие операции над объектами требуют их выделения. Выделить объект можно, установив указатель мыши на объекте и щелкнув левой кнопкой. Объект будет размещен в рамке из темных линий (рис. 3.15).

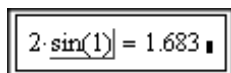


Рис. 3.15. Выделение блока щелчком левой клавиши мыши

Команда Cut (Вырезать) меню Edit (Правка) или одноименная кнопка на панели инструментов пересылает выделенный объект в буфер обмена. Команда Cut реализуется также комбинациями клавиш Shift+Del и Ctrl+X.

Пересылаемый командой Cut объект поступает в буфер, а его изображение в окне редактирования исчезает. Для пересылки объекта в буфер подходит и средство ранних версий системы — клавиша F3.

Объект или группу объектов можно целиком выделить и другим путем. Подведите указатель мыши к объекту, нажмите левую кнопку мыши и, не отпуская ее, передвиньте указатель мыши по диагонали. На экране появится прямоугольник из пунктирных черных линий, размеры которого задаются перемещением мыши. Как только в этот прямоугольник попадет хотя бы часть какого-либо объекта, объект окажется обведенным прямоугольником из сплошных черных линий (рис. 3.16). Как только левая кнопка мыши будет отпущена, основной прямоугольник исчезнет, но все объекты останутся выделенными пунктирными прямоугольниками.

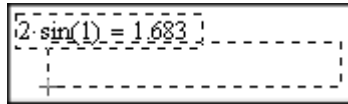


Рис. 3.16. Выделение объекта его захватом

Так можно выделить сразу несколько объектов (рис. 3.17). Чтобы выделить несколько объектов по отдельности, щелкайте на них поочередно при нажатой клавише Ctrl или Shift. Расположение выделяемых объектов и порядок их выделения при этом могут быть любыми.

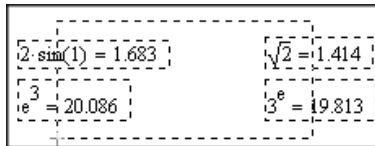


Рис. 3.17. Выделение четырех блоков

Как видно из рис. 3.17, группа выделенных объектов ведет себя как единый блок. Его можно перетаскивать, что наглядно иллюстрирует рис. 3.18. На нем выделенный блок из трех объектов показан в процессе его переноса мышью. Для переноса выделенного блока достаточно подвести указатель мыши к границе выделения любого объекта блока и добиться, чтобы указатель превратился в темное изображение ладони (если указатель приобретает вид двухсторонней стрелки, это означает возможность изменения размера выделенного блока). Это будет означать возможность перемещения блока.

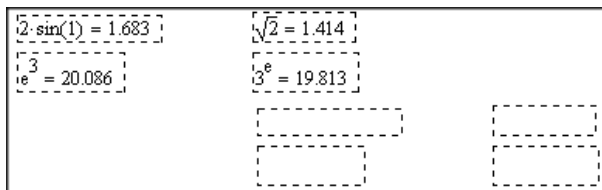


Рис. 3.18. Выделение и перенос четырех блоков

Такой метод перемещения объектов получил название *Drag and Drop* (тащи и отпускай), или *перетаскивания*. В данном случае он реализован в пределах одного окна документа. Однако возможности метода перетаскивания куда шире. Его можно использовать для перемещения группы объектов Mathcad из одного окна в другое окно, причем как Mathcad, так и вообще другого приложения (например, в окно текстового редактора Word).

Обратите внимание, что после переноса блока из одного окна в другое меняются элементы форматирования, в частности размер символов. Они задаются параметрами форматирования, которые установлены в новом окне. Этого, естественно, не происходит, если перенос осуществляется в пределах одного окна.

Как уже отмечалось, в новых версиях упрощена процедура выделения выражений мышью. Теперь с ее помощью можно легко выделять даже отдельные фрагменты формул, что иллюстрирует рис. 3.19, на котором выделено выражение в функции пользователя $f(x)$. Это открывает обширные возможности по конструированию формул из набранных ранее фрагментов. Сохранены и возможности изменения размера выделения с помощью клавиш управления курсором и пробела. Например, для расширения выделения на все выражение удобно нажать клавишу Пробел.

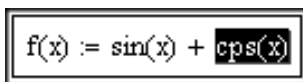


Рис. 3.19. Пример выделения мышью части математической формулы

В текстовых надписях можно выделить мышью любой фрагмент текста, вплоть до одного символа. Аналогично в формулах можно выделить любую переменную. Однако нельзя выделять отдельные символы в именах функций, входящих в математические выражения, — имена выделяются целиком. Щелчок мышью за пределами выделенных объектов снимает все выделения.

3.4.5. Выделение всех объектов

Для выделения всех объектов используется команда Save All (Выделить все) в позиции Edit меню. Она выделяет все блоки документа, обводя их темными пунктирными линиями (рис. 3.20). Это позволяет видеть не только положение блоков, но и оценить степень их перекрытия. Как правило, оно нежелательно.

При переносе объектов или их редактировании возможно наложение одних объектов на другие, при этом части некоторых объектов или даже объекты целиком могут оказаться скрытыми. Выделение всех объектов полезно для выявления этой ситуации.

3.4.6. Копирование объекта в буфер обмена

Команда Copy (Копировать) меню Edit (Правка) аналогична команде Cut (Вырезать), но с одним существенным различием — выделенный объект (или блок объектов) переносится в буфер, но не удаляется из окна редактирования. Эта операция обычно используется в том случае, когда нужно перенести заданный объект в несколько документов, оставив при этом оригинал на своем месте, либо если нужно объект или блок объектов воспроизвести в другом месте документа. Команда Copy реализуется также комбинациями клавиш F4 и Ctrl+V.

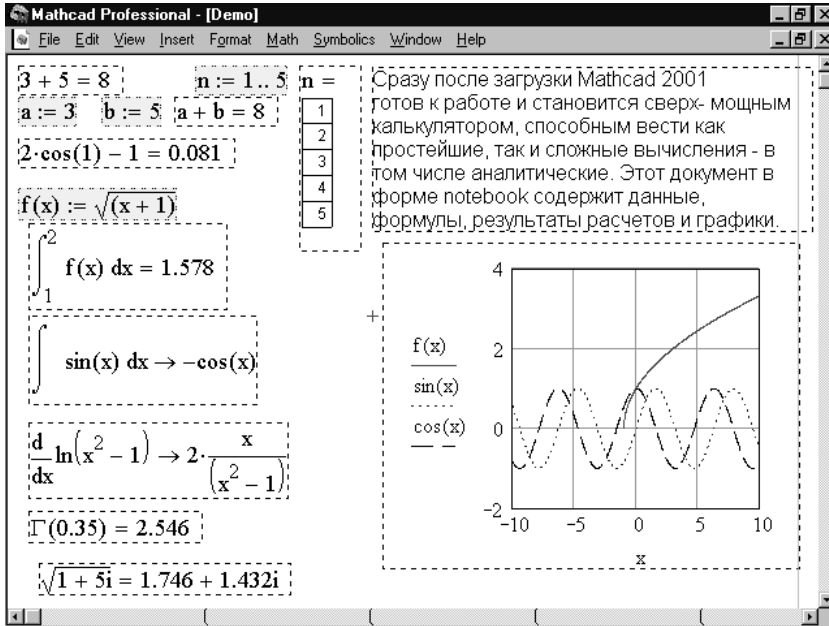


Рис. 3.20. Пример выделения всех блоков документа

Особенно полезна операция *копирования* при составлении документа из частей других документов. Если переключиться в другое окно, то скопированная и помещенная в буфер информация может быть помещена в это окно с помощью описанной ниже команды Paste (Вставить).

Для переноса ряда объектов из одного документа в другой не обязательно использовать буфер обмена. Можно перетаскивать объекты из одного окна в другое, зацепив их указателем мыши в форме ладони. Примеры этого уже приводились выше.

3.4.7. Вставка объекта из буфера обмена

Команда Paste (Вставить) меню Edit (Правка) переносит объекты из буфера в окно и располагает их в месте расположения курсора. Помещенные в окно объекты выделяются пунктирными прямоугольниками. Для снятия выделения достаточно шелкнуть мышью вне области выделения (или нажать клавишу Enter). Команда Paste реализуется также клавишами F4, Shift+Ins или Ctrl+V.

При использовании операции *вставки* содержимое буфера сохраняется и команду можно использовать неоднократно, что особенно удобно для многократного дублирования объектов, когда, например, какая-либо формула или математическое выражение требуются в тексте документа несколько раз.

3.4.8. Специальная вставка

Команда Paste Special (Специальная вставка) меню Edit (Правка) позволяет вставлять в документ данные, представленные файлами различных форматов. Выбор форматов выполняется в обычном диалоговом окне. Доступны различные форматы: текстовый, Mathcad, RTF и Bitmap. Команда Paste Special расширяет

возможности подготовки многокомпонентных документов. С ее помощью можно, например, подготовив рисунок в графическом редакторе Paint, перенести его в документ Mathcad. В остальном применение команды Paste Special подобно применению команды Paste (Вставить).

Рис. 3.21 иллюстрирует технику выполнения специальной вставки. В одном из документов выделены и скопированы в буфер обмена два объекта. Затем сделано активным пустое окно (слева), курсором ввода намечено место вставки и выбрана команда Paste Special. Она вызывает появление окна вставки, в котором можно выбрать доступный формат вставки. Переключатель Insert Link (Вставить связь) позволяет не вставлять объект в документ, а реализовать связь с ним. Этот прием обычно используется для сокращения размера файла документа.

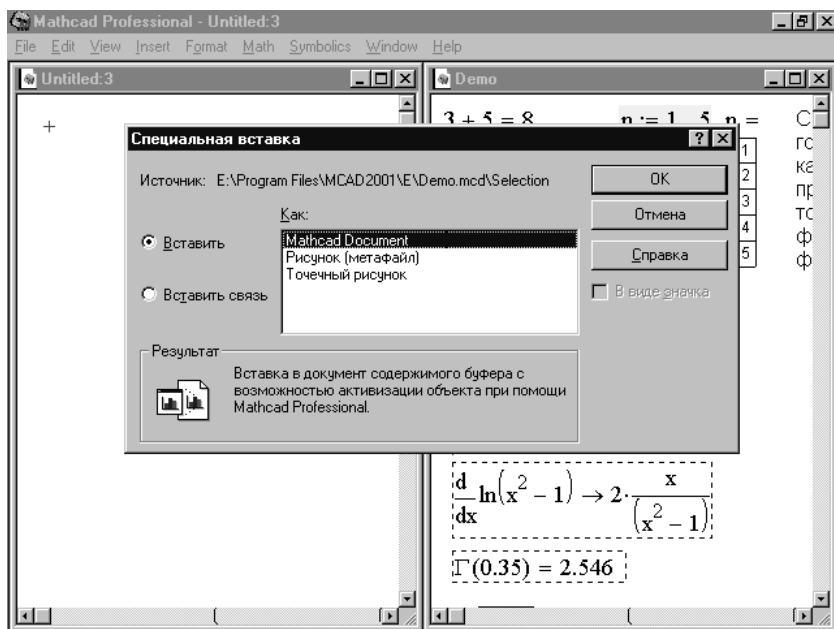


Рис. 3.21. Первый этап специальной вставки

Поскольку в данном случае объектами вставки являются объекты системы Mathcad, в окне специальной вставки надо указать тип объекта — Mathcad Document (Документ Mathcad). Причины появления в диалоговом окне специальной вставки русскоязычных надписей уже обсуждались — эти окна являются общими для различных приложений и создаются не ими, а операционной системой Windows.

Теперь, щелкнув на кнопке ОК, можно поместить содержимое буфера в другое окно документа. Отметим, что курсор ввода задает положение верхнего левого угла верхнего объекта. Объекты форматируются в соответствии с параметрами окна, в которое они вставлены. После вставки объекты остаются выделенными рамкой из пунктирных линий.

Как уже упоминалось, формат объектов специальной вставки может быть самым разным, в частности допускается вставка рисунков. Для иллюстрации этого сделаем следующий эксперимент. С помощью любой программы просмотра рисунков выведем на экран какой-нибудь рисунок (в нашем случае это изображение микропроцессора Pentium 4) и поместим его в буфер обмена, нажав клавишу Print

Screen. Теперь откроем новое окно Mathcad, выполним операцию специальной вставки и выберем формат вставки BitMap (Точечный рисунок). Щелчок на кнопке ОК вставляет рисунок в открытое окно (рис. 3.22).



Рис. 3.22. Пример вставки точечного рисунка командой специальной вставки

Вставленный рисунок можно выделить мышью, при этом он будет обрамлен рамкой с черными прямоугольниками для изменения размера рисунка. Его можно также перетаскивать по полю окна документа. На рисунке показан результат уменьшения рисунка и его перемещения (исходный рисунок имел разрешение 1024×960 точек и в экран не вмещался).

Возможность помещать в окно Mathcad рисунки весьма ценна при подготовке электронных книг и уроков в системе Mathcad. Следует обратить внимание, что четкость рисунков падает при уменьшении их размеров, поэтому рекомендуется выполнять манипуляции с рисунками с помощью специальных графических редакторов, после чего уже готовый рисунок нужного размера вставлять в документ Mathcad.

3.4.9. Удаление выделенных объектов

Команда Delete (Удалить) меню Edit (Правка) служит для удаления выделенного объекта (в том числе блока выделенных объектов). В отличие от команды Cut (Вырезать), ее применение не загружает буфер обмена и предотвращает нехватку оперативной памяти.

ВНИМАНИЕ Не забывайте, что с помощью команды Undo (Отменить) можно отменить только последнюю выполненную операцию, поэтому будьте осторожны с командой Delete — можно легко удалить нужный объект. А если после удаления была выполнена еще какая-либо операция, восстановить удаленный объект не удастся (придется создавать заново).

3.4.10. Поиск подстроки

Команда Find (Найти) меню Edit (Правка) служит для поиска фрагментов текста или имен переменных в математических выражениях. При ее выборе появля-

ется диалоговое окно Find для ввода искомого фрагмента — подстроки (рис. 3.23). При обнаружении указанной подстроки в формульном блоке курсор устанавливается в конец обнаруженной подстроки, после чего можно приступить к ее редактированию. В текстовом блоке найденная подстрока выделяется, как показано на заднем плане рисунка.

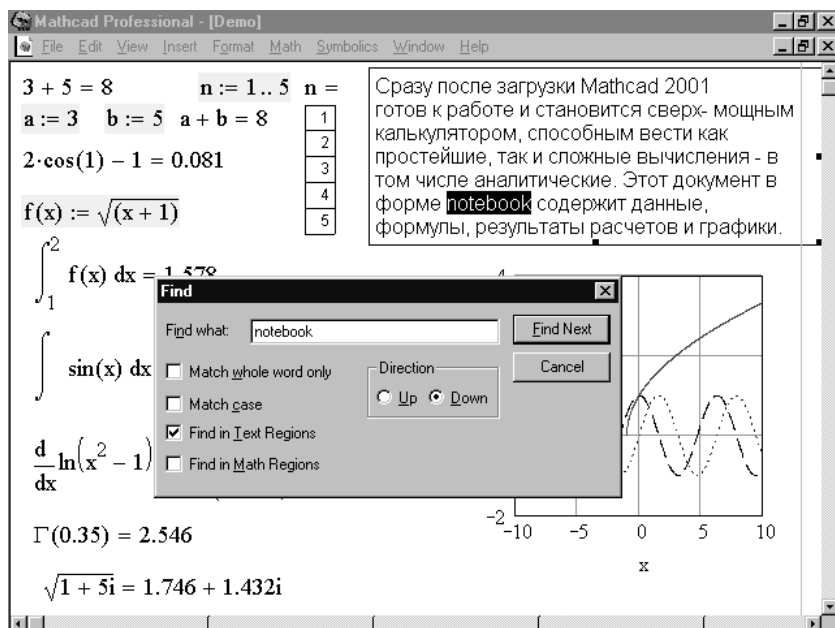


Рис. 3.23. Поиск заданного фрагмента текста (слова notebook)

В группе Direction (Направление) диалогового окна Find имеется два переключателя: Up (Вверх) и Down (Вниз). Первый задает поиск от текущего положения курсора вверх до первого обнаружения подстроки, второй — поиск от текущего положения курсора вниз. Иницирует поиск щелчок на кнопке Find Next (Найти далее). Если подстрока поиска не обнаружена, появляется сообщение Stop Not found (Не найдено). Кроме перечисленных элементов управления в окне Find имеются следующие флажки, назначение которых очевидно по их названиям:

- Match whole word only (Только слово целиком);
- Match case (Учитывать регистр);
- Find in Text Regions (Искать в текстовых блоках);
- Find in Math Regions (Искать в формульных блоках).

3.4.11. Замена подстроки

Иногда в ходе редактирования документа пользователь заранее знает, какая именно часть документа должна измениться. Например, часто возникает необходимость исправить имя какой-либо переменной или скорректировать заведомо известную формулу. Возможно, потребуется изменение какой-либо фразы текстового сообщения. При этом весьма полезна операция поиска изменяемого фрагмента по всему документу, особенно когда документ велик и заданное изменение надо повторить несколько раз.

Если вы хотите не только найти заданную подстроку, но и заменить ее другой, используйте команду Replace (Заменить) меню Edit (Правка). Замену можно проводить в тексте или математическом выражении. При выполнении этой команды появляется окно с двумя полями: Find what (Найти что), в которое вводится искомая строка, и Replace with (Заменить на), в которое вводится строка замены (рис. 3.24).

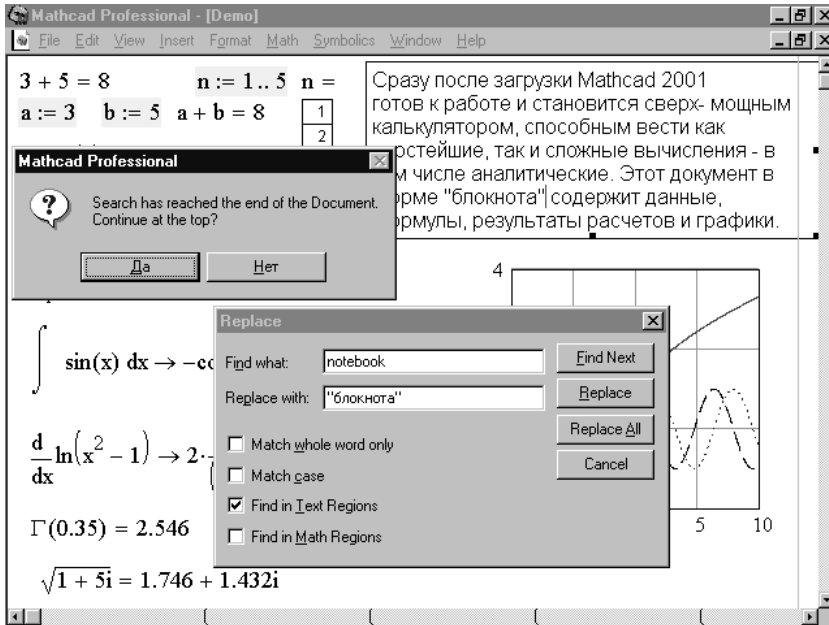


Рис. 3.24. Замена слова с помощью команды Replace

Иницирует поиск щелчок на кнопке Find Next (Найти далее). Чтобы заменить найденную строку, нужно щелкнуть на кнопке Replace (Заменить), а чтобы заменить подобные строки во всем документе — на кнопке Replace All (Заменить все). Как и в случае с операцией поиска (см. подраздел «Поиск подстроки»), операцию замены можно применять неоднократно. К сожалению, в русских текстах иногда могут возникнуть трудности при поиске и замене слов, поскольку система поиска и замены создавалась для англоязычных текстов. Тем не менее обычно система работает и в этом случае.

Помимо окна поиска и замены на рис. 3.24 показано окно с сообщением о достижении конца документа и запросом о необходимости продолжения поиска с начала. Щелчок на кнопке Yes (Да) санкционирует продолжение поиска. Если поиск не приводит к успеху, появляется окно с сообщением об этом (рис. 3.25).



Рис. 3.25. Сообщение об отсутствии искомого объекта в документе

При работе со сложными многостраничными документами операция поиска и замены оказывается очень полезной, позволяя упростить их редактирование.

ВНИМАНИЕ Помните, что *Mathcad* — англоязычная система. Поэтому правильный поиск и замена подстрок гарантируются только для англоязычных текстов. Для русскоязычных текстов поиск приносит результат не всегда, даже если искомая подстрока гарантированно находится в документе.

3.4.12. Переход к заданной странице документа

Документ с достаточно сложными расчетами может занимать множество страниц. Для быстрого перехода к нужной странице используется команда *Go To Page* (Перейти к странице) меню *Edit* (Правка). При выборе этой команды появляется одноименное диалоговое окно, показанное на рис. 3.26.

В этом окне с помощью переключателей надо указать страницу, к которой надо перейти: *First Page* (К первой странице), *Last Page* (К последней странице) или *Page Number* (По номеру). В последнем случае в поле рядом с переключателем требуется ввести номер искомой страницы. В двух полях в нижней части окна отображается общее число страниц в документе и номер текущей страницы. Кнопка *OK* инициирует переход, а кнопка *Cancel* (Отмена) отменяет его.

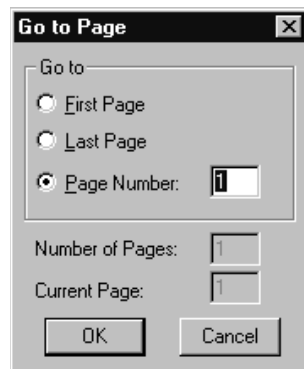


Рис. 3.26.
Диалоговое окно *Go To Page*

3.4.13. Проверка орфографии

Для проверки орфографии в англоязычных текстах служит команда *Check Spelling* (Проверка орфографии) меню *Edit* (Правка). Она обеспечивает проверку путем сравнения слов в документе с набором эталонных слов в специальном словаре системы. Процесс проверки иллюстрирует рис. 3.27. Здесь было намеренно искажено слово «notebook».

Как видно из приведенного примера, система орфографического контроля не только выявляет ошибочные слова и выделяет их, но и предлагает варианты замены. Если ошибки не обнаружены или все исправлены, появляется окно с сообщением о завершении проверки.

Кнопки в окне проверки орфографии имеют следующее назначение:

- *Ignore* (Пропустить) — игнорировать найденное слово с ошибкой;
- *Change* (Заменить) — заменить слово подходящим из списка;
- *Add* (Добавить) — добавить слово в словарь;
- *Suggest* (Предложить) — представить расширенный список слов для замены;
- *Cancel* (Отмена) — прекратить проверку.

Добавление слов в словарь расширяет его и позволяет проводить проверку специальных терминов, возможно, отсутствующих в словаре. Однако эту операцию следует проводить осторожно — словарь легко засорить случайно введенными неверными словами, что снижает эффективность проверки орфографии.

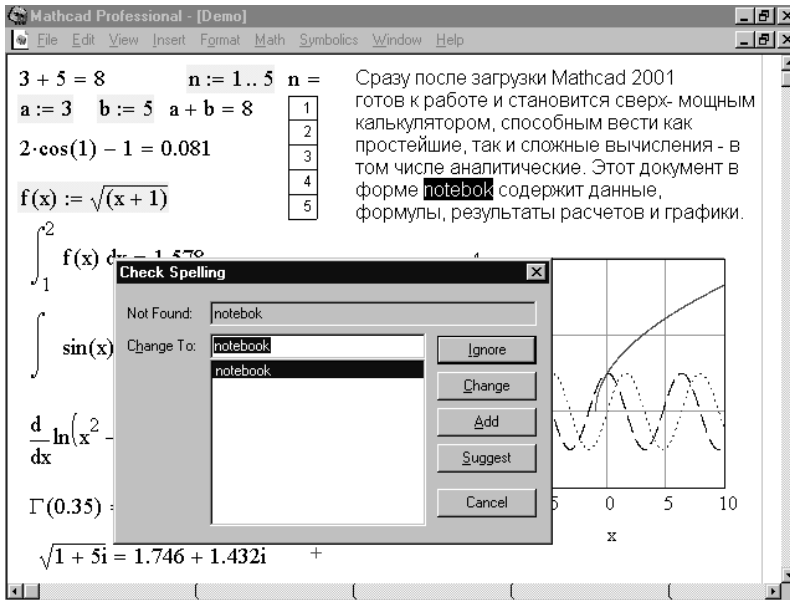


Рис. 3.27. Пример проверки орфографии

ПРИМЕЧАНИЕ Как уже говорилось, ценность системы орфографического контроля заметно снижается из-за невозможности проверки русскоязычных текстов. При этом все русские слова отмечаются как ошибочные. Однако эту проблему легко обойти. Тексты можно готовить в редакторе Microsoft Word (он имеет мощную русскоязычную систему орфографического и грамматического контроля) и затем включать их в документы Mathcad.

3.4.14. Редактирование связей

Команда Links (Связи) меню Edit (Правка) используется для редактирования связей в документе. Связывание объектов, как описывалось выше в подразделе «Специальная вставка», выполняется с применением команды Paste Special (Специальная вставка), а также команды Object (Объект) меню Insert (Вставка), о которой будет рассказано в разделе «Вставка объектов» главы 4. Если в окне специальной вставки при внедрении объекта был установлен переключатель Insert Link (Вставить связь), то в документ вставляется не сам объект, а связь с ним. Именно такие связи (если они есть в документе) помогает редактировать команда Links. Она позволяет менять объекты-источники и прерывать связь. При связывании объектов правильные связи устанавливаются автоматически, поэтому, если не переписывать файл-источник в другую папку, редактировать связи не требуется. Если в документе нет связанных объектов, окно редактирования связей оказывается пустым, а соответствующие кнопки — недоступными. Именно такой случай показан на рис. 3.28.

3.4.15. Редактирование внедренных объектов

С помощью команды Object (Объект) меню Insert (Вставка), о которой будет рассказано в разделе «Вставка объектов» главы 4, в документ можно внедрять объ-



Рис. 3.28. Окно редактирования связей объектов

екты из других приложений. Если внедренный объект выделен (прямоугольником из сплошных линий), становится доступной команда Object (Объект) в меню Edit (Правка). При ее выборе выделенный объект открывается в том приложении, в котором он был создан.

3.5. Вставка в документ объектов

3.5.1. Подменю вставки Insert

Комментарии, формулы, математические выражения, графики и т. д. — все это разновидности *объектов*. Введение любого объекта в окно редактирования называется *вставкой* (insert). Mathcad реализует различные механизмы вставки — от простого вывода шаблона объекта до вставки объекта со связыванием его с приложением, создавшим объект, и возможностью редактирования объекта с помощью этого приложения. В Mathcad 2001i все команды вставки включены в меню Insert (Вставка):

- Graph (График) — раскрытие подменю вставки шаблонов графики;
- Matrix (Матрица) — вставка шаблонов матриц и векторов;
- Function (Функция) — вставка шаблонов встроенных функций;
- Unit (Единицы) — вставка единиц измерения размерных величин;
- Picture (Рисунок) — вставка шаблона импортируемого рисунка;
- Area (Область) — вставка области, которую в дальнейшем можно скрыть или заблокировать (см. подразделы «Блокировка области» и «Скрытие области» в разделе «Форматирование электронных документов» главы 5);
- Math Region (Математический блок) — вставка в текстовую область шаблона математической области;
- Text Region (Текстовый блок) — вставка текстовой области;
- Page Break (Разрыв страницы) — вставка символа разрыва страницы;
- Hyperlink (Гиперссылка) — вставка гиперссылки;
- Reference (Ссылка) — вставка кнопки обращения к заданному файлу путем щелчка на ней;
- Component (Компонент) — вставка других компонентов системы;
- Object (Объект) — вставка объекта с установлением динамической связи с породившим его приложением.

Раскрытое меню Insert показано на рис. 3.29.

Помимо команд вставки шаблонов двумерных и трехмерных графиков, применение которых рассматривается в следующей главе, в меню Insert (Вставка) сосредоточены команды вставки многих других объектов. Именно эти команды и рассматриваются в данном разделе.

3.5.2. Вставка шаблона матриц и векторов

Команда Matrix (Матрицы) меню Insert (Вставка) обеспечивает вывод шаблона для векторов или матриц. Как известно, матрица является именованным объектом, заданным в виде массива данных. Mathcad использует одномерные массивы — векторы и двумерные — собственно матрицы.

Матрица характеризуется числом строк (rows) и столбцов (columns). Таким образом, число элементов матрицы (или ее размерность) равно $rows \cdot columns$. В Mathcad элементами матрицы могут быть числа, константы, переменные и даже математические выражения. Соответственно матрицы могут быть численными и символьными.

При выполнении команды Matrix в текущем окне документа появляется диалоговое окно, позволяющее задать размерность вектора или матрицы (рис. 3.30). Для этого нужно указать число строк в поле Rows (Строки) и число столбцов в поле Columns (Столбцы). Нажав клавишу Enter или щелкнув на кнопке Insert (Вставить), можно вывести шаблон матрицы или вектора (у вектора один из параметров размерности равен 1).

Шаблон матрицы (рис. 3.31) содержит места ввода элементов (числовых или символьных) вектора или матрицы. Место ввода можно сделать активным, щелкнув на нем мышью. С помощью клавиш перемещения курсора можно поочередно ввести все элементы вектора или матрицы.

Пока идет ввод элементов вектора или матрицы, пустые места ввода отображаются без каких-либо комментариев. Однако если закончить ввод до полного заполнения шаблона, система выведет сообщение об ошибке, при этом незаполненный шаблон будет окрашен в красный цвет. Аналогично, несуществующая матрица или ошибочно указанный индекс отображаются красным цветом.

Если щелкнуть на кнопке Insert (Вставить) при заполненном шаблоне матрицы, то матрица расширяется и ее размер увеличивается. Кнопка Delete (Удалить) позволяет ликвидировать расширение матрицы, удалив из нее строку или столбец.

Каждый элемент матрицы характеризуется индексированной переменной, а его положение в матрице определяется двумя индексами — номером строки и номером столбца. Для задания индексированной переменной прежде всего надо ввести

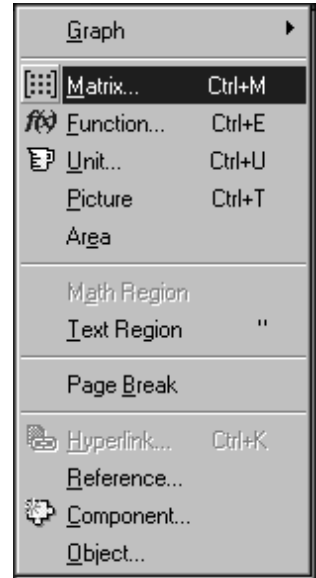


Рис. 3.29. Меню Insert

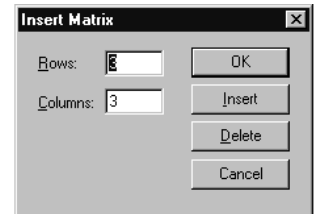


Рис. 3.30. Диалоговое окно для вставки матрицы

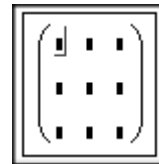


Рис. 3.31. Шаблон матрицы

имя переменной, а затем перейти к набору индексов нажатием клавиши `]`. Сначала указывается индекс строки, а затем через запятую — индекс столбца.

Врожденная в одну строку или в один столбец матрица является вектором. Его элементы — индексированные переменные с одним индексом. Векторы могут быть двух типов: вектор-столбец и вектор-строка. Нижняя граница индексов задается значением системной переменной `ORIGIN`. Обычно ее значение задают равным 0 или 1.

3.5.3. Вставка функций

Как правило, при подготовке документов имена функций пользователь вводит самостоятельно. Поскольку функций, встроенных в систему, очень много (около 300), запомнить их все довольно сложно. Это и не нужно, поскольку в Mathcad есть средство для ввода имен функций из их общего списка (каталога). Команда `Function` (Функция) меню `Insert` (Вставка) выводит диалоговое окно (см. рис. 1.21) с полным списком встроенных в ядро системы функций, разбитым по тематическим разделам с кратким описанием назначения каждой функции. Это позволяет корректно ввести шаблон с именем функции в место расположения курсора в математическом выражении.

Обратите внимание на то, что запись каждой функции дается в полной синтаксической форме. Это исключает довольно частые ошибки в формулах, вызванные неточностями записи функций. В выведенном шаблоне функции имеются места для ввода параметров.

Набор функций может пополняться при загрузке в Mathcad пакетов расширения и библиотек новых функций. При этом создаются новые разделы функций и в них размещаются новые функции. Использование их аналогично использованию изначально присущих Mathcad встроенных функций.

3.5.4. Вставка единиц измерения размерных величин

Команда `Unit` (Единицы) меню `Insert` (Вставка) служит для задания единиц измерения размерных величин. Размерные величины широко используются в физических и научно-технических расчетах.

При выборе команды `Unit` появляется диалоговое окно со списком размерных величин (`Dimension`) и относящихся к ним единиц измерения (`Unit`). В поле `System` отображается текущая система единиц измерения (рис. 3.32). По умолчанию используется система `SI` (СИ).

ПРИМЕЧАНИЕ Смену используемой системы единиц обеспечивает команда `Options` (Параметры) меню `Math` (Вычисления).

Окно вставки единиц измерения размерных величин имеет три кнопки:

- `Insert` (Вставить) — вставляет выбранную единицу измерения, при этом окно вставки единиц измерения остается открытым;

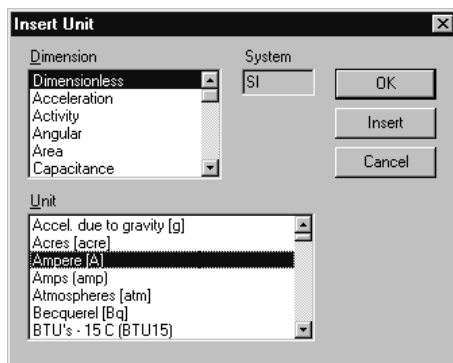


Рис. 3.32. Диалоговое окно вставки единиц измерения размерных величин

- ОК — вставляет выбранную единицу и закрывает окно;
- Cancel (Отмена) — отменяет операцию вставки единиц измерения размерных величин.

При работе с размерными величинами Mathcad производит необходимые преобразования и выводит числовые значения величин вместе с единицами их измерения. Они указываются после числового значения соответствующей величины.

ПРИМЕЧАНИЕ *Размерные величины часто требуются в физических и химических расчетах. Однако не следует слишком увлекаться их применением, поскольку такие величины могут служить источником ряда неточностей или даже ошибок. Система Mathcad 2001 избавлена от большинства из них. Тем не менее следует по возможности нормировать значения величин и записывать математические выражения преимущественно в безразмерной форме.*

3.5.5. Вставка шаблона импортируемого рисунка

Система Mathcad дает возможность импортировать графические файлы, создаваемые другими графическими системами, например, графическими редакторами класса Paint, системами AutoCAD, PCAD и др. Это позволяет включать в документы Mathcad высококачественные графические иллюстрации, что крайне необходимо при создании электронных книг.

Перед импортом файла его надо подготовить в соответствующем приложении и записать в текущую папку. Файл может иметь расширение .bmp (BitMap). Воспринимаются также файлы с расширением .msc из прежних версий системы Mathcad.

Для импорта файла с рисунком достаточно выбрать команду Picture (Рисунок) в меню Insert (Вставка), при этом на экране появится шаблон рисунка с единственным местом ввода в левом нижнем углу. В этом месте ввода надо записать в кавычках путь к графическому файлу, его имя и, отведя указатель мыши в сторону от рисунка, щелкнуть левой кнопкой мыши. На рис. 3.33 показан документ с рисунком из файла `ibmpc.bmp`, на котором представлен теперь уже музейный экспонат — первый персональный компьютер фирмы IBM, легендарный IBM PC.



Рис. 3.33. Пример импорта графического изображения

Импортируемый рисунок, как и любой другой, можно перетаскивать, изменять его размер, выделять, помещать в буфер обмена и т. д. Для форматирования рисунка служит панель Picture (Рисунок), с помощью которой можно осуществлять операции поворота и масштабирования рисунков, изменения яркости и контрастности и т. д. Работа с панелью Picture подробно описана в следующей главе.

Помимо импорта рисунков из файлов, система Mathcad допускает вставку рисунков из буфера обмена. Для этого сначала в исходном приложении надо ввести рисунок в буфер обмена, например, используя команду Copy (Копировать) или Cut (Вырезать). Затем, перейдя к работе с системой Mathcad, нужно установить курсор на место верхнего левого угла будущего рисунка и выбрать команду Paste (Вставить) в меню Edit (Правка) или нажать клавишу F4.

3.5.6. Вставка текстового комментария

Как отмечалось ранее, текстовые области вставляются в документы Mathcad для создания словесных комментариев. В Mathcad 2001 PRO оставлена команда вставки расширяемой текстовой области — Text Region (Текстовый блок). Перед ее применением нужно установить курсор в то место окна редактирования, откуда должен начинаться текст — это левый верхний угол будущей текстовой области. После этого выбирается команда Text Region в меню Insert (Вставка) и на месте курсора появляется прямоугольник (шаблон текстовой области) с текстовым курсором внутри него. На рис. 3.34 показан пример создания текстового блока.

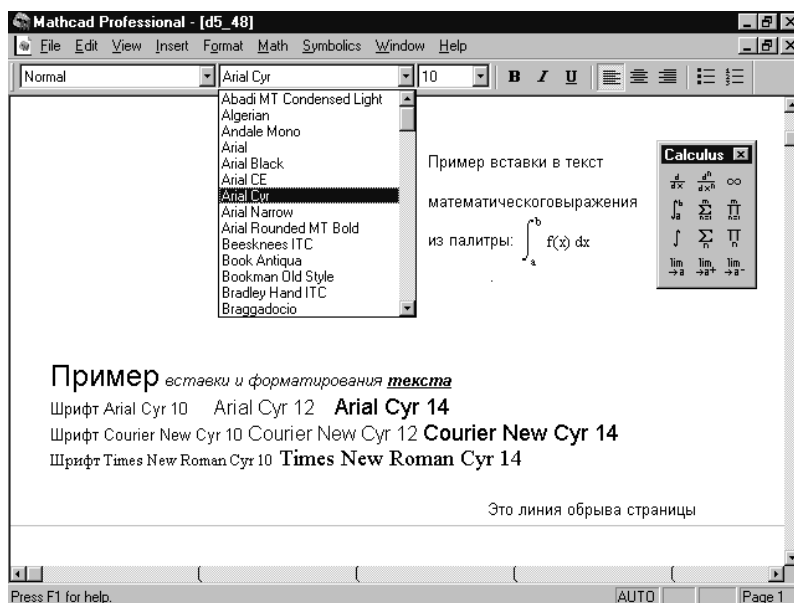


Рис. 3.34. Пример создания текстового блока

Перед вводом русскоязычного текста рекомендуется выбрать подходящий шрифт. У русскоязычных (кириллических) шрифтов в названии обычно имеется слово «Cyr». Если в начале ввода выяснилось, что введена не буква русского алфавита, то при попытке ее удаления шаблон текстовой области исчезает и его приходится выводить снова.

3.5.7. Вставка математических выражений в текстовый комментарий

Нередко в текстовый комментарий требуется вставить математическое выражение или формулу. Для этого надо установить курсор ввода в нужное место текстового комментария и в меню Insert (Вставка) выбрать команду Math Region (Математический блок). При этом на месте курсора ввода появится шаблон в виде маленького черного квадрата. В него можно ввести любое выражение или формулу с помощью палитры математических знаков или набрать выражение на клавиатуре. В правом верхнем углу окна документа на рис. 3.34 показан пример подобной вставки.

Документы, представленные на рисунках в данной книге, содержат минимальные текстовые и формульные комментарии, поскольку все необходимые пояснения даны в тексте книги. Кроме того, это позволяет «втиснуть» в окна документов больше содержательного материала. Однако законченные документы Mathcad, особенно предназначенные для учебных целей, должны содержать достаточно подробные комментарии с пояснениями примененных алгоритмов и методов решения задач.

3.5.8. Вставка линии разрыва страницы

Большие документы нередко занимают много страниц. Бывает крайне неудобно, когда при печати таких документов какой-нибудь график или таблица обрываются линией разрыва страниц. Недопустимы и ситуации, когда заголовок раздела оказывается на одной странице, а описание раздела начинается с другой страницы.

Для предотвращения таких ситуаций желательно прервать страницу в заданном месте. Для этого служит команда Page Break (Разрыв страницы) меню Insert (Вставка), вставляющая в документ горизонтальную линию разрыва страницы (на рис. 3.34 такая линия показана внизу окна редактирования). Когда линия разрыва встречается при печати, печать продолжается со следующей страницы.

ПРИМЕЧАНИЕ *Линию разрыва страницы, как и любой объект Mathcad, можно удалить, выделив ее мышью (линия при этом обрамляется пунктирным длинным прямоугольником) и воспользовавшись клавишей Delete или командой Cut (в последнем случае линия разрыва страниц попадает в буфер обмена и может быть перенесена на новое место в документе).*

3.5.9. Вставка гипертекстовой ссылки

Гиперссылки — важный элемент сложных документов, обучающих программ и справочных систем. Гиперссылки бывают двух видов — гипертекстовые ссылки и гипермедиассылки. Напомним, что гипертекстовая ссылка в системе Mathcad представляется подчеркнутым словом, при щелчке на котором вызывается связанный со ссылкой документ (файл). Гипермедиассылка — более сложный объект на основе какого-либо знака или рисунка.

Для создания гипертекстовой ссылки должен быть создан текстовый блок, в котором нужно выделить некоторый фрагмент текста — чаще всего слово. После того как фрагмент текста выделен, надо выбрать команду Hyperlink (Гиперссылка) в меню Insert (Вставка). Она выводит диалоговое окно Insert Hyperlink (Вставка гиперссылки), показанное на заднем плане рис. 3.35. В верхнем свободном поле следует указать полное имя файла, который будет вызываться данной гипертекстовой ссылкой.

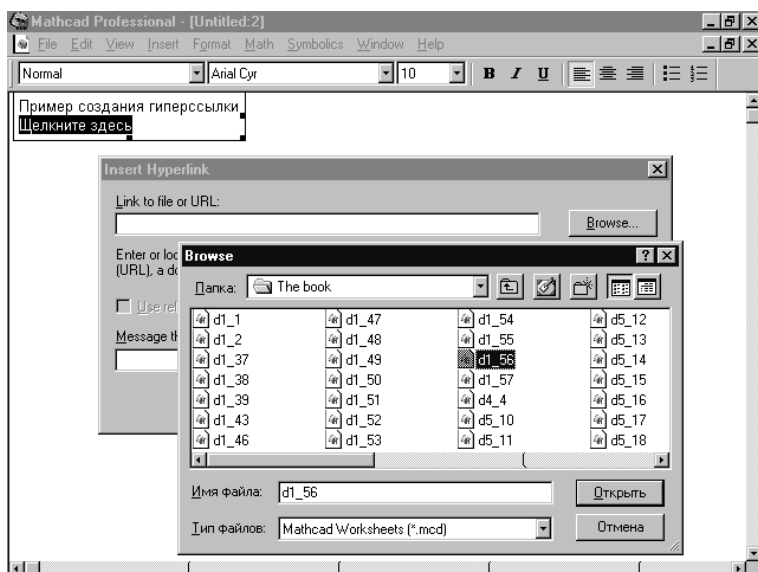


Рис. 3.35. Подготовка к заданию гипертекстовой ссылки

Обычно точное имя файла (с полным путем доступа к нему) запоминается трудно. Поэтому, щелкнув на кнопке Browse (Обзор), можно вызвать окно поиска файла, показанное на переднем плане рисунка. Пусть это будет один из файлов учебного курса по системе Mathcad 2000, что продемонстрирует совместимость файлов для последних версий системы.

После нахождения файла его полное имя появляется в верхнем поле окна создания гипертекстовой ссылки, и в нижнее поле этого диалогового окна можно ввести текст сообщения, которое будет появляться в строке состояния (внизу экрана системы) при наведении на гипертекстовую ссылку указателя мыши (рис. 3.36). Для завершения создания гипертекстовой ссылки достаточно щелкнуть на кнопке OK окна создания гипертекстовой ссылки. При отказе от ее создания надо щелкнуть на кнопке Cancel (Отмена).

Сообщения, появляющиеся в строке состояния при наведении указателя мыши на гипертекстовую ссылку, при большом числе гипертекстовых ссылок очень полезны, поскольку позволяют судить об их назначении и использовать те из них, которые действительно необходимы. Если в окне создания гиперссылки снят флажок Display as pop-up document (Открывать как дополнительный документ), то вызываемый гиперссылкой документ заменяет текущий. При установленном флажке окно нового документа выводится поверх исходного.

Для перехода по гипертекстовой ссылке достаточно щелкнуть на ней мышью. Текстовое сообщение, в котором имеется гипертекстовая ссылка, будет выделено, и на экране появится вызываемый этой ссылкой документ (рис. 3.37). В данном случае он выводится поверх исходного документа. В строке состояния можно увидеть текстовый комментарий к гипертекстовой ссылке.

После ознакомления с вызываемым документом его можно закрыть. До закрытия документа возможно его редактирование. Объектами гипертекстовых ссылок могут быть не только документы. Возможно, к примеру, задать гиперссылку на рисунок, создаваемый графическим редактором, на электронную таблицу и т. д. Все это позволяет готовить вполне современные и удобные в применении доку-

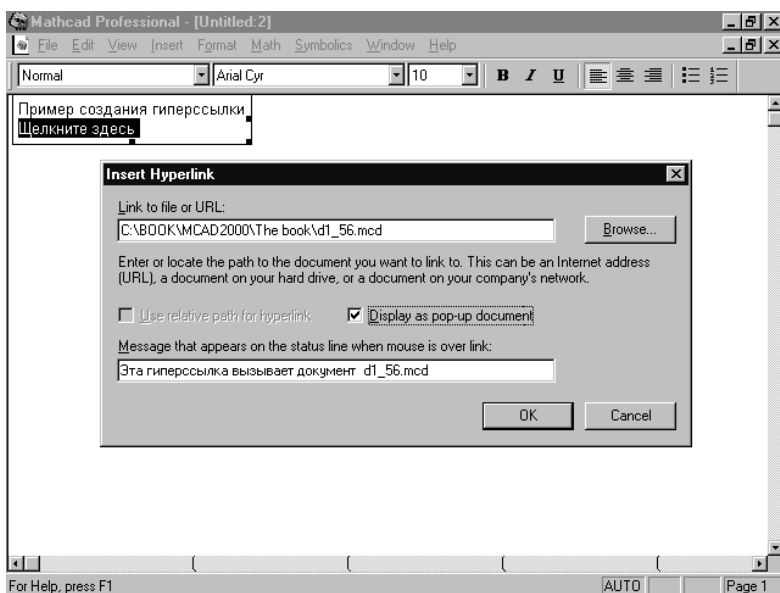


Рис. 3.36. Завершение создания гипертекстовой ссылки

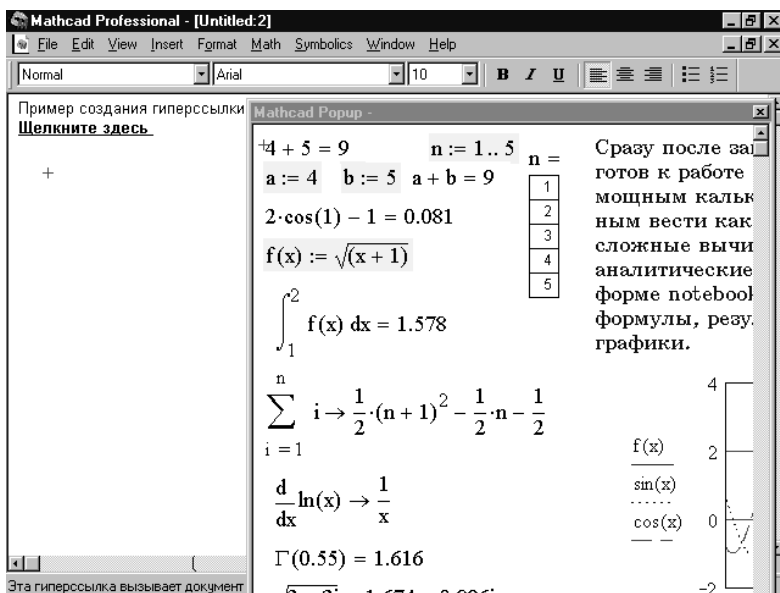


Рис. 3.37. Использование гипертекстовой ссылки

менты не только расчетного, но и информационно-справочного, а также учебного характера.

Гипертекстовая ссылка может быть ликвидирована. Для этого надо щелкнуть на кнопке Remove Link (Удалить связь), которая появляется в окне создания гиперссылки, если выделить уже созданную гипертекстовую ссылку и выбрать команду Insert Hyperlink (рис. 3.38).

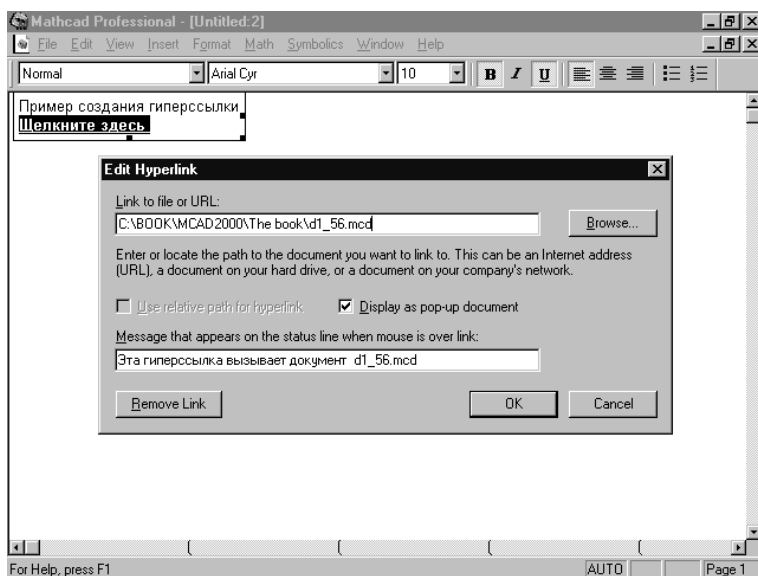


Рис. 3.38. Окно редактирования и удаления гипертекстовой ссылки

Можно также отредактировать гипертекстовую ссылку в этом окне. Применение гипертекстовых ссылок позволяет пользователю создавать сложные документы, имеющие вид электронных книг с многочисленными перекрестными связями.

3.5.10. Вставка гипермедиассылки

Есть еще один удобный способ обращения к заданным файлам, реализуемый путем создания ассоциированного с ними графического объекта — кнопки со стрелкой и/или поясняющей надписью. Для этого курсор ввода устанавливается в нужное место и в меню Insert (Вставка) выбирается команда Reference (Ссылка). Она выводит окно задания такой ссылки, показанное на рис. 3.39.

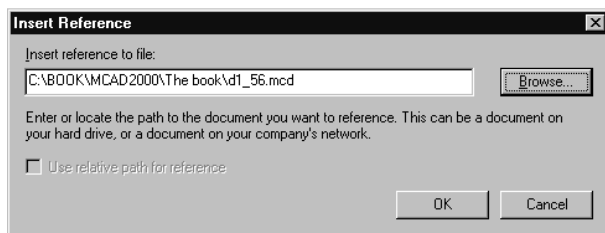


Рис. 3.39. Задание гипермедиассылки в виде кнопки, ассоциированной с файлом

В этом окне надо либо явно указать полное имя файла, либо найти файл, воспользовавшись кнопкой Browse. Щелкнув затем на кнопке ОК, можно вывести по месту расположения курсора ввода кнопку гиперссылки с указанным после нее именем вызываемого ею файла. Щелчок на этой кнопке ведет к появлению окна с вызываемым документом, которое располагается поверх исходного окна (рис. 3.40). Удаляется эта ссылка, как и любой объект на экране, выделением и выбором команды Delete (Удалить) меню Edit (Правка) или нажатием клавиши F3.

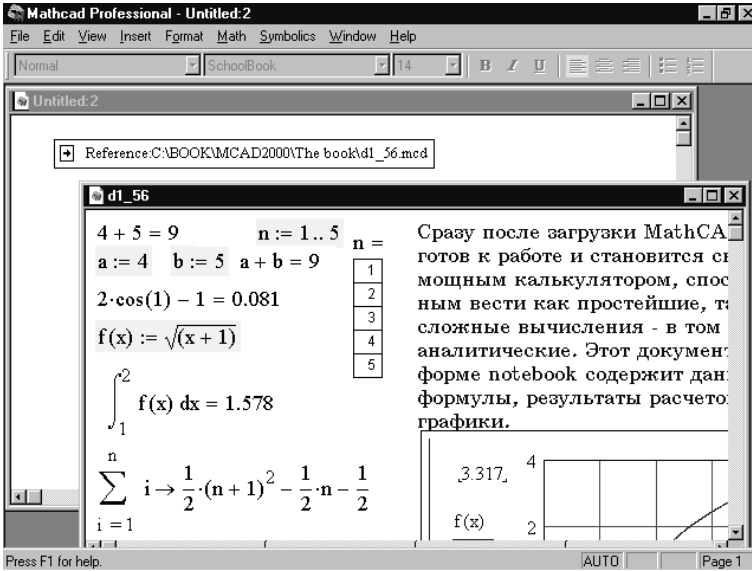


Рис. 3.40. Использование гипермедиаадреса в виде кнопки

Два вида гиперссылок (гипертекстовые ссылки и гипермедиаадреса), реализованных в системе Mathcad, расширяют возможности подготовки сложных документов. При этом гипермедиаадреса очень широко используются в электронных книгах, как создаваемых самим пользователем, так и в книгах, которые специально разработаны для расширения системы Mathcad (см. главы 13, 14 и 15).

3.5.11. Вставка объектов

Команда Object (Объект), предназначенная для вставки объектов, пожалуй, самая мощная из команд вставок. Впрочем, она имеется в большинстве серьезных приложений для операционных систем класса Windows. Благодаря этой команде легко устанавливается связь с самыми различными приложениями этой операционной системы, в том числе довольно далекими от математических систем, такими как текстовый процессор Word или база данных Access из пакета Microsoft Office.

После выбора команды Object выводится стандартное окно с перечнем приложений, с которыми может быть реализована объектная связь (рис. 3.41).

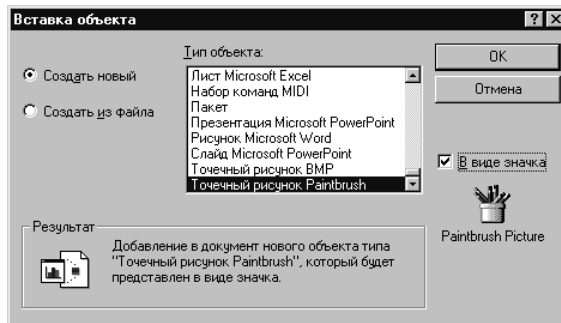


Рис. 3.41. Окно объектной связи с приложениями

Можно создать новый объект или объект, связанный с файлом (в последнем случае в окне появляется кнопка Browse (Обзор), которая позволяет найти подходящий файл). Можно также создать новый объект с помощью связанного с ним приложения — на рис. 3.42 показан такой случай на примере графического редактора Paint. Слева виден значок, представляющий объект, созданный в окне графического редактора.

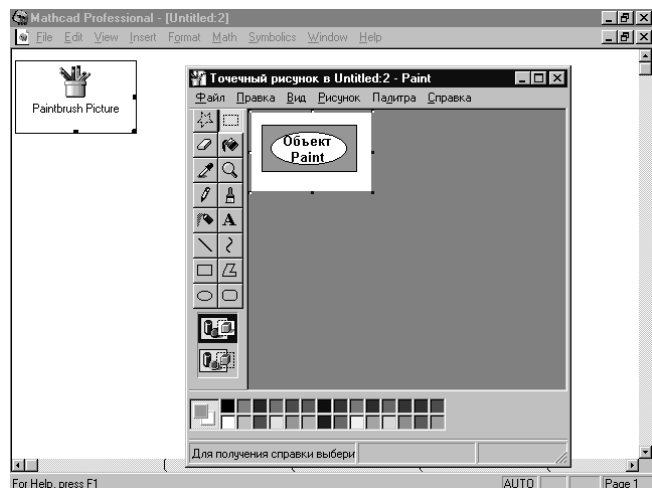


Рис. 3.42. Установление объектной связи с графическим редактором Paint

После закрытия окна графического редактора в окне Mathcad остается значок, представляющий объект Paint. Стоит активизировать этот значок, как в окне документов Mathcad снова появится окно редактора Paint. Таким образом, полученный объект может редактироваться в среде Paint без его специального вызова.

Практически точно так же организуется связь с файлом. Если файл не ассоциирован с порождающим его приложением, появится сообщение о невозможности вставки файла, показанное на рис. 3.43. Такое сообщение часто встречается при попытке использовать в качестве объекта внешние файлы, связь которых с порождающими их приложениями утрачена.

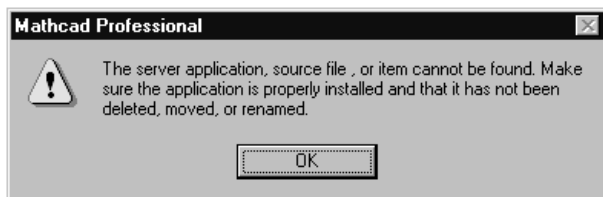


Рис. 3.43. Окно с предупреждением о невозможности вставки объекта

ПРИМЕЧАНИЕ Объекты для связи с Mathcad могут быть самыми разнообразными — помимо текстовых и графических данных это могут быть таблицы и записи из баз данных, музыкальные фрагменты, видеоклипы и т. д. По существу таким образом возможно связать воедино самые различные приложения и обеспечить между ними полноценный обмен данными.

3.6. Особенности вставок в системе Mathcad 11/12

3.6.1. Позиция Insert меню Mathcad 11/12

Позиция Insert меню Mathcad 11/12 претерпела заметные изменения — см. рис. 3.44. В ней появились две новые позиции — списки:

- Data — выводит список вывода шаблона таблицы и окон импорта и экспорта файлов;
- Control — выводит список графических элементов контроля.

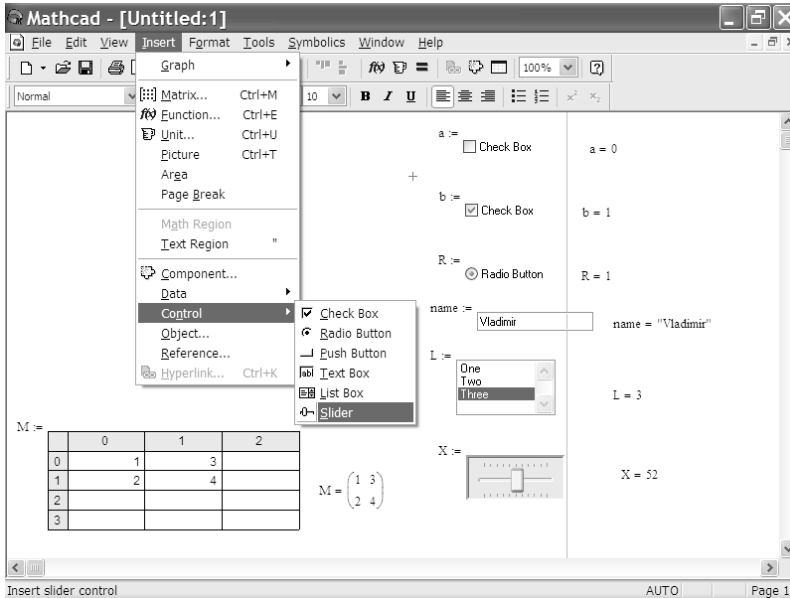


Рис. 3.44. Подменю позиции Insert меню Mathcad 11

3.6.2. Примеры вставки элементов контроля

Применение шаблона таблицы и графических элементов контроля показано на рис. 3.44. К числу этих элементов относятся элементы контроля за знаком птички в блоке, радиокнопка, панель задания имени, переменная — список и ползунковый регулятор, движок которого перемещается мышью. Применение этих элементов контроля разнообразит вид документов и обеспечивает улучшенный контроль за исполнением документов. Стало возможным создание документов с интерактивным управлением.

3.7. Управление вычислениями в документах

3.7.1. Позиция Math меню Mathcad 2001i

Документы Mathcad прежде всего создаются для выполнения тех или иных вычислений. Для управления вычислительным процессом в Mathcad служит меню

Math (Вычисления), в котором сосредоточены команды управления вычислительным процессом (рис. 3.45).

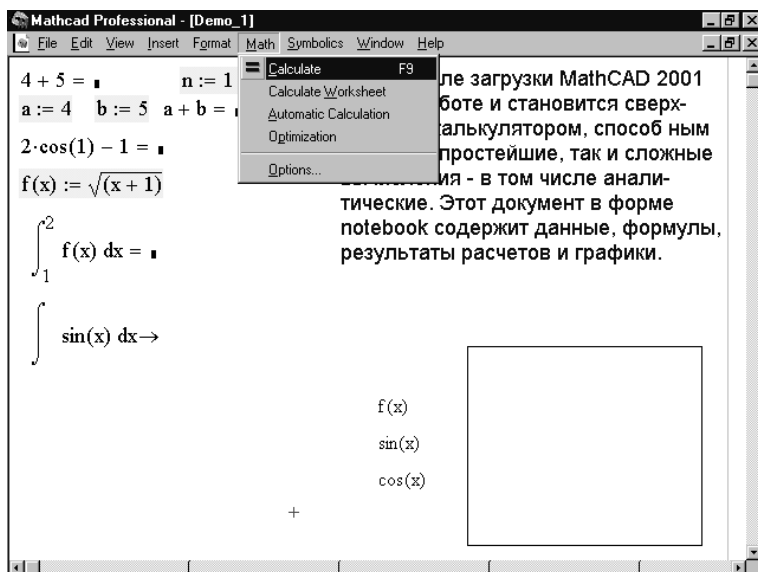


Рис. 3.45. Документ в режиме ручных вычислений и раскрытое меню Math

Команды меню Math перечислены ниже:

- Calculate (Вычислить) — вычисления в пределах видимой части документа;
- Calculate Worksheet (Вычислить в документе) — вычисления всех блоков в документе;
- Automatic Calculate (Вычислить автоматически) — установка режима автоматических вычислений;
- Optimization (Оптимизация) — оптимизация вычислений;
- Options (Параметры) — вывод окна установки параметров вычислений.

Рассмотрим особенности указанных выше команд.

3.7.2. Ручные вычисления

По умолчанию Mathcad работает в режиме автоматических вычислений. При этом вычисления автоматически начинаются, как только тот или иной блок текущего документа попадает в видимую область окна.

Однако иногда полезно выполнять вычисления в ручном режиме — например, если вычисления при каждом изменении исходных данных и выражений выполняются заново и долго. Ручной режим вводится либо командой Calculate (Вычислить) меню Math (Вычисления), либо кнопкой \blacksquare на панели инструментов, либо клавишей F9. Разумеется, такой режим возможен, если отключен режим автоматических вычислений.

Рис. 3.45 иллюстрирует работу Mathcad в ручном режиме вычислений. Видно, что вычислений не происходит. Везде после оператора вывода \blacksquare стоит черный прямоугольник — шаблон вывода. На рис. 3.45 показано подменю Math в откры-

том виде. В нем видны команды управления системы Mathcad 2001i (напоминаем, что у Mathcad 11 позиция Math в меню вообще отсутствует).

Чтобы начались вычисления, надо ввести команду Calculate любым из описанных выше способов. Вид документа после этого показан на рис. 3.46. Нетрудно заметить, что теперь в пределах видимой части документа вычисления выполнены.

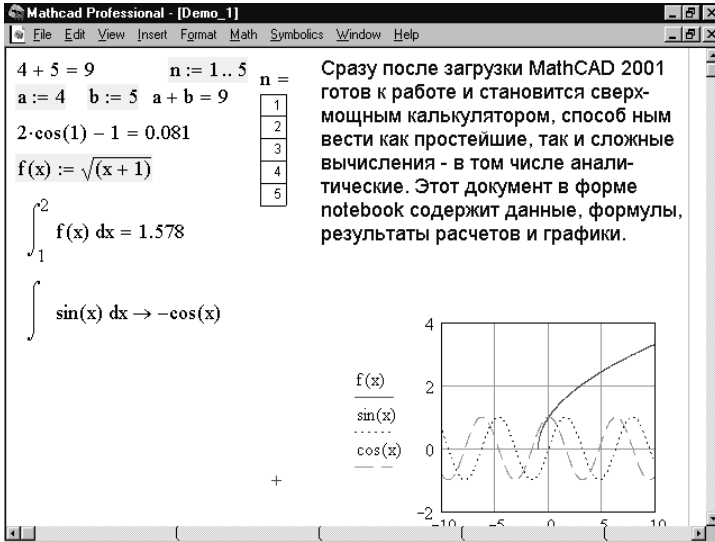


Рис. 3.46. Документ в ручном режиме вычислений после ввода команды Calculate

Однако если перейти в ту часть документа, которая не была видна на экране, то можно обнаружить, что вычисления в этих блоках не выполнялись. Таким образом, команда Calculate обрабатывает только те блоки, которые видны на экране. Для исполнения последующих блоков надо либо выполнять команду Calculate заново, либо сразу использовать команду Calculate Worksheet (Вычислить в документе), которая позволяет выполнить вычисления во всем документе от начала до конца, а не только в его видимой части.

3.7.3. Автоматические вычисления

Команда Automatic Calculation (Автоматические вычисления) меню Math (Вычисления) обеспечивает переключение между ручным и автоматическим режимами вычислений. Если против названия этой команды присутствует галочка, поддерживается автоматический режим вычислений, если галочки нет — ручной режим.

Как правило, автоматический режим вычислений намного удобнее ручного, поэтому он установлен по умолчанию. На время обработки блок заключается в зеленую рамку, а на экране появляются песочные часы. Впрочем, большинство вычислений происходит настолько быстро, что эти признаки не удается даже заметить.

Чтобы прекратить вычисления, надо нажать клавишу Esc.

3.7.4. Оптимизация вычислений

Оптимизация вычислений достигается заменой сложной функции или математического выражения их аналитическим представлением (если, конечно, его удастся получить). Для иллюстрации техники оптимизации рассмотрим рис. 3.47.

$$I(a) := \int_0^a \int_0^a \int_0^a (x^2 + y^2) \cdot z \, dx \, dy \, dz$$

a := 0,0.5..10

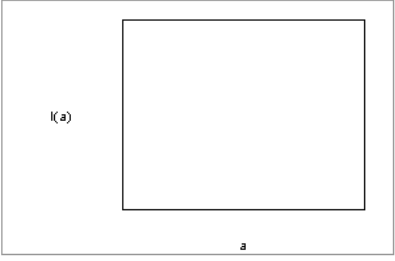


Рис. 3.47. Пример, требующий оптимизации вычислений

Рисунок иллюстрирует задание сложной функции $I(a)$ — тройного интеграла, у которого верхний предел a является аргументом функции. Поскольку вычисления производятся численным методом, то они достаточно трудоемки и задержка выдачи результата заметна даже на скоростных ПК с процессором Pentium. Так, график функции $I(a)$ для 21 точки на ПК класса Pentium II 350 строится примерно 30 секунд. Стоит немного изменить пределы изменения аргумента a или просто перестроить график (удалив аргумент a в месте ввода шаблона графика у оси X и затем вновь введя a), и вы увидите, что повторное построение графика займет то же время.

Причина этого очевидна — интеграл вычисляется сложным численным методом с адаптацией решения под ход функции и заданную погрешность вычислений. И это происходит для каждой из 21 точек графика, заданных ранжированной переменной a . Пока идут вычисления, график обрамляется зеленой рамкой и на экране появляются песочные часы (к сожалению, на рисунке они не видны).

Mathcad имеет радикальное средство по повышению эффективности подобных вычислений, названное *оптимизацией*. Суть его проста — вместо сложных численных вычислений получается аналитическое выражение, по которому и проводятся последующие вычисления. При этом скорость и время вычислений нередко сокращаются в десятки раз!

Для включения оптимизации достаточно выбрать команду Optimization (Оптимизация) в меню Math (Вычисления). Признаком оптимизации выражения является появление правее оптимизируемых выражений (в нашем случае это функция с тройным интегралом) красной звездочки (рис. 3.48). Кроме того, установив в оптимизированное выражение указатель мыши и щелкнув ее левой кнопкой, можно наблюдать появление окна с оптимизированным выражением (на рисунке это окно также показано). Нетрудно заметить, что в нашем примере довольно сложный интеграл заменен достаточно простым аналитическим выражением, что и объясняет фантастическое уменьшение времени вычислений — график строится в доли секунды!

Очень редко, но бывают случаи, когда исходное выражение превращается в не менее сложное «оптимизированное» выражение. В таких случаях оптимизация оказывается менее эффективной. Оптимизацию можно задать в тексте документа специальными словами — директивами, но эта возможность будет рассмотрена в дальнейшем (см. главу 9). Установка птички против команды оптимизации в

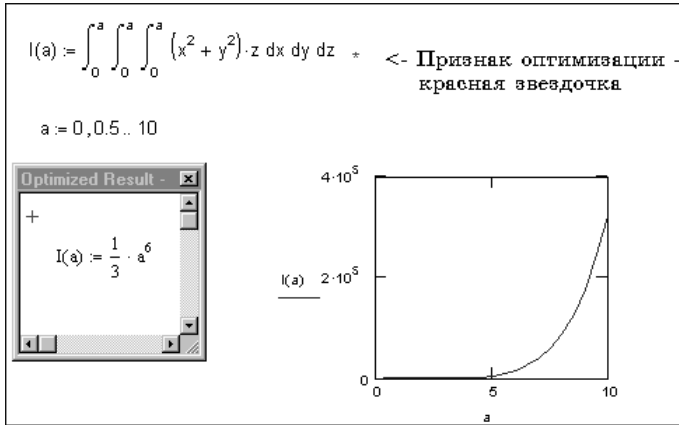


Рис. 3.48. Пример применения оптимизации для построения графика сложной функции

меню при отсутствии выделенных блоков означает, что оптимизация задается для всех блоков документа.

ПРИМЕЧАНИЕ Вычисления с оптимизацией нередко ведут к устранению ошибок, которые появляются при обычных вычислениях. Например, нередки ошибки, связанные с переполнением разрядной сетки системы Mathcad при значительных промежуточных расчетах. Эта ситуация встречается довольно часто при вычислении бесконечных сумм и произведений. Нередко вычисления с оптимизацией дают более точные результаты, чем встроенные функции для численных вычислений (например определенных интегралов).

3.7.5. Запрет вычисления выражений

Иногда необходимо использовать вычислительные блоки в качестве комментариев. В этом случае они должны быть невычисляемыми и иметь статус пассивных выражений. При этом блоки помечаются значком черного прямоугольника. Для такой отметки достаточно установить курсор ввода в соответствующем выражении и командой Format (Формат) ▶ Properties (Свойства) вызвать окно свойств математического выражения (см. также раздел «Установка свойств» в главе 5). В этом окне на вкладке Calculation (Вычисление) надо установить флажок Disable Evaluation (Запретить выполнение). Другой способ — щелкнуть правой кнопкой мыши и в контекстном меню выбрать команду Disable Evaluation.

Чтобы вернуть выражению статус вычисляемого, надо снять флажок Disable Evaluation или в контекстном меню выбрать команду Enable Evaluation (Разрешить выполнение) — она будет в этом меню, если блок помечен как невычисляемый.

3.8. Установка опций вычислений

3.8.1. Обзор опций вычислений

Если выбрать команду Options (Параметры) в меню Math (Вычисления), на экран будет выведено диалоговое окно, представленное на рис. 3.49.

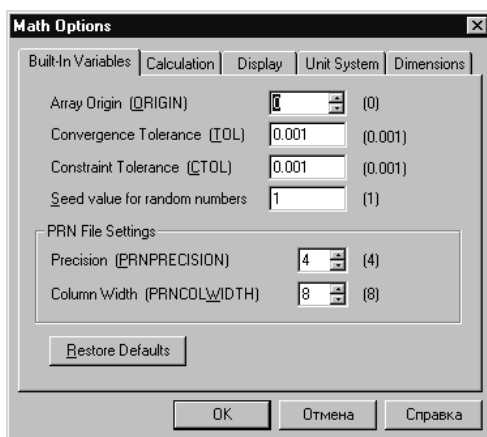


Рис. 3.49. Диалоговое окно установки параметров вычислений

Это окно имеет следующие вкладки:

- Built-in Variables (Встроенные переменные) — параметры встроенных (системных) переменных;
- Calculation (Вычисления) — параметры вычислений;
- Display (Отображение) — параметры отображения;
- Unit System (Система единиц) — установка системы единиц для размерных величин;
- Dimensions (Размерности) — установка размерности величин.

Рассмотрим назначение параметров на каждой из этих вкладок подробнее.

3.8.2. Установка значений системных переменных

В системе Mathcad имеется несколько так называемых *системных переменных*. У них есть свои имена и значения, присваиваемые системой по умолчанию. Действие системных переменных глобальное, то есть они доступны в любом документе программы и изменять их значения можно также в любом месте программы.

Вкладка Build-In Variables (Встроенные переменные), показанная на рис. 3.49, служит для установки значений системных переменных (в скобках показаны значения по умолчанию):

- ORIGIN (0) — нижняя граница значения индексов векторов и матриц;
- TOL (0.001) — погрешность численных расчетов;
- CTOL (0.001) — погрешность задания ограничений;
- PRNCOLWIDTH (8) — число столбцов для инструкции WRITEPRN;
- PRNPRECISION (4) — число десятичных знаков, используемых для записи численных данных в инструкции WRITEPRN.

Из этих переменных особенно важной является переменная TOL, поскольку она задает погрешность многих численных методов, реализованных в системе Mathcad. Большинство этих методов — адаптивные итерационные методы. Итерации при этом выполняются, пока погрешность метода не станет меньше TOL. Это отчетливо видно из приведенного ниже примера, где функция root используется для нахождения решения уравнения $x^2 - 2 = 0$:

$x := 1$	$\sqrt{2} = 1.4142135624$
TOL := 0.1	$\text{root}(x^2 - 2, x) = 1.4066543438$
TOL := 0.01	$\text{root}(x^2 - 2, x) = 1.4138276373$
TOL := 0.0001	$\text{root}(x^2 - 2, x) = 1.4142156833$
TOL := 0.000001	$\text{root}(x^2 - 2, x) = 1.4142135621$

Нетрудно заметить, что к точному решению (корень квадратный из двух) вычисления приближаются по мере уменьшения заданного значения погрешности TOL. Таким образом, неверное задание значений переменной TOL (и использование ее значения по умолчанию) может привести к значительным погрешностям при численных расчетах.

В Mathcad 2001i/11 предусмотрена также установка базы генератора случайных чисел — поле Seed value for random numbers (по умолчанию 1).

Значения системных переменных можно вывести непосредственно в документе, указав после их имени знак =.

3.8.3. Опции вычислений

Вкладка Calculation (Вычисления) показана на рис. 3.50. Это самая простая вкладка в диалоговом окне параметров.

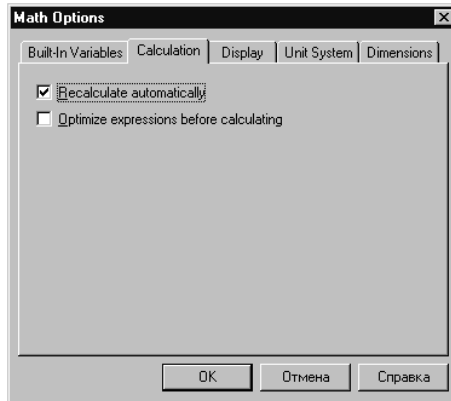


Рис. 3.50. Окно параметров с открытой вкладкой Calculation

На этой вкладке имеется всего два флажка:

- Recalculate automatically (Пересчитывать автоматически) — автоматическое вычисление математических выражений при их изменении или изменении входящих в них объектов;
- Optimize expressions before calculating (Оптимизировать перед вычислением) — оптимизация вычислений (см. выше подраздел «Оптимизация вычислений»).

3.8.4. Опции отображения результатов вычислений

Вкладка Display (Отображение), показанная на рис. 3.51, служит для установки параметров отображения объектов на экране.

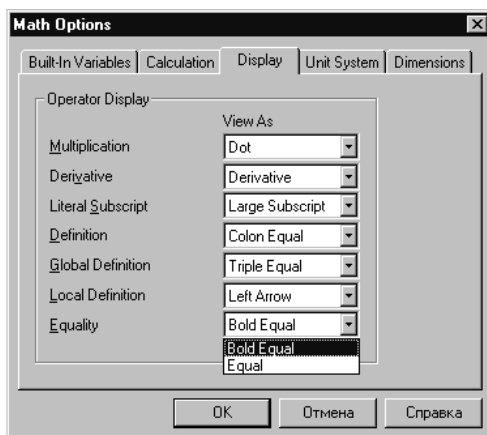


Рис. 3.51. Окно параметров с открытой вкладкой Display

На этой вкладке находится семь раскрывающихся списков для выбора способов отображения объектов и операторов на экране, в частности операторов умножения, вычисления производных и др.

3.8.5. Выбор системы единиц

На рис. 3.52 показана вкладка Unit System (Система единиц), которая служит для выбора систем размерных единиц.

В этом окне с помощью переключателей можно выбрать одну из следующих систем единиц для размерных величин: SI (International), MKS, CGS, U. S. и None (не используется ни одна из систем).

3.8.6. Форматирование размерных единиц

Для изменения формата размерных единиц служит вкладка Dimensions (Размерности), показанная на рис. 3.53.

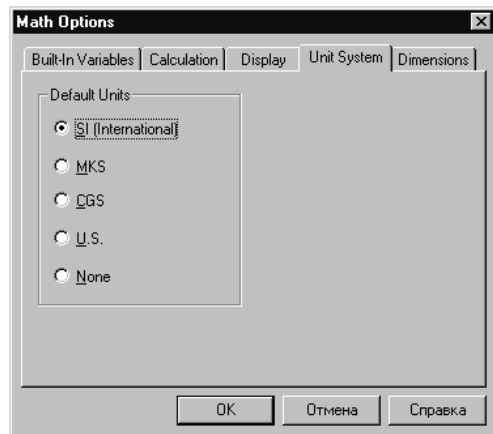


Рис. 3.52. Вкладка выбора системы размерных единиц

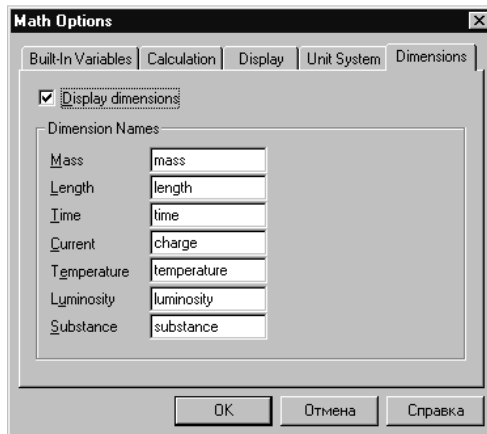


Рис. 3.53. Изменение формата размерных единиц

В этом окне выведен перечень имен размерных единиц, который при необходимости может редактироваться. Для этого надо установить флажок `Display dimensions` (Просмотр размерностей).

3.9. Ошибки в документах и отладка вычислений

3.9.1. Виды ошибок в документах Mathcad

Как совершенно верно подмечено в поговорке «Не ошибается только тот, кто ничего не делает», вычисления в системах Mathcad, увы, часто сопровождаются ошибками. О роли ошибок и путях их устранения впору написать большую книгу, но мы вынуждены ограничиться лишь коротким их описанием.

Есть ошибки, в которых система Mathcad неповинна. Прежде всего, это ошибки в записях алгоритмов пользователем. Например, если в какой-то формуле пользователь вместо функции $\sin(x)$ запишет $\ln(x)$, то никакая система компьютерной математики такую грубейшую ошибку не обнаружит, поскольку обе записи *синтаксически* абсолютно верные. Это типичная ошибка пользователя, не понявшего разницы между вычислением синуса x и логарифма x .

Другой распространенный вид ошибок — синтаксические ошибки, или, попросту говоря, неправильные записи операторов и функций. Например, совершенно правильная с точки зрения математики запись $\arcsin(x)$, представляющая обратный тригонометрический синус, является грубейшей синтаксической ошибкой на входном языке системы Mathcad, в котором эта функция обозначена проще — $\text{asin}(x)$.

К счастью, ошибки такого рода Mathcad отлавливает, и довольно неплохо. Однако лучше их не допускать уже на уровне записи математических выражений. В сомнительных случаях стоит ввести шаблон функции, запись которой вызывает у вас неуверенность в ее правильности. Напоминаем, что список всех встроенных функций появляется при щелчке на кнопке $f(x)$ панели инструментов.

Особенно коварны в Mathcad ошибки, связанные с использованием индексов. Часто путается подстрочное написание символов в именах переменных с индексами массивов — векторов и матриц. Нередко пользователи путают векторы с ранжированными переменными и не следят за разницей между скалярными и векторными величинами. Несоответствие типов переменных — еще один повод для многочисленных ошибок. Например, нельзя применять арифметические операторы для переменных, имеющих строковые значения.

ПРИМЕЧАНИЕ *Первое время Mathcad может «забросать» пользователя сообщениями об ошибках, но по мере освоения системы такие сообщения будут появляться все реже и реже. Кроме того, пользователь, знающий математику, меньше рискует получить сообщение об ошибке, чем пользователь, «хромающий» в школе или в вузе по этому предмету.*

3.9.2. Индикация ошибок

В рассматриваемых в данной книге версиях системы Mathcad индикация ошибок задается двумя признаками:

- ошибочное выражение или его часть окрашиваются ярко-красным цветом;

- при наведении указателя мыши на область ошибки появляется всплывающая подсказка с сообщением о сути ошибки.

К сожалению, на черно-белом рисунке 3.54 видна только всплывающая подсказка. На этом рисунке показана довольно типичная ошибка — подстановка в функцию пользователя для численных расчетов строкового значения одной из переменных.

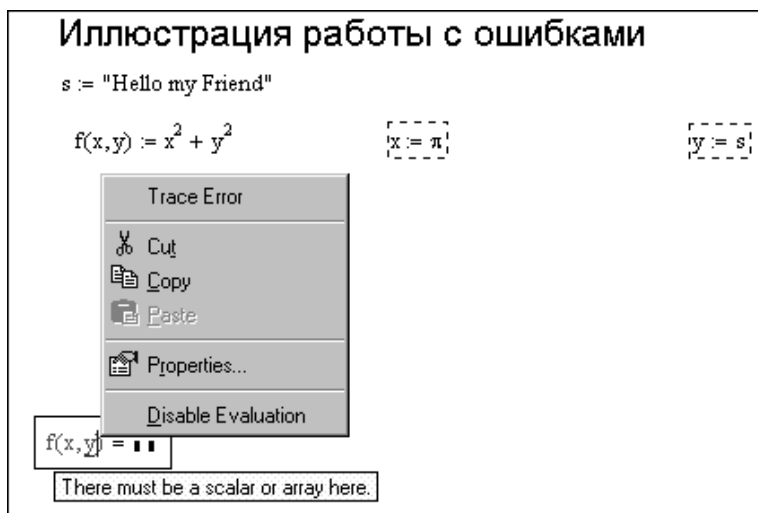


Рис. 3.54. Пример индикации ошибки в документе

Полный перечень ошибок насчитывает многие десятки их видов. Список наиболее распространенных сообщений об ошибках приведен в приложении 5.

3.9.3. Трассировка ошибок

Вообще говоря, тестирование документов на предмет выявления ошибок — ахиллесова пята систем класса Mathcad. В этом отношении они заметно уступают современным языкам программирования, имеющим средства трассировки и отладки программ. Это не случайно, ведь синтаксические ошибки являются далеко не самыми коварными и распространенными в математических расчетах.

Тем не менее в Mathcad 2001i/11/12 имеется трассировщик ошибок, который позволяет «вылавливать» ошибки на различных шагах вычислений. Если установить указатель мыши в область ошибочного выражения и щелкнуть правой кнопкой мыши, появится контекстное меню (см. рис. 3.54), в котором имеется команда трассировки ошибок — Trace Error (Трассировка ошибок). Выбрав эту команду, можно наблюдать появление удлиненного окна трассировки ошибок (рис. 3.55).

Окно трассировки имеет четыре кнопки для перемещения по трассе ошибок с очевидным назначением (перемещение вперед, к следующей или последней ошибке, либо назад, к предыдущей или первой ошибке). Пятая кнопка, Close (Закрыть), закрывает окно трассировки. В нашем случае щелчок на кнопке Previous (Предыдущая) ведет к обнаружению ошибки в задании функции пользователя (рис. 3.56) — в ней для вычислений использована переменная, которой было присвоено значение символьной переменной s .

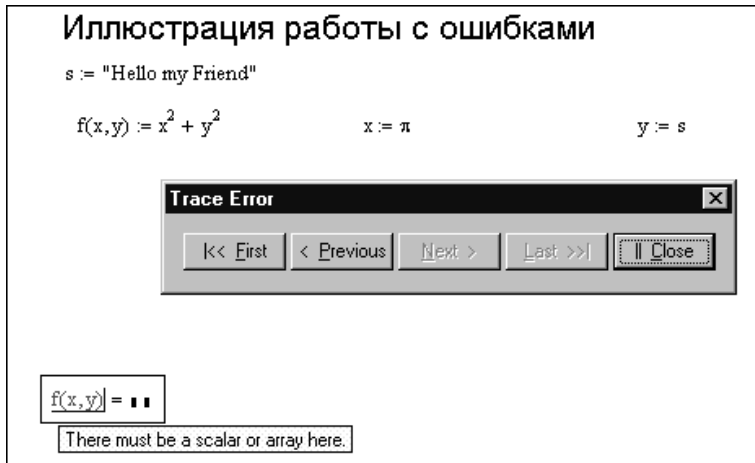


Рис. 3.55. Окно трассировки ошибок



Рис. 3.56. Обнаружена ошибка в предшествующем выражении

В целом подобная трассировка достаточно примитивна, но для начинающих пользователей, безусловно, полезна. Однако польза от нее будет только в том случае, если внимательно анализировать сообщения об ошибках (см. приложение 5).

В общем случае лучший способ отладки документов — тщательный контроль вычислений на каждом их этапе с учетом особенностей конкретного алгоритма решаемой задачи. Разумеется, это требует от пользователя четкого понимания деталей работы этого алгоритма.

Заканчивая этот раздел, надо отметить, что существует ошибочное мнение о том, что для работы с системой Mathcad никакой математической подготовки не требуется. На самом деле это далеко не так. Без знания основ математики применение Mathcad не только бесполезно, но порою и вредно. Оно создает иллюзорное и неверное представление о простоте решения сложных задач. На самом деле Mathcad *помогает* пользователю в решении его задач, а вовсе не решает их за него. Об этом стоит всегда помнить, работая с системами класса Mathcad.

3.10. Работа с командами позиции Tools меню Mathcad 11/12

3.10.1. Обзор команд позиции Tools меню Mathcad 11/12

В системе Mathcad 11/12 введена новая позиция Tools (Инструменты) меню. Ее команды для Mathcad 11 представлены на рис. 3.57.

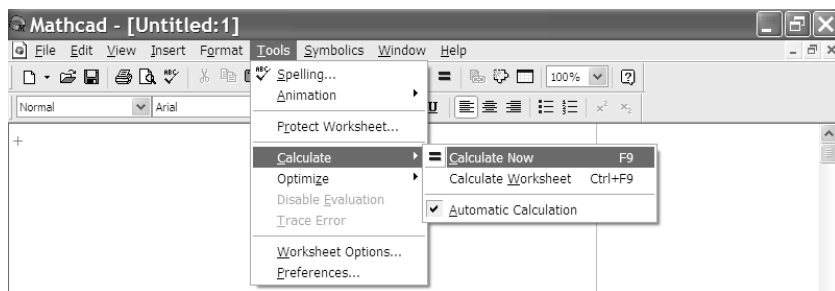


Рис. 3.57. Команды позиции Tools меню Mathcad 11

В позицию Tools входят следующие команды:

- Spelling... — включение орфографического контроля;
- Animation — выполнение анимации (команды Record... и Playback...);
- Protect WorkSheet... — открытие окна установки защиты документа;
- Calculate — команды управления вычислениями (Calculate Now F9 — пуск вычислений текущей страницы, Calculate Worksheet Ctrl+F9 — пуск вычислений по всему документу и Automatic Calculation — автоматические вычисления);
- Optimize — оптимизация (Equation — уравнения, Worksheet — текущего документа и Show Optomization — показ оптимизации);
- Disable Evaluation — управление (включение и выключение) вычислением уравнений;
- Trace Error — трассировка ошибок;
- Worksheet options... — открытие окна установки опций документа;
- Preferences... — открытие окна Preferences (см. раздел 2.6.5).

Большинство команд этой позиции меню уже описывалось ранее, поэтому остановимся на тех командах, которые специфичны для системы Mathcad 11. При этом команды анимации, как относящиеся к графике, мы рассмотрим в следующей главе. Из новых возможностей можно отметить комбинацию клавиш Ctrl+F9 для команды пуска вычислений по всему документу. В Mathcad 12 к описанным позициям меню добавилась новая позиция License, обеспечивающая доступ к данным лицензирования.

3.10.2. Защита документов Mathcad 11/12

Система Mathcad часто используется в учебных целях. В таких случаях важно обеспечить парольную защиту документов. Иначе школьники или студенты групп, занимающихся в компьютерных классах, ухитряются легко списывать работы, выполненные их предшественниками или даже преподавателями. Этой «лафе» при-

шел конец с применением новой версии системы Mathcad 11. В ней командой Protect WorkSheet... предусмотрена парольная защита текущего документа. Эта команды вызывает открытие окна защиты документа, представленного на рис. 3.58.

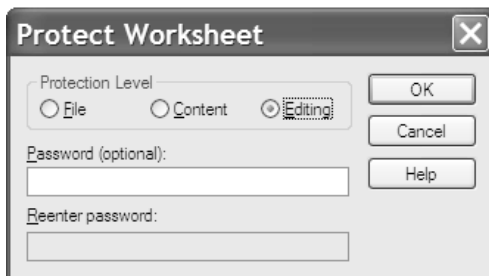


Рис. 3.58. Окно ввода пароля для защиты документа Mathcad 11

В этом окне предусмотрены опции защиты файла (File), содержимого (Content) и операций редактирования (Editing). Все, что нужно для защиты, это ввести пароль (password) и повторить его ввод для контроля.

3.10.3. Опции вычислений в документах Mathcad 11/12

Команда Worksheet Options открывает окно установки опций вычислений, показанное на рис. 3.59 с открытой вкладкой Build-in Variables (Системные переменные).

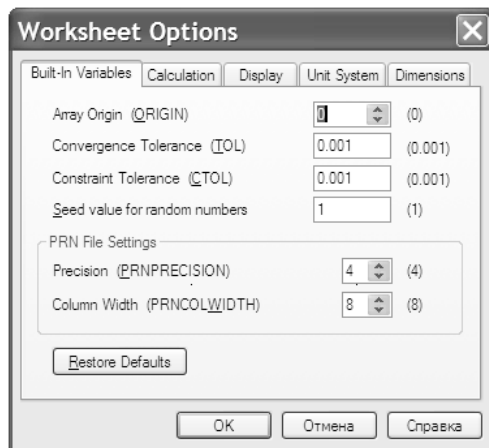


Рис. 3.59. Окно установки опций вычислений Mathcad 11/12

Нетрудно заметить, что за исключением несколько более изысканного внешнего вида это окно ничем не отличается от окна этой команды, показанного на рис. 3.49 для системы Mathcad 2001i. Поэтому описание назначения всех вкладок окна Worksheet Options вполне соответствует описанию, приведенному в разделе 3.8.

Глава 4. Графика

Для создания графиков в системе Mathcad имеется программный графический процессор. Основное внимание при его разработке было уделено обеспечению простоты задания графиков и их модификации с помощью соответствующих параметров. Процессор позволяет строить самые разные графики, например, в декартовой и полярной системе координат, графики поверхностей, трехмерные фигуры, графики уровней и т. д.

4.1. Виды графиков

Графики в системе Mathcad являются такими же объектами, как и константы, переменные, текстовые блоки, формулы и так далее. В позиции Graph подменю вставок Insert имеется список следующих шаблонов графики (рис. 4.1):

- X-Y Plot (Декартов график);
- Polar Plot (Полярный график);
- 3D Plot Wizard (Мастер трехмерной графики);
- Surface Plot (График поверхности);
- Contour Plot (Контурный график);
- 3D Scatter Plot (Точечный график);
- 3D Bar Plot (Трехмерная гистограмма);
- Vector Field Plot (Векторное поле).

Рассмотрим эти виды графиков более подробно.

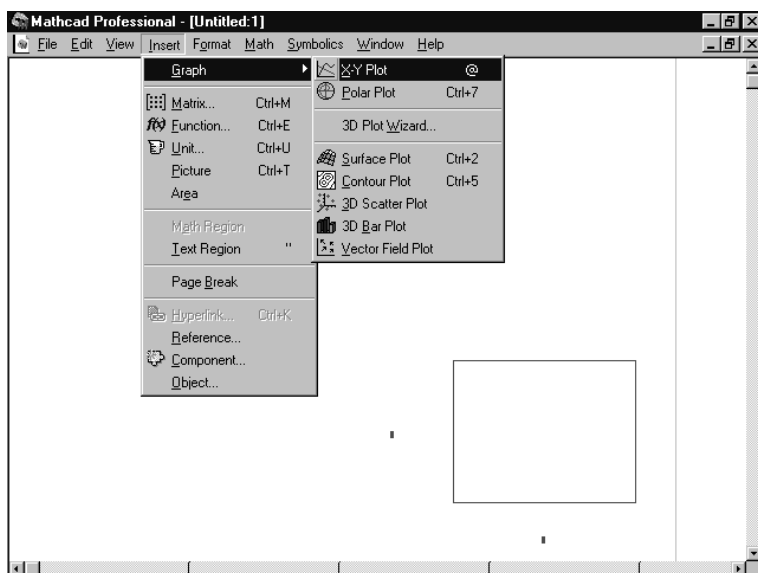


Рис. 4.1. Подменю со списком графиков системы Mathcad

4.2. Двумерный график

4.2.1. Вставка шаблона двумерного графика

Графики в системе Mathcad могут иметь различные размеры и перемещаться в окне редактирования документа. Для вывода шаблона двумерной графики в декартовой системе координат служит команда X-Y Plot (Декартов график) или клавиша @. Она выводит в текущем положении курсора шаблон двумерного графика. Напоминаем, что это обычный график на плоскости с воображаемыми (или действительно нарисованными) осями X (горизонтальная) и Y (вертикальная), расположенными под прямым углом друг к другу.

Каждая точка графика в декартовой системе координат характеризуется своими координатами x и $y = f(x)$, где x — абсцисса точки, а y — ее ордината. Точки соединяются друг с другом разнообразными линиями (сплошной, пунктирной и т. д.). Могут быть показаны исходные (узловые) точки графика в виде жирных точек, квадратиков, кружков и т. д. Возможно также построение на одном рисунке графиков нескольких функций. Эти особенности представления графиков задаются их форматированием, которое будет рассмотрено позже (см. главу 5).

Незаполненный шаблон графика представляет собой большой пустой прямоугольник с местами ввода данных в виде темных маленьких прямоугольников, расположенных около осей абсцисс и ординат будущего графика. В них необходимо ввести выражения, задающие координаты точек графика по осям X (горизонтальная ось) и Y (вертикальная ось). В общем случае это могут быть функции некоторой переменной x .

Если строятся графики нескольких функций в одном шаблоне, то функции следует разделять запятыми. Для выделения данных в шаблоне удобно использовать клавиши перемещения курсора. Можно также выделить данные в шаблоне с помощью мыши.

Если график уже построен, то при его выделении появляются крайние места ввода с автоматически введенными числами, которые служат для указания предельных значений абсцисс и ординат, то есть задают масштаб графика. В принципе, можно не заполнять эти места ввода самостоятельно, но тогда масштаб, скорее всего, окажется не вполне удобным, например, будет представлен неокругленными десятичными числами, обеспечивающими максимальный размер графика. Тем не менее рекомендуется всегда вначале использовать автоматическое масштабирование и лишь затем выбрать более подходящий масштаб.

4.2.2. Особенности построения графиков функции одной переменной

Для наиболее распространенных графиков в декартовой системе координат Mathcad предусматривает два способа построения графиков функций одной переменной $f(x)$:

- упрощенный способ без задания ранжированной переменной x (пределы изменения x автоматически задаются от -10 до 10);
- обычный способ с заданием ранжированной переменной x .

Напомним, что для упрощенного построения двумерного графика некоторой функции $f(x)$ надо ввести выражение для правой части этой функции, отметить его курсором ввода (синим уголком) и затем вывести шаблон двумерного графика.

Останется ввести x в место ввода горизонтальной оси и, отведя указатель мыши в сторону и щелкнув левой кнопкой, получить готовый график. Таким же образом можно строить на одном рисунке и графики нескольких функций — просто опишите их у вертикальной оси, используя запятые для разделения функций. Графики будут построены линиями разного типа и цвета.

При обычном способе построения графиков необходимо ввести саму функцию и интервал изменения ее аргумента (например, x). Впрочем, простые функции, если они в дальнейшем не используются, можно указать непосредственно в шаблоне графика. Далее надо приблизительно наметить место верхнего левого угла графика и установить на это место курсор. Затем следует выбрать команду X-Y Plot (Декартов график) в подменю Graph (График) меню Insert (Вставка) или нажать клавишу @.

Начинающие пользователи обычно задают ранжированную переменную x целочисленной, например, $x := -15..15$. При этом они забывают, что в данном случае график задается небольшим числом точных целочисленных значений x : $-15, -14, -13, \dots, -1, 0, 1, 2, 3, \dots, 14, 15$. В некоторых случаях (рис. 4.2 сверху) это ведет к грубому искажению формы графиков.

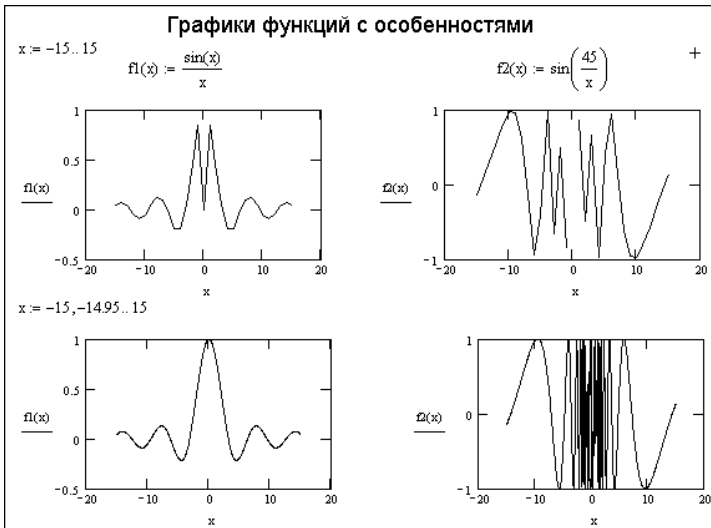


Рис. 4.2. Построение графиков функций

В левой части рисунка построены графики функции $\sin(x)/x$, имеющей устранимую особенность типа $0/0 \rightarrow 1$ в точке $x = 0$. Однако система Mathcad ведет себя в данном случае как типичная система для численных расчетов — вне зависимости от значения знаменателя дроби она считает значение дроби равным нулю, если числитель равен нулю. При указанном задании переменной x в точке $x = 0$ получаем значение функции $f1(x) = 0$, что и дает провал первого графика в точке $x = 0$. Интересно отметить, что в Mathcad 11 введена новая функция $\text{sinc}(z) = \sin(z)/z$ которая учитывает устранимую особенность в точке $z = 0$ и может использоваться с комплексным аргументом.

Графики функции $\sin(45/x)$, показанные в правой части рисунка, при $x \rightarrow 0$ дают резкое повышение частоты синусоиды. При недостаточно малом шаге изменения x , равном 1, особенность этой функции практически незаметна.

Простейшим способом избавиться от указанных недостатков графиков является их построение при более мелком шаге изменения x . Например, если задать x как $x:=-15, -14.95..15$, шаг изменения x будет равен 0.05. Указание в числе -14.95 разделительной точки означает переход от целочисленного представления x к представлению в виде принципиально приближенных вещественных чисел с плавающей точкой. В результате числитель выражения $\sin(x)/x$ хоть немного, но отличается от нуля в точке примерного равенства x нулю. Результат налицо — нижние графики на рисунке уже дают хорошее представление о форме и особенностях функций.

4.2.3. Простейшие приемы форматирования двумерных графиков

Чтобы построить график в автоматическом режиме вычислений, достаточно вывести указатель мыши за пределы графического объекта и щелкнуть левой кнопкой. В «ручном» режиме вычислений для этого нужно еще нажать клавишу F9. При построении во время вычислений ординат функции область графика покрывается зеленой штриховкой, затем графики функций появляются внутри шаблона (см. рис. 4.2).

Если что-либо в построенном графике не вполне удовлетворяет пользователя, можно применить описанные в главе 5 команды изменения формата графиков. Эти команды позволяют изменять заданные по умолчанию параметры графиков. Заметим, что окно задания форматов графиков появляется, если дважды щелкнуть мышью на графике либо щелкнуть один раз, если график выделен.

Графики можно перемещать по полю окна документа и изменять в размерах. Для этого надо выделить график, щелкнув на нем мышью. Можно поступить и по-иному — поместить указатель мыши вблизи графика и, нажав левую кнопку мыши, перемещать указатель в направлении графика, захватывая его рамкой выделения в виде прямоугольника из черных пунктирных линий. Как только график окажется внутри пунктирного прямоугольника, надо отпустить кнопку мыши. График будет выделен.

Стоит поместить указатель мыши вблизи границы выделения, форма указателя изменится — вместо маленького красного крестика он приобретет вид ладони руки. Если теперь начать перемещать мышью при нажатой левой кнопке, то весь шаблон графика будет перемещаться. Установите его в нужное место и отпустите левую кнопку мыши. Рисунок окажется на новом месте.

Для изменения размеров рисунка нужно подвести указатель мыши к маркерам изменения размера. Эти маркеры имеют вид маленьких черных прямоугольников. Указатель при этом приобретет форму двухсторонней стрелки, указывающей, в каких направлениях можно изменять размер рисунка. Нажав левую кнопку мыши и захватив соответствующую сторону или угол шаблона рисунка, можно, не отпуская кнопки, растягивать или сжимать шаблон. После того как кнопка будет отпущена, размер рисунка изменится. Сжимать и растягивать графики можно в вертикальном, горизонтальном и диагональном направлениях.

Если при выделенном рисунке нажать клавишу F3, рисунок будет перенесен в буфер обмена. Переместив курсор в новое место и нажав клавишу F4, можно вставить рисунок на новое место. Кроме того, ряд команд форматирования графиков имеется в контекстном меню. Оно появляется при щелчке на графике правой кнопкой мыши.

Полезно отметить, что помимо команд для операций с буфером обмена в контекстном меню имеются команды трассировки графиков Trace (Трассировка), из-

менения масштаба выделенной части графика Zoom (Масштаб) и вывода графиков или текстовых надписей поверх изображения Bring to Front (На передний план) или под ним Send to Back (На задний план). Это позволяет создавать сложные графики с поясняющими надписями, которые невозможно задать при использовании обычных команд форматирования. Эти возможности мы проиллюстрируем по мере описания графических средств системы Mathcad.

4.2.4. Графики с параметрическим заданием функций

Допускается строить двумерные графики с параметрическим заданием функций по осям координат. При этом в местах ввода могут стоять произвольные функции одной переменной x . На рис. 4.3 показаны 4 графика, наглядно иллюстрирующие, что, в сущности, двумерная графика — это графика с параметрическим заданием функций по осям координат.

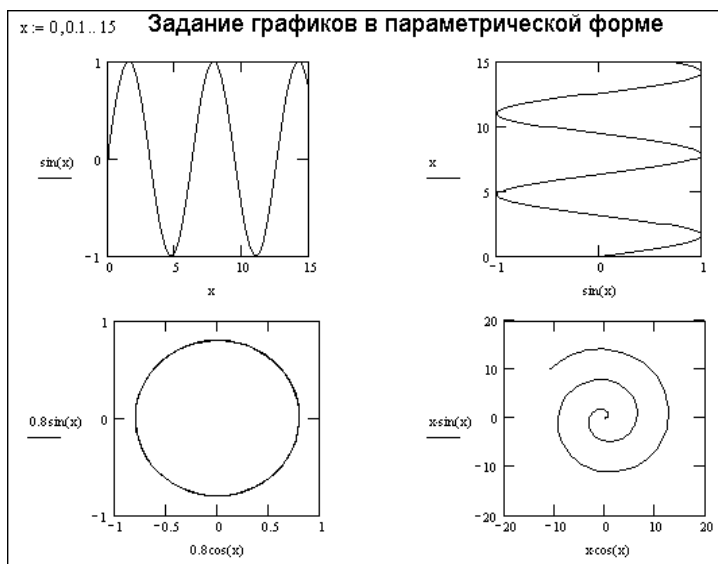


Рис. 4.3. Задание графиков в параметрической форме

Два первых графика на рисунке иллюстрируют эффект от перемены функций, заданных по осям X и Y . В результате функция $\sin(x)$ оказывается повернутой на угол 90° . На третьем графике показано построение параметрически заданной окружности по формулам $R\cos(x)$ и $R\sin(x)$, где $R = 0.8$ — радиус окружности. На последнем графике показано построение спирали по формулам $x\cos(x)$ и $x\sin(x)$.

Все графики, представленные на рисунке, строятся после задания независимой переменной x как ранжированной. Однако можно и не задавать переменную x . В этом случае диапазон ее изменения автоматически устанавливается в интервале от -10 до 10 .

4.2.5. Построение графиков ряда функций на одном рисунке

Как уже отмечалось, допускается построение графиков ряда функций с перечислением их через запятую в месте ввода шаблона по оси Y . Однако допускается

задание различных функций не только по оси Y , но и по оси X . Все это означает возможность построения нескольких графиков разного типа на одном рисунке, что и иллюстрирует рис. 4.4.

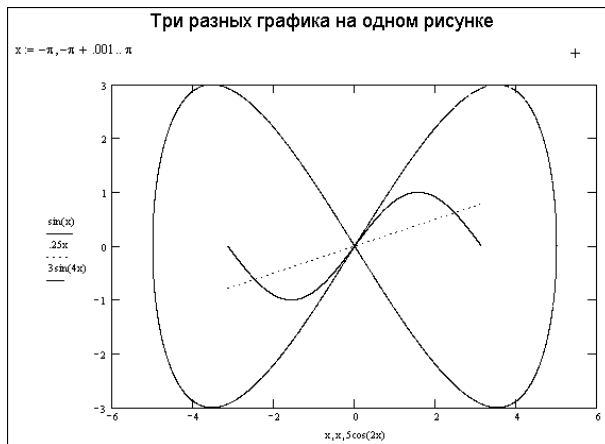


Рис. 4.4. Графики трех функций на одном рисунке — одна из них задана параметрически

Параметрически заданная функция на рисунке образует так называемую фигуру Лиссажу, которую наблюдают на экране осциллографа, подавая на отклоняющие пластины X и Y синусоидальные сигналы с кратными частотами (в нашем случае кратность равна $2/4$ и фигура напоминает знак бесконечности).

4.2.6. Полулогарифмические и логарифмические графики

В ряде случаев графики в линейном масштабе, установленном по умолчанию, бывают недостаточно представительными и не выявляют математических закономерностей построения той или иной функции. Например, график логарифмической функции в линейном масштабе имеет резко нелинейный вид с крутым участком при малых значениях x и довольно плоским при больших значениях x (рис. 4.5).

При построении полулогарифмического графика (масштаб по оси y линейный, а по оси x логарифмический) график логарифмической функции превращается в прямую. На рисунке снизу представлены также графики функций x и x^2 в логарифмическом масштабе по обеим осям — иногда его называют двойным логарифмическим масштабом. Обратите внимание на то, что оба графика имеют вид прямых, но с разными углами наклона.

Графики в логарифмическом и полулогарифмическом масштабе строятся как обычные графики. Однако их надо форматировать с помощью окна форматирования. На рисунке это окно показано в правом нижнем углу окна документа. Для его вызова достаточно установить указатель мыши на графике и дважды щелкнуть левой кнопкой мыши.

На вкладке X - Y Axes (Оси X - Y) окна форматирования двумерных графиков нужно установить флажок Log Scale (Логарифмический масштаб). С другими возможностями панели форматирования графиков мы познакомимся позже в главе 5, посвященной форматированию объектов, в том числе графиков.

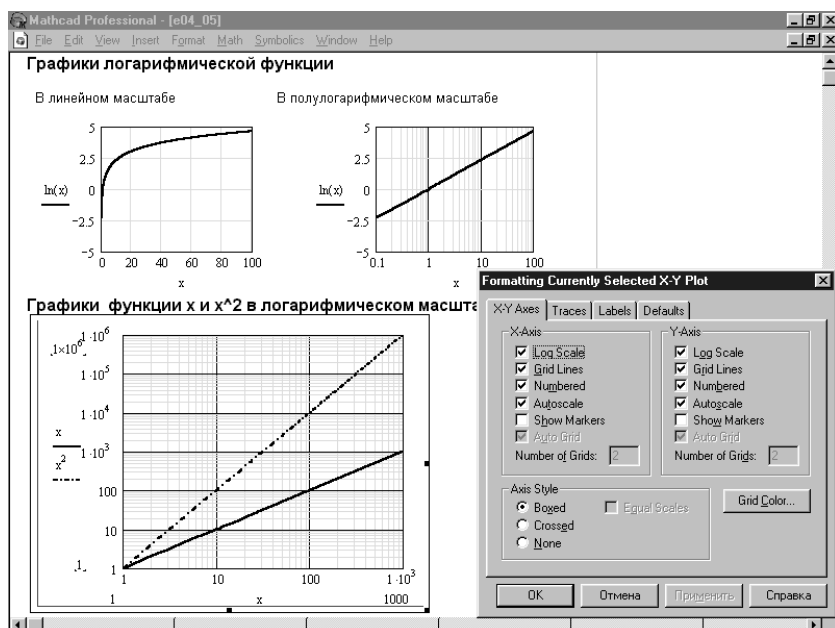


Рис. 4.5. Графики с разным масштабом

Графики такого типа довольно широко используются в радиотехнических расчетах для построения логарифмических амплитудно-частотных характеристик — ЛАЧХ. Они применяются также для анализа устойчивости радиотехнических устройств и систем.

4.2.7. Построение графиков в полярной системе координат

В полярной системе координат каждая точка задается углом \bar{w} и длиной его радиус-вектора $R(\bar{w})$. График функции обычно строится при изменении угла \bar{w} в определенных пределах, чаще всего от 0 до 2. Выбор команды Polar Plot (Полярный график) в подменю Graph (График) меню Insert (Вставка) или нажатие комбинации клавиш $\text{Ctrl}+7$ выводит шаблон таких графиков, показанный на рис. 4.6. Этот шаблон имеет форму окружности и содержит места ввода данных. На рисунке также показана всплывающая подсказка с сообщением о необходимости заполнения мест ввода для построения графика.

Перед построением таких графиков надо задать пределы изменения ранжированной переменной \bar{w} (она может иметь и другое имя). После вывода шаблона следует ввести \bar{w} в место ввода снизу и функцию $R(\bar{w})$ в место ввода слева, а также указать нижний предел изменения длины радиус-вектора $R(\bar{w})$ — R_{\min} — в место ввода справа снизу и верхний предел — R_{\max} — в месте ввода справа сверху. Эти места ввода становятся видимыми при выделении графика.

На рис. 4.7 показано построение ряда графиков в полярной системе координат — от одиночной окружности радиуса 0.8 до трех кривых на одном графике.

Надо также отметить возможность прямого построения графиков функций в полярной системе координат без определения диапазона изменения независимой переменной x (в этом случае в качестве независимой переменной надо указывать именно x , а не \bar{w}).

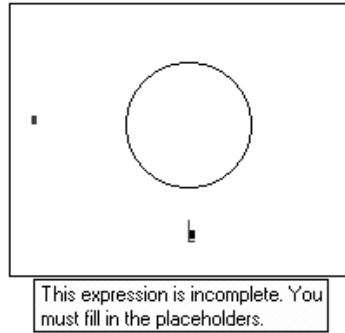


Рис. 4.6. Шаблон полярного графика



Рис. 4.7. Примеры построения графиков функций в полярной системе координат

При прямом построении графика достаточно просто заполнить место ввода функции. Саму функцию надо описать ее уравнениями, которые вписываются в соответствующие места ввода. Можно также задать построение графиков нескольких функций в одном шаблоне (рис. 4.8). Стоит щелкнуть мышью вне области графика, как последний будет построен.

Для преобразования координат двумерных графиков можно использовать вполне очевидные функции: $xu2rol$ (декартовы координаты преобразуются в полярные) и $rol2xu$ (полярные координаты преобразуются в декартовы).

4.3. Трехмерный график поверхности

Трехмерная графика системы Mathcad при правильном ее применении способна поразить воображение многих пользователей. Она дает прекрасное представле-



Рис. 4.8. Пример прямого построения графиков функций в полярной системе координат

ние о сложных поверхностях и объемных фигурах, расположенных в пространстве. Особенно впечатляет трехмерная графика с функциональной окраской поверхностей и фигур и с построением ряда пересекающихся объектов в пространстве.

4.3.1. Построение поверхностей по матрице аппликат их точек

До появления Mathcad 2000 при построении графика поверхности, представленной функцией $z(x, y)$ двух переменных, приходилось предварительно определять матрицу M аппликат (высот z) ее точек. Разумеется, этот способ возможен в любой версии Mathcad.

Поскольку элементы матрицы M — индексированные переменные с целочисленными индексами, то перед созданием матрицы требуется задать индексы в виде ранжированных переменных с целочисленными значениями, а затем уже из них формировать сетку значений x и y — координат для аппликат $z(x, y)$. Значения x и y при этом обычно должны быть вещественными числами, нередко как положительными, так и отрицательными.

Все это приводит к усложнению алгоритма подготовки данных для построения трехмерного графика поверхности, особенно большие трудности возникают у школьников и студентов. Они не всегда понимают, зачем все эти сложности и как задать область изменения x и y , чтобы была построена нужная часть поверхности.

После выполнения указанных выше определений командой Surface Plot (График поверхности) подменю Graph (График) меню Insert (Вставка) вводится шаблон графика, левый верхний угол которого помещается в место расположения курсора. Шаблон в свою очередь содержит единственное место ввода — темный прямоугольник у левого нижнего угла основного шаблона. В него надо занести имя матрицы аппликат поверхности. После этого надо установить указатель мыши в стороне от графического блока и щелкнуть левой кнопкой. На рис. 4.9 показан пример такого построения.

В данном случае построена поверхность в виде «проволочного каркаса» со всеми видимыми линиями.

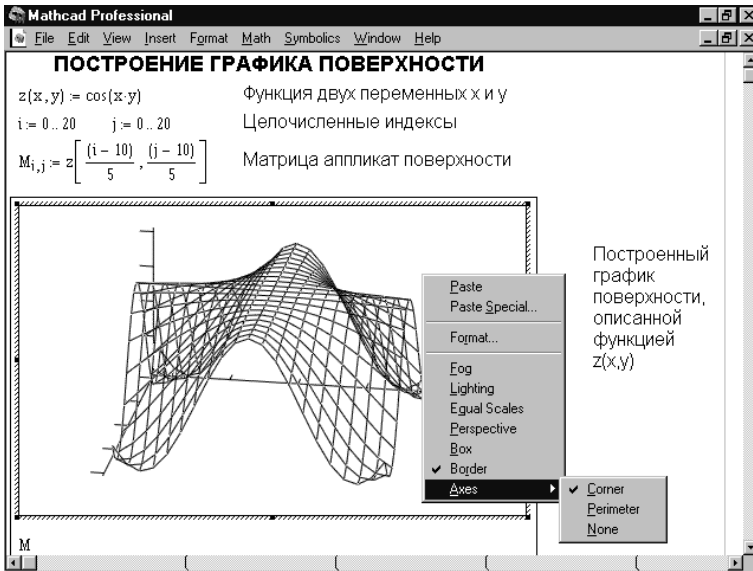


Рис. 4.9. Задание и построение поверхности без удаления невидимых линий

Наглядность представления поверхностей и трехмерных фигур зависит от множества факторов: масштаба построений, углов поворота фигуры относительно осей, применения алгоритма удаления невидимых линий или отказа от него, использования функциональной окраски и т. д. Для изменения этих параметров следует отформатировать график (см. главу 5). Для этого, в частности, можно использовать контекстное меню, также показанное на рисунке.

Фрагмент документа Mathcad, показанный на рис. 4.10, иллюстрирует применение алгоритма удаления невидимых линий и задание функциональной окраски поверхности для описанного выше графика.

Нетрудно заметить, что применение алгоритма удаления невидимых линий делает рисунок поверхности намного более наглядным. Дальнейшее повышение наглядности восприятия трехмерных графиков обеспечивается применением функциональной окраски. По существу она дает дополнительную информацию о третьем измерении, в нашем случае — чем ниже расположена поверхность фигуры, тем она темнее.

С помощью команд изменения формата можно получить множество разновидностей трехмерной графики. В частности, возможен вывод координатных осей, «параллелепипеда», обрамляющего фигуру, и иных деталей подобных графиков, например титульных надписей. Детали форматирования мы рассмотрим в главе 5.

Поскольку график строится на основе матрицы, содержащей только координаты высот фигуры, то истинные масштабы по осям X и Y неизвестны и на рисунках не проставляются. Возможно, впрочем, выводить порядковые номера элементов матриц в заданном направлении (по X и по Y). Необходимо следить за тем, как сформировать векторы X и Y , чтобы фигура выглядела естественно и была видна нужная часть фигуры в пространстве. Все это несколько затрудняет быстрое создание графиков трехмерных поверхностей нужного вида.

Для преобразования координат в Mathcad служат функции: $xyz2cyl$ (декартовы координаты преобразуются в цилиндрические), $cyl2xyz$ (цилиндрические координаты преобразуются в декартовы).

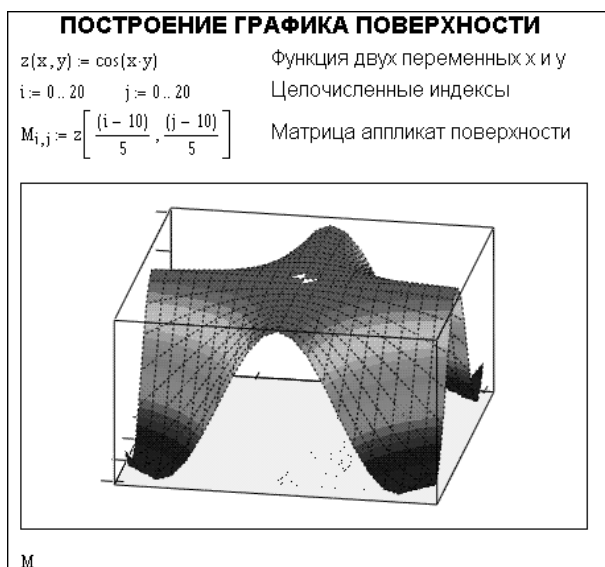


Рис. 4.10. Задание и построение поверхности с удалением невидимых линий и использованием функциональной окраски

наты преобразуются в декартовы), $xyz2sph$ (декартовы координаты преобразуются в сферические) и $sph2xyz$ (сферические координаты преобразуются в декартовы). Переход из одной системы координат в другие возможен и при форматировании графиков.

4.3.2. Построение параметрически заданных поверхностей

Большие возможности дает несколько иной способ задания поверхностей — в параметрическом виде. При этом приходится формировать три матрицы X , Y и Z и указывать их в шаблоне в виде (X, Y, Z) . Скобки необходимы, поскольку в противном случае Mathcad попытается построить три поверхности по данным матриц X , Y и Z .

На рис. 4.11 показаны построенные таким способом сферы — одна при параметрах форматирования, заданных по умолчанию, другая (см. главу 5) — после простого форматирования путем введения обрамляющего параллелепипеда, применения алгоритма удаления невидимых линий и использования функциональной окраски, зависящей от значений координаты x .

Подобный способ построения пространственных фигур открывает новые возможности для наглядной визуализации трехмерных объектов различной формы.

4.3.3. Построение трехмерных фигур с вырезом

Параметрическая форма задания трехмерных фигур открывает еще одну возможность — представление объемных фигур с вырезом. Такие фигуры отличаются повышенной наглядностью, ибо в вырезе видна внутренняя структура фигур. Все, что надо для такого построения, — ограничить диапазон изменения параметрических углов, сделав его меньше обычного значения 2π . Этот прием иллюстрирует рис. 4.12.

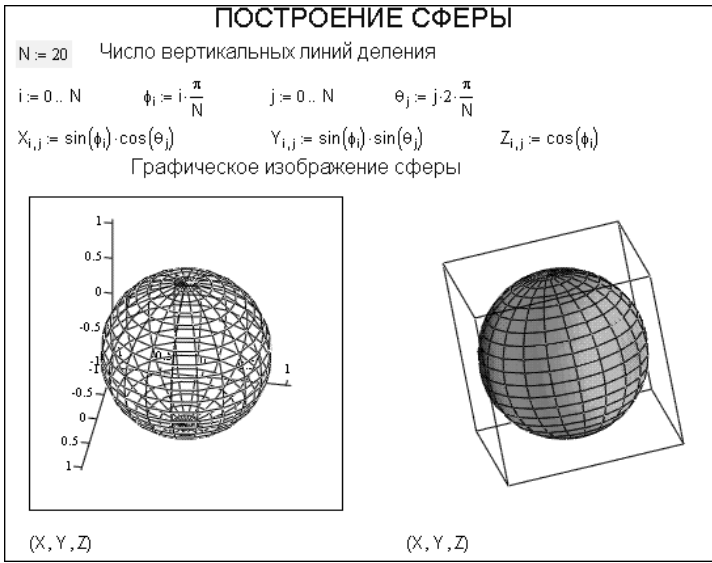


Рис. 4.11. Построение сферы, заданной параметрически

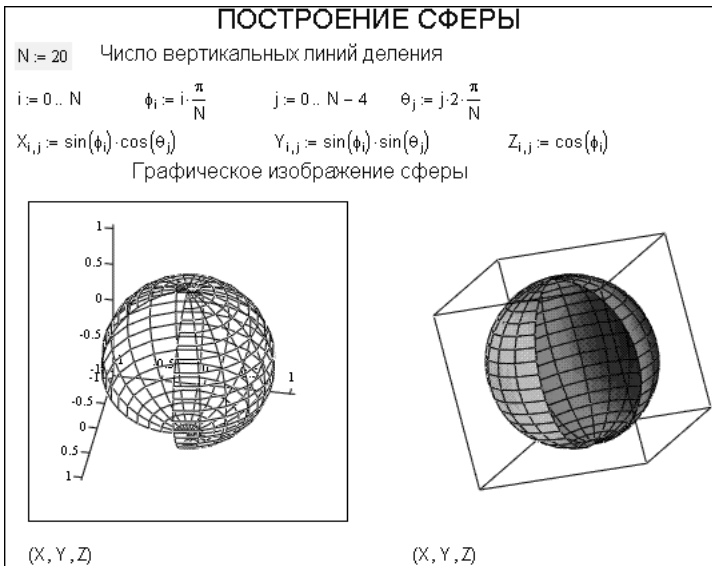


Рис. 4.12. Построение сферы с вырезом

Рекомендуется внимательно сравнить рис. 4.11 и 4.12. Вся разница (в аналитических выражениях) заключается в уменьшении диапазона изменения индекса j — на рис. 4.12 он меняется от 0 до $N-4$, а на рис. 4.11 — от 0 до N .

4.3.4. Построение трехмерных графиков без задания матрицы

Mathcad 2001i/11/12 обладает принципиально новой возможностью — допускается построение трехмерных графиков без задания матрицы аппликат поверхно-

стей. В результате построение графиков поверхностей выполняется столь же просто, как и построение двумерных графиков. Фрагмент документа Mathcad, показанный на рис. 4.13, иллюстрирует эту возможность.

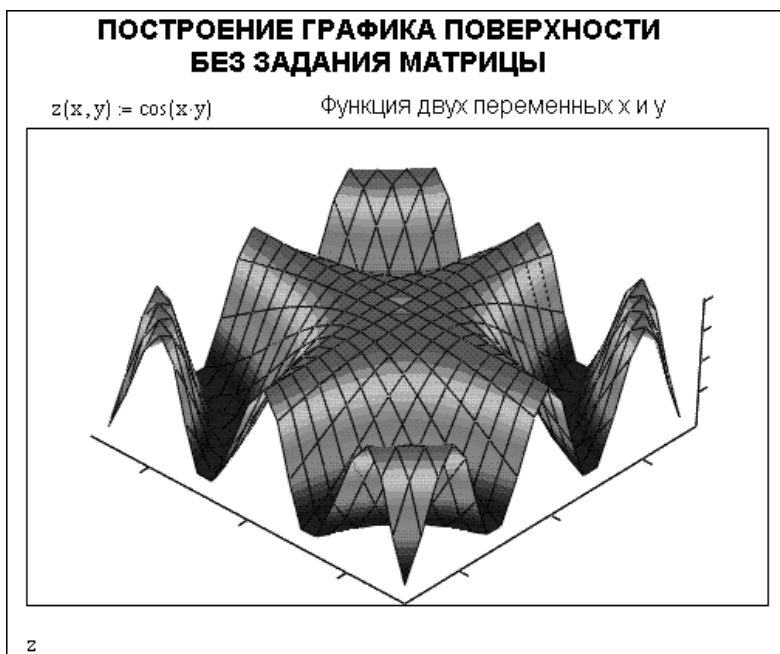


Рис. 4.13. Построение графика поверхности в системе Mathcad 2001 без задания матрицы

В данном случае для построения оказалось достаточным задать функцию двух переменных $z(x,y) := \cos(xy)$ и указать ее имя z в месте ввода шаблона графики. Единственным недостатком такого упрощенного метода построения поверхностей является неопределенность в масштабировании, поэтому для получения приемлемого вида графиков требуется форматирование (см. главу 5). Впрочем, в любом случае получение графика в достаточно эффектном виде всегда требует его форматирования.

4.3.5. Построение графика поверхности, заданной в векторной параметрической форме

Описанный выше новый метод быстрого построения поверхности может иметь множество вариантов. Один из них — задание поверхности в векторной параметрической форме. Пример такого построения иллюстрирует рис. 4.14. Строится фигура, напоминающая бублик (тор).

Обратите внимание на особую наглядность задания поверхности в такой форме с помощью единственной формулы и простоту построения графика. В данном случае для создания исходного графика (на рисунке слева) не требуется никаких промежуточных операций. Вид графика (на рисунке справа) можно улучшить путем форматирования и поворота графика мышью (см. главу 5).

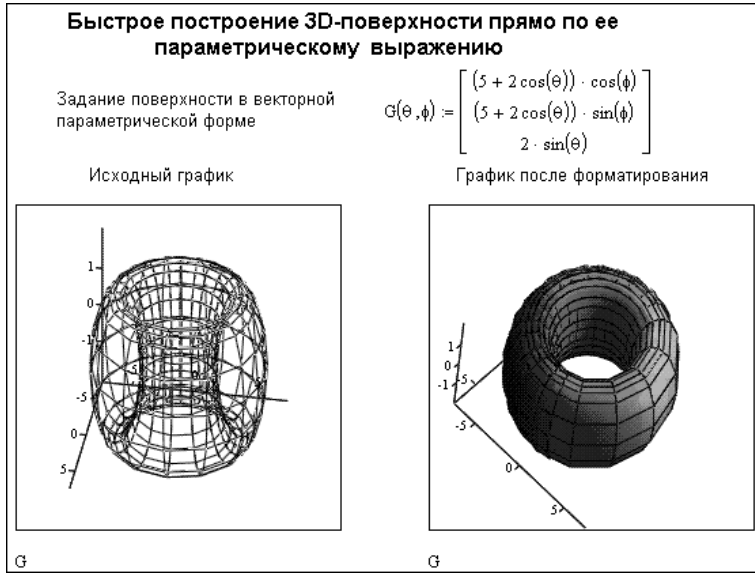


Рис. 4.14. Построение графика поверхности при задании ее в векторной параметрической форме

4.3.6. Применение новой функции CreateMesh

Mathcad 2001i/11/12 поддерживает новую графическую функцию для задания поверхностей:

`CreateMesh(F, s0, s1, t0, t1, sgrid, tgrid, fmap)`

Эта функция возвращает массив из трех матриц, представляющих координаты x , y и z для функции F , определенной в векторной параметрической форме в качестве функции двух параметров $sgrid$ и $tgrid$. Аргументы $s0$, $s1$, $t0$ и $t1$ определяют пределы изменения переменных $sgrid$ и $tgrid$. Аргумент $fmap$ — трехэлементный вектор значений, задающих число линий в сетке изображаемой функции. Создаваемый функцией `CreateMesh` массив можно использовать для ввода в шаблон трехмерной графики поверхности. В качестве примера рассмотрим построение поверхности, заданной функцией $\sin(uv)$.

1. Задайте функцию двух переменных следующим образом:

$$H(u, v) := \sin(uv)$$

2. Задайте пределы изменения переменных:

```
u0 := -2
u1 := 2
v0 := -2
v1 := 2
```

3. Используя функцию `CreateMesh`, создайте матрицу аппликат поверхности:

$$C := \text{CreateMesh}(H, u0, u1, v0, v1)$$

4. Выберите команду `Insert (Вставка) ▶ Graph (График) ▶ Surface Plot (График поверхности)`.

5. В единственное место ввода появившегося шаблона трехмерного графика введите имя матрицы аппликата поверхности, созданной на шаге 3.
6. Щелкните в документе вне области графика. Поверхность будет построена.
7. При необходимости отформатируйте график (подробно команды форматирования будут описаны в главе 5).

Построение поверхности с применением функции `CreateMesh` иллюстрирует фрагмент документа Mathcad, показанный рис. 4.15.

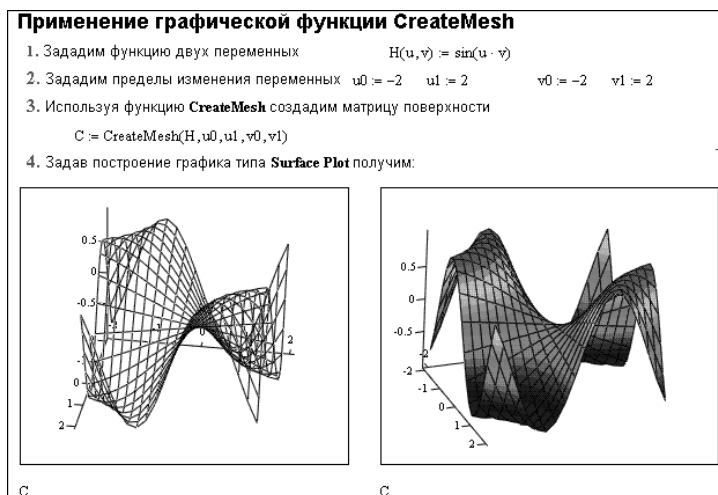


Рис. 4.15. Пример построения графика поверхности с применением функции `CreateMesh`

Рисунок иллюстрирует технологию применения этой функции. Построение слева дано при форматировании по умолчанию, а справа — после ввода функциональной окраски и поворота фигуры мышью (см. главу 5).

4.3.7. Построение объемной фигуры, образованной вращением кривой

Еще один полезный пример применения функции `CreateMesh` — построение объемной фигуры, которая получается вращением кривой, заданной функцией $f(x)$, вокруг оси X или Y (рис. 4.16).

Слева на рисунке показана исходная кривая, заданная функцией $f(x)$, а справа — объемная фигура, построенная с применением форматирования для повышения наглядности графика (подробно команды форматирования будут описаны в главе 5).

4.3.8. Построение полиэдров

В Mathcad 2000 появилась функция для построения объемных фигур полиэдров — `Polyhedron(name)`, где `name` — имя фигуры. Имя ряда фигур можно задавать явно, например `Polyhedron("cube")`, или в виде номера (см. ниже). Рис. 4.17 иллюстрирует применение данной функции для построения октаэдра с его заданием в явном виде. В данном случае переменной `name` присваивается значение `"octahedron"`:

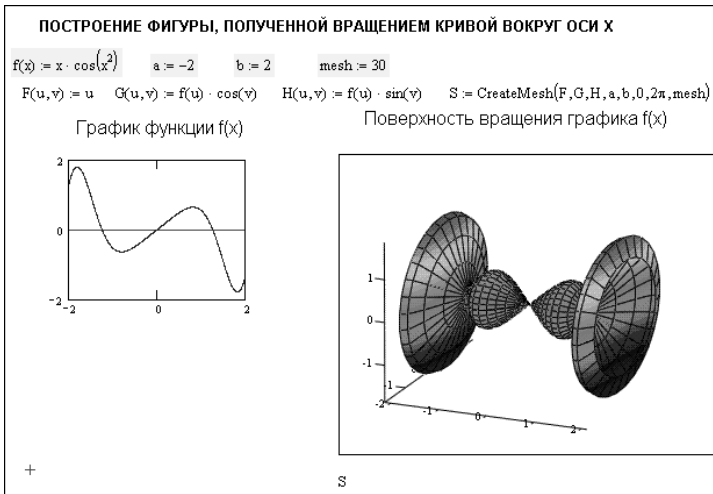


Рис. 4.16. Построение объемной фигуры, полученной вращением кривой

```
name := "octahedron"
```

Далее в место ввода шаблона трехмерной поверхности вводится функция `Polyhedron(name)`.

Построенная фигура может форматироваться, как и другие графики трехмерных поверхностей (подробно команды форматирования будут описаны в главе 5), а также поворачиваться и масштабироваться с помощью мыши.

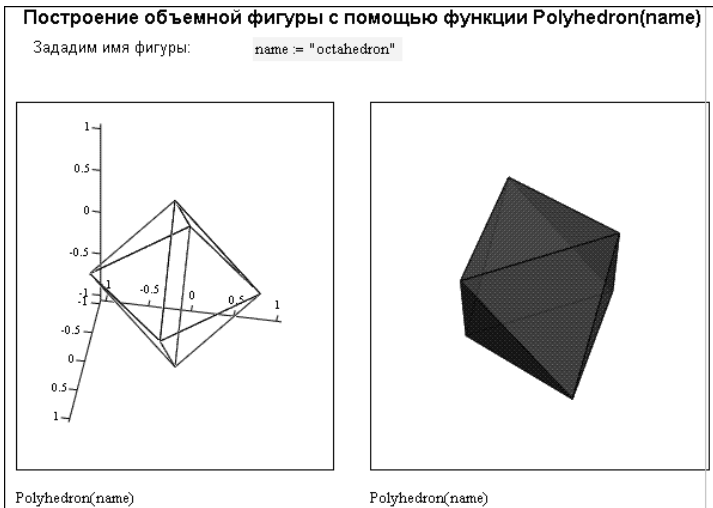


Рис. 4.17. Построение объемной фигуры — октаэдра

Как уже упоминалось, помимо явного задания имени фигуры можно задавать ее в виде параметра "#N", где N — номер фигуры. Для получения имени фигуры по ее номеру можно использовать функцию `PolyLookup`. Например, как показано на рис. 4.18, вызов функции `PolyLookup("#12")` позволил получить следующие данные:

- "cuboctahedron" — первое имя фигуры;
- "rhombic dodecahedron" — второе имя фигуры;
- "2|3 4" — описатель фигуры.

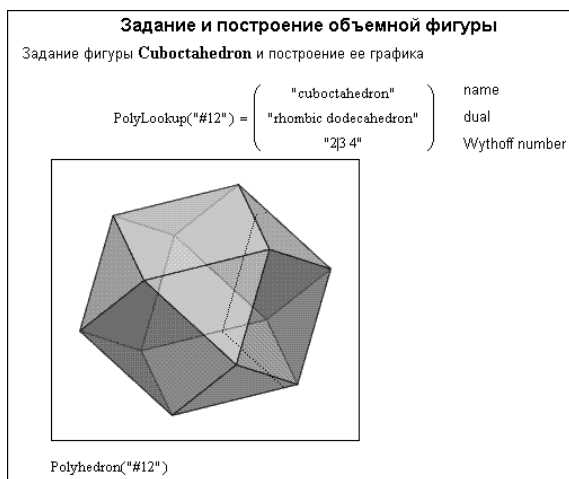


Рис. 4.18. Получение имени фигуры по ее номеру

Большое число примеров применения функции Polyhedron можно найти в справочной системе Mathcad — раздел Polyhedron (Полиэдры) таблиц Reference Tables (Таблицы ссылок). На рис. 4.19 показана страница этого раздела, посвященная регулярным и квазирегулярным полиэдрам. Всего в таблицах имеется около 60 полиэдров.

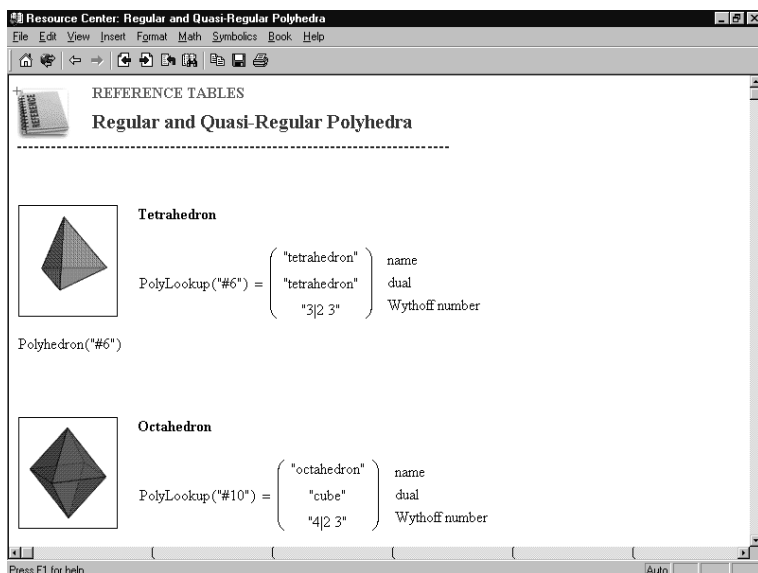


Рис. 4.19. Страница справочных таблиц, посвященная регулярным и квазирегулярным полиэдрам

Наличие превосходных средств построения большого числа полиэдров позволяет эффектно использовать Mathcad в преподавании аналитической геометрии в пространстве и стереометрии.

4.3.9. Построение на одном графике нескольких трехмерных объектов

Новые версии Mathcad предоставляют возможность построения нескольких пересекающихся или непересекающихся поверхностей на одном графике. Для этого достаточно перечислить функции задания поверхностей в местах ввода шаблона графика через запятую.

На рис. 4.20 показано построение октаэдра (см. выше подраздел «Построение полиэдров»), как бы находящегося внутри поверхности, заданной функцией $z(x, y) := -\cos(xy)$.

Место ввода шаблона трехмерного графика должно быть заполнено следующим образом:

```
z, Polyhedron("Octahedron")
```



Рис. 4.20. Построение октаэдра внутри поверхности

Рис. 4.21 иллюстрирует построение на одном рисунке трех пересекающихся поверхностей. Рисунок отформатирован с применением функциональной окраски. В данном случае поверхности заданы следующими функциями z_1 , z_2 и z_3 . Их определение дано над рисунком. Место ввода шаблона трехмерного графика должно быть заполнено следующим образом:

```
z1, z2, z3
```

Возможно построение на одном графике поверхностей с разными способами описания. Пример такого построения дан на рис. 4.22. Единственное место ввода шаблона трехмерного графика должно быть заполнено следующим образом: $1.25G, Z$. Множитель 1.25 увеличивает размеры фигуры G .

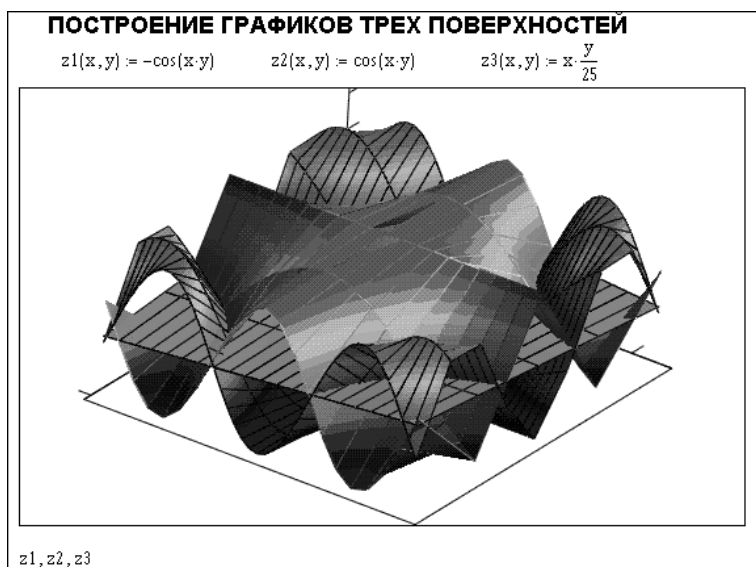


Рис. 4.21. Построение на одном рисунке трех пересекающихся поверхностей



Рис. 4.22. Построение на одном графике трехмерных поверхностей с разными способами задания

Приведенными примерами возможности трехмерной графики далеко не исчерпываются. Ниже рассматривается ряд других типов таких графиков. Кроме того, форматированием можно обеспечить дополнительные свойства графиков, например, их представление в перспективе, в цилиндрической и сферической системах координат и др. Все эти возможности мы рассмотрим в дальнейшем при описании форматирования графиков (см. главу 5).

4.4. Контурный трехмерный график

Еще один широко распространенный тип графиков для представления поверхностей — график из линий равного уровня. Такие графики широко применяются, например, в картографии. Команда Contour Plot (Контурный график) подменю Graph (График) меню Insert (Вставка) служит для ввода шаблона таких графиков.

4.4.1. Стандартный способ построения контурных графиков

Рис. 4.23 иллюстрирует стандартный способ (с заданием матрицы аппликат) построения контурного графика в виде линий равного уровня в системе Mathcad 2001.

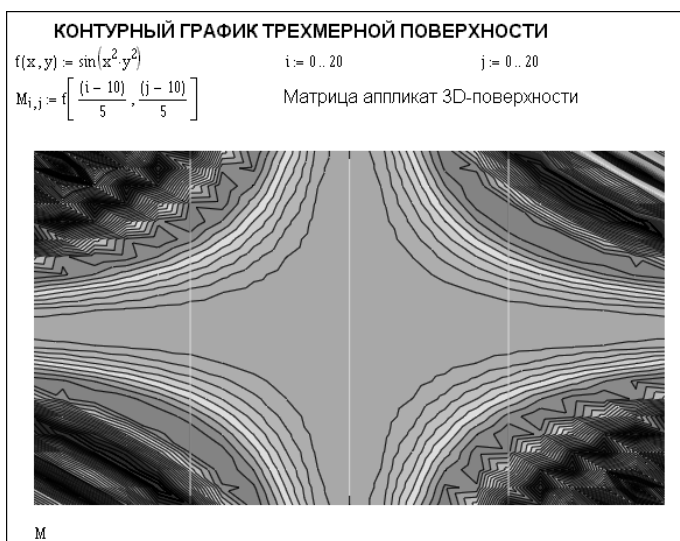


Рис. 4.23. Контурный график поверхности, построенный по ее матрице аппликат

Обычно функциональная окраска такого графика, вводимая из окна форматирования, существенно улучшает вид контурных графиков. Но при необходимости ее можно убрать и оставить только контурные линии.

4.4.2. Построение контурных графиков без явного задания матрицы

В Mathcad 2001i/11/12 есть более удобный способ построения контурных графиков — без задания в явном виде матрицы узловых точек, используемых для построения кривых. В этом случае (рис. 4.24) достаточно в месте ввода указать имя функции.

График отформатирован с оцифровкой линий равного уровня (подробно команды форматирования будут описаны в главе 5). В место ввода шаблона контурного графика должно быть введено имя функции (f). Представление графика с оцифровкой линий удобно для количественных оценок. Однако в большинстве случаев указание числовых значений уровней загромождает график.

Иногда контурные графики получаются даже более информативными, чем просто трехмерные поверхности. У последних одни части поверхности нередко за-

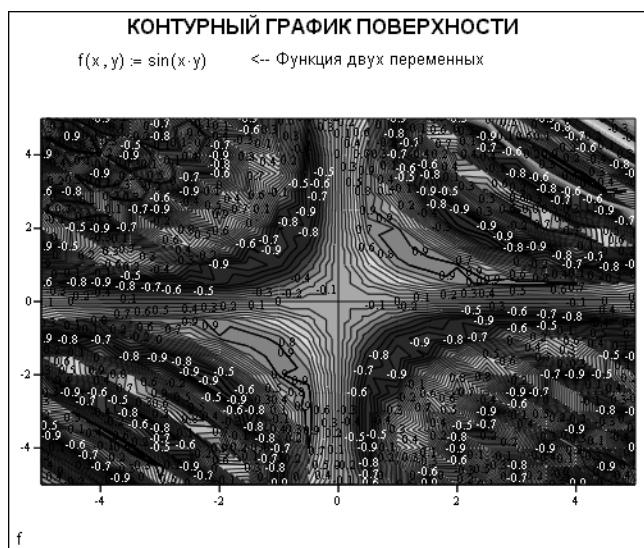


Рис. 4.24. Контурный график поверхности, построенный без задания матрицы

крывают другие. Например, пик на переднем плане может закрыть меньшие пики или впадины на заднем плане. У контурных графиков такого эффекта нет, и на них легко обнаруживаются все пики и впадины (правда, при достаточно большом числе линий равного уровня и малом расстоянии между ними).

4.5. Точечный трехмерный график

4.5.1. Определение точечного графика

Нередко трехмерные поверхности представляют в виде находящихся в пространстве точек, кружочков или иных фигур. Каждая из этих фигур несет информацию о геометрическом положении ее центра в трехмерном пространстве. Такой график создается командой 3D Scatter Plot (Точечный график) подменю Graph (График) меню Insert (Вставка).

Размеры точек, их вид и окраску можно изменять с помощью команд форматирования трехмерного графика (см. главу 5). Обычно неплохо выглядят графики с малыми фигурами, расположенными внутри параллелепипеда. Однако в целом наглядность таких графиков не очень высока. Их имеет смысл использовать не для показа трехмерных поверхностей, а лишь для размещения на поверхностях небольшого числа объектов. Объекты могут быть представлены не только в виде точек, но и в виде иных фигур — крестиков, окружностей, квадратов и др.

4.5.2. Построение точечного графика с заданием матрицы аппликат точек

Сначала рассмотрим пример построения точечного графика при явном задании аппликат точек с помощью матрицы M . Благодаря явному заданию матрицы аппликат точек этот способ пригоден для любых версий Mathcad. В место ввода шаблона точечного графика должно быть введено имя матрицы аппликат поверхности (M). Вид графика показан на рис. 4.25.

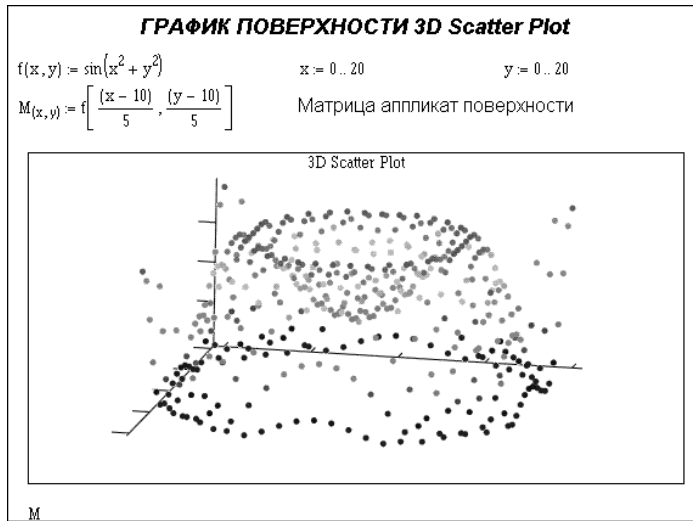


Рис. 4.25. График трехмерной поверхности в виде разбросанных в пространстве точек

Графики этого типа имеют ограниченное применение.

4.5.3. Построение точечного графика с заданием только функции поверхности

Mathcad позволяет строить точечные графики без явного задания матрицы аппликат отображаемых точек. Достаточно указать в шаблоне графика имя функции двух переменных, задающей аппликаты точек (рис. 4.26). В место ввода шаблона точечного графика должно быть введено имя функции f .

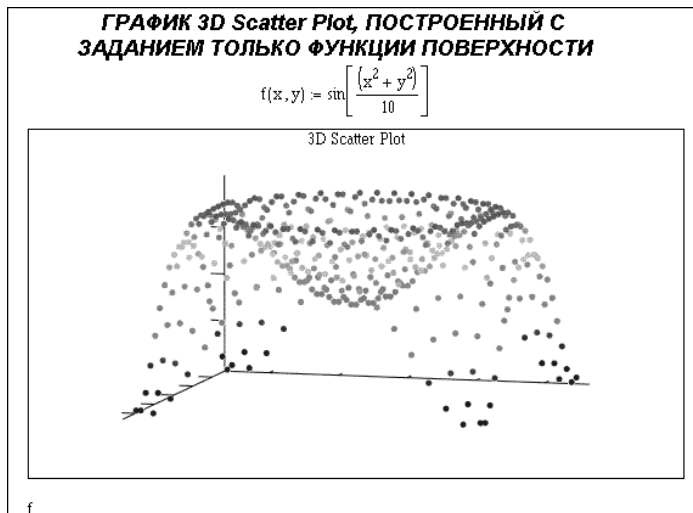


Рис. 4.26. Построение точечного графика с заданием только функции поверхности

Этот способ имеет то неудобство, что масштабы осей, установленные по умолчанию, не всегда приемлемы, поэтому требуется либо нормировать функцию, либо переформатировать график, изменив масштабы осей. Разумеется, формирование графика позволяет получить и иные эффекты, например, можно придать точкам функциональную окраску, вывести титульную надпись и др. (подробно команды форматирования будут описаны в главе 5).

4.5.4. Применение функции CreateSpace

Новые версии Mathcad имеют еще одну новую графическую функцию:

```
CreateSpace(F, t0, t1, tgrid, fmap)
```

Эта функция отличается от функции CreateMesh (см. подраздел «Применение новой функции CreateMesh» в разделе «Трехмерный график поверхности») только тем, что заданная в векторном виде функция F представляет собой набор функций одной переменной t , причем параметры $t0$ и $t1$ определяют пределы ее изменения, а переменная $tgrid$ — число линий сетки. Необязательный параметр $fmap$ в виде списка трех переменных позволяет задать тип координатной системы, например прямоугольная система (cartesian) задается по умолчанию следующим образом:

```
fmap_default(e1, e2, e3) := c(e1, e2, e3)
```

Вместо первого параметра функции CreateMesh (вектора F) можно задать список из трех функций $f1$, $f2$ и $f3$ переменной t .

С помощью функции CreateSpace удобно строить точечные графики в виде пространственных спиралей и иных подобных геометрических образов. Для примера рассмотрим алгоритм использования функции CreateSpace для построения точечного графика пространственной спирали.

1. Введите в векторной форме функцию $H(t)$ одной переменной t , как показано ниже:

$$H(t) := \begin{pmatrix} t \cdot \sin(t) \\ t \cdot \cos(t) \\ t \end{pmatrix}$$

2. Задайте пределы изменения переменных и число точек графика:

```
t0 := 0
t1 := 16
tgrid := 160
```

3. С помощью функции CreateSpace создайте матрицу аппликат:

```
C := CreateSpace(H, t0, t1, tgrid)
```

4. Выберите команду Insert (Вставка) ▶ Graph (График) ▶ 3D Scatter Plot (Точечный график).
5. В единственное место ввода появившегося шаблона точечного графика введите имя матрицы аппликат, созданной на шаге 3.
6. Щелкните в документе вне области графика. График будет построен.
7. При необходимости отформатируйте график (подробно команды форматирования будут описаны в главе 5).

График, показанный на рис. 4.27 слева, получен без форматирования, а справа — после форматирования.

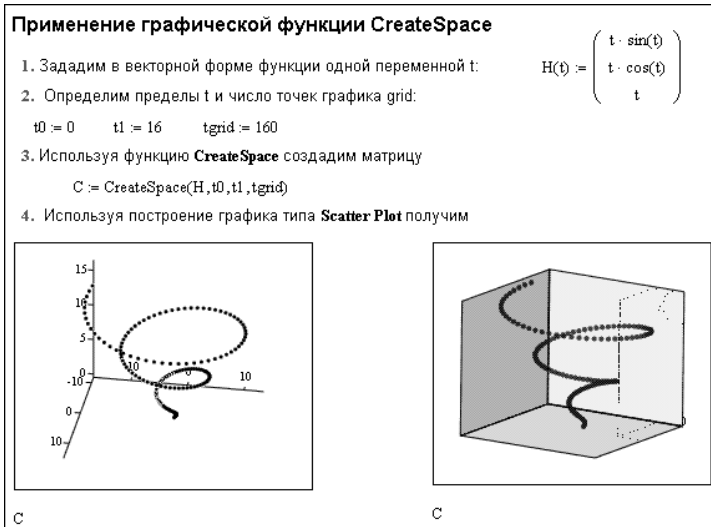


Рис. 4.27. Построение точечного графика пространственной спирали

4.6. Трехмерная гистограмма

Весьма распространенной формой представления поверхностей является ряд трехмерных столбиков, высота которых определяется значением координаты $z(x, y)$. Для этого используется команда 3D Bar Plot (Трехмерная гистограмма) подменю Graph (График) меню Insert (Вставка).

4.6.1. Обычное построение гистограмм

Рис. 4.28 иллюстрирует стандартную технологию построения таких графиков — с заданием матрицы аппликат поверхности, представляемой столбиками. В место ввода шаблона трехмерной гистограммы должно быть введено имя матрицы аппликат поверхности (M).

Подобные графики широко применяются при представлении сложных статистических данных. Обычно их построение считается «высшим пилотажем», но Mathcad превращает его в обыденную операцию. При этом, как видно на рисунке, представление столбиками даже сложных трехмерных поверхностей является достаточно наглядным. В отношении трехмерной гистограммы возможны все операции форматирования, включая поворот графиков мышью (подробно команды форматирования будут описаны в главе 5).

4.6.2. Построение трехмерных гистограмм с заданием только функции поверхности

Mathcad 2001i/11/12 дает возможность упростить построение трехмерных гистограмм за счет указания в месте ввода графика только имени функции. Пример такого построения дан на рис. 4.29.

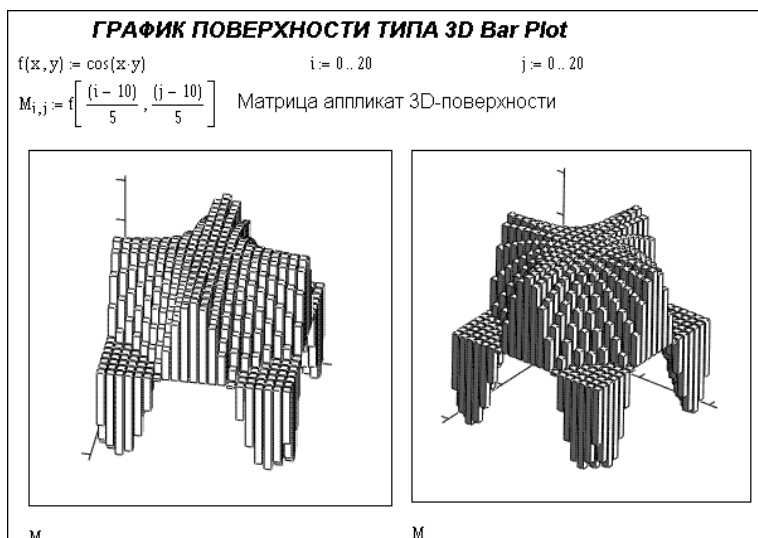


Рис. 4.28. Представление поверхности трехмерными столбиками

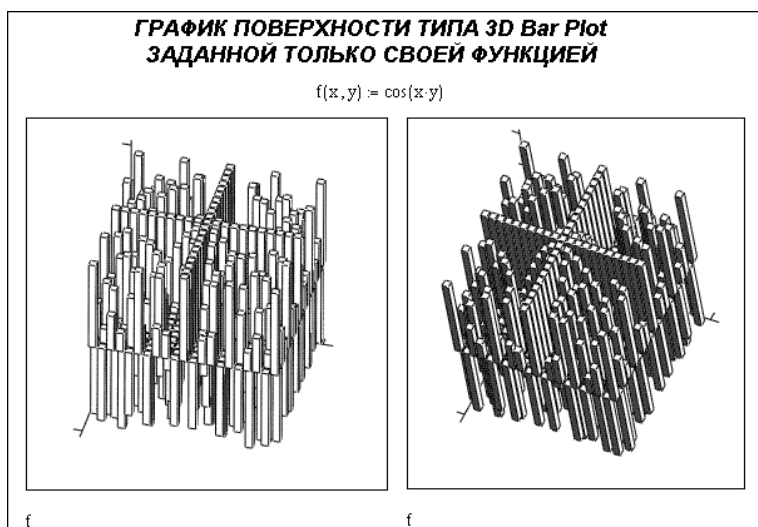


Рис. 4.29. Пример трехмерной гистограммы с заданием только функции поверхности

В отношении трехмерной гистограммы возможны все операции форматирования, включая поворот графиков мышью (подробно команды форматирования будут описаны в главе 5).

4.7. Трехмерный график в векторном представлении

Еще один вид представления поверхности — векторное представление. Оно задается набором коротких стрелочек — векторов. Стрелка обращена острием в сто-

рону нарастания высоты поверхности, а плотность расположения стрелок зависит от скорости этого нарастания. Для построения используется команда Vector Field Plot (Векторное поле) подменю Graph (График) меню Insert (Вставка).

4.7.1. Обычное построение графиков векторного поля

На рис. 4.30 показан пример графика векторного поля. В место ввода шаблона векторного поля должно быть введено имя матрицы аппликат поверхности (M).

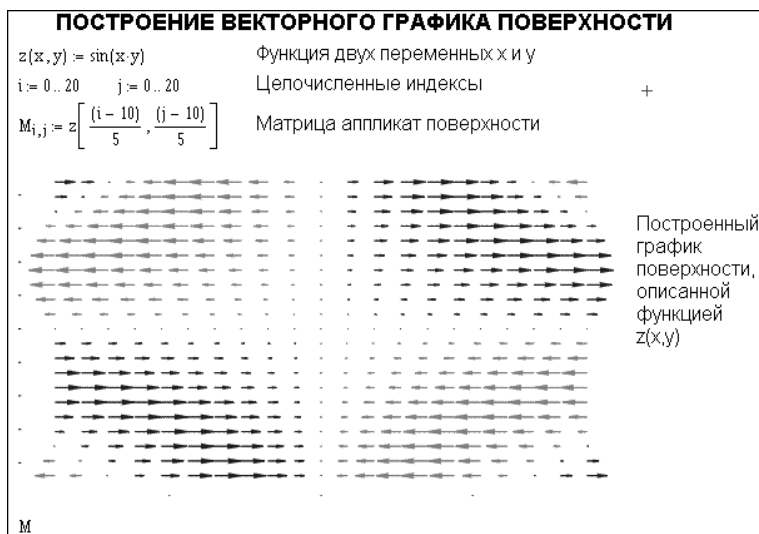


Рис. 4.30. Представление поверхности векторами

Эти графики применяются редко из-за трудности построения множества стрелок, для каждой из которых надо рассчитывать градиент поля. Но Mathcad обеспечивает возможность широкого применения графиков этого вида. Они особенно удобны для представления электромагнитных, тепловых, гравитационных и иных полей.

4.7.2. Построение графика векторного поля, заданного в параметрической форме

Еще один способ построения графиков векторных полей заключается в задании поверхности в параметрической форме. Этот случай представлен на рис. 4.31. В данном случае в место ввода шаблона графика векторного поля вводятся в скобках имена матриц M и N, а исходная формула для построения выглядит следующим образом:

$$V(x, y) := \left[\frac{y^2}{(1-x^2) \cdot \sqrt{y}} \right]$$

Место ввода шаблона векторного поля должно быть заполнено следующим образом: (M,N).

Ряд примеров на построение графиков различного типа можно найти в быстрых «шпаргалках» QuickSheets.

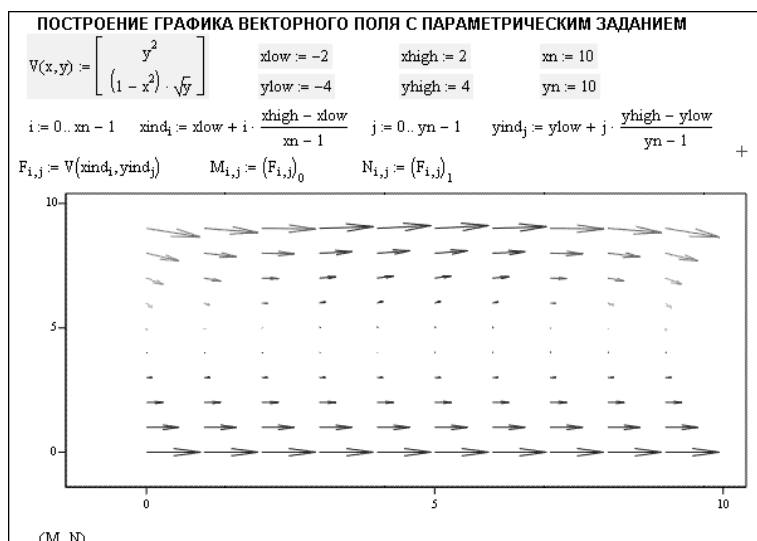


Рис. 4.31. График векторного поля поверхности, заданной в параметрическом виде

4.8. Специальные приемы построения трехмерных графиков

4.8.1. Построение трехмерных графиков мастером

Форматирование трехмерных графиков в Mathcad — довольно сложный процесс, поскольку число параметров форматирования достигает многих десятков (см. главу 5). Чтобы упростить создание трехмерных графиков, начиная с Mathcad 2000 используется специальный мастер, разбивающий процедуру построения и форматирования графика на несколько вполне очевидных этапов.

Для создания трехмерного графика с помощью мастера вначале надо ввести функцию или матрицу поверхности, например так:

$$f(x,y) := \sin(0.2xy)$$

Далее требуется выбрать команду Insert (Вставка) ▶ Graph (График) ▶ 3D Plot Wizard (Мастер трехмерной графики). В окне документа появится первое окно мастера — Plot Type (Тип графика), показанное на рис. 4.32.

В этом окне с помощью переключателей требуется выбрать необходимый тип графика. В нашем случае установите переключатель Surface Plot (График поверхности) и щелкните на кнопке Next (Далее), чтобы перейти к следующему окну мастера — Appearance (Вид), показанному на рис. 4.33. В нем установите переключатель Fill Surface and Draw Lines (Окрашенная поверхность с линиями).

Щелкнув на кнопке Next (Далее), можно вывести последнее окно мастера — Coloring (Задание цветовой гаммы), показанное на рис. 4.34. Здесь установите переключатель Color using lighting (Цвет определяется источником света). Вместо кнопки Next (Далее) в этом окне появляется кнопка Finish (Готово), что указывает на завершение операций с мастером. Щелкните на ней, если считаете правильными введенные параметры. Кнопка Back (Назад) позволяет вернуться к предыдущему

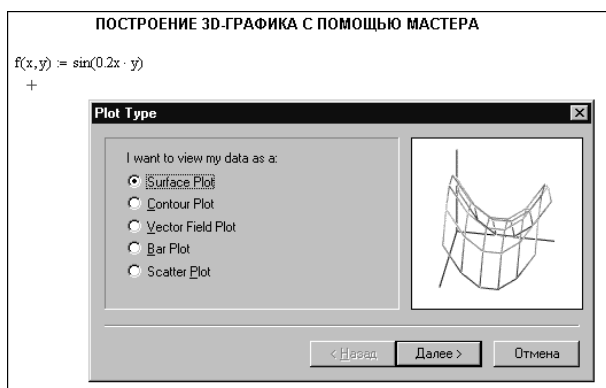


Рис. 4.32. Окно Plot Type мастера создания трехмерных графиков

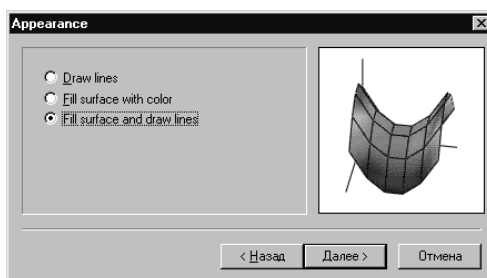


Рис. 4.33. Окно Appearance мастера создания трехмерных графиков

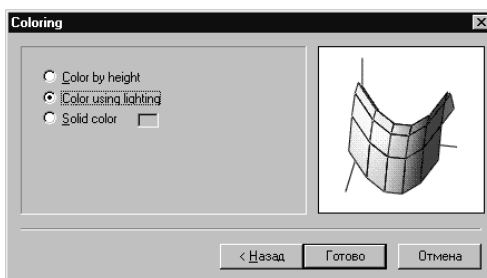


Рис. 4.34. Завершающее окно мастера — Coloring

му окну, если вы решите изменить введенные ранее параметры. А если вы вообще передумали строить график, щелкните на кнопке Cancel (Отмена).

При работе с мастером все вводимые параметры наглядно отображаются в области предварительного просмотра. Однако это не ваш график. Для демонстрации параметров форматирования используется не относящийся к делу рисунок параболической поверхности. По завершении работы с мастером на месте курсора ввода (красный крестик) появится шаблон трехмерного графика, но пока без самого графика. Для его построения надо в месте ввода шаблона графика указать имя той матрицы или функции, для которой должен быть построен график (в нашем случае — это f). Затем надо щелкнуть левой кнопкой мыши вне области графика — график будет тут же построен, что и иллюстрирует рис. 4.35.

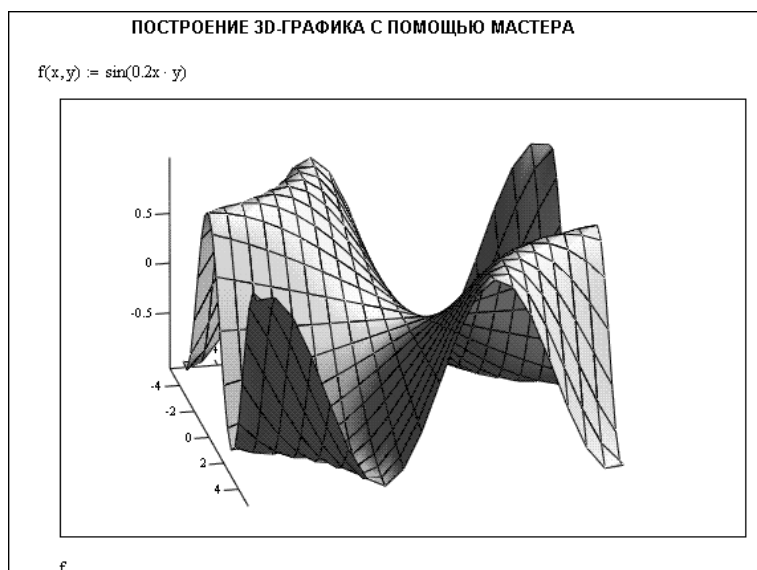


Рис. 4.35. Готовый график, построенный с помощью мастера

Разумеется, построенный график можно подвергнуть добавочному форматированию всеми возможными способами. К примеру, график, показанный на рисунке, увеличен в размере.

Высокая наглядность процесса создания трехмерных графиков с помощью мастера делает его применение предпочтительным, особенно для пользователей, впервые приступающих к освоению премудростей системы Mathcad. Однако следует отметить, что мастер реализует не все возможности форматирования. Например, он не позволяет задать тип координатных осей, установить обрамление графика параллелепипедом, окрасить плоскости и т. д. Более полно форматирование графиков будет рассмотрено в главе 5.

4.8.2. Трехмерный «лоскутный» график

Хотя в подменю Graph (График) меню Insert (Вставка) нет соответствующей команды, существует возможность построения еще одного типа графика — это так называемый «лоскутный» график. Для его построения нужно построить трехмерный график любого типа, а затем дважды щелкнуть мышью в области графика. В открывшемся окне форматирования перейдите на вкладку General (Общие), установите переключатель Patch Plot («Лоскутный» график) и щелкните на кнопке ОК. График будет тут же перестроен, а окно форматирования закрыто. Например, на рис. 4.36 показан «лоскутный» график, построенный на основе графика поверхности, созданного в разделе «Трехмерный график поверхности» (см. рис. 4.9).

СОВЕТ *Переключатели в нижней части вкладки General окна форматирования можно использовать не только для построения «лоскутного» графика, но и в более общем плане — для оперативной смены типа трехмерного графика.*

Как нетрудно заметить, «лоскутный» график представляет собой поверхность в виде прямоугольников или иных простых фигур.

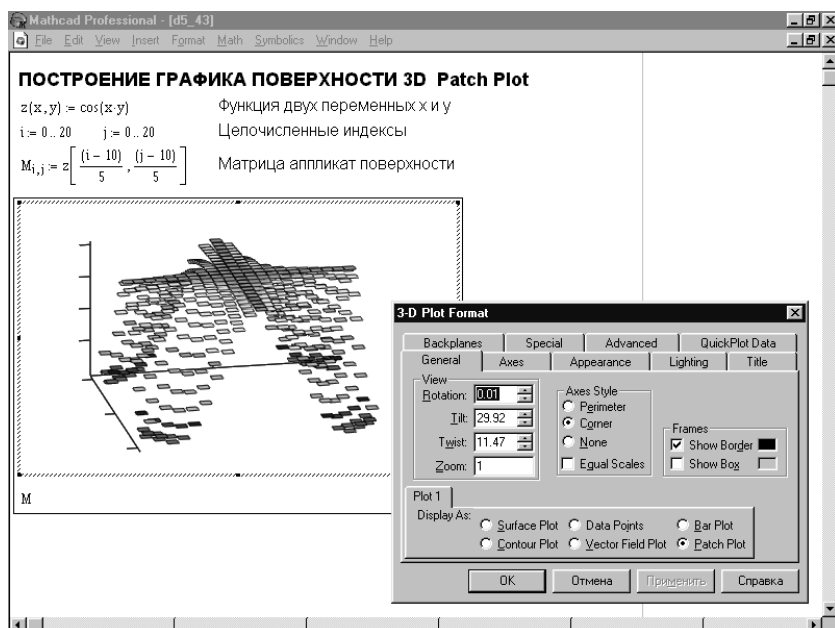


Рис. 4.36. Пример построения и форматирования графика типа Patch Plot

Подробно окно форматирования трехмерных графиков будет рассмотрено в разделе «Форматирование трехмерных графиков» главы 5.

4.8.3. Два пересекающихся в пространстве тора

Рис. 4.37 иллюстрирует построение двух пересекающихся в пространстве торов. Здесь интересно то, что математически задается один тор с параметрическим заданием построения, но в месте ввода данные для построения тора указываются дважды. В итоге получается весьма реалистичное изображение двух пересекающихся торов.

Еще одна примечательная деталь показанного на рисунке графика — форматирование с использованием световых эффектов, имитирующих блики света при освещении торов от внешнего источника света (подробно команды форматирования будут описаны в главе 5). К сожалению, на черно-белых рисунках книги не видна игра цветов, но блики света отчетливо заметны.

4.8.4. Представление функций двух переменных графиками векторного поля

Иногда желательно представление функций двух переменных графиками векторного поля, составленными из стрелок. Есть два распространенных вида таких графиков: в виде градиентного поля и поля Гамильтона. Их реализация в виде программных модулей представлена на рис. 4.38.

Читателю, желающему разобраться с представленными программными модулями, надо обратиться к главе 10. На рис. 4.39 показаны графики, построенные с помощью модулей и тестовых функций, представленных на рис. 4.38.

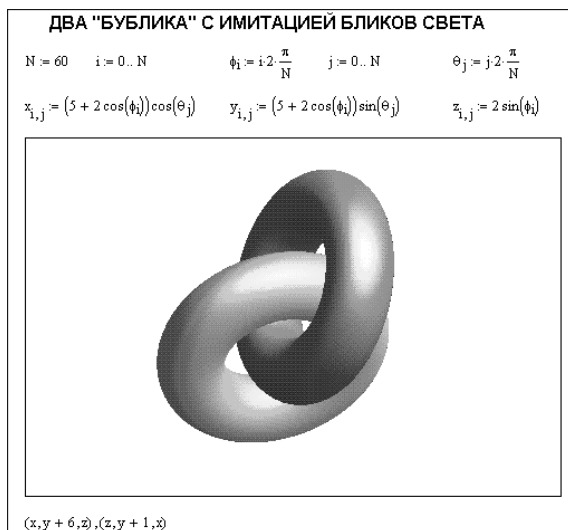


Рис. 4.37. Построение двух пересекающихся в пространстве торов

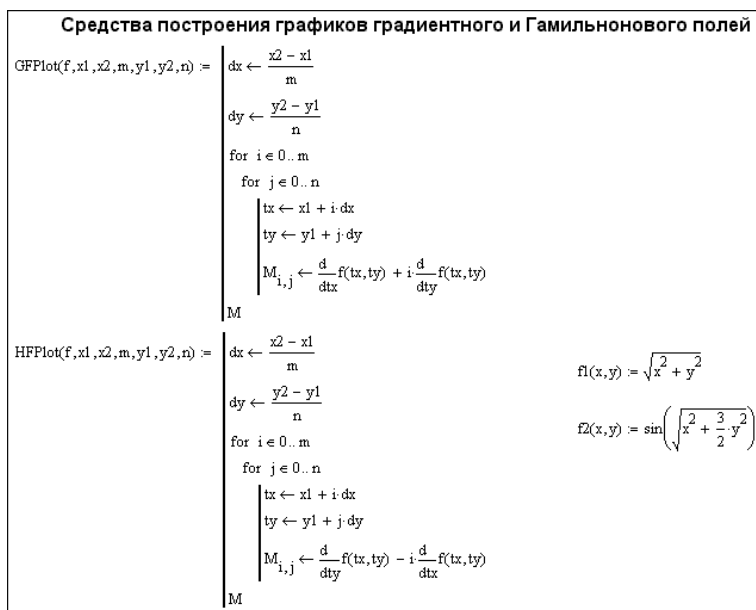


Рис. 4.38. Программные модели для построения графиков функций двух переменных в виде градиентного поля и поля Гамильтона

4.9. Техника анимации (оживления) графиков

4.9.1. Принципы анимации графиков

Анимация (или «оживление») графиков — возможность, присущая Mathcad. Принцип анимации достаточно прост. В системе имеется встроенная переменная

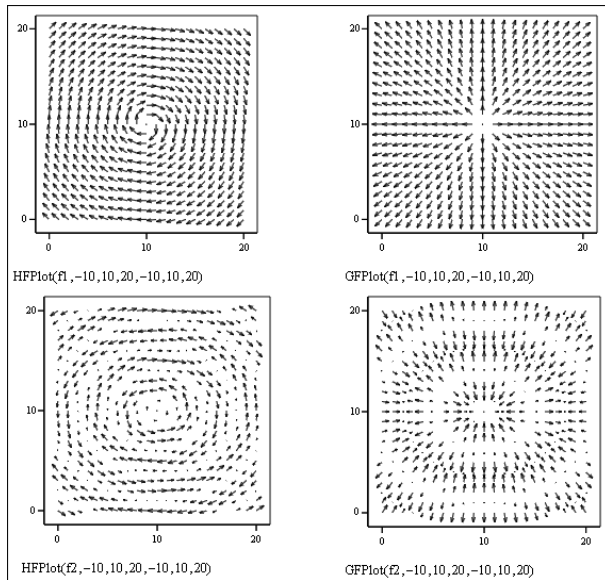


Рис. 4.39. Примеры построения графиков функций двух переменных в виде градиентного поля и поля Гамильтона

FRAME, принимающая целочисленные значения (по умолчанию она меняется от 0 до 9 с шагом 1). Любая функция, график которой планируется наблюдать в развитии, должна быть функцией этой переменной, идентифицирующей, по существу, просто номер текущего кадра. Диапазон изменения переменной FRAME задается в диалоговом окне команды Animate (Анимация) меню View (Вид). Не следует пытаться присваивать переменной FRAME значения иным путем.

При создании анимационных рисунков все кадры строятся с одинаковыми координатами углов и, следовательно, с одинаковыми размерами и одинаковым положением на экране. Их вывод один за другим с заданной скоростью (по умолчанию 10 кадров в секунду) и создает «живую» картинку. Просмотр анимации (созданной последовательности кадров) осуществляется с помощью специального проигрывателя.

4.9.2. Подготовка к анимации

Для построения анимационного графика вначале системная переменная задается FRAME как ранжированная, затем задается функция, у которой переменная FRAME должна определять вид каждого кадра анимированного графика. После этого в специальном диалоговом окне задают три основных параметра: начальное значение переменной FRAME, ее конечное значение и частоту смены кадров. Для примера давайте анимируем контурный график некоторой сложной функции. Процедура подготовки к анимации графика выглядит следующим образом.

1. Задайте системную переменную FRAME как ранжированную, например так:

$$g := 2 \cdot \text{FRAME}$$

2. Задайте функцию, одним из параметров которой (определяющим вид каждого кадра анимации) будет заданная на первом шаге переменная. Пусть это будет следующая функция:

$$A(u, v) := \sin\left(\frac{u}{5}\right)^2 \cdot \cos\left(\frac{u+v}{8}\right)^2 + \exp\left[-\frac{(u-5-g)^2 + (v-10-g)^2}{50}\right]$$

3. Выбором команды Contour Plot (Контурный график) в подменю Graph (График) меню Insert (Вставка) выведите шаблон контурного графика (подробно процедура построения контурного графика описана в разделе «Контурный трехмерный график» этой главы).
4. В единственное место ввода шаблона контурного графика введите имя функции, заданной на шаге 2 (это имя A).
5. Выбором команды Animate (Анимация) в меню View (Вид) выведите диалоговое окно для задания параметров анимации.
6. Задайте следующие параметры анимации (как показано на рис. 4.40):
 - начальное значение переменной FRAME — 0;
 - конечное значение переменной FRAME — 9;
 - частота смены кадров — 10 кадров в секунду.
7. Выделите мышью нужный фрагмент изображения. Можно выделить любую часть графика и даже расположенные около него объекты, например формулы. В этом случае они тоже будут отображаться при воспроизведении анимации.

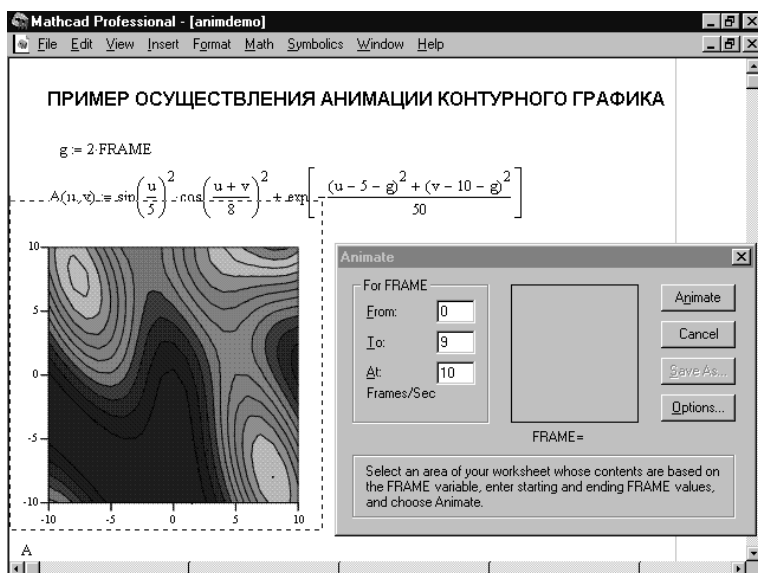


Рис. 4.40. Подготовка к построению анимированного графика

На этом подготовительные операции к созданию анимации заканчиваются.

4.9.3. Создание кадров изображения

Щелчок на кнопке Animate (Анимировать) окна параметров анимации приводит к созданию последовательности анимационных кадров. При этом эти кадры будут видны в области просмотра окна, а под этой областью можно наблюдать изменение переменной FRAME.

С помощью кнопки Options (Параметры) можно выбрать формат сжатия видеофайлов и систему работы с ними. Кроме Microsoft Video 1.1 возможна работа и с рядом других видеосистем, разумеется, если они установлены. Чем больше конечное значение переменной FRAME и выше частота кадров, тем более плавно происходит считывание, но увеличивается объем AVI-файлов.

По окончании создания серии кадров для анимации в окне появится проигрыватель анимационных кадров (рис. 4.41).

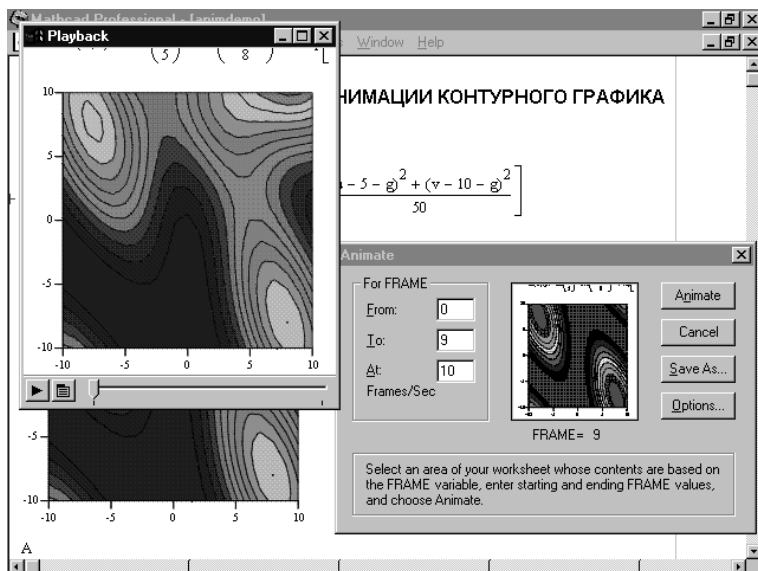


Рис. 4.41. Просмотр анимационных кадров с помощью проигрывателя

На этом создание анимационного рисунка заканчивается и можно приступить к его просмотру.

4.9.4. Воспроизведение анимированного рисунка

Для просмотра анимированных рисунков используется специальный проигрыватель Playback. Щелкнув в окне проигрывателя на кнопке с изображением треугольника, можно наблюдать изменение графика во времени.

Щелкнув на кнопке Save As (Сохранить как) в диалоговом окне Animate (Анимация), можно вызвать стандартное окно записи файлов на диск. В этом окне можно найти папку, в которую будет помещен записываемый файл. Файл записывается с расширением .avi, принятым для файлов программной видеосистемы Microsoft Video for Windows. До записи в файл кадры видеоролика хранятся в оперативной памяти ПК, что ограничивает их число.

При необходимости размер окна проигрывателя можно уменьшить в два раза или, напротив, увеличить в два раза. Это делается с помощью меню, появляющегося при щелчке в окне проигрывателя на кнопке с изображением экрана (вторая слева).

При воспроизведении анимированных рисунков рекомендуется отключить все параметры автоматического масштабирования графиков. Автоматическое изменение масштаба подчас ведет к скачкообразному изменению размеров графика, хотя

на деле он должен меняться без скачков, точнее говоря, со скачками, но определяемыми только изменением переменной $FRAME = 0, 1, 2, 3$ и так далее. Скачки размеров фигур в ходе воспроизведения анимации порою могут озадачить даже опытных пользователей, но куда хуже, если не очень опытные пользователи принимают их «за чистую монету».

4.9.5. Вызов проигрывателя

Теперь перейдем к описанию самой «крутой» возможности графики системы Mathcad — просмотру произвольных видеофильмов. Это может быть боевик, который позволяет немного скрасить выполнение нудных вычислений, или специально подобранная вставка для пояснения сути электронного урока.

Любой видеофайл с расширением *.avi* (не обязательно с описанной выше анимированной графикой) может быть просмотрен с помощью проигрывателя, запускаемого командой Playback (Воспроизвести) меню View (Вид). При этом проигрыватель появляется в окне Mathcad вначале в уменьшенном виде (рис. 4.42).

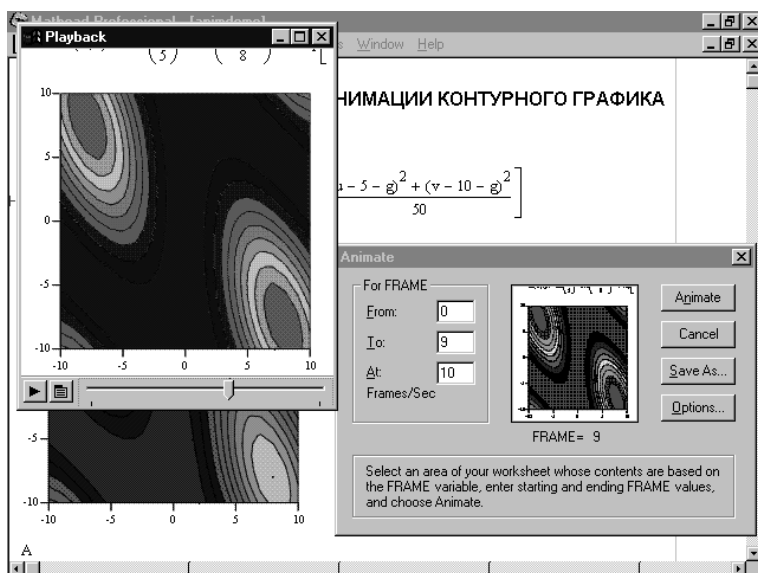


Рис. 4.42. Проигрыватель после вызова командой Playback

Как уже отмечалось, проигрыватель управляется всего двумя кнопками. Первая (с изображением треугольника) запускает проигрыватель. Однако для воспроизведения какого-либо видеоклипа его файл должен быть загружен. Для загрузки файлов используется вторая кнопка. Она выводит на экран меню, показанное на рис. 4.43.

Ниже перечислены пункты этого меню:

- View (Вид) — подменю управления размером окна просмотра;
- Volume (Громкость) — ползунок (регулятор) управления громкостью (если AVI-файл содержит звук);
- Speed (Скорость) — ползунок (регулятор) управления скоростью воспроизведения;

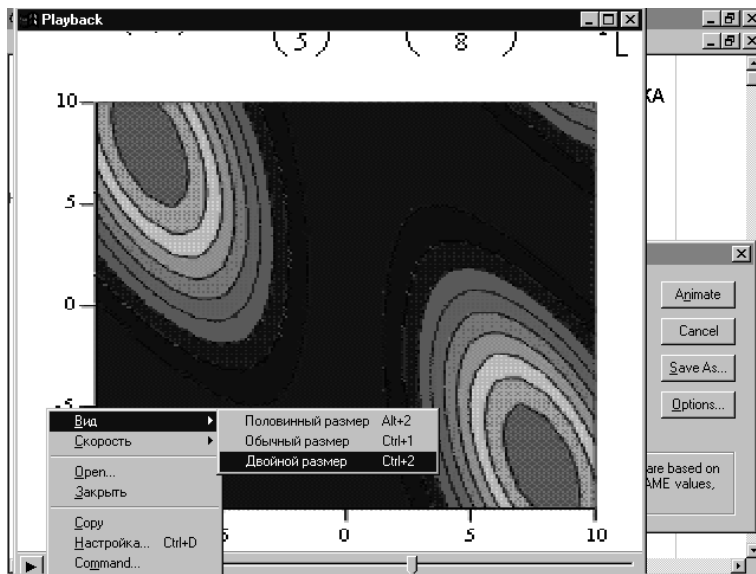


Рис. 4.43. Воспроизведение видеофильма проигрывателем Playback в среде Mathcad

- Open (Открыть) — вывод стандартного диалогового окна загрузки файлов;
- Close (Закреть) — закрытие текущего окна проигрывателя;
- Copy (Копировать) — копирование фильма в буфер обмена;
- Configure (Настройка) — вывод диалогового окна настройки;
- Command (Команда) — управление из командной строки.

ПРИМЕЧАНИЕ Поскольку проигрыватель Playback не относится к системе Mathcad, имена команд зависят от его версии. В данном случае приводятся команды как для локализованной, так и для нелокализованной версий проигрывателя. Некоторые из команд (например, команда отображения регулятора громкости) появляются только в том случае, когда они необходимы.

Выбор команды Open (Открыть) позволяет открыть стандартное диалоговое окно для загрузки файлов. Кроме кнопок в окне проигрывателя расположен индикатор воспроизведения в виде шкалы с ползунком.

На рис. 4.44 показано окно Video Properties (Свойства видео), и ющееся при выборе команды (Настройка) или при нажатии комбинации клавиш Ctrl+D. Это окно позволяет с помощью переключателей задать видеорежим:

- Video Mode (В окне) — обычный для Windows оконный режим (размер окна можно выбрать в раскрывающемся списке);
- Full Screen (Во весь экран) — изображение занимает весь экран.

Применение проигрывателя заметно облегчает построение эффектных обучающих программ, в частности подготовку видеороликов по работе с самой системой. К сожалению, AVI-файлы требуют достаточно большого объема памяти на жестком диске, поэтому приходится делать их довольно короткими. В то же время множество готовых видеофильмов можно найти на компакт-дисках и в Интернете.

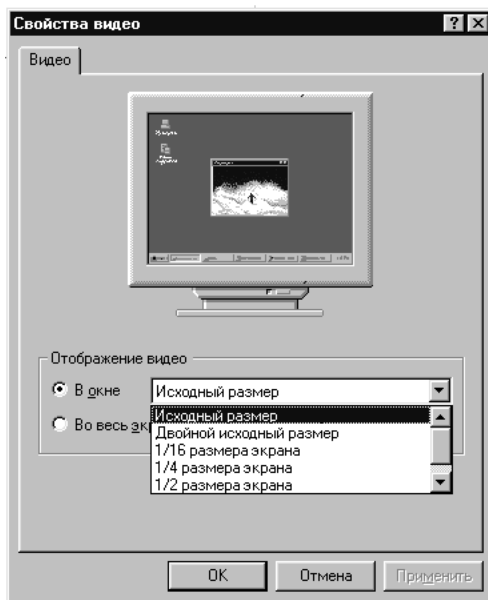


Рис. 4.44. Диалоговое окно свойств видео

Глава 5. Форматирование объектов

Вид документов с различными блоками в значительной степени зависит от их форматирования. Многие параметры форматирования заданы по умолчанию, но дополнительная настройка параметров форматирования позволяет целенаправленно менять вид документов и их блоков как в процессе их подготовки, так и после их создания. В этой главе вы изучите многочисленные параметры форматирования самых различных объектов.

5.1. Понятие о форматировании объектов документов

5.1.1. Для чего нужно форматирование

Под форматом объектов документов подразумевается некоторый набор их характеристик, таких как размеры изображения объекта на экране, размеры и стиль символов математических выражений и текстовых комментариев, цвет и т. д. Начинаящие пользователи системы Mathcad, как правило, вполне удовлетворены теми форматами, которые заданы по умолчанию. И это не случайно — форматы выбраны с учетом устоявшейся практики научно-технических вычислений и технических характеристик современных ПК.

Однако по мере освоения системы те или иные форматы могут потребовать изменений. К примеру, пользователя может не удовлетворить точность представления дробных чисел всего тремя цифрами после десятичной точки или отсутствие изображения осей на графике, а может быть, напротив, их присутствие. Может потребоваться изменение размеров математических символов в формулах или шрифтов в текстовых комментариях.

5.1.2. Позиция Format меню

Все команды для изменения формата сведены в меню Format (Формат):

- Equations (Выражения) — задание формата выражений;
- Result (Результат) — задание формата чисел результата;
- Text (Текст) — задание формата текста;
- Paragraph (Абзац) — задание формата абзацев;
- Tabs (Табуляция) — установка позиций табуляции;
- Style (Стиль) — задание стиля;
- Properties (Свойства) — задание свойств;
- Graph (График) — подменю с командами форматирования графиков:
 - X-Y Plot (Декартов график);
 - Polar Plot (Полярный график);
 - 3D Plot (Трехмерный график);
 - Trace (Трассировка);
 - Zoom (Масштаб);
- Color (Цвет) — подменю с командами задания цвета:
 - Background (Фон);

- Highlight (Выделение);
- Annotation Color (Цвет комментариев);
- Use Default Palette (Использовать палитру по умолчанию);
- Optimize Palette (Оптимизировать палитру);
- Separate Regions (Разделение областей) — разделение областей (блоков);
- Align Regions (Выравнивание областей) — подменю с командами задания взаимного расположения областей:
 - Across (По горизонтали);
 - Down (Вниз);
- Area (Область) — подменю установки атрибутов областей:
 - Lock (Блокировка области);
 - Unlock (Разблокировка области);
 - Collapse (Скрытие области);
 - Expand (Раскрытие области);
- Headers/Footers (Колонтитулы) — создание колонтитулов;
- Repaginate Now (Подгонка страниц) — перестройка страниц так, чтобы ни один из блоков не попал на линию раздела страниц.

Раскрытое меню Format показано на рис. 5.1. Состав этого меню в системах Mathcad 2001i/11/12 одинаков.

Все перечисленные выше команды доступны не только через меню Format. Обычно соответствующие диалоговые окна для настройки параметров форматирования можно вызвать, дважды щелкнув на соответствующем объекте. Еще один путь доступа к параметрам форматирования — контекстные меню. Ниже рассмотрено действие всех этих команд вне зависимости от способа их вызова.

5.1.3. Форматирование математических выражений

Математические выражения имеют довольно сложную структуру — они содержат переменные, константы, операторы и специальные знаки [32]. Иногда вид математических выражений не соответствует привычным для нас образам. Чаще всего это связано с отказом от применения типовых параметров форматирования, установленных в Mathcad по умолчанию.

Шрифты для математических символов привязаны к определенным их классам, например к числовым константам, переменным, надписям на графиках и т. д. Для переменных и надписей на графиках Mathcad по умолчанию использует шрифт Times New Roman размером 10 пунктов. Однако с помощью команды Equations (Выражения) меню Format (Формат) можно назначить для переменных, надписей, чисел и других символов в математических выражениях иной шрифт и размер шрифта. Эта команда выводит диалоговое окно Equation Format (Формат выражений), показанное на рис. 5.2.



Рис. 5.1. Меню Format

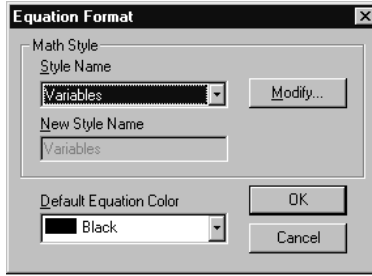


Рис. 5.2. Окно форматирования математических выражений

Это окно позволяет выбрать группу объектов в математических выражениях, шрифтовое оформление которых подлежит модификации. К таким объектам относятся переменные (Variables), константы (Constant) и объекты пользователя (User N, где N — номер группы от 1 до 7).

Выбрав в раскрывающемся списке Style Name (Имя стиля) один из указанных выше объектов (см. рис. 5.2), можно щелкнуть на кнопке Modify (Изменить), чтобы открыть стандартное диалоговое окно для настройки параметров шрифта. На рис. 5.3 показано окно выбора шрифта для переменных (Variables).

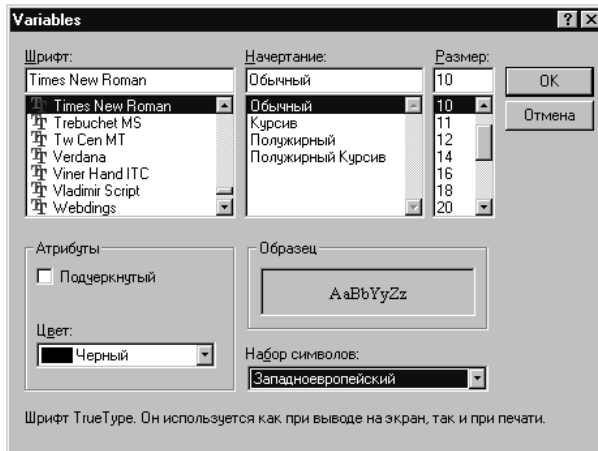


Рис. 5.3. Окно выбора шрифта для переменных

Для всех объектов выбранного стиля устанавливаются единые параметры шрифтов. Это позволяет выбрать стиль представления математических выражений, наиболее удобный для пользователя.

5.2. Форматирование результатов вычислений

5.2.1. Установки формата действительных чисел

Команда Result (Результат) меню Format (Формат) выводит диалоговое окно параметров форматирования результатов (рис. 5.4).

Это окно содержит четыре вкладки. На рисунке открыта вкладка Number Format (Формат чисел), позволяющая задать формат представления чисел результата:

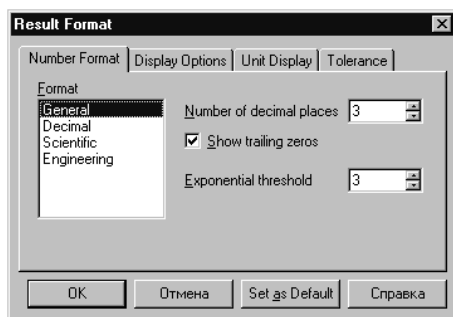


Рис. 5.4. Окно установки формата чисел

General (Общий), Decimal (Десятичный), Scientific (Научный) и Engineering (Инженерный). Это типичные форматы, знакомые пользователям калькуляторов. В поле со счетчиком Number of Decimal places (Число десятичных разрядов) можно задать число разрядов после десятичной точки (по умолчанию 3). В поле со счетчиком Exponential Threshold (Экспоненциальный порог) задается порог n от 0 до 15, при превышении которого (то есть при превышении 10^n) число отображается с указанием порядка. Например, при $n = 3$ (по умолчанию) $1000/7$ будет представлено как 142.857, а $10000/7$ как $1.429 \cdot 10^3$. Флажок Show trailing zeros (Показывать завершающие нули) позволяет задать необходимость дополнения чисел результата завершающими нулями, если число цифр после десятичной точки меньше указанного в поле Number of Decimal places. Например, при числе десятичных разрядов $n = 3$ результат выражения $5/2$ будет выглядеть как 2.500.

На следующей вкладке Display Options (Параметры отображения), представленной на рис. 5.5, задается основание системы счисления для представления чисел (раскрывающийся список Radix): Decimal (Десятичное), Binary (Двоичное), Hexadecimal (Шестнадцатеричное) или Octal (Восьмеричное). На рисунке также показаны примеры вывода чисел в этих форматах. Шестнадцатеричные числа отмечаются в конце символом «h», восьмеричные — символом «o», двоичные — символом «b». Эти числа находят применение в основном для описания адресной системы компьютеров. Флажок Expand nested arrays (Раскрывать вложенные массивы) обеспечивает раскрытие (расширение) встроенных в массив массивов (Mathead 2001 допускает применение в качестве элементов массива других массивов).

5.2.2. Представление комплексных чисел

Комплексное число задается в алгебраической форме как $Z = a + ib$, где $a = \text{Re}(Z)$ — действительная часть комплексного числа Z и $b = \text{Im}(Z)$ — мнимая часть Z . Формат вывода мнимой единицы (j или i) также выбирается на вкладке Display Options (Параметры отображения) в раскрывающемся списке Imaginary value (Мнимая единица).

Если $\text{Re}(Z) / \text{Im}(Z) > 10^n$, комплексное число Z выводится как действительное, а если $\text{Im}(Z) / \text{Re}(Z) > 10^n$ — как мнимое. Значение n задается в поле со счетчиком Complex Threshold (Порог для комплексных чисел) на вкладке Tolerance (Допуск).

5.2.3. Представление размерных величин

Следующая вкладка — Unit Display (Отображение размерных величин), показанная на рис. 5.6, позволяет задать параметры отображения размерных величин.

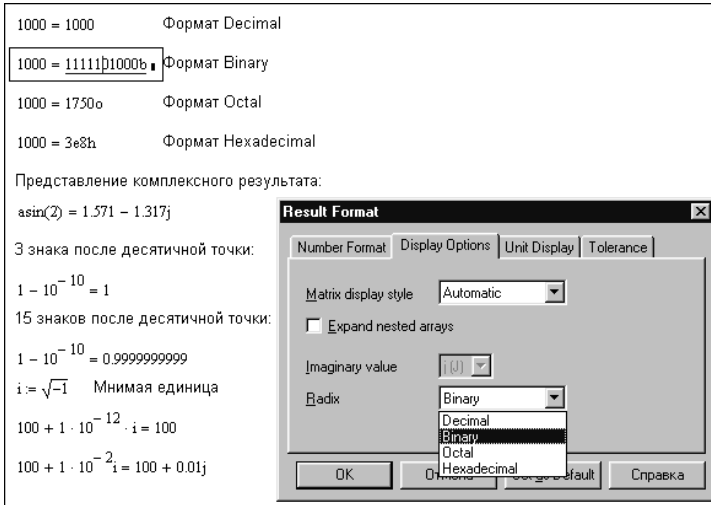


Рис. 5.5. Параметры отображения чисел

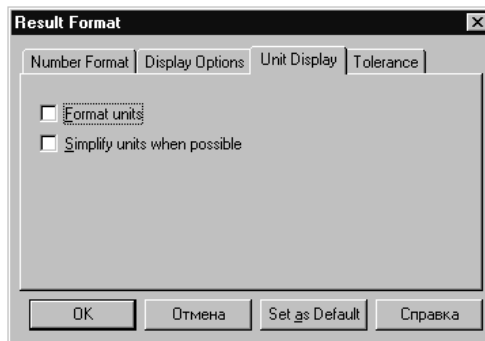


Рис. 5.6. Окно форматирования результатов с открытой вкладкой Unit Display

Флажок **Format units** (Форматировать размерные величины) задает форматы размерных величин в виде дроби, если такие величины имеют отрицательный показатель степени.

Ниже представлены простые примеры работы с размерными величинами из области электротехники:

$$E := 10 \text{ volt}$$

$$R := 2.5 \text{ ohm}$$

$$I := \frac{E}{R}$$

$$I = 4 \text{ A}$$

$$E \cdot I = 40 \text{ W}$$

$$\frac{E^2}{R} = 40 \text{ W}$$

Функция упрощения символьных вычислений не всегда работает с размерными величинами. Установка флажка *Simplify units when possible* (Упрощать размерные величины по возможности) допускает вывод размерных величин только в том случае, когда это возможно.

5.2.4. Представление результатов численных расчетов

Последняя вкладка *Tolerance* (Допуск), показанная на рис. 5.7, позволяет задать допустимую границу для комплексных чисел *Complex threshold* (Порог для комплексных чисел) и допустимую границу для малых действительных чисел *Zero threshold* (Нулевой порог).

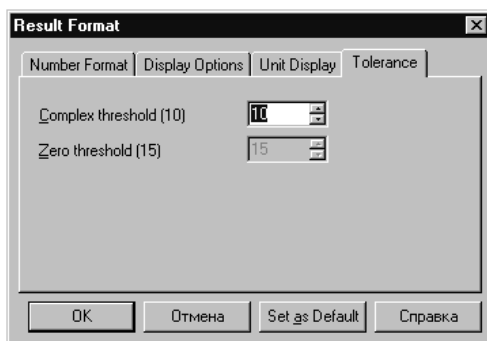


Рис. 5.7. Окно форматирования результатов с открытой вкладкой *Tolerance*

Как уже говорилось, в поле со счетчиком *Exponential threshold* на вкладке *Number Format* задается порог n от 0 до 15, при превышении которого (то есть при превышении 10^n) число отображается с указанием порядка. Если же число по модулю меньше указанного в поле со счетчиком *Zero threshold*, то число представляется в виде нулей. Все это особенно важно при научно-технических расчетах, когда излишняя точность представления чисел ни к чему хорошему не ведет.

Описание параметра *Complex threshold* приводилось выше, в подразделе «Представление комплексных чисел».

5.2.5. Представление массивов

Формат вывода массивов чисел выбирается в раскрывающемся списке *Matrix display style* (Стиль отображения матриц) на вкладке *Display Options* (Параметры отображения). Допустимые варианты:

- *Matrix* — в виде матрицы с длинными скругленными скобками;
- *Table* — в виде электронной таблицы с линейками прокрутки при выделении;
- *Automatic* — автоматически, в зависимости от размера массива.

На рис. 5.8 представлены образцы задания и вывода матриц — малых сверху и большой снизу, представленных случайными числами.

Отображение больших массивов в виде электронной таблицы, несомненно, более компактно и удобно. Однако такое представление не совсем соответствует традиционной матричной форме представления двумерных массивов, принятой в математике. Следует обратить внимание на то, что при выделении таблицы появляются затемненные области номеров строк и столбцов, в которых видны индек-

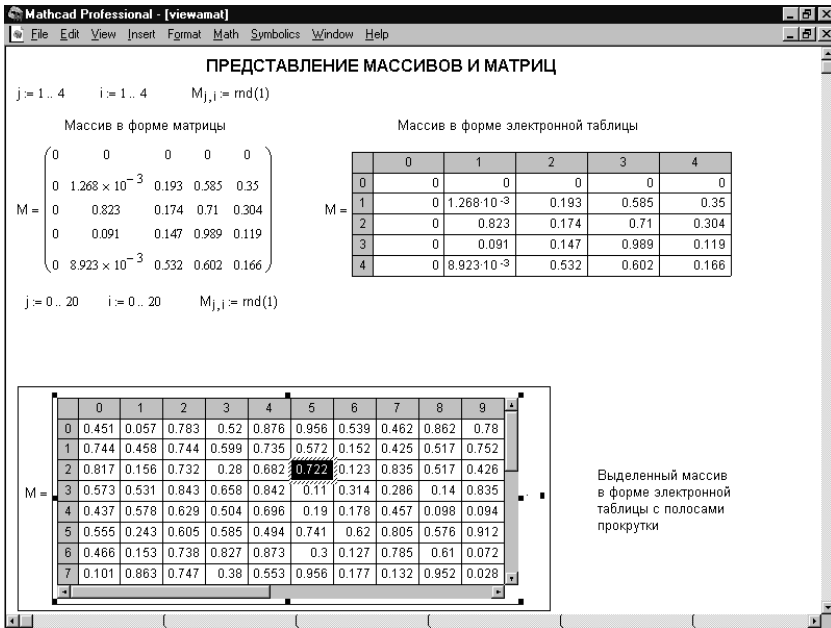


Рис. 5.8. Представление массивов чисел в виде матрицы и электронной таблицы

сы ячеек таблицы (элементов массива) и полосы прокрутки, дающие доступ к любой ячейке таблицы.

Начальное значение индексов для строк и столбцов матриц задается значением системной переменной ORIGIN, которая может принимать значения 0 (по умолчанию) или 1. Если при значении ORIGIN, равном 0, задавать индексы с 1, то элементы, имеющие индекс строк и столбцов, равный 0, будут иметь нулевые значения. Этот вариант также показан на рис. 5.8, который иллюстрирует возможность поэлементного задания матриц.

5.3. Форматирование текста

Для изменения параметров форматирования текста предназначены несколько команд меню Format (Формат). Например, команда Text (Текст) позволяет выбрать шрифт и настроить параметры шрифта, команда Paragraph (Абзац) предназначена для настройки параметров абзацев, а команда Style (Стиль) — для задания стиля текстовых комментариев.

5.3.1. Изменение параметров шрифта

Команда Text (Текст) доступна в меню Format (Формат) только в том случае, если указатель мыши находится в текстовой области. Она позволяет выбрать шрифт в имеющемся списке, изменив его для выделенной области текста. При выборе этой команды открывается диалоговое окно Text Format (Формат текста) со списком шрифтов (рис. 5.9).

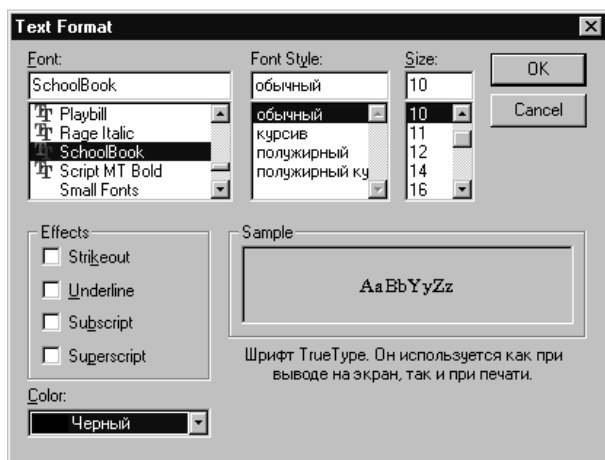


Рис. 5.9. Диалоговое окно форматирования текста

Это окно содержит три списка:

- Font (Шрифт) — список шрифтов;
- Font Style (Стиль шрифта) — список начертаний шрифта;
- Size (Размер) — список размеров шрифта.

Список шрифтов определяется тем, какие шрифты установлены в системе Windows. Начертание шрифта может быть следующим: Normal (Обычный), Bold (Полужирный), Italic (Курсив). Эти три вида начертаний можно использовать в любой комбинации с любым шрифтом. Они могут быстро вводиться с помощью одноименных кнопок на панели форматирования.

Имеется также группа флажков Effects (Эффекты):

- Strikeout (Зачеркнутый) — перечеркнутые посередине символы;
- Underline (Подчеркнутый) — подчеркнутые снизу символы;
- Subscript (Подстрочный) — подстрочные символы;
- Superscript (Надстрочный) — надстрочные символы.

Раскрывающийся список Color (Цвет) позволяет выбрать цвет символов. Поле Sample (Образец) дает возможность наблюдать за написанием нескольких букв выбранным шрифтом с использованием установленных вами параметров. Как обычно, кнопка ОК диалогового окна фиксирует выбор, а кнопка Cancel (Отмена) позволяет отказаться от модификации текстовых надписей.

5.3.2. Форматирование абзацев

Если указатель мыши находится в текстовом блоке, то для форматирования абзаца можно использовать команду Paragraph (Абзац) меню Format (Формат). Она выводит диалоговое окно, показанное на рис. 5.10 снизу.

В полях группы Indent (Отступы) окна форматирования абзацев можно ввести значения отступов в дюймах:

- Left (Слева) — установка отступа слева;
- Right (Справа) — установка отступов справа;

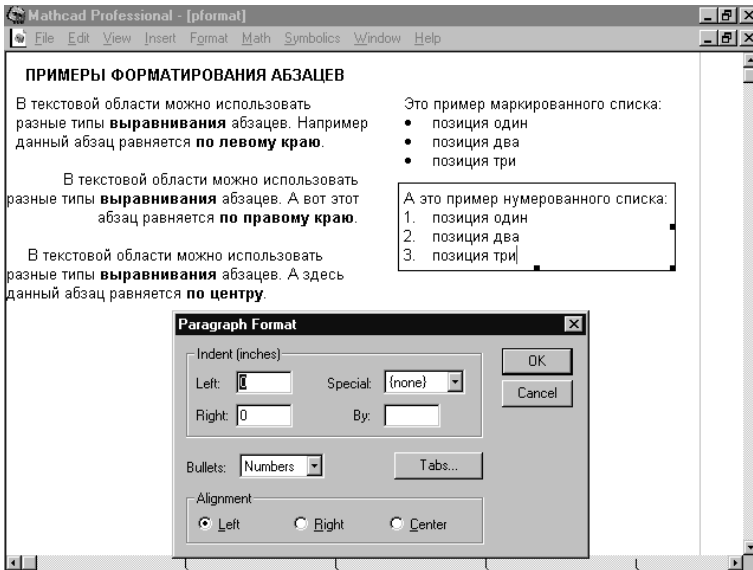


Рис. 5.10. Окно установки формата абзаца

- Special (Специальное) — специальная установка отступов (например, можно задать отступ только для первой строки). Точное значение вводится в поле By (На).

С помощью переключателей группы Alignment (Выравнивание) внизу окна можно задать способ выравнивания строк:

- Left (Влево) — по левому краю абзаца;
- Right (Вправо) — по правому краю абзаца;
- Center (По центру) — по центру абзаца.

Возможно также форматирование текстовых строк в виде списков (раскрывающийся список Bullets):

- None (Отсутствует) — нет форматирования;
- Bullets (Маркеры) — маркированный список;
- Number (Нумерация) — нумерованный список.

Действие кнопки Tabs (Табуляция) рассмотрено ниже. На рис. 5.10 показаны примеры различного выравнивания текста. При выравнивании текста влево задан абзацный отступ для первой строки. В прежних версиях Mathcad было два типа текстовых блоков — с шириной, меньшей ширины страницы, и с полной шириной, равной ширине страницы. В новой версии остался лишь один блок — с его помощью можно задавать как узкие надписи, так и надписи во всю ширину страницы.

5.3.3. Установка позиций табуляции

Табуляцией называют перемещение текстового курсора внутри текстового блока на заданную позицию с помощью клавиши Tab. Для установки позиций табуляции используется диалоговое окно, показанное на рис. 5.11, для открытия которого нужно выбрать команду Tabs (Табуляция) в меню Format (Формат) или (см. выше подраздел «Форматирование абзацев») щелкнуть на одноименной

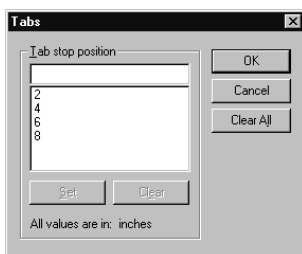


Рис. 5.11. Окно установки позиций табуляции

кнопке в диалоговом окне Paragraph Format (Формат абзаца). Позиции табуляции представляются списком, а соответствующие значения вводятся (кнопка Set) в верхнем поле. Можно стирать как отдельные позиции, так и весь список позиций табуляции с помощью кнопок Clear (Очистить) и Clear All (Очистить все).

5.3.4. Задание стиля текстовых комментариев

Под стилем текстовых комментариев подразумевается именованная совокупность параметров форматирования текстовых объектов — заголовков различного уровня, абзацев, списков и т. д. Подготовка текстовых комментариев в определенном стиле необходима при создании документов на высоком полиграфическом уровне.

Команда Style (Стиль) меню Format (Формат) позволяет устанавливать стили для различных текстовых объектов. При ее выборе открывается диалоговое окно Text Styles (Стили текста), показанное на рис. 5.12 сверху.

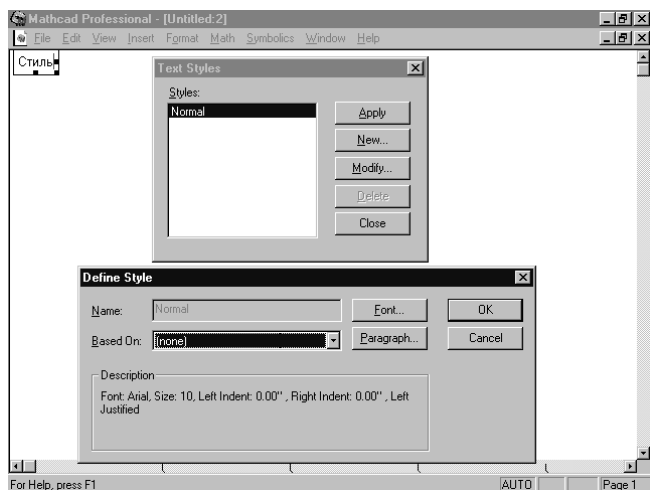


Рис. 5.12. Окна задания стиля текстовых комментариев

В окне выбора стилей имеются перечень стилей и ряд кнопок для выполнения следующих операций:

- Apply (Применить) — применить выбранный стиль;
- New (Создать) — создать новый стиль;
- Modify (Изменить) — модифицировать выбранный стиль;

- Delete (Удалить) — удалить выбранный стиль;
- Close (Закреть) — закрыть окно и отказаться от выбора стиля.

Для создания нового стиля щелкните на кнопке New, а для изменения имеющегося — на кнопке Modify. На рис. 5.12 снизу показано дополнительное окно, которое появляется при щелчке на кнопке Modify (окно, появляющееся при щелчке на кнопке New, отличается только возможностью доступа к полю Name для ввода имени нового стиля).

В окне модификации стиля можно задать параметры шрифта (кнопка Font) и абзаца (кнопка Paragraph). При создании нового стиля он дополняет уже существующий список стилей. По умолчанию задан единственный стиль — normal.

5.3.5. Надписи на переднем и заднем плане

Форматирование графиков предусматривает вывод титульной надписи сверху или снизу графиков, а также надписей по осям (см. подраздел «Форматирование надписей на графиках» в разделе «Форматирование двумерных графиков в полярной системе координат» и подраздел «Параметры надписей» в разделе «Форматирование трехмерных графиков»). Однако этот вид форматирования имеет серьезные недостатки — размер надписей ограничен, место их фиксировано и смена шрифта не предусмотрена. В частности, не удается сделать надписи на русском языке. В то же время этих ограничений нет в текстовых блоках.

Однако до появления версии Mathcad 2000/2001 наложение текстовых блоков на рисунки вело к подавлению вывода надписей. Другими словами, надписи располагались на заднем плане.

В Mathcad 2000/2001 (и, разумеется, в Mathcad 2001i/11/12) появилась возможность обойти этот недостаток. В контекстном меню текстового блока имеются две относящиеся к надписям команды — Bring to Front (На передний план) и Send to Back (На задний план). После выбора команды Bring to Front надпись выводится поверх рисунка.

Описанная возможность весьма ценна, поскольку позволяет украсить рисунок надписями с любым шрифтом, стилем и местом расположения. Единственный недостаток такого метода — при изменении размера и местоположения рисунка нарушается взаимное расположение элементов графика и текстовых надписей. Впрочем, его несложно восстановить перетаскиванием текстовых блоков.

5.4. Установка свойств объектов

5.4.1. Установка свойств математического выражения

Одно из определяющих понятий в операционных системах Windows — свойства различных объектов. Это могут быть цвета выделений, характер выполнения математических выражений и т. д. Для установки свойств предназначена команда Properties (Свойства) меню Format (Формат). Это контекстно-зависимая команда, и открывающиеся диалоговые окна могут несколько отличаться для разных объектов. Выбор объектов, как обычно, задается установкой в них курсора.

На рис. 5.13 слева показано окно установки свойств, когда в качестве объекта выбрано математическое выражение.

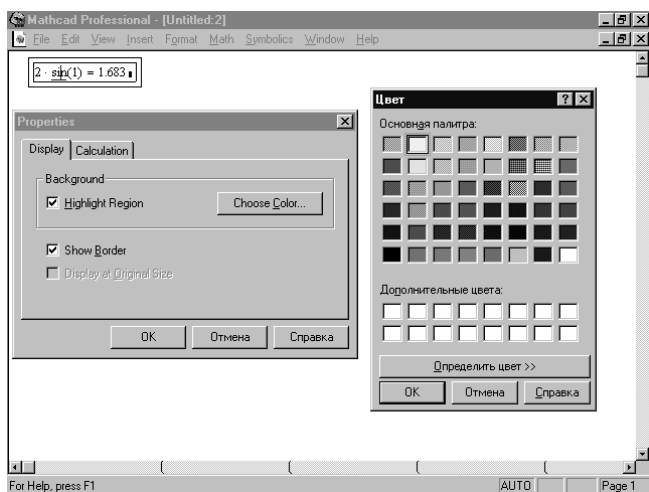


Рис. 5.13. Установка свойств математического выражения

Это окно имеет две вкладки:

- Display (Отображение) — свойства вывода;
- Calculation (Вычисления) — свойства вычислений.

К свойствам вывода в данном случае относится лишь наличие или отсутствие подкраски фона математического выражения (флажок Hight Region), а также наличие или отсутствие рамки блока (флажок Show Border). Активизируя кнопку Choose Color (Выбор цвета), можно вызвать окно задания цвета фона выражений (на рис. 5.13 справа). Кнопка Define Color (Определить цвет) расширяет это окно и вводит в него палитру цветов (рис. 5.14), которая позволяет получать разнообразные цвета для цветового оформления выражений.

Вкладка Calculation (рис. 5.15) позволяет отключить или включить вычисление математических выражений. Напоминаем, что обычно математические выражения вычисляются автоматически. Однако, установив флажок Disable Evaluation (Запретить выполнение), можно сделать выражение пассивным, то есть, по существу, просто комментарием.

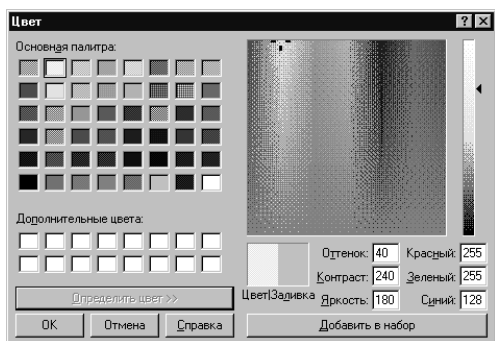


Рис. 5.14. Окно выбора цветов в расширенном виде

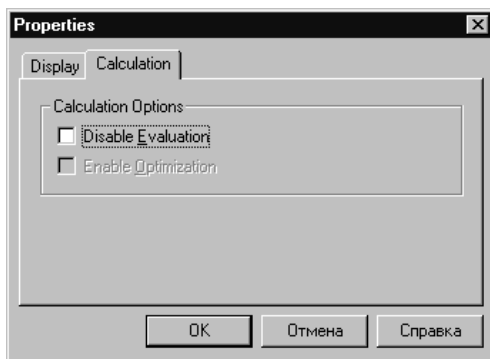


Рис. 5.15. Окно свойств с открытой вкладкой Calculation

5.4.2. Установка свойств оптимизации

Еще одно важное свойство математических выражений — их оптимизация. Флажок **Enable Optimization** (Разрешить оптимизацию) включает режим оптимизации, если он для этого выражения возможен (в противном случае этот флажок недоступен). При этом выражение, если оно вычисляется, представляется в аналитическом виде всегда, когда это возможно. Это свойство может привести к заметному ускорению вычислений в тех случаях, когда оптимизированное выражение проще исходного. Особенно заметен выигрыш при вычислении сложных функций (например, вычисления интегралов или производных) численными методами.

5.5. Форматирование двумерных графиков

5.5.1. Форматирование графиков в прямоугольной системе координат

Подменю **Graph** (График) меню **Format** (Формат) открывает доступ к командам форматирования графиков. Начнем с команды **X-Y Plot** (Декартов график). Она выводит диалоговое окно с параметрами форматирования двумерных графиков (рис. 5.16). Следует помнить, что для изменения формата уже построенного графика необходимо его выделить. Выделенный график обводится сплошной линией с маркерами изменения размера.

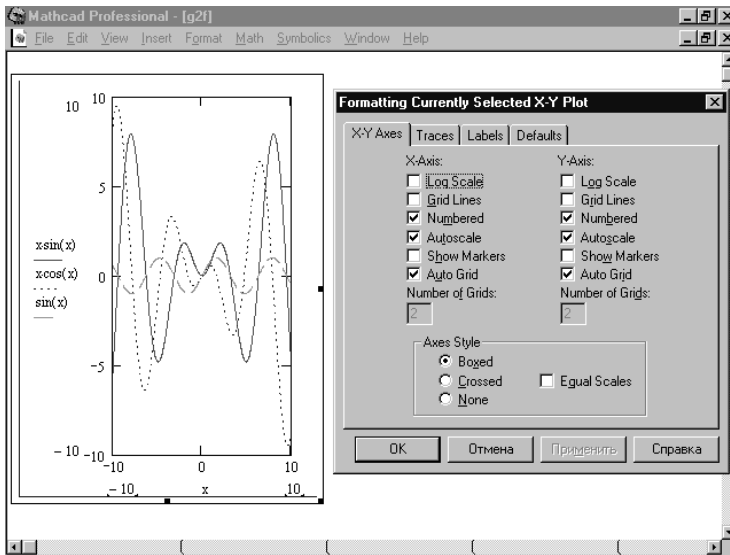


Рис. 5.16. Окно форматирования двумерного графика

Как видно из рисунка, диалоговое окно форматирования имеет четыре вкладки:

- X-Y Axes (Оси X-Y) — задание параметров отображения осей;
- Traces (Линии) — задание параметров отображения линий графика;
- Labels (Метки) — задание параметров отображения меток (надписей) у осей;

- Defaults (По умолчанию) — задание параметров по умолчанию.

Важно отметить, что все параметры форматирования относятся к выделенному графику и могут при необходимости изменяться.

5.5.2. Форматирование осей

На вкладке X-Y Axes представлены следующие основные параметры, относящиеся к осям X и Y (Axis X и Axis Y):

- Log Scale (Логарифмический масштаб) — установка логарифмического масштаба;
- Grid Lines (Линии сетки) — установка линий сетки;
- Numbered (Пронумеровать) — установка цифровых данных по осям;
- Autoscale (Автомасштаб) — автоматическое масштабирование графика;
- Show Markers (Нанести риски) — установка делений по осям;
- Auto Grid (Автосетка) — автоматическая установка масштабных линий;
- Number of Grids (Число интервалов) — установка заданного числа масштабных линий.

Все эти параметры достаточно очевидны. Отметим лишь, что если снят флажок Grid Lines, то масштабная сетка графика не строится, хотя на осях размещаются короткие деления. При выводе масштабной сетки можно сменить цвет ее линий (по умолчанию он зеленый). Для этого есть кнопка Grid Color..., открывающая окно установки цвета. Флажок Numbered обеспечивает вывод цифр у осей. Флажок Autoscale обеспечивает автоматическое масштабирование графика. Это полезно, например, для округления нижнего и верхнего пределов изменений значений абсцисс и ординат, которые при автоматическом выборе масштаба могут оказаться десятичными числами с дробной частью.

Переключатели группы Axes Style (Стиль осей) позволяют задать стиль отображения координатных осей:

- Boxed (Рамка) — оси в виде прямоугольника;
- Crossed (Крест) — оси в виде креста;
- None (Нет) — отсутствие осей.

Флажок Equal Scales (Равные масштабы) позволяет установить одинаковые масштабы по осям графика.

Внизу вкладки имеются следующие кнопки:

- OK — закрыть окно и применить выбранные параметры к выделенному графику;
- Cancel (Отмена) — закрыть окно без применения выбранных параметров;
- Apply (Применить) — применить выбранные параметры к выделенному графику;
- Help (Справка) — вывод подсказки.

На рис. 5.17 показано, как влияет форматирование осей на вид исходного графика (см. рис. 5.16). В данном случае изменены расположение и нумерация делений на осях, а также выведены линии сетки.

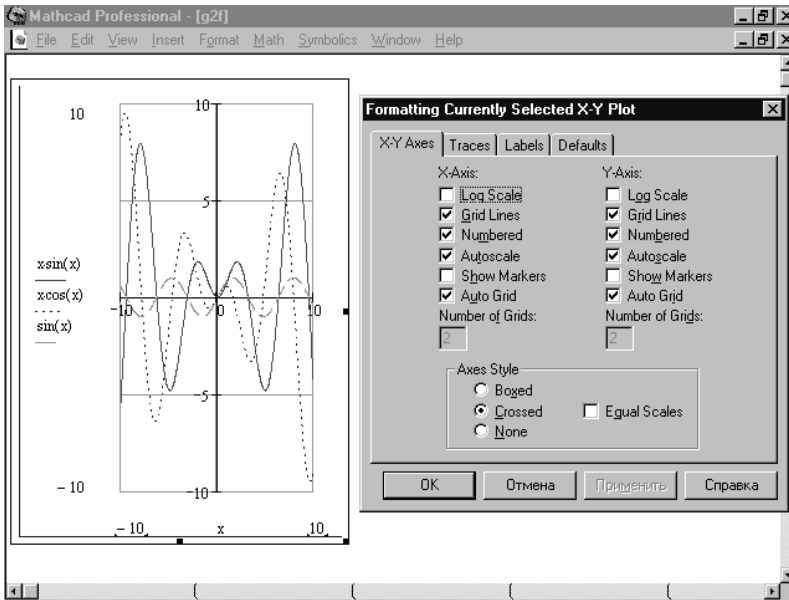


Рис. 5.17. Пример форматирования осей двумерного графика

5.5.3. Форматирование линий графиков

Следующая вкладка Traces (Линии), показанная на рис. 5.18, служит для управления отображением линий, из которых строится график.

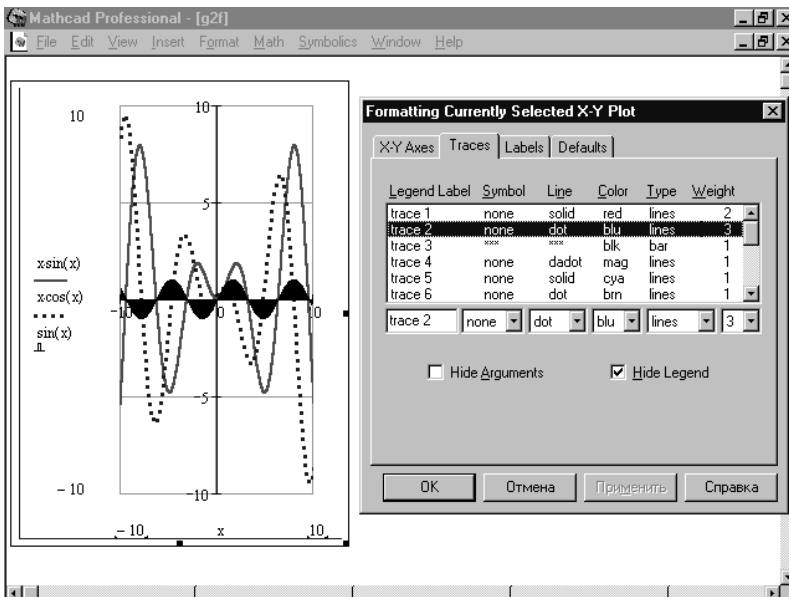


Рис. 5.18. Вкладка Traces

Вкладка позволяет управлять следующими параметрами линий графика:

- Legend Label (Метка легенды) — метка линий у оси ординат;
- Symbol (Символ) — символ, который помещается на линию, для отметки базовых точек графика;
- Line (Линия) — тип линий (сплошная, пунктирная и др.);
- Color (Цвет) — цвета линий и базовых точек;
- Type (Тип) — тип графика;
- Weight (Толщина) — толщина линий.

Узловые точки графиков (точки, для которых вычисляются координаты) часто требуется выделить какой-либо фигурой — кружком, крестиком, прямоугольником и т. д. Раскрывающийся список Symbol позволяет выбрать следующие отметки для базовых точек графика каждой из функций:

- none — без отметки;
- x's — наклонный крестик;
- +'x — прямой крестик;
- box — квадрат;
- dmnd — ромб;
- o's — окружность.

Графики отдельных функций можно также выделять, используя для их построения линии различного типа. Раскрывающийся список Line позволяет выбрать следующие типы линий:

- none — линия не строится;
- solid — непрерывная линия;
- dash — пунктирная линия;
- dadot — штрихпунктирная линия.

Другой распространенный способ выделения линий, относящихся к различным кривым на графике, заключается в изменении их цвета. Разумеется, этот метод эффективен при применении цветного дисплея и цветного (например струйного) принтера для распечатки графиков. Раскрывающийся список Color позволяет выбрать следующие основные цвета линий и базовых точек:

- red — красный;
- blu — синий;
- grn — зеленый;
- cya — голубой;
- brn — коричневый;
- blc — черный.

Важное значение имеет и тип графика. Раскрывающийся список Type позволяет выбрать тип графика:

- line — линии;
- points — точки;
- err — вертикальные черточки с оценкой интервала погрешностей;
- bar — в виде столбцов гистограмм;
- step — ступенчатая линия;
- draw — прямые от точки до точки.

Возможные конфликты между типом графика и типом линий автоматически устраняются. При этом приоритет отдается типу графика, а конфликтные типы линий или точек отмечаются тремя звездочками.

Два флажка в нижней части вкладки связаны с возможностью удаления с графика вспомогательных надписей:

- Hide Argument (Скрыть аргумент) — скрывает математические выражения на осях графика;
- Hide Legend (Скрыть легенду) — скрывает описание обозначений на графике.

На рис. 5.18 представлены результаты переформатирования линий графиков, показанных ранее на рис. 5.17.

5.5.4. Форматирование надписей на графиках

Вкладка Labels (Метки) позволяет вводить в рисунок дополнительные надписи, что иллюстрирует рис. 5.19 (см. также подраздел «Надписи на переднем и заднем плане» в разделе «Форматирование текста»). Эта вкладка появляется только при выделении уже созданных графиков.

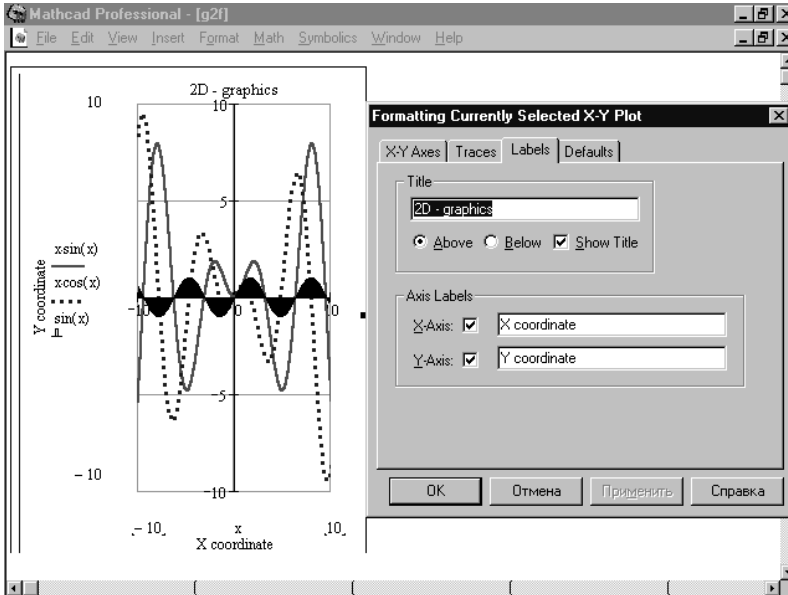


Рис. 5.19. Вкладка задания дополнительных надписей

Для ввода надписей служат следующие поля:

- Title (Заголовок) — установка титульной надписи к рисунку;
- X-Axis (Ось X) — установка надписи по оси X;
- Y-Axis (Ось Y) — установка надписи по оси Y.

В группе Title находятся переключатели Above (Сверху) и Below (Снизу) для установки титульной надписи либо над рисунком, либо под ним. Кроме того, флажок Show Title (Показать заголовок) позволяет отображать или скрывать титульную надпись.

5.5.5. Задание параметров по умолчанию

Вкладка Defaults (По умолчанию), как показано на рис. 5.20, служит для задания параметров форматирования графиков по умолчанию. Щелчок на кнопке Change to Defaults (Вернуть значения по умолчанию) позволяет отказаться от параметров, заданных на предыдущих вкладках, и вернуть их значения, установленные по умолчанию. Установка флажка Use for Defaults (Использовать для значений по умолчанию) позволяет сделать параметры, заданные на предыдущих вкладках, параметрами по умолчанию.

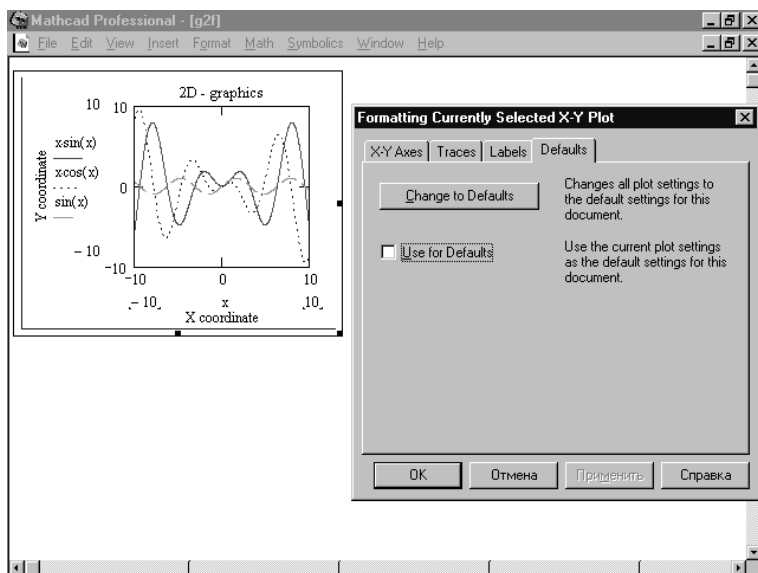


Рис. 5.20. Вкладка настройки параметров по умолчанию

5.5.6. Примеры форматирования двумерных графиков

Влияние некоторых параметров было показано на предыдущих рисунках. Напомним, что для назначения заданных параметров без закрытия диалогового окна форматирования служит кнопка Apply (Применить).

В целом перечисленные параметры предоставляют возможность создавать графики самых различных видов, удовлетворяющие любым запросам пользователя (см., например, шесть вариантов одного и того же графика на рис. 5.21).

Различные форматы позволяют уверенно различать кривые на графике и соотносить их с той или иной функцией, а введение дополнительных надписей (титальной и по осям) — разнообразить графики и повысить их информативность.

5.5.7. Трассировка графиков

Если щелкнуть в области графика правой кнопкой мыши, появляется контекстное меню. В нем есть интересная команда Trace (Трассировка), которую можно также найти в подменю Graph (График) меню Format (Формат).

Эта команда выводит окно трассировки двумерных графиков, представленное на рис. 5.22.

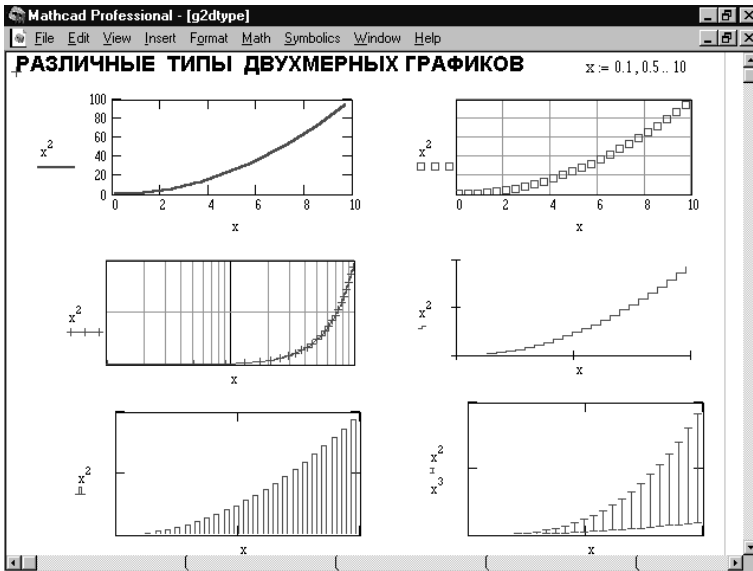


Рис. 5.21. Графики функции, построенные с применением различных форматов

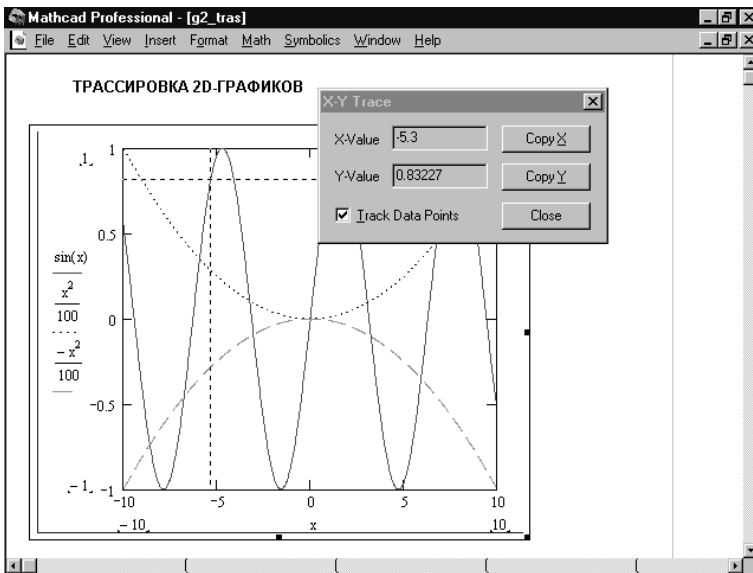


Рис. 5.22. Пример трассировки двумерного графика

Трассировка начинает работать после выделения графика. При этом в окне графика появляется большое перекрестие из двух черных пунктирных линий. С помощью указателя мыши его можно перемещать по графику с дискретностью, определяемой заданным шагом изменения абсциссы x . При этом координаты текущей точки ближайшей линии графика, на которую установлено перекрестие, отображаются в окне трассировки. Это позволяет в первом приближении выявить координаты особых точек графика.

Кнопки *Сору X* (Копировать X) и *Сору Y* (Копировать Y) позволяют занести соответствующие координаты текущей точки графика в буфер обмена. Кнопка *Close* (Заккрыть) завершает трассировку и закрывает окно трассировки.

Если установлен флажок *Trace Data Points* (Трассировка по точкам данных), то при трассировке указатель автоматически устанавливается на точку ближайшей кривой, отслеживая ее ход. При снятом флажке указатель может быть установлен в любую точку области графика, при этом координаты этой точки отображаются в окне трассировки.

5.5.8. Просмотр участков двумерных графиков

Некоторые графики представляют собой довольно любопытные кривые. Попробуйте, к примеру, разобраться в том, какой вид имеет кривая функции $x \cdot \sin(1/x)$. Нетрудно понять, что поведение функции при значении x , стремящемся к нулю, будет необычным. Но каким?

Выбор в подменю *Graph* (График) меню *Format* (Формат) команды *Zoom* (Масштаб) ведет к открытию диалогового окна, с помощью которого можно увеличить любой участок графика (применительно к функциям времени такую возможность часто называют «лупой времени»). Чтобы воспользоваться этим окном, надо предварительно выделить фрагмент графика функции (рис. 5.23).

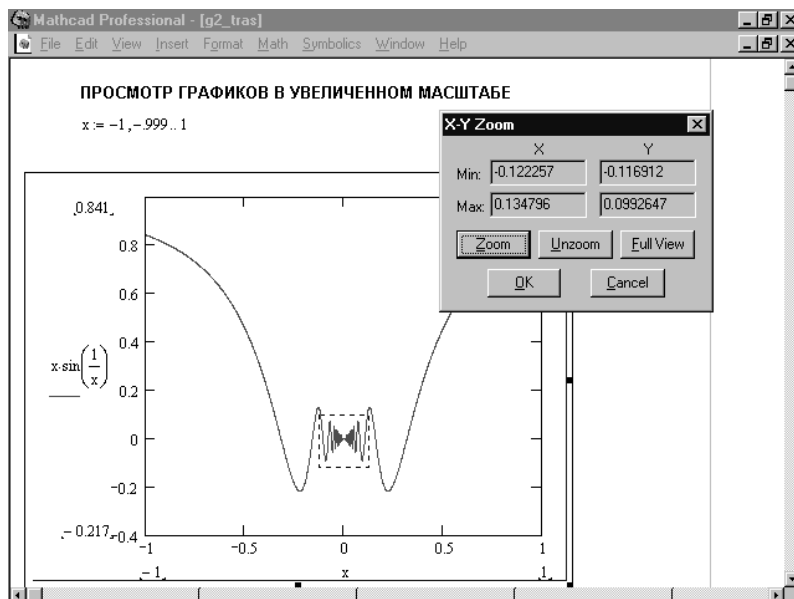


Рис. 5.23. Подготовка к просмотру фрагмента графика

При этом в окне просмотра отображаются минимальные и максимальные значения координат X и Y, определяющие область просмотра. Кнопки *Zoom*, *Unzoom* и *Full View* позволяют, соответственно, увеличить выделенную часть графика, снять выделение и вернуться к просмотру всего графика. На рис. 5.24 показан случай, когда был выполнен щелчок на кнопке *Zoom*.

Нетрудно заметить, что выделенный участок графика увеличивается до размера всего окна просмотра. При этом поведение функции становится более понятным.

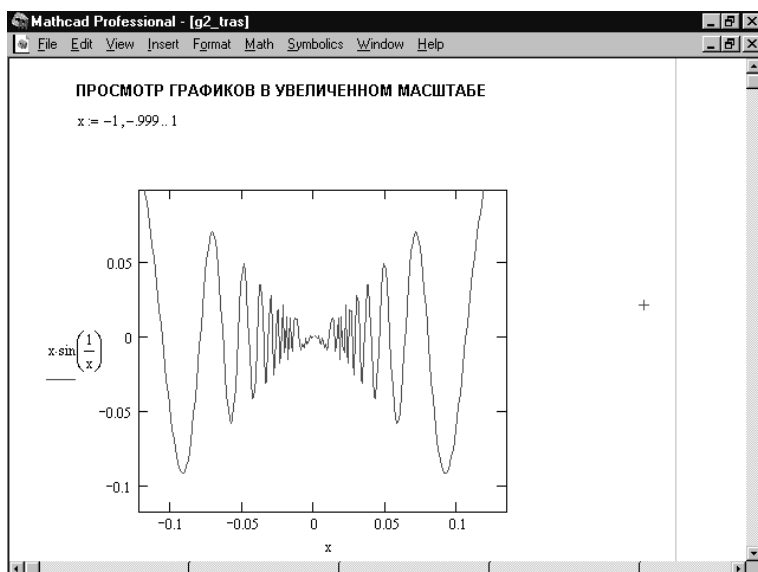


Рис. 5.24. Просмотр выделенного фрагмента двумерного графика

5.6. Форматирование графиков в полярной системе координат

5.6.1. Установка форматов графиков в полярной системе координат

Команда Polar Plot (Полярный график) подменю Graph (График) меню Format (Формат) обеспечивает задание форматов графиков в полярной системе координат. При ее выполнении выводится диалоговое окно, показанное на рис. 5.25.

Окно содержит четыре вкладки:

- Polar Axes (Полярные оси) — задание параметров отображения осей;
- Traces (Линии) — задание параметров отображения линий графика;
- Labels (Метки) — задание параметров отображения меток (надписей) у осей;
- Defaults (По умолчанию) — задание параметров по умолчанию.

На рис. 5.25 открыта вкладка Polar Axes. На этой вкладке задаются форматы отображения радиус-вектора (Radial) и его угла (Angular). Возможные варианты были описаны ранее (см. раздел «Форматирование двумерных графиков в прямоугольной системе координат»), и нет смысла их повторять. Переключатели в группе Axes Style (Стиль осей) позволяют задать стиль отображения координатных осей либо в виде окружности вокруг графика — Perimeter (Обрамление), либо в виде расположенных крестом осей — Crossed (Визир), либо без представления координатных осей — None (Отсутствует).

Параметры вкладок Traces, Labels и Defaults аналогичны описанным ранее, хотя вкладка Labels упрощена — на ней предусмотрена установка только дополнительной титульной надписи.

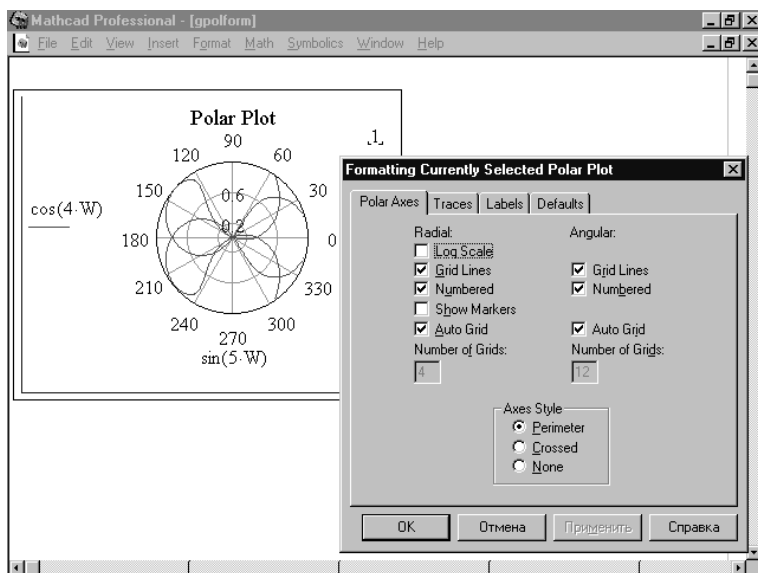


Рис. 5.25. Окно установки форматов графиков в полярной системе координат

5.6.2. Трассировка графика в полярной системе координат

Как и у графиков в декартовой системе координат, в полярной системе координат также предусмотрена возможность трассировки графиков с помощью графического маркера. Она реализуется командой Format (Формат) ▶ Graph (График) ▶ Trace (Трассировка), которая выводит диалоговое окно, показанное на рис. 5.26.

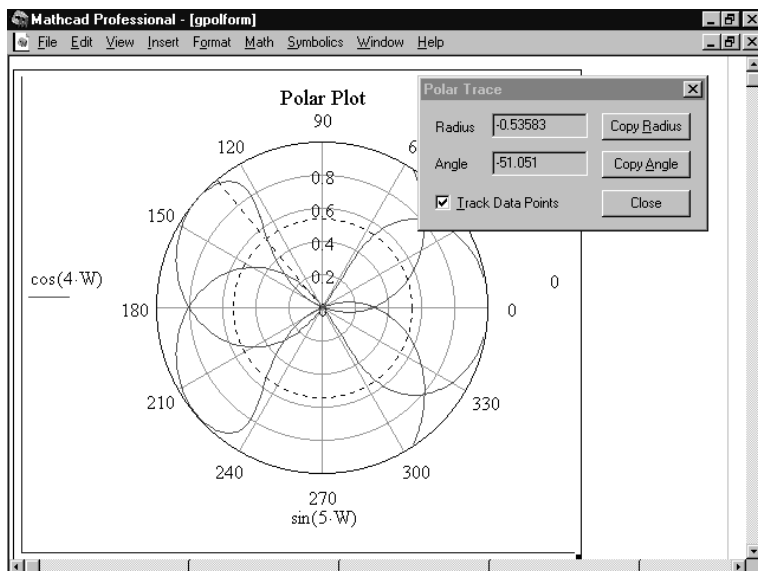


Рис. 5.26. Трассировка графика в полярной системе координат

Трассировка в данном случае выполняется вращательными движениями мыши и перемещением ее указателя по кругу. На графике появляются радиальный отрезок пунктирной прямой и пунктирная окружность. Отрезок прямой задает угол радиус-вектора, а окружность — положение его конца, то есть расстояние от центра координатной системы до точки пересечения вращающегося отрезка с окружностью. Координаты (угол и расстояние) отображаются в окне трассировки.

Трассировка возможна как без слежения за кривой рисунка, так и со слежением. Для осуществления слежения надо установить флажок Track Data Points (Трассировка по точкам данных). С помощью кнопок Copy Angle (Копировать угол) и Copy Radius (Копировать радиус) положение конца радиус-вектора текущей точки (значения угла и длины) можно скопировать в буфер обмена. Затем с помощью команды Paste (Вставка) меню Edit (Правка) можно, если это требуется, перенести скопированное значение в математическое выражение.

5.6.3. Просмотр участков двумерных графиков в полярной системе координат

Команда Format (Формат) ▶ Graph (График) ▶ Zoom (Масштаб) позволяет просматривать выделенную часть графика в полярной системе координат. Выделение производится изменением радиуса выделяющей пунктирной окружности (рис. 5.27).

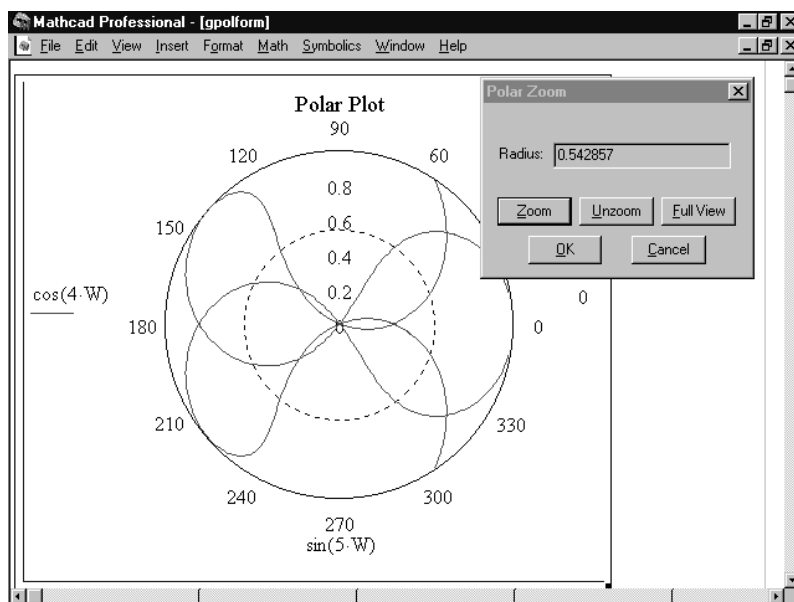


Рис. 5.27. Выделение части графика в полярной системе координат

Щелкнув на кнопке Zoom в открывшемся диалоговом окне, можно получить график выделенной части с увеличением (рис. 5.28).

Назначение остальных кнопок окна просмотра то же, что и у окна просмотра графиков в декартовой системе координат (см. подраздел «Просмотр участков двумерных графиков» в разделе «Форматирование двумерных графиков в прямоугольной системе координат»).

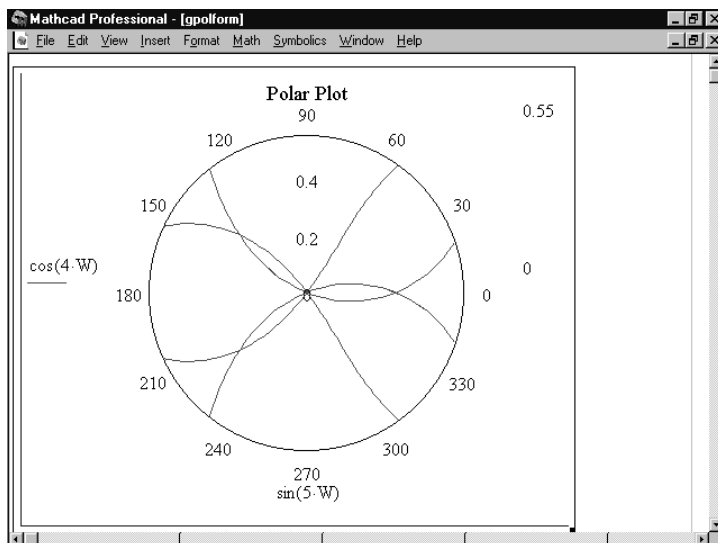


Рис. 5.28. Просмотр части графика в полярной системе координат

5.7. Форматирование трехмерных графиков

5.7.1. Окно установки форматов трехмерных графиков

Команда 3D Plot (Трехмерный график) подменю Graph (График) меню Format (Формат) служит для установки форматов представления трехмерных графиков. Она вызывает появление окна форматирования, показанного на рис. 5.29.



Рис. 5.29. Окно форматирования трехмерного графика

Как можно видеть, это окно имеет ряд вкладок:

- General (Общие) — общие параметры изображения;
- Axes (Оси) — параметры координатных осей (тип, толщина и цвет линий осей, число отметок, их нумерация, масштаб и др.);
- Appearance (Вид) — параметры отображения графика (цвет линий и тип точек, используемых при построении фигур и поверхностей);
- Backplanes (Грани) — параметры граней;
- Lighting (Освещение) — параметры условий и схемы освещения;
- Special (Специальные) — специальные параметры (контурные линии, столбцы, интерполяция по цвету и др.);
- Title (Заголовок) — титульные надписи и их параметры;
- Advanced (Дополнительно) — дополнительные параметры (перспектива, световые эффекты, качество печати и др.);
- QuickPlot Data (Быстрое построение графика по данным) — параметры быстрого построения графиков.

Это окно может быть вызвано и другими способами — двойным щелчком на трехмерном графике, либо из контекстного меню.

5.7.2. Общие параметры трехмерных графиков

Рассмотрим вкладку General (см. рис. 5.29), предназначенную для задания самых важных параметров: углах обзора фигуры, стиле осей и внешнем оформлении графика. Наиболее важная группа параметров расположена внизу окна — это переключатели для выбора типа трехмерного графика:

- Surface Plot (График поверхности);
- Contour Plot (Контурный график);
- Data Points (Точки данных);
- Vector Field Plot (Векторное поле);
- Bar Plot (Гистограмма);
- Patch Plot («Лоскутный» график).

Под переключателями типа фигур имеются традиционные кнопки: OK, Cancel (Отмена), Apply (Применить) и Help (Справка).

- В группе View (Вид) представлены следующие поля:
 - Rotation (Поворот) — угол поворота (от 0 до 360 градусов);
 - Tilt (Наклон) — угол наклона (от 0 до 180 градусов);
 - Twist (Вращение) — угол вращения (от 0 до 360 градусов);
 - Zoom (Масштаб) — относительный размер (по умолчанию 1).
- Группа параметров Axes Style (Стиль осей) позволяет задать стиль отображения осей:
 - переключатель Perimeter (Периметр) — по периметру;
 - переключатель Corner (Угол) — в углу;
 - переключатель None (Отсутствуют) — без вывода осей;
 - флажок Equal Scales (Равные масштабы) задает равные масштабы по всем осям.

- Группа флажков Frames (Обрамление) позволяет задать параметры обрамления:
 - Show Border (Показать границы) — рамка вокруг графика;
 - Show Box (Показать параллелепипед) — параллелепипед, обрамляющий график.

На рис. 5.30 показан пример изменения общих параметров графика, представленного ранее на рис. 5.29.

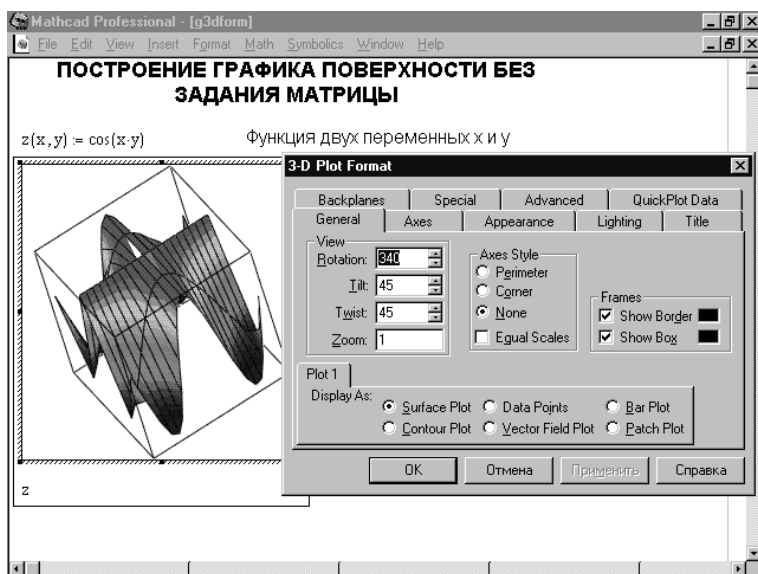


Рис. 5.30. Пример форматирования с помощью вкладки General

На рисунке изменены углы поворота фигуры, ее обрамление, убраны координатные оси.

5.7.3. Параметры отображения осей трехмерных графиков

Вкладка Axes (Оси) служит для задания параметров координатных осей трехмерного графика. Окно форматирования с открытой вкладкой Axes показано на рис. 5.31.

Внутри этой вкладки имеется еще три: X Axis, Y Axis и Z Axis. На этих вкладках задаются идентичные параметры для каждой из координатных осей X, Y и Z.

- Группа параметров Grids (Сетка) позволяет установить формат координатной сетки:
 - флажок Draw Lines (Линии сетки) — вывод линий сетки;
 - флажок Draw Tics (Построение делений) — вывод делений на осях;
 - флажок Auto Grid (Автосетка) — автоматический выбор числа линий;
 - поле Line Color (Цвет линий) — задание цвета линий;
 - поле Number (Число) — задание количества делений;
 - поле Line Weight (Толщина линий) — задание толщины линий сетки.

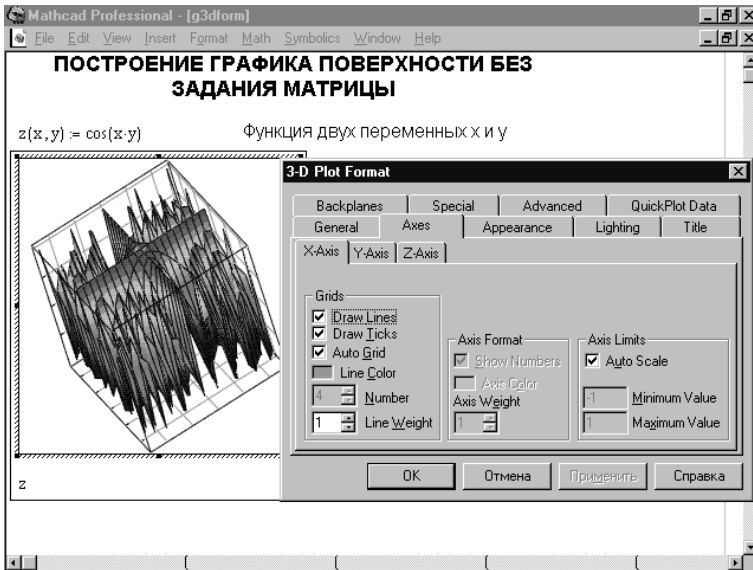


Рис. 5.31. Окно форматирования трехмерных графиков с открытой вкладкой *Axes*

- Группа полей *Axis Format* (Формат осей) позволяет установить формат координатных осей:
 - *Show Numbers* (Показать числа) — оцифровка осей;
 - *Axis Color* (Цвет осей) — задание цвета осей;
 - *Axis Weight* (Толщина линий) — установка толщины линий осей.
- Группа параметров *Axis Limits* (Предельные значения по осям) позволяет задать пределы изменения координат:
 - флажок *Auto Scale* (Автоматическое масштабирование) — автоматическая установка масштаба;
 - поле *Minimum Value* (Минимальное значение) — минимальное значение координаты;
 - поле *Maximum Value* (Максимальное значение) — максимальное значение координаты.

Пример применения параметров форматирования координатных осей дан на рис. 5.31.

5.7.4. Параметры внешнего вида трехмерных графиков

Окно форматирования трехмерных графиков с открытой вкладкой *Appearance* (Вид) представлено на рис. 5.32.

В этом окне имеется три группы параметров:

- *Fill Options* (Параметры закрашивания) — установка параметров окраски поверхностей и контурных линий;
- *Line Options* (Параметры линий) — установка параметров отображения линий и их окраски;

- Point Options (Параметры точек) — установка параметров представления точек разными символами и их окраски.

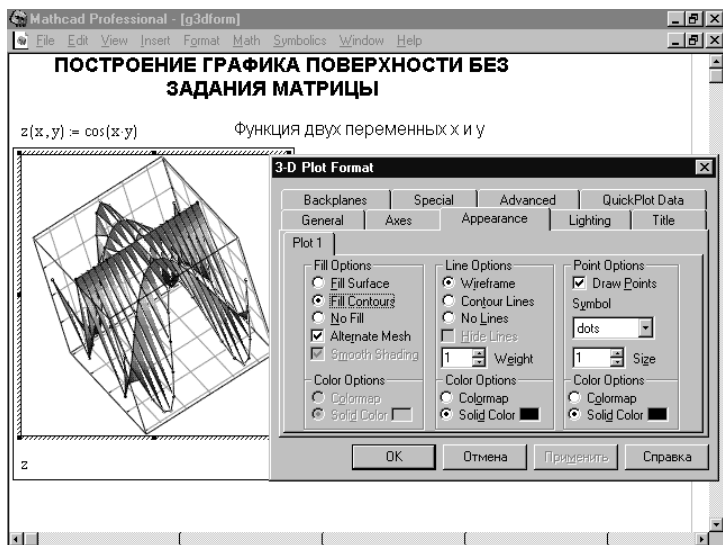


Рис. 5.32. Окно форматирования трехмерных графиков с открытой вкладкой Appearance

В каждой группе имеются переключатели для выбора схемы окраски Colormap (Цветовая карта) или Solid Color (Основной цвет). Рисунок дает представление о том, насколько параметры данной группы влияют на вид графика (за исключением цветовых параметров, влияние которых по черно-белому рисунку оценить трудно).

5.7.5. Параметры надписей трехмерных графиков

Вкладка Title (Заголовок), как показано на рис. 5.33, позволяет расположить на графике введенную в поле Graph Title (Заголовок графика) титульную надпись. Переключатели Above (Сверху) и Below (Снизу) устанавливают местоположение надписи относительно рисунка, переключатель Hide (Скрыть) позволяет отказаться от вывода титульной надписи.

Выведенные таким образом надписи органично вписываются в график, причем они не исчезают при перестройке графиков, что иногда случается с надписями в виде текстовых блоков, наложенных на блоки графики (см. подраздел «Надписи на переднем и заднем плане» в разделе «Форматирование текста»).

5.7.6. Параметры освещения трехмерных графиков

Вкладка Lighting (Освещение), показанная на рис. 5.34, позволяет задать эффект освещения трехмерной поверхности или фигуры. Нередко это придает таким объектам более реалистичный вид.

Группа параметров Lighting вкладки позволяет включить освещение (флажок Wna Lig hting) и выбрать схему освещения. Имеется возможность установить параметры осветителя (в том числе удаленного в бесконечность), учесть диффузию

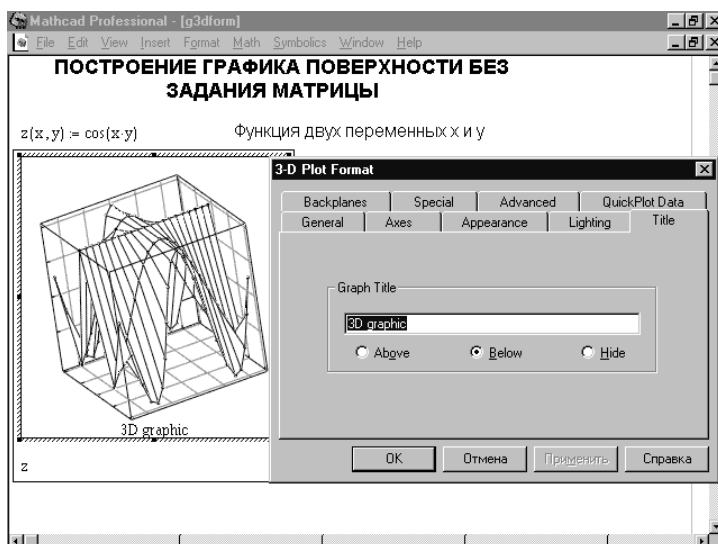


Рис. 5.33. Вкладка задания и вывода титульных надписей для трехмерных графиков

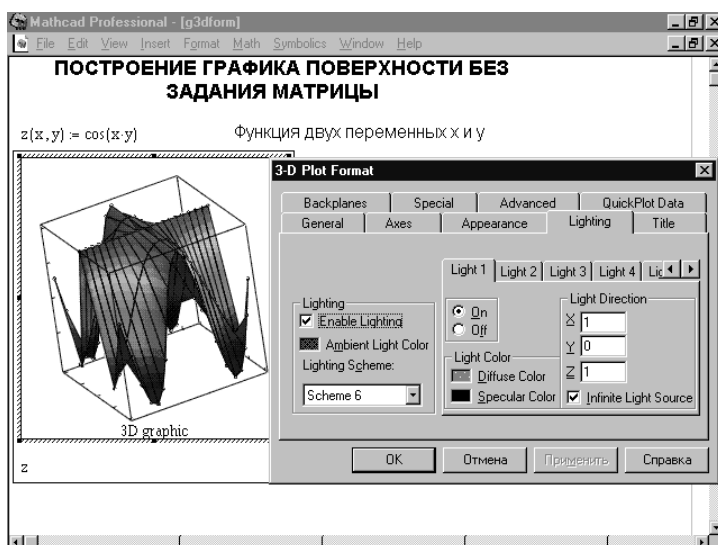


Рис. 5.34. Вкладка Lighting

света, а также выбрать некоторые другие параметры. Рисунок в некоторой степени иллюстрирует влияние параметров этой вкладки.

5.7.7. Параметры форматирования граней

Вкладка Backplanes (Грани) состоит из трех вкладок с параметрами форматирования граней трехмерного рисунка: XY-Backplane, YZ-Backplane и XZ-Backplanes (рис. 5.35).

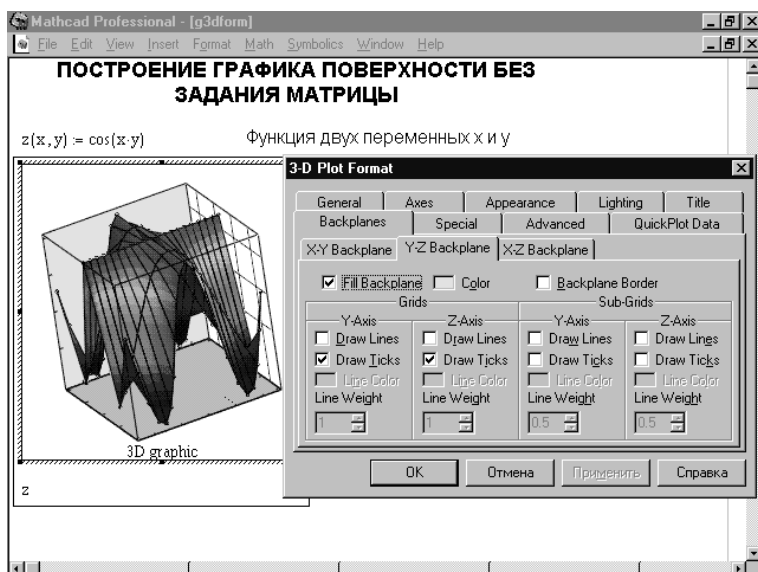


Рис. 5.35. Вкладка форматирования границ

На этих вкладках имеется два основных флажка:

- Fill Backplane (Закрашивание грани) — закрашка соответствующей грани;
- Backplane Border (Обрамление грани) — задание обрамления грани.

Кроме того, имеются группы параметров для задания сетки на гранях: Grid (Сетка) и Subgrid (Подсетка). При установке цветов появляется диалоговое окно выбора цвета. Варианты выбора цвета определяются возможностями применяемого видеоадаптера. Пример форматирования графика с помощью вкладки Backplanes представлен на рисунке.

5.7.8. Специальные эффекты трехмерных графиков

Вкладка Special (Специальные), показанная на рис. 5.36, служит для задания различных специальных эффектов.

На вкладке имеется четыре группы параметров:

- Contour Options (Параметры контурных изображений) — задание необходимости вывода, а также окраски и количества контурных линий;
- Bar Plot Layout (Разметка гистограммы) — управление видом графика в виде столбиковых диаграмм (гистограмм);
- Interpolated Mesh (Интерполяция поверхности) — задание интерполяции поверхности по заданному числу линий по строкам и столбцам (по умолчанию это число равно 21);
- Connectivity (Связывание) — связь функциональной окраски с изменяющимися параметрами (например, координатами).

Параметры этой вкладки являются контекстно-зависимыми, поэтому их изменение возможно только для определенных видов графики. Например, доступ к параметрам группы Bar Plot Layout возможен только для гистограмм, то есть (см. выше подраздел «Общие параметры») только при установке переключателя

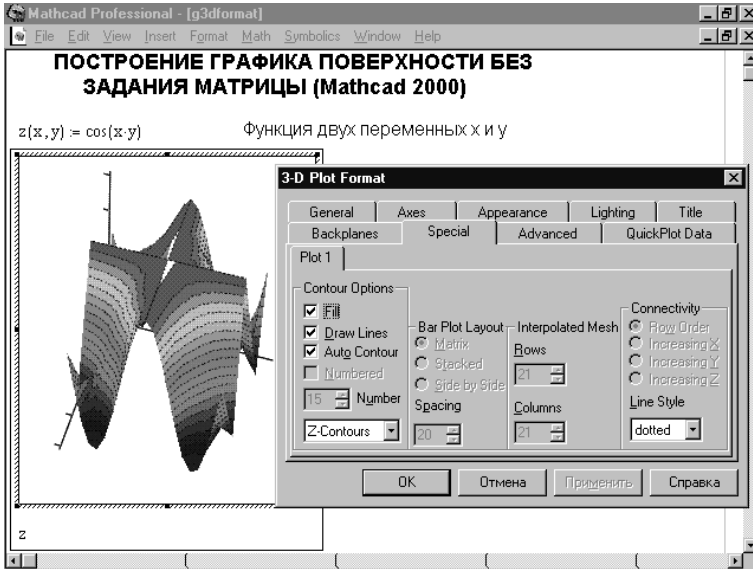


Рис. 5.36. Вкладка установки специальных эффектов

Bar Plot (Гистограмма) на вкладке General (Общие). Пример форматирования с помощью этой вкладки представлен на рисунке.

5.7.9. Дополнительные параметры трехмерных графиков

Вкладка Advanced (Дополнительно), показанная на рис. 5.37, служит для задания дополнительных параметров.

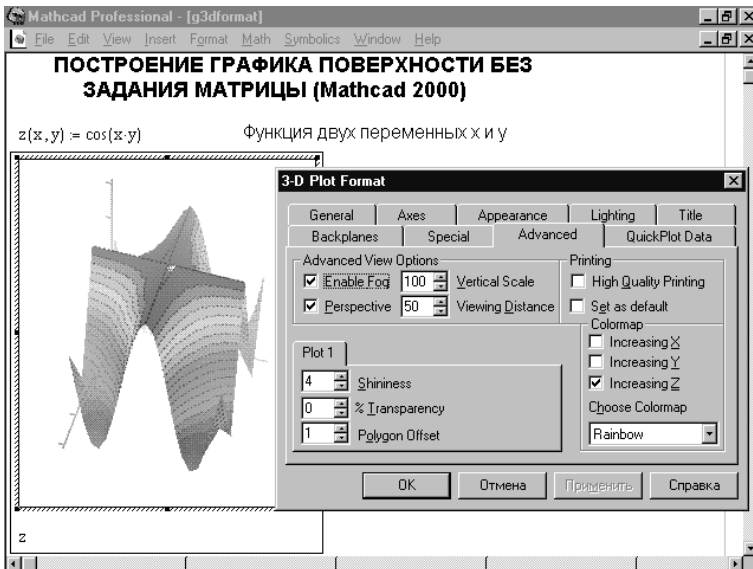


Рис. 5.37. Окно форматирования трехмерного графика с открытой вкладкой Advanced

Наиболее важные параметры собраны в группе Advanced View Options (Дополнительные параметры отображения):

- Enable Fog (Включение дымки) — включение эффекта дымки (тумана);
- Perspective (Перспектива) — отображение поверхности (фигуры) в перспективе;
- Vertical Scale (Вертикальный размер) — установка масштаба по вертикали;
- Viewing Distance (Расстояние до глаза) — установка расстояния, с которого рассматривается фигура.

В группе Printing (Печать) имеется только два флажка, задающие печать с обычным и повышенным качеством.

Группа параметров Colormap (Цветовая карта) позволяет задать параметры функциональной окраски по возрастанию значений координат точек вдоль оси X, Y или Z.

5.7.10. Параметры быстрого построения графиков

Вкладка Quick Plot Data (Быстрое построение графика по данным), показанная на рис. 5.38, имеется только в Mathcad 2000/2001. Она позволяет настроить основные параметры для быстрого построения трехмерных графиков без задания матриц аппликат поверхностей.



Рис. 5.38. Окно форматирования трехмерных графиков с открытой вкладкой Quick Plot Data

Эта вкладка содержит три группы параметров:

- Range 1 (Пределы 1) — задание пределов по одному параметру;
- Range 2 (Пределы 2) — задание пределов по другому параметру;
- Coordinate System (Система координат) — выбор одной из трех систем координат.

Пределы изменения параметров задаются от начального значения (поле Start) до конечного (поле End). Также задается число линий, которыми представлена поверхность (поле со счетчиком # of Grids). Возможен выбор одной из трех систем координат: Cartesian (Прямоугольная), Spherical (Сферическая) и Cylindrical (Цилиндрическая).

Обратите внимание, сколь необычно выглядит исходная фигура (см. рис. 5.29) в сферической (рис. 5.38) или цилиндрической (рис. 5.39) системе координат.

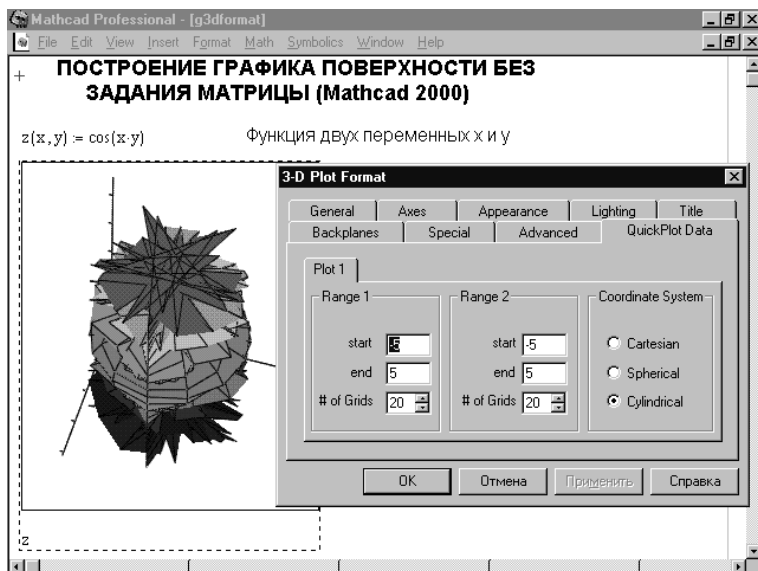


Рис. 5.39. Фигура в цилиндрической системе координат

Две дополнительные системы координат расширяют возможности графики Mathcad. Например, чтобы построить сферу в сферической системе координат, достаточно задать исходное выражение $z(x, y) = R$, где R — радиус сферы. Рис. 5.40 иллюстрирует построение сферы единичного радиуса.

На рис. 5.41 представлен еще один пример на построение графика функции в цилиндрической системе координат. На этот раз строится график функции $z(x, y) = xy$.

Более мощные математические системы, например Maple, имеют возможность построения трехмерных графиков в гораздо большем числе систем координат. Однако надо отметить, что, используя известные формулы преобразования координат из одной системы в другую, в Mathcad можно выполнить построение графиков и в других системах координат, кроме тех, которые указаны на вкладке Quick Plot Data (Быстрое построение графика по данным).

5.8. Форматирование электронных документов

Mathcad позволяет готовить электронные документы превосходного качества. Это могут быть просто хорошо выглядящие статьи и отчеты, уроки по различным дисциплинам и, наконец, электронные книги. Как уже было показано (см. по-

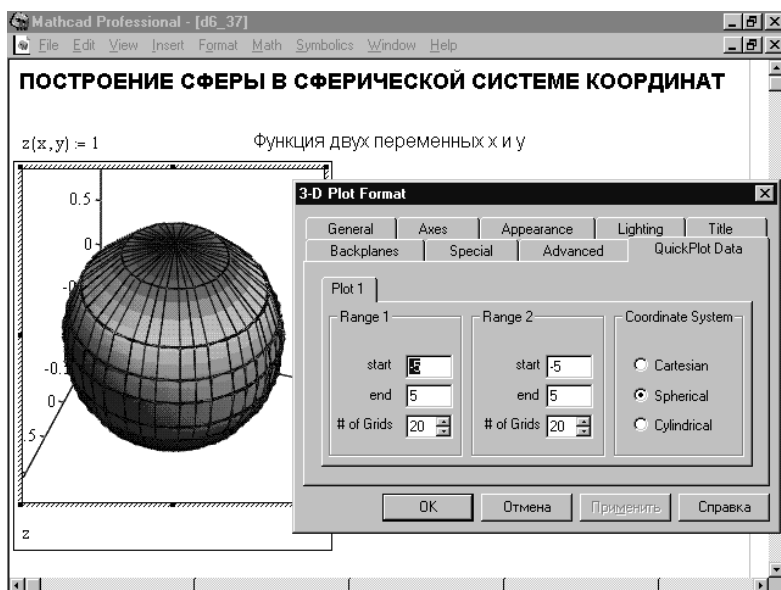


Рис. 5.40. Построение сферы единичного радиуса

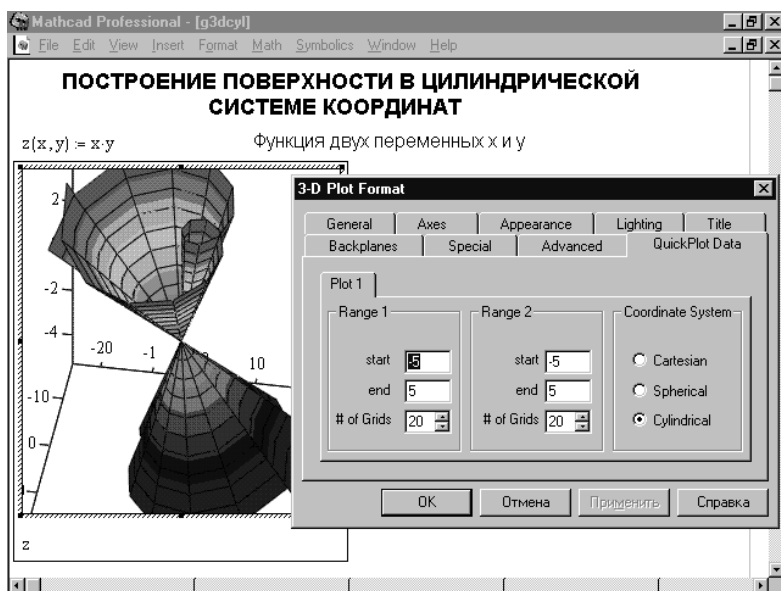


Рис. 5.41. Построение график функции $z(x, y) = xy$

дразделы «Вставка гипертекстовой ссылки» и «Вставка гипермедиассылки» в разделе «Вставка неграфических шаблонов» главы 4), в них могут использоваться гиперссылки — гипертекстовые и гипермедиа. С их помощью можно обращаться к другим документам, рисункам, видеофильмам и т. д. В этом разделе мы изучим некоторые дополнительные возможности Mathcad в области создания электронных документов.

5.8.1. Форматирование цветов

Команды для изменения параметров цвета собраны в подменю Color (Цвет) меню Format (Формат):

- Background (Фон) — цвет фона окна;
- Highlight (Выделение) — цвет выделяемых выражений;
- Annotation Color (Цвет комментариев) — цвет вводимых в график комментариев.

Любая из этих команд выводит стандартное диалоговое окно, показанное ранее на рис. 5.14. Каждое такое окно обладает строкой заголовка с названием соответствующей операции. Помимо перечисленных команд есть еще две, которые доступны только в том случае, если видеосистема компьютера поддерживает 256 цветов и более:

- Use Default Palette (Использовать палитру по умолчанию) — задание исходной цветовой палитры;
- Optimize Palette (Оптимизировать палитру) — оптимизация цветовой палитры для растровых рисунков.

Если в документе отсутствуют рисунки, то при выборе последней команды выводится сообщение, представленное на рис. 5.42.



Рис. 5.42. Сообщение об отсутствии объекта для оптимизации цветовой палитры

Подменю Color открывает доступ к командам цветового оформления документов, позволяющим придать им наглядный и эстетически приятный вид, удовлетворяющий самым изысканным запросам. Однако если предполагается печать документов монохромным принтером или их просмотр на монохромном дисплее, то к цветовым выделениям надо относиться с осторожностью.

5.8.2. Разделение областей документов

Как уже отмечалось, области различных объектов могут налагаться в документах друг на друга (рис. 5.43). Это может не только испортить вид документа, но и привести к возникновению ошибочных ситуаций из-за нарушения требуемой последовательности обработки блоков.

Перемещение блоков вручную в сложном, плотно заполненном блоками документе весьма трудоемко и неудобно. Для решения проблемы в меню Format (Формат) имеется команда Separate Regions (Разделение областей), обеспечивающая автоматическое разделение перекрывающихся областей. Эту команду можно ввести также нажатием комбинации клавиш Ctrl+S. На рис. 5.44 показано действие команды Separate Regions на документ, представленный на рис. 5.43.

К сожалению, команда Separate Regions действует довольно грубо, что прекрасно видно на рисунке. Подчас возможно даже нарушение последовательности

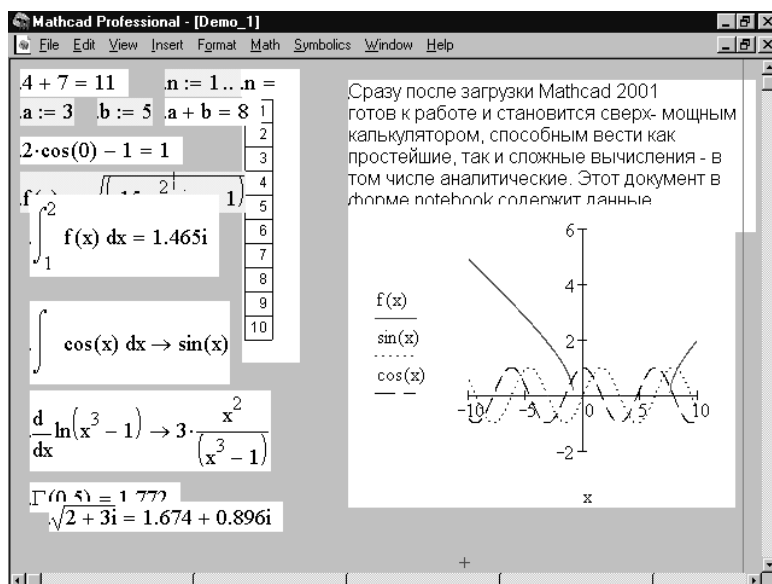


Рис. 5.43. Документ с перекрывающимися областями

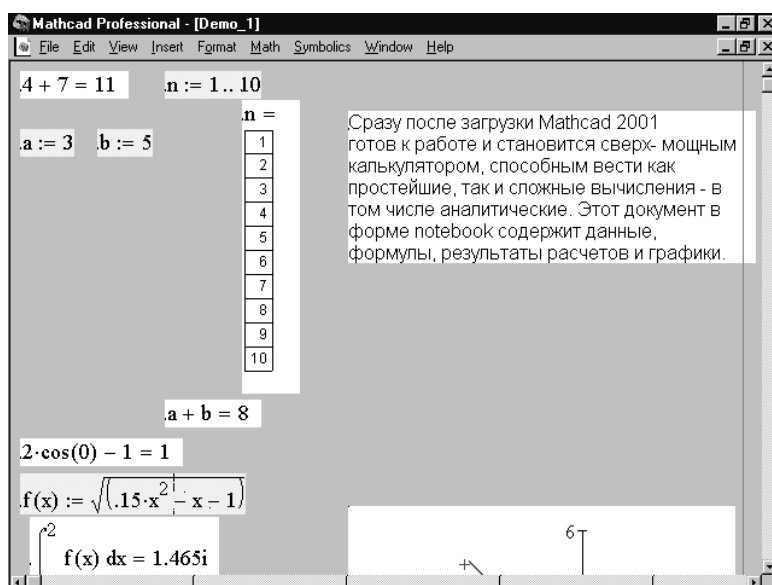


Рис. 5.44. Документ после выполнения команды *Separate Regions*

обработки блоков, поэтому необходимость в ручной коррекции документа, как правило, остается.

5.8.3. Выравнивание областей

Команды подменю *Align Regions* (Выравнивание областей) меню *Format* (Формат) служат для специального выравнивания областей:

- Across (По горизонтали) — выровнять выделенные области вдоль горизонтальной линии, расположенной посередине между верхними краями самой верхней и самой нижней из выделенных областей;
- Down (Вниз) — выровнять выделенные области вдоль вертикальной линии, расположенной посередине между левыми краями самой правой и самой левой из выделенных областей.

Эти операции применяются как элементы комбинированного форматирования документов. Блоки, подлежащие выравниванию с помощью этих команд, должны быть предварительно выделены. На рис. 5.45 показан фрагмент документа с произвольным расположением трех блоков после их выделения.

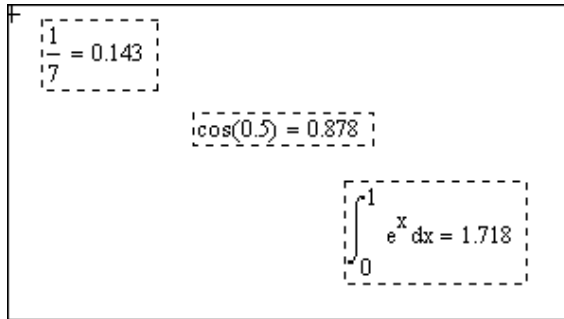


Рис. 5.45. Пример документа с произвольно расположенными блоками

Чтобы расположить блоки рядом друг с другом, надо использовать команду Across. Результат выполнения этой команды представлен на рис. 5.46. Все блоки оказались расположенными рядом друг с другом и выровненными по средней линии.

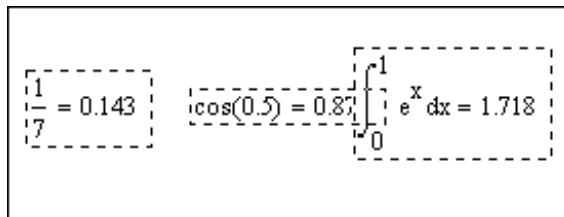


Рис. 5.46. Пример выравнивания блоков по горизонтали

Эти команды надо применять осторожно, поскольку при неудачной операции блоки могут перекрываться (небольшое перекрытие видно и на рисунке). Если в данном примере применить команду Down, то это приведет к полному наложению блоков. Вообще говоря, при подготовке документов полезно сразу избегать наложения блоков друг на друга.

5.8.4. Блокировка области

В ряде случаев желательно создание в документе некоторых заблокированных (то есть недоступных для редактирования) областей. Это может потребоваться, например, в обучающих программах, где в такие области можно помещать примеры, которые запрещено изменять учащимся.

Для задания будущей закрытой области используется команда Area (Область) меню Insert (Вставка), которая выводит на экран две характерные горизонтальные линии со значками в виде черных треугольников, направленных друг к другу (рис. 5.47). Линии могут выделяться (см. верхнюю линию на рисунке) и перемещаться, например, для расширения или сужения области. В область можно вводить математические выражения, которые затем будут блокироваться.

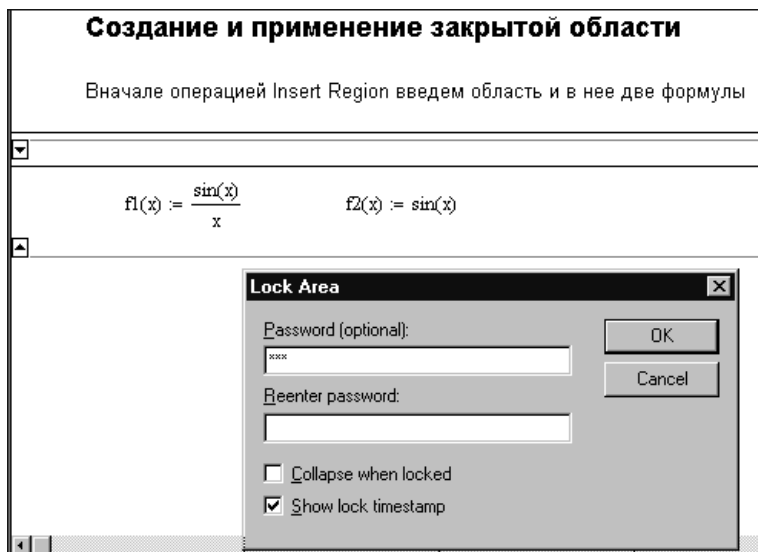


Рис. 5.47. Подготовка к блокировке области

В подменю Area (Область) меню Format (Формат) для блокировки и разблокировки областей предназначены следующие команды:

- Lock (Блокировка области) — заблокировать область;
- Unlock (Разблокировка области) — открыть заблокированную область.

После выбора команды Lock появляется окно (показано в нижней части рисунка) с предложением ввести и подтвердить пароль.

Если не ввести пароль, то область будет заблокирована навсегда. Ввод пароля означает, что в дальнейшем область можно будет сделать открытой. Обычно стоит предусмотреть такую возможность.

ПРИМЕЧАНИЕ При вводе пароля, как обычно, вместо символов нажимаемых клавиш на экран выводятся звездочки. Пароль надо выбирать такой, чтобы его можно было запомнить (неплохо записать его в свою записную книжку).

После ввода пароля и его подтверждения выделенная область блокируется. При этом на линиях, обозначающих заблокированную область, появляются значки с изображением замка и возможность модернизации области прекращается (рис. 5.48).

Как видно на рисунке, рядом с изображениями замков выводятся данные о времени создания заблокированной области. Введенные в заблокированную область определения (например, показанные на рисунке функции пользователя f_1 и f_2) можно использовать в нижеследующих объектах документа (например для по-

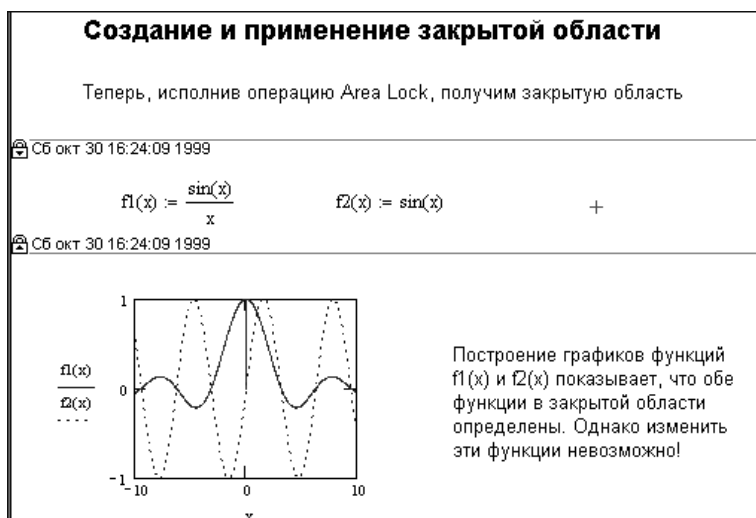


Рис. 5.48. Фрагмент документа с заблокированной областью

строения графиков функций). Однако если попытаться модифицировать введенное в заблокированную область математическое выражение или перетащить его на другое место, то из этого просто ничего не выйдет. Таким образом, доступ в заблокированную область невозможен.

5.8.5. Открытие заблокированной области

Команда Unlock (Разблокировка области) позволяет в случае необходимости открыть заблокированную область. После выбора этой команды появляется окно ввода пароля (рис. 5.49).



Рис. 5.49. Открытие заблокированной области

Если в нем ввести ранее заданный пароль, то замки в ограничителях области «откроются» и она станет открытой. Выделив ограничители, их можно удалить, например, нажатием клавиши F3 или командой Delete (Удалить) меню Edit (Правка). Ограничители задаются парой и стираются одновременно.

5.8.6. Скрытие области

Для создания полноценных электронных книг и учебников Mathcad 2001 имеет еще одну новую возможность — возможность создания скрытых областей. Подготовка к скрытию области выполняется так же, как к ее блокировке (см. выше подраздел «Блокировка области»). После создания ограничителей и заполнения области объектами получается документ, фрагмент которого показан на рис. 5.50.



Рис. 5.50. Фрагмент исходного документа для создания скрытой области

Теперь вместо команды блокировки области в подменю Area (Область) меню Format (Формат) выбирается команда Collapse (Скрытие области). При этом вся область вместе с содержащимися в ней объектами «сжимается» в одну горизонтальную линию (рис. 5.51).

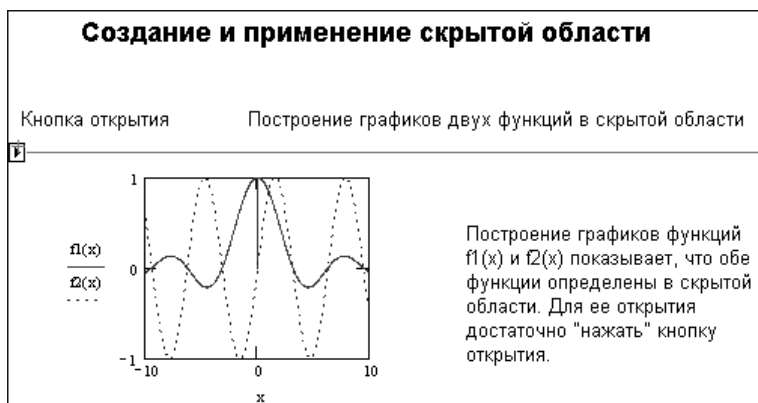


Рис. 5.51. Фрагмент документа со скрытой областью

Несмотря на то, что объекты в скрытой области не видны, они вполне функциональны. В этом убеждает рисунок, на котором строится график функций f_1 и f_2 , определения которых находятся в скрытой области.

5.8.7. Открытие скрытой области

Для открытия скрытой области достаточно щелкнуть на значке в начале линии, представляющей скрытую область, или выбрать команду Expand (Раскрытие области) в подменю Area (Область) меню Format (Формат). Документ при этом возвращается к исходному виду (см. рис. 5.50).

Таким образом, можно скрывать целые разделы документов, оставляя только их заголовки. Если пользователю надо увидеть соответствующий раздел, достаточно щелкнуть мышью на значке соответствующей скрытой области.

5.8.8. Создание колонтитулов

При печати документов нередко требуется ввести в заголовок или в нижнюю строку *каждой страницы документа* некоторую служебную информацию или надпись, например имя файла, номер страницы, дату создания и др. Такие надписи называют *колонтитулами* (верхними и нижними).

Эта возможность реализуется командой Headers/Footers (Колонтитулы) меню Format (Формат). Команда выводит диалоговое окно, показанное на рис. 5.52. Это окно содержит две вкладки: Header (Верхний колонтитул) и Footer (Нижний колонтитул).

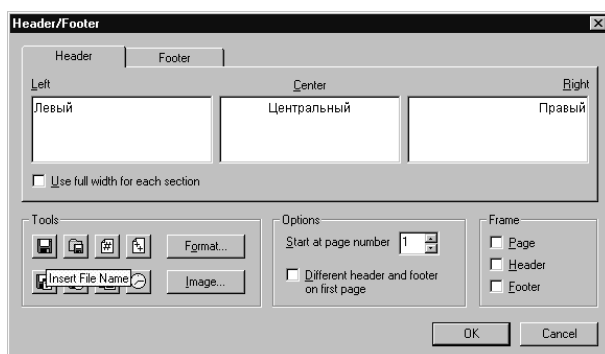


Рис. 5.52. Диалоговое окно установки колонтитулов с открытой вкладкой Header

Окно на рисунке показано с открытой вкладкой Header. Эта вкладка содержит три поля для ввода надписей в верхней части каждой страницы: Left (Слева), Center (По центру) и Right (Справа). Соответственно, и введенные надписи будут печататься в верхнем колонтитуле с выравниванием влево, по центру или вправо.

Имеется ряд кнопок для автоматического занесения в колонтитулы некоторых данных:

- Insert File Name (Вставка имени файла);
- Insert File Patch (Вставка пути к файлу);
- Insert Page Number (Вставка номера страницы);
- Insert Number Pages (Вставка числа страниц);
- Insert Date Last Save (Вставка даты последнего сохранения);
- Insert Time Last Save (Вставка времени последнего сохранения);
- Insert Date (Вставка даты);
- Insert Time (Вставка времени).

Эти данные вставляются в виде полей, но при просмотре и печати страницы отображаются в явном виде. Кнопка Format (Формат) дает возможность выбора подходящего шрифта для текста, вводимого в область колонтитула. В качестве колонтитула можно также выбрать рисунок из файла. Для этого служит кнопка Image...

ge (Рисунок), которая открывает обычное окно загрузки и поиска необходимого файла в файловой системе.

Вторая вкладка — Footer — служит для настройки нижнего колонтитула. Она содержит те же элементы управления и те же инструментальные средства, что и вкладка для настройки верхнего колонтитула.

Колонтитулы в окне редактирования документов не видны. Они появляются только в режиме предварительного просмотра (рис. 5.53) и на распечатках документов.

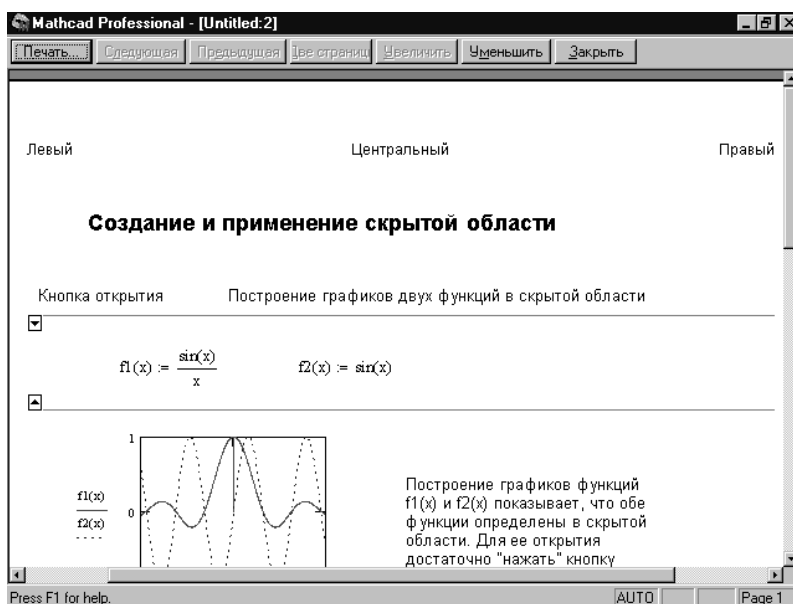


Рис. 5.53. Документ с колонтитулами в режиме предварительного просмотра

Вводом колонтитулов завершается набор средств Mathcad, которые позволяют готовить электронные документы в форме статей, отчетов, диссертаций и книг с высоким качеством любых объектов документов — текстов, математических формул, таблиц, графиков и рисунков. Впрочем, в части подготовки текстов преимущество за текстовыми процессорами класса Word все еще остается.

5.8.9. Замена латинских букв на греческие

К специфическим операциям форматирования можно отнести замену латинских букв на греческие. Как уже отмечалось, греческие буквы можно вводить непосредственно из палитры греческих букв. Однако для этого надо пользоваться мышью, что не всегда удобно.

Есть и второй, порой более быстрый, путь ввода греческих букв — достаточно ввести ассоциированную с греческой латинскую букву и нажать клавиши Ctrl+G. В результате латинская буква заменится греческой. На рис. 5.54 представлена таблица соответствия латинских и греческих букв.

Большинство греческих букв легко ассоциируются с латинскими. Помните также, что есть комбинации клавиш для ввода математических спецсимволов. Так, нажатие клавиш Ctrl+P вводит константу «пи», а Ctrl+Z — знак бесконечности.

α	a	η	h	\omicron	o	ω	v
β	b	ι	i	π	p	ω	w
χ	c	ψ	j	θ	q	ξ	x
δ	d	κ	k	ρ	r	ψ	y
ϵ	e	λ	l	σ	s	ζ	z
ϕ	f	μ	m	τ	t		
γ	g	ν	n	υ	u		
Δ	A	H	H	O	O	ζ	V
B	B	I	I	П	P	Ω	W
X	C	J	J	Q	Q	Ξ	X
Δ	D	K	K	P	R	Ψ	Y
E	E	Δ	L	Σ	S	Z	Z
Φ	F	M	M	T	T		
Γ	G	N	N	Υ	U		

Рис. 5.54. Таблица соответствия греческих (сверху) и латинских букв

5.9. Форматирование изображений и рисунков

5.9.1. Подготовка к форматированию рисунков

Mathcad позволяет вставлять в документ рисунки, подготовленные с помощью внешних программных средств. Для форматирования таких рисунков служит специальная панель инструментов — Picture (Рисунок). Она предлагает удобный интерактивный способ форматирования изображений.

Проведем следующий эксперимент. Включите в текущую папку ваших документов файл arrow1.bmp (он находится в папке qsheet «быстрых шпаргалок» системы Mathcad 2001). С помощью функции READBMP этот файл можно считать в массив S и затем вывести изображение этого массива (рис. 5.55, слева).

Наведя указатель мыши на рисунок и щелкнув ее левой кнопкой, можно наблюдать (в Windows 98) выделение рисунка и появление панели инструментов Picture, предназначенной для форматирования рисунков.

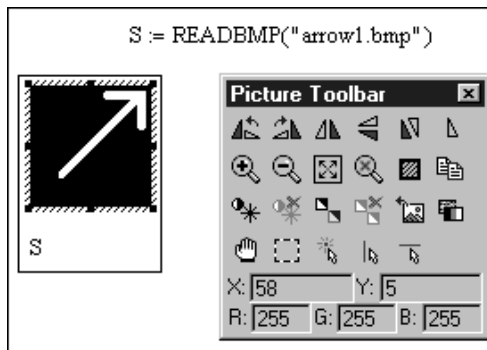


Рис. 5.55. Фрагмент документа с рисунком стрелки (слева) и панелью инструментов Picture

5.9.2. Повороты рисунков

Верхний ряд кнопок панели Picture (Рисунок) служит для поворота изображения (рис. 5.56). Возможны повороты против часовой стрелки, по часовой стрелке, зеркальное отражение вокруг вертикальной оси, вокруг горизонтальной оси и транспонирование изображения. Последняя кнопка отменяет все операции поворота и возвращает исходное изображение.

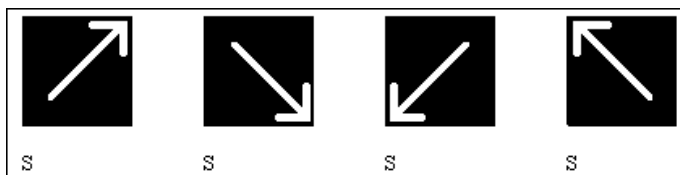


Рис. 5.56. Варианты поворота изображения (стрелки)

5.9.3. Масштабирование рисунков и операции с буфером

Вторая группа кнопок панели Picture (Рисунок) служит для масштабирования рисунков. Это кнопки увеличения или уменьшения размера рисунков, доведение размеров рисунка до размеров окна и сброс операций масштабирования.

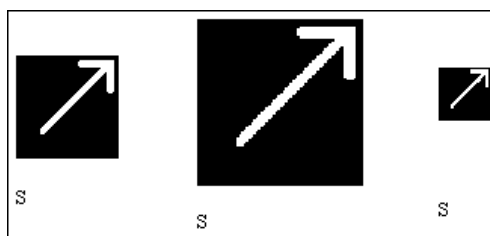


Рис. 5.57. Управление масштабом изображения (стрелки)

Две последние кнопки второй группы служат для работы с буфером обмена операционной системы Windows. Одна кнопка — Select All (Выделить все) — выделяет весь рисунок, а другая — Copy (Копировать) — копирует его в буфер. Разместив указатель мыши в некотором (пустом) месте документа и нажав клавишу F4 или выбрав команду Paste (Вставить) в меню Edit (Правка), можно разместить в этом месте скопированный в буфер рисунок.

5.9.4. Управление передаточной характеристикой по яркости

Третья группа кнопок панели Picture (Рисунок) служит для управления передаточной характеристикой изображений по яркости. На рис. 5.58 показан фрагмент документа с изображением из файла topo.bmp. Под ним представлено окно с передаточной характеристикой изображения по яркости, которое получается при активизации первой кнопки из указанной группы.

В этом окне имеется гистограмма распределения яркости и передаточная характеристика по яркости. Она линейная и имеет участки насыщения. Границы линейной области можно перетаскивать мышью, что позволяет менять яркость и контрастность изображения. Вторая кнопка Picture панели позволяет отказаться

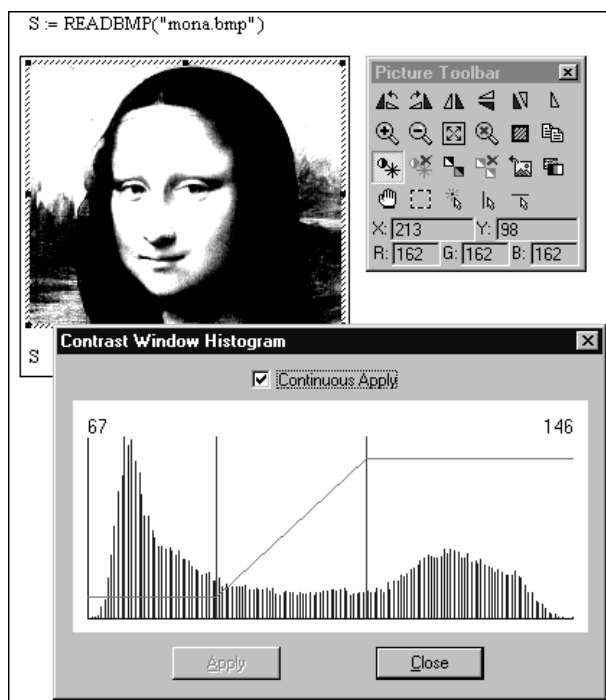


Рис. 5.58. Установка передаточной характеристики по яркости

от сделанных изменений яркости и контрастности. Нелинейная гамма-коррекция изображений не предусмотрена.

Следующие две кнопки этой группы служат для подобной настройки монохромных изображений (рис. 5.59).

В данном случае под гистограммой яркости рисунка предусмотрена шкала оттенков серого цвета.

Две последние кнопки этой группы служат соответственно для отмены всех настроек дисплея и для перестройки изображения.

5.9.5. Управление перемещением и выделением фрагментов изображения

Последняя группа кнопок панели Picture (Рисунок) начинается с кнопки со значком ладони. Эта кнопка позволяет при нажатой левой кнопке мыши плавно перемещать область изображения в его окне. На рис. 5.60 показан результат такого перемещения изображения вправо и вниз. Следующая кнопка (со значком прямоугольника из пунктирных линий) позволяет выделить часть изображения — эта операция также показана на рисунке.

5.9.6. Определение координат пиксела

Последние три кнопки данной группы служат для определения координат пиксела при произвольном положении указателя мыши, при положении указателя мыши на вертикали и при положении указателя мыши на горизонтали (рис. 5.61). Координаты текущей точки отображаются в полях внизу панели Picture (Рисунок).

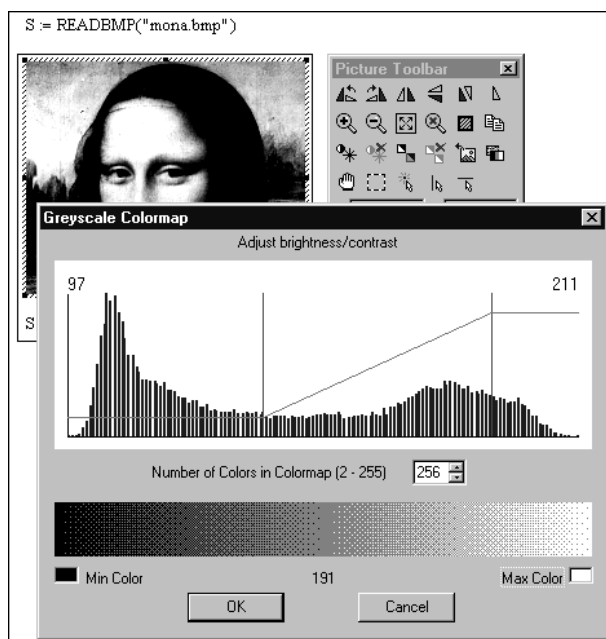


Рис. 5.59. Настройка монохромного изображения

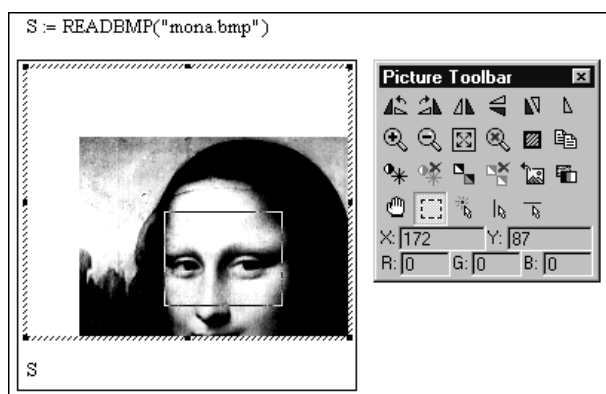


Рис. 5.60. Перемещение изображения и выделение его части

Соответствующая линия (горизонтальная или вертикальная) перемещается при нажатой левой кнопке мыши и фиксируется при ее отпуске. Перемещение указателя мыши после этого задает недостающую координату. Координаты текущей точки отображаются в полях внизу панели Picture. На рисунке показан случай определения горизонтальной координаты кончика носа Моны Лизы.

5.9.7. Использование для форматирования рисунков контекстного меню

Для форматирования рисунков наряду с панелью инструментов Picture (Рисунок) можно использовать контекстное меню, которое появляется при щелчке на

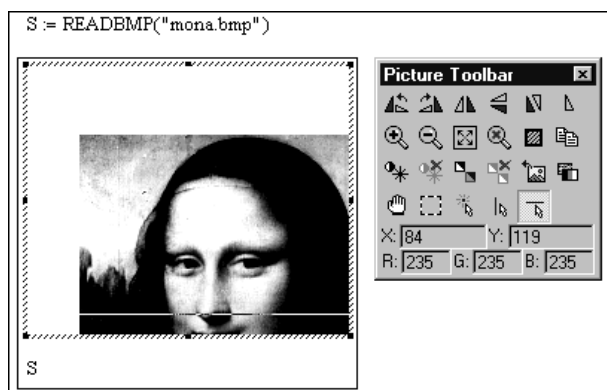


Рис. 5.61. Определение координат пиксела на заданной горизонтали

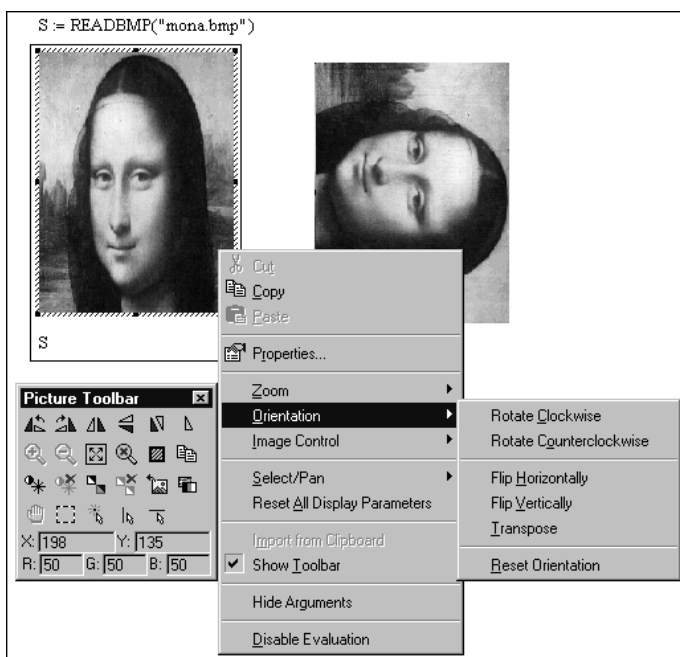


Рис. 5.62. Контекстное меню изображения

изображении правой кнопкой мыши. На рис. 5.62 показано это меню вместе с панелью Picture.

Нетрудно заметить, что контекстное меню предлагает практически тот же набор команд форматирования изображений, что панель инструментов Picture.

В целом возможности форматирования рисунков, особенно цветных, в системе Mathcad достаточно скромны, хотя и позволяют делать наиболее распространенные операции. Очевидно, что для получения рисунков предельно высокого качества их обработку стоит осуществлять в специализированных графических редакторах и затем уже импортировать их в документы системы Mathcad.

Контекстное меню можно использовать и для работы с трехмерными графиками. Пример этого показан на рис. 5.63. В качестве примера показана работа с над-

писью, которая частично находится сзади графика. Командой Bring to Front можно вывести надпись поверх рисунка. Это полезный прием для создания графиков с поясняющими надписями.

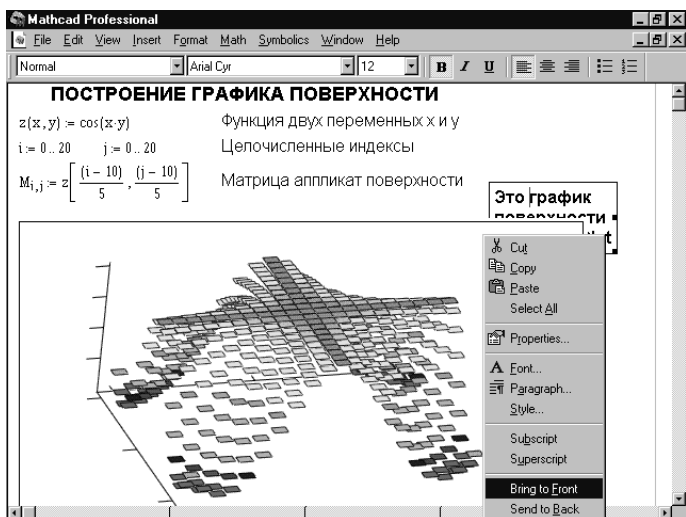


Рис. 5.63. Контекстное меню трехмерного графика

Глава 6. Обычные вычисления и математический анализ

В этой главе мы познакомимся с поддерживаемыми в системе операторами и функциями, константами и переменными, специальными математическими знаками и единицами измерения размерных величин, матрицами и векторами, в общем, со всем тем, что обеспечивает эффективное выполнение обычных и самых массовых вычислений в системе Mathcad. И прежде всего вычислений, относящихся к математическому анализу.

6.1. Входной язык системы Mathcad

6.1.1. Алфавит входного языка системы Mathcad

Общение с пользователем системы Mathcad осуществляется с помощью *математически и визуально ориентированного* входного языка общения с системой, который является типичным языком *визуального программирования*. Большинство операторов и функций входного языка знакомо пользователю по курсу математики. Благодаря этому большая часть расчетов в Mathcad не требует программирования в общепринятом смысле этого слова. Тем не менее все описанные в этой главе вычислительные средства могут использоваться как сами по себе, так и в составе программных модулей системы Mathcad.

Алфавит входного языка — это совокупность символов и слов, которые используются при задании команд и функций, необходимых для решения интересующего пользователя класса задач. Алфавит системы Mathcad содержит:

- малые и большие латинские буквы;
- малые и большие греческие буквы;
- арабские цифры от 0 до 9;
- системные переменные;
- математические операторы;
- имена встроенных функций;
- спецзнаки;
- малые и большие буквы кириллицы (при работе с русифицированными документами).

К укрупненным элементам языка относятся типы данных, операторы, встроенные функции, функции пользователя, процедуры и управляющие структуры. Все эти элементы присущи входному языку системы Mathcad, как и любому другому языку программирования. Кроме того, все знаки, которые можно ввести с помощью палитр математических знаков, также относятся к алфавиту Mathcad.

6.1.2. Применение чисел и констант

К важнейшим *типам данных* в системе Mathcad относятся константы, обычные и системные переменные, массивы (векторы и матрицы) и данные файлового типа.

Константы — имеющие уникальные имена объекты, хранящие некоторые значения, которые определяются в процессе загрузки системы. Mathcad поддерживает следующие типы констант:

- целочисленные константы (0, 1, 23, -45 и т. д.);
- вещественные числа с мантиссой и порядком (например, $12.3 \cdot 10^{-5}$ — десятичная константа с мантиссой 12.3 и порядком -5);
- символьные константы, имеющие численные значение (например, π и e);
- восьмеричные числа (идентифицируются латинской буквой O — от слова *octal* — восьмеричное);
- шестнадцатеричные числа, образуемые цифрами 0, 1, 2, ..., A, B, C, D, E, F (имеют в конце отличительный признак в виде буквы h или H, от слова *hexagonal* — шестнадцатеричное; если число начинается с буквы, то перед ней надо ввести ноль);
- комплексные числа $Z = \text{Re}Z + i \cdot \text{Im}Z$ или $Z = \text{Re}Z + j \cdot \text{Im}Z$, где $\text{Re}Z$ — действительная часть комплексного числа Z , $\text{Im}Z$ — его мнимая часть, а символы i или j обозначают мнимую единицу, то есть корень квадратный из -1 ;
- системные константы, хранящие определенные параметры системы (например погрешность TOL, нижнее значение индексов в массивах ORIGIN и др.);
- строковые константы — любые цепочки символов, заключенные в кавычки, например, "string", "2 + 3" (арифметические выражения в строковых константах рассматриваются как текст и не вычисляются);
- единицы измерения физических величин;
- константу NaN, означающую отсутствие физического результата (есть только Mathcad 12).

Примеры применения некоторых констант приведены ниже (напоминаем, что знак = является оператором вывода):

$$\begin{array}{ll} \pi = 3.142 & e = 2.718 \\ \text{TOL} = 1 \times 10^{-3} & \text{ORIGIN} = 0 \\ z := 2 + 3i & z = 2 + 3i \end{array}$$

Для проведения физических расчетов в системе Mathcad может применяться особый вид констант — *единицы измерения размерных величин*. Помимо своего числового значения, они характеризуются еще и указанием на то, к какой физической величине они относятся. При необходимости Mathcad выполняет физические расчеты с соответствующим преобразованием размерных величин. В Mathcad 2001i/11/12 предпринят ряд мер для более корректной и быстрой работы с размерными переменными.

Уместно сразу отметить, что знак умножения * при выводе числа на экран меняется на привычную математикам точку в середине строки, а операция возведения в степень (с применением спецзнака ^) отображается путем представления порядка в виде надстрочного символа. Знак деления / заменяется горизонтальной чертой. Десятичные числа имеют основание 10. Диапазон их возможных значений лежит в пределах от 10^{307} до 10^{-307} (это машинная бесконечность и машинный ноль).

6.1.3 Новая константа NaN в Mathcad 12

Особая константа NaN, введенная в Mathcad 12, означает отсутствие физического результата, т. е. результат является ничем. Примеры применения этой константы даны ниже:

```

a := 1
b := NaN
IsNaN(a) = 0
IsNaN(b) = 1

```

Функция $\text{IsNaN}(x)$ контролирует x на значение NaN , если x имеет значение NaN , то возвращается логическая 1, иначе возвращается логический 0 (см. приведенные выше примеры). Константа NaN часто используется как заполнитель элементов векторов и матриц, которым не были присвоены численные, символьные или строковые значения.

6.1.4. Переменные

Переменные также являются имеющими уникальные имена объектами. Однако, в отличие от констант, они вначале не определены, а после определения могут принимать любые значения в пределах своего типа. Имена констант, переменных и иных объектов называют *идентификаторами*. В системе Mathcad тип переменной определяется ее значением. Так, переменные могут быть числовыми, строковыми, символьными и так далее, поэтому тип переменной предварительно не задается.

Имена (идентификаторы) могут иметь практически любую длину, и в них могут входить любые латинские и греческие буквы, а также цифры. Однако начинаться идентификатор может только с буквы, например: x , x_1 , α , $X_coordinate$. Кроме того, в идентификаторах недопустимо использование пробелов. Некоторые спецсимволы (например, знак подчеркивания « $_$ ») могут входить в состав идентификаторов, другие (например, знаки арифметических действий) недопустимы, поскольку ведут к неоднозначности идентификации переменной. Значения строковых переменных задаются в кавычках, например: $N := \text{"My_name"}$.

Нельзя использовать в идентификаторах буквы русского языка. Строчные и прописные буквы в идентификаторах различаются. Идентификаторы должны быть уникальными, то есть они не могут совпадать с именами встроенных или определенных пользователем функций. Полезно сверять подозрительные имена со списком всех функций, который можно получить, щелкнув на кнопке $f(x)$ панели инструментов.

Назначение переменных в Mathcad соответствует назначению переменных в математике. Уже само их название говорит о том, что значения переменных могут меняться. Они используются для обобщенного представления данных определенного типа. К примеру, вычисление значения выражения $(2 + 3)$, равного 5, носит весьма частный характер. Куда более общим является вычисление значения выражения $(x + y)$ как суммы значений двух переменных — x и y . В частности, при $x = 1$ и $y = 3$ для $(x + y)$ получим значение 4, но уже при $x = 3$ и $y = 5$ вычисление $(x + y)$ даст уже 8.

Mathcad позволяет задавать и более сложные имена, например $X_coordinate$, V_{cub} и т. д. Длина идентификатора не ограничена. Внутри имен переменных надо избегать знаков, которые могут трактоваться как операторы (например, пробел, $*$, $/$, $+$, $-$ и т. д.). Имена должны начинаться с буквы (допустимо a_2 , но не $2a$ и т. д.).

6.1.5. Операторы и операнды

Операторы — элементы языка, предназначенные для создания математических выражений совместно с данными, именуемыми *операндами*. Это знаки арифметических операций, вычисления сумм и произведений членов ряда, производных, интег-

ралов и т. д. После указания операндов (аргументов соответствующих операторов) операторы становятся исполняемыми программными блоками. Mathcad позволяет задавать и пользовательские операторы. Операторы в Mathcad вводятся с помощью шаблонов, которые, в свою очередь, имеют места ввода для операндов (см. главу 2).

6.1.6. Встроенные функции

Mathcad поддерживает множество *встроенных функций*, то есть функций, определенных в самой системе и готовых к немедленному использованию. Функции обладают свойством возвращать некоторое значение в ответ на обращение к ним по имени с указанием аргумента (или списка аргументов). Они возвращают символьное или числовое значение, вектор или матрицу. В систему встроен ряд элементарных математических функций, например, функция вычисления синуса $\sin(x)$, натурального логарифма $\ln(x)$ и т. д.

Функции вводятся своими именами, принятыми в системе Mathcad. Иногда эти имена несколько отличаются от стандартных обозначений — в основном некоторым сокращением. К примеру, $\arctg(x)$ в Mathcad обозначается как $\operatorname{atan}(x)$. При вводе функции посредством палитры функций появляется шаблон для ввода параметров. Если на имени функции установлен курсор ввода, то нажатие клавиши F1 открывает страницу справки по указанной функции.

Возможно также задание *функций пользователя*, которые создаются самим пользователем. Благодаря функциям пользователя обеспечивается расширение входного языка Mathcad и его адаптация к специфическим задачам пользователя.

Набор функций системы Mathcad может дополняться с помощью корректирующих систему Patch-файлов, загружаемых библиотек функций, пакетов расширений и электронных книг. Их можно приобрести или получить из Интернета. Позже мы вернемся к описанию функций пользователя и опишем их более подробно (см. раздел «Функции пользователя»).

6.1.7. Математические выражения и приоритет операций

Функции, наряду с операторами, могут входить в математические выражения. Например, рассмотрим выражение:

$$Y := 5 \cdot \ln(x) + 1$$

В этом выражении Y — переменная, 1 и 5 — числовые константы, \cdot и $+$ — операторы, $\ln(x)$ — встроенная функция. Заметим, что для вычисления этого выражения перед ним должно быть определено значение x .

Исполнение математических выражений осуществляется с учетом общепринятого приоритета операций. Вначале исполняются встроенные функции, затем логические операторы, далее операции возведения в степень, умножения и деления и последними операции сложения и вычитания. Для изменения приоритета операций используются скобки. Например, $2+3*4=14$ (поскольку вначале вычисляется $3*4$), а $(2+3)*4=20$.

При выполнении символьных операций константы π и e используются только в символьном виде. Это значит, что их численные значения не вычисляются при выводе результатов *символьных* вычислений. К примеру, число $2 \cdot \pi = 6.283\dots$ при использовании знака символьного вычисления \rightarrow выводится как $2 \cdot \pi$, а не как приближенное численное значение 6.283, при использовании же знака $=$ будет получено 6.283.

6.2. Применение переменных

6.2.1. Присваивание переменным значений

Задание значений переменным называется *присваиванием*. В отличие от классической записи операции присваивания в математике, в Mathcad знак равенства = можно использовать как оператор присваивания только один раз, при первом присваивании значения новой переменной любого типа. Основное назначение этого оператора — вывод значения переменной или выражения, после которых и ставится знак равенства. До первого присваивания переменная не определена (не имеет значения) и поэтому знак = срабатывает как оператор присваивания, например:

$$x = 2 \quad y = 3 \quad x + y = 5$$

В последнем случае переменные уже определены, поэтому едва вы успеете ввести знак =, как тут же появится результат сложения x и y — число 5. Mathcad «знает», что поскольку переменные x и y уже определены, то знак = надо использовать для вывода результата.

Основным знаком присваивания является составной знак :=, вводимый двоеточием. Его можно применять как при первом присваивании значения переменной, так и при любой следующей по порядку операции присваивания:

$$x := 3 \quad y := 2 \quad x + y = 5$$

Так какой же оператор (:= или =) правильнее использовать для присваивания переменным значений? В новой версии Mathcad, пожалуй, правильнее все же использовать знак =. Это позволяет избежать переопределения некоторой переменной, которую пользователь ранее отвел под константу. Если вы попытаетесь переопределить ее вводом оператора =, то Mathcad проигнорирует ваши усилия и выведет значение этой переменной.

Существует также «жирный» знак равенства, который используется в логических операциях сравнения и в записях уравнений. Он вводится нажатием клавиш Ctrl и =.

Надо отметить, что попытка использования неопределенной переменной ведет к выводу сообщения об ошибке «This variable or function is not defined above» (эта переменная или функция не определена ранее), при этом переменная окрашивается в ярко-красный цвет.

6.2.2. Локальное и глобальное присваивание

Если переменной присваивается значение с помощью оператора := (или =), то такое присваивание является *локальным* и действует лишь до очередной операции присваивания. Если операция присваивания не проводилась, то переменная будет неопределенной и ее нельзя будет использовать в численных расчетах (но в символьных можно).

Однако с помощью знака ≡ (три горизонтальные черточки) можно обеспечить *глобальное присваивание*, когда переменная получает и имеет заданное значение независимо от того, в каком месте документа стоит оператор глобального присваивания. К примеру, если переменной присвоено таким образом некоторое значение в самом конце документа, то она будет иметь это же значение и в начале документа. Разумеется, в дальнейшем значение переменной можно изменить и с по-

мощью оператора локального присваивания. Особенности присваивания иллюстрирует фрагмент документа, показанный на рис. 6.1.

СТАТУС ПРИСВАИВАНИЯ ЗНАЧЕНИЙ ПЕРЕМЕННЫМ	
$x =$ ■	Переменная x не определена и поэтому
$\ln(x) =$ ■	недоступна для применения
$x = 2$	Переменной x присвоено значение 2
$\exp(x) = 7.389$	и она стала доступной
$x = 2$	Вывод значения переменной x
$y = 50$	Переменная y доступна, поскольку она определена глобально в конце документа
$\ln(y) = 3.912$	
$y = 123$	Значение переменной y может быть
$y = 123$	переопределено локально
$e = 2.718$	Системная переменная e - основание натурального логарифма
$e = 5$	Системная переменная может быть переопределена
$e = 5$	(в данном случае значение e изменено на 5)
$y = 50$	Переменной y глобально присвоено значение 50

Рис. 6.1. Особенности локального и глобального присваивания переменным их численных значений

Статус присваивания не следует путать со статусом самих переменных. Все описанные переменные являются *глобальными*, поскольку их можно определять в любом месте документа и в любом месте изменять их значение. *Локальные переменные* в системе Mathcad содержатся в телах функций пользователя, которые будут рассмотрены ниже (см. раздел «Функции пользователя»).

Как уже отмечалось, использование незаданной переменной в каком-либо математическом выражении является ошибкой и ведет к окраске ее имени ярко-красным цветом (в предшествующих версиях — черным цветом). Установив указатель мыши на имени переменной, во всплывающей подсказке можно получить дополнительное текстовое пояснение о сути ошибки (см. Приложение 5). Использование переменных, значения которых не определены, может привести к различным ошибкам. Все они диагностируются и требуют исправления для продолжения работы.

6.2.3. Применение размерных переменных

Данные и переменные могут быть и *размерными*, то есть характеризоваться не только своим значением, но и указанием физической величины, значение которой они хранят (рис. 6.2). Для присваивания таким переменным значений используются обычные знаки присваивания, но после численного значения со знаком умножения или через пробел указывается единица измерения. Ее удобно выбирать из окна размерных величин, которое появляется при выборе команды Unit (Единицы) в меню Insert (Вставка) или щелчке на кнопке с изображением мерной кружки стандартной панели инструментов.

Mathcad допускает ввод единиц измерения размерных величин символами национальных алфавитов (в том числе буквами русского алфавита), а также позво-

ПРИМЕРЫ ОПЕРАЦИЙ С РАЗМЕРНЫМИ ВЕЛИЧИНАМИ

Заданы единицы длины, массы и времени:

$$\text{cm} \equiv 1\text{L} \quad \text{g} \equiv 1\text{m} \quad \text{ns} \equiv 1 \cdot 10^{-9} \cdot \text{sec}$$

Примеры вычислений с размерными величинами:

$$\text{dyne} \equiv \text{g} \cdot \frac{\text{cm}}{\text{sec}^2} \quad \text{dyne} = 1 \text{ m}^2 \text{ sec}^{-2}$$

$$\text{kg} \equiv 1000 \cdot \text{g} \quad \text{m} \equiv 100 \cdot \text{cm}$$

$$\text{P} := 100 \cdot \frac{\text{kg}}{\text{m}^2} \quad \text{P} = 10 \text{ m}^{-1}$$

$$\text{Tf} \equiv 20 \cdot \text{ns} \quad \text{Tf} = 2 \times 10^{-8} \text{ sec}$$

Проверка переменных с помощью функции UnitsOf (только для Mathcad 2000):

$$x := 123 \quad \text{UnitsOf(P)} = 1 \text{ m}^{-1} \quad \text{UnitsOf}(x) = 1$$

Удаление размерности (только для Mathcad 2000):

$$p := \frac{\text{P}}{\text{UnitsOf(P)}} \quad p = 10$$

Рис. 6.2. Примеры задания и применения размерных переменных

ляет задавать собственные единицы измерения размерных величин. В процессе вычислений Mathcad следит за соответствием размерных величин и выдает сигнал ошибки в случае нарушения такого соответствия: «The units in this expression do not match» (размерности в этом выражении не совпадают). В Mathcad 12 работа с размерными переменными существенно улучшена, осуществляется более тщательный контроль и не допускается применение несовместимых размерностей.

Проведение расчетов с размерными величинами и переменными особенно удобно при решении различных физических задач. При этом правильная размерность результата является дополнительным фактором, свидетельствующим о корректности таких расчетов. Тем не менее переход к безразмерным вычислениям чаще всего является признаком высокой математической культуры при проведении вычислений. Поэтому без особой надобности вводить размерные величины не стоит.

ВНИМАНИЕ Единицы измерения в Mathcad англоязычные. Кроме того, нередко Mathcad вместо ожидаемой единицы выдает ее эквивалент, как очевидный, например Ω вместо ohm (омы), так и далеко не очевидный, например $\text{kg} \cdot \text{m}^2 \cdot \text{s}^{-3} \cdot \text{A}^{-2}$ для единицы измерения сопротивления. В этом случае полезно включить автоматический перевод единиц измерения в более простую форму, установив флажок **Simplify units when possible** (Упрощать размерные величины по возможности) в окне форматирования результатов вычислений (см. подраздел «Представление размерных величин» в разделе «Форматирование результатов» главы 5). Работа с размерными величинами в Mathcad 12 заметно улучшена, но и тут возможны неожиданности.

6.2.4. Задание ранжированных переменных

До сих пор мы рассматривали переменные, которые имеют единственное значение. Однако в математике часто возникает необходимость в задании некоторого ряда значений — чаще всего упорядоченного. Например, для вычисления факториала $N! = 1 \cdot 2 \cdot \dots \cdot (N - 1) \cdot N$ нужно сформировать ряд чисел от 1 до N с ша-

гом 1 и перемножить их. Также упорядоченный ряд значений какой-то переменной (например, абсциссы x) нужен для построения графика функции — Mathcad строит графики функций по точкам, соединяя их отрезками прямой.

Для создания таких рядов в Mathcad используются так называемые *ранжированные переменные*. Иногда они заменяют управляющие структуры — циклы, однако полноценной такая замена все же не является (в частности потому, что не предусмотрен выбор любого значения ранжированной переменной).

В самом простом случае для создания ранжированной переменной используется выражение

```
Name := Nbegin..Nend
```

Здесь Name — имя переменной, Nbegin — ее начальное значение, Nend — конечное значение, .. — символ, указывающий на изменение переменной в заданных пределах (он вводится знаком точки с запятой ;). Если Nbegin < Nend, то шаг изменения переменной будет равен +1, в противном случае — -1.

Для создания ранжированной переменной общего вида используется выражение

```
Name := Nbegin, (Nbegin + Step)..Nend
```

Здесь Step — заданный шаг изменения переменной (он должен быть положительным, если Nbegin < Nend, или отрицательным в противном случае). С помощью этой формы задания ранжированной переменной также можно задавать как возрастающие, так и уменьшающиеся значения ранжированных переменных.

Мы уже неоднократно применяли ранжированные переменные для построения двумерных и трехмерных графиков функций с одной и двумя независимыми переменными. В дальнейшем мы будем часто применять такие переменные и в других применениях.

ПРИМЕЧАНИЕ *Полезно знать, что Mathcad позволяет задавать и ранжированные размерные переменные. Каждый компонент такой переменной — размерная величина.*

6.2.5. Применение ранжированных переменных

Важно отметить, что задание ранжированных переменных (range variable) эквивалентно заданию конечных циклов. Явно задаются начальное значение, шаг изменения и конечное значение, неявно — обращение к оператору цикла, вычисляющему все промежуточные значения ранжированной переменной при каждом обращении к ней.

Ранжированные переменные удобно использовать при определении индексированных переменных и векторов, а также при построении графиков.

Привыкшие к обычному программированию пользователи часто забывают, что ранжированная переменная — это не обычная переменная, в каждый данный момент имеющая одно значение. Они пытаются выполнять с такими переменными действия, корректные лишь для обычных переменных. Например, задав ранжированную переменную $i := 1..100$, пытаются использовать присваивание вида $f := i^2$, где f — обычная переменная. Это приводит к ошибке «Illegal context. Press F1 for Help» — система (в русском переводе) сообщит: «Неправильный контекст. Нажмите F1 для справки». Однако можно использовать следующее присваивание:

```
fi := i2
```


В этом случае будет получен новый *вектор* с именем f , элементы которого в нашем случае являются квадратами значений ранжированной переменной i . Более подробно особенности задания и применения векторов рассматриваются ниже (см. раздел «Массивы, векторы и матрицы»).

На рис. 6.3. представлена ранжированная переменная J (ряд значений тока в амперах — amp). К сожалению, нетрудно заметить, что при выводе произошла замена одной единицы тока — amp — на эквивалентное ей, но непривычное отношение coul/sec (Кулон/с). Это говорит о том, что в применении ранжированных размерных величин в Mathcad пока не все гладко.

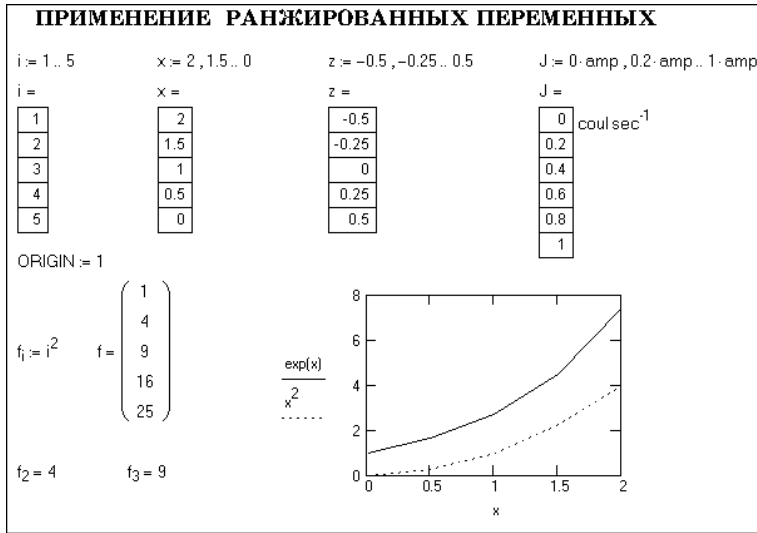


Рис. 6.3. Примеры применения ранжированных переменных

6.2.6. Применение таблиц вывода

Ранжированные переменные широко применяются для представления функций в виде таблиц, а также для построения их графиков. Если после некоторого выражения с ранжированной переменной поставить знак равенства, то после щелчка мышью на экран будет выведена таблица значений этого выражения. Несколько таких таблиц показано на рис. 6.3.

Полезно учитывать некоторые свойства таблиц вывода.

- Если количество значений переменной (выражения, вектора и т. п.) и, соответственно, строк в таблице вывода больше 16, то выводятся первые 16 строк. Если указатель мыши находится в пределах таблицы, то щелчок приводит к появлению вертикальной полосы прокрутки, позволяющей просмотреть все строки таблицы.
- Числа в таблицах можно задавать в требуемом формате с помощью операций задания формата чисел.
- В старых версиях Mathcad при выводе значений размерного выражения в виде таблицы все ячейки таблицы содержали единицу измерения; начиная с Mathcad 8.0, единица измерения выводится только в заголовке таблицы, что, несомненно, удобнее.

Примеры вывода вектора и одномерного массива давления представлены ниже:

$$i := 1..5 \quad V_i := i \quad L_i := i \cdot \text{atm}$$

$$V = \begin{pmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix} \quad L = \begin{pmatrix} 0 \\ 1.013 \times 10^5 \\ 2.026 \times 10^5 \\ 3.04 \times 10^5 \\ 4.053 \times 10^5 \\ 5.066 \times 10^5 \end{pmatrix} \text{Pa}$$

Обратите внимание на то, что при выводе массива давлений Mathcad изменил одну единицу давления atm (атмосфера) на другую Pa (килопаскаль).

6.3. Работа с файловыми данными

6.3.1. Файлы как данные

Mathcad может успешно использоваться в современных физических установках для обработки данных эксперимента. Но для этого необходима связь между такими установками (точнее, их программным обеспечением) и системой Mathcad. Для осуществления подобной связи Mathcad имеет специальный тип данных — *файловые данные*. В сущности, это те же векторы и матрицы, но с элементами, которые могут записываться в виде файлов текстового формата, имеющих свои имена. Файлы данных в системе представляют собой запись матриц в их естественной форме. Имеются операции считывания таких файлов, позволяющие присваивать содержащиеся в них данные переменным, векторам или матрицам.

В ходе создания файла система считывает значения элементов вектора или матрицы поэлементно (для матриц — слева направо и сверху вниз) и по ходу считывания преобразует числовые значения элементов в их символьные эквиваленты, использующие ASCII-коды цифр. Эти символьные значения и записываются в виде файлов данных. Как уже отмечалось, в Mathcad 12 в качестве значения элементов векторов и матриц может использоваться константа NaN.

6.3.2. Применение функций для работы с файлами

Во всех версиях Mathcad для Windows существует ряд простых файловых операций (функций). Создаваемые или используемые ими файлы легко просмотреть в любом текстовом редакторе, воспринимающем тексты в виде ASCII-кодов. При считывании файлов система обеспечивает обратное преобразование символьных представлений в числовые значения элементов. Такая форма задания облегчает «стыковку» Mathcad с программами, написанными на любом языке программирования, даже таком простом, как Бейсик (или Паскаль), и обеспечивает взаимный обмен файлами текстового формата.

Начнем рассмотрение данных операций с операций считывания и записи неструктурированных файлов, которые были в прежних версиях Mathcad (до Mathcad 2001):

```
READ ("Имя_файла") и WRITE ("Имя_файла") .
```

Была также функция

`APPEND("Имя_файла"),`

которая дописывала данные в конец уже существующего файла.

ВНИМАНИЕ Функции `READ`, `WRITE` и `APPEND` в новых версиях *Mathcad 2001i/11/12* уже отсутствуют. Поэтому применение старых документов, в которых они применяются, будет приводить к ошибкам и отказу в выполнении операций с этими функциями. Применяйте вместо них описанные ниже функции.

Следующая функция считывает структурированные данные в виде двумерного массива — матрицы:

`READPRN("Имя_файла")`

Функция `READPRN` возвращает матрицу, значения элементов которой однозначно связаны со значениями элементов файла. Или, точнее, каждая строка или столбец возвращаемой матрицы подобны соответствующим строкам или столбцам текстового представления файла.

А для записи матричного выражения (или матрицы) в файл с указанным именем используется функция:

`WRITEPRN("Имя_файла")`

Структура текста файла подобна структуре матрицы. Следующая функция дописывает данные в уже существующий матричный файл:

`APPENDPRN("Имя_файла")`

Следует особо отметить, что при работе с векторами и матрицами, имеющими комплексные элементы, эти операции используют расширенные векторы или матрицы, элементы которых — действительные числа.

На рис. 6.4 показан документ с типовыми операциями файлового типа. Этот рисунок иллюстрирует эффективные возможности занесения данных в файл и их последующего считывания.

Полезно просмотреть в текстовом редакторе созданный *Mathcad* файл данных. Нетрудно заметить, что данные представлены наглядно и естественно, поэтому та-

РАБОТА С ДАННЫМИ ФАЙЛОВОГО ТИПА		
$M1 := \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$	<code>WRITEPRN("DMAT.prn") := M1</code>	Создание файла данных DMAT матричного типа (матрица M1)
$M := \text{READPRN}("DMAT.prn")$	$M = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$	Считывание файла данных DMAT и создание матрицы M
$i := \sqrt{-1}$	$N := \begin{pmatrix} 1 + 2i & 3 + 4i \\ 5 + 6i & 7 + 8i \end{pmatrix}$	Создание матрицы N с комплексными элементами
<code>WRITEPRN("DCMAT.prn") := N</code>		Запись матрицы N в файл DCMAT
$P := \text{READPRN}("DCMAT.prn")$		Считывание файла DCMAT и создание матрицы P
$P = \begin{pmatrix} 1 + 2i & 3 + 4i \\ 5 + 6i & 7 + 8i \end{pmatrix}$		Вывод матрицы P

Рис. 6.4. Основные операции с данными файлового типа

кой файл легко обрабатывать (создавать или модифицировать) с помощью программ, написанных на любых других языках программирования, что создает предпосылки для обеспечения легкого обмена данными через файлы.

Следующая операция используется для считывания RGB-изображений:

```
READRGB ("Имя_файла")
```

Эта операция возвращает массив из трех подмассивов, несущих данные о разложении цветного изображения на три основных цвета — красный, зеленый и синий.

Благодаря введению данных файлового типа возможности системы Mathcad существенно расширяются. Она может использоваться в сложных программных комплексах. Например, некоторая специализированная программа может управлять экспериментальной установкой и формировать файлы данных, которые в дальнейшем обрабатываются системой Mathcad с использованием ее мощных и гибких математических возможностей. При этом предусмотрена обработка множества пакетов данных.

ПРИМЕЧАНИЕ В Mathcad 2001i/11/12 данные файлового типа существенно расширены. Помимо упомянутых функций имеется ряд функций для так называемого графического процессора, позволяющих работать с цветовыми параметрами изображений — например, с разложением их на отдельные цвета, управлением интенсивностью отдельных составляющих цвета, их насыщенностью и яркостью. В приложении 4 можно найти данные об этих функциях. Кроме того, добавлена возможность записи и считывания файлов в формате HTML с поддержкой технологии корректной передачи математических текстов через Интернет — MathML. В Mathcad 12 появилась возможность сохранения файлов в формате XML.

6.3.3. Новые функции для работы с бинарными файлами в Mathcad 11/12

Системы Mathcad 2001i/11 имеют множество файлов для работы с изображениями. Все они перечислены в Приложении В. В системах Mathcad 11/12 введены две новые функции для работы с бинарными файлами.

Для считывания бинарного файла используется функция

```
READBIN(file, type[, [endian] [, cols] [, skip] [, maxrows]])
```

Эта функция считывает бинарный файл со спецификацией type строчного типа. Файл может иметь тип utit16, unit32, double (64-разрядные числа), float (32-разрядные числа) и byte (8-разрядные числа). Необязательные параметры уточняют структуру считываемых бинарных данных: тип данных признака конца файла endian (0 или 1, по умолчанию 0), число столбцов в строке входного файла (cols ≥ 1, по умолчанию 1), число байт, пропущенных с начала файла skip (по умолчанию 0), и число строк, лимитирующих вход maxrows.

Другая функция

```
WRITEBIN(file, type)
```

записывает в файл матрицу бинарного файла (например, WRITEBIN(file, type):=A), файл может иметь тип utit16, unit32, double (64-разрядные числа), float (32-разрядные числа) и byte (8-разрядные числа).

6.4. Вычисление математических выражений

6.4.1. Операторы и операнды

Вычисление *математических выражений* — одна из самых распространенных задач, которые решаются в системе Mathcad. В простейшем случае выражения состоят из констант, переменных операторов с операндами и функций с их параметрами. Для вычисления выражения в численном виде после него достаточно установить знак оператора численного вывода $=$, а для вычисления в символьном виде — знак оператора символьного вывода \rightarrow .

Операторы — элементы языка, предназначенные для создания математических выражений. Это знаки арифметических операций, вычисления сумм и произведений членов ряда, производных, интегралов и т. д. После указания аргументов (*операндов*) операторы становятся исполняемыми программными блоками.

6.4.2. Арифметические операторы

Арифметические операторы предназначены для выполнения арифметических действий над численными величинами (операндами) и конструирования математических выражений [32]. Полный перечень операторов системы Mathcad приведен в Приложении 2. Самыми распространенными являются операторы арифметических действий (+, -, * и /), возведения в степень ^, извлечения квадратного корня и др.

На рис. 6.5 показан фрагмент документа с примерами использования арифметических операторов. Здесь представлены простые операции над данными скалярного типа. Обратите внимание на то, что такие операции используют общепринятый приоритет операций, который можно изменять с помощью скобок. При этом, если число пар скобок больше 1, то лишь внутренние скобки выводятся как круглые — внешние скобки выводятся как квадратные.

Операции со скалярными данными:			
$X := 2$	$a := 5$	$Y := a \cdot X$	$Y = 10$
$\sqrt{50} = 7.071$	$ -10 = 10$	$10! = 3.629 \times 10^6$	$-Y = -10$
$F := \frac{(1+3)}{(2+7)} + 5^2$		$F = 25.444$	
$H := 6 \cdot [1 + [2 + [3 + (4 + 5)]]]$		$H = 90$	

Рис. 6.5. Примеры применения арифметических операторов

Для работы с комплексными величинами необходимо в виде корня квадратного из -1 указать обозначение мнимой единицы (i или j). Например, при использовании идентификатора i это делается следующим образом:

$$i := \sqrt{-1}$$

Соответствующий фрагмент документа показан на рис. 6.6.

Многие встроенные функции Mathcad, в частности элементарные, могут иметь комплексный аргумент.

Операции с комплексными данными:			
$i := \sqrt{-1}$	Задание мнимой единицы		
$Z := -3 + 2 \cdot i$	$ Z = 3.606$	$\arg(Z) = 2.554$	
$\sqrt{Z} = 0.55 + 1.817i$	$\sqrt{-5} = 2.236i$	$2 \cdot Z = -6 + 4i$	
$Z1 := (1 + 2 \cdot i)$	$Z2 := (3 + 4 \cdot i)$		
$Z1 + Z2 = 4 + 6i$	$Z1 - Z2 = -2 - 2i$	$Z1 \cdot Z2 = -5 + 10i$	$\frac{Z1}{Z2} = 0.44 + 0.08i$
$\forall C0 := Z1$	$\forall C1 := Z2$	$\forall C = \begin{pmatrix} 1 + 2i \\ 3 + 4i \end{pmatrix}$	

Рис. 6.6. Примеры арифметических операций с комплексными числами

6.4.3. Применение операторов отношения

Ряд операторов предназначен для сравнения двух величин. Они называются *операторами отношения*.

Оператор	Клавиши	Наименование операции
$X > Y$	X > Y	X больше Y
$X < Y$	X < Y	X меньше Y
$X \geq Y$	X Ctrl) Y	X больше или равно Y
$X \leq Y$	X Ctrl (Y	X меньше или равно Y
$X \neq Y$	X Ctrl # Y	X не равно Y
$X = Y$	X Ctrl = Y	X равно Y

Не следует путать оператор сравнения (знак равенства) с похожим знаком вывода значений переменных. В системе Mathcad знак равенства как оператор отношения имеет больший размер и более жирное начертание, чем обычный знак равенства — оператор вывода. Все операторы отношения могут вводиться самостоятельно в месте расположения курсора. В этом случае по обе стороны от них появляются маленькие темные прямоугольники. Они являются местами ввода для подлежащих сравнению выражений.

Необходимо отметить, что выражения с операторами отношения возвращают логическое значение, соответствующее выполнению или невыполнению условия, заданного оператором. Эти значения в системе Mathcad являются логической единицей (1), если условие выполнено, и логическим нулем (0), если оно не выполнено. Математически значения логической единицы и нуля совпадают со значениями числовых констант 1 и 0. Например:

- $2 > 1 = 1$ — условие выполнено, результат 1;
- $1 > 2 = 0$ — условие не выполнено, результат 0.

Указанное свойство операторов отношения позволяет строить не совсем обычные выражения, содержащие в себе операторы отношения, например:

$$2 \cdot (5 > 0) = 2$$

Нетрудно понять, что выражение $(5 > 0)$ возвращает единицу, поэтому результат вычисления такого выражения даст число 2. Операторы отношения часто используются совместно с условными функциями, которые будут описаны далее.

6.4.4. Применение логических операторов

Поскольку логические операторы легко имитируются с помощью функции Хевисайда (x) (см. далее раздел «Функции с условиями сравнения»), долгое время они в систему Mathcad не вводились. Однако уже в Mathcad 2001 введен минимум таких операторов. Они находятся в палитре Boolean (Булевы) и имеют обычные обозначения. Действие логических операторов задается *таблицей истинности*.

not \neg	and \wedge	or \vee	xor \otimes
$0 \neg = 1$	$0 \wedge 0 = 0$	$0 \vee 0 = 0$	$0 \otimes 0 = 0$
$1 \neg = 0$	$1 \wedge 0 = 0$	$1 \vee 0 = 1$	$1 \otimes 0 = 1$
	$0 \wedge 1 = 0$	$0 \vee 1 = 1$	$0 \otimes 1 = 1$
	$1 \wedge 1 = 1$	$1 \vee 1 = 1$	$1 \otimes 1 = 0$

Аргументами операторов должны быть логические нули (0) или логические единицы (1). Их иногда называют значениями FALSE (ложь) и TRUE (истина). Можно сказать, что данные операторы реализуют *четкую логику*. Однако эксперимент показал, что в роли логической единицы может выступать любое вещественное число, даже отрицательное. Например, для оператора not справедливо следующее: $1.23 \neg = 0$. Иногда это позволяет реализовать так называемую *нечеткую логику*.

6.5. Решение типовых задач математического анализа

Математический анализ находит широкое применение в решении учебных и научно-технических задач [29—34]. Ниже рассмотрены основные возможности систем Mathcad в решении задач математического анализа.

6.5.1. Простые примеры на применение операторов математического анализа

К числу наиболее распространенных задач относятся задачи математического анализа — вычисление сумм, произведений, производных, пределов, интегралов и т. д. Они могут решаться как в численном, так и в аналитическом (символьном) виде. Система Mathcad содержит ряд расширенных арифметических операторов. Это операторы вычисления суммы (клавиша \$) и произведения (клавиша #) членов ряда, а также операторы производной (клавиша ?) и определенного интеграла (клавиша &). Нажатие указанных клавиш приводит к выводу на экран шаблона соответствующего оператора.

Шаблон того или иного оператора содержит места ввода, подлежащие заполнению операндами — числовыми или символьными значениями. Вычисление иницируется вводом знака равенства (=). Ниже даны основы решения таких задач на основе расширенных операторов системы Mathcad. Например, вычисление суммы членов ряда выполняется следующим образом:

$i := 1..8$

$$\sum_i i = 36$$

Разумеется, для вызова шаблона можно использовать и соответствующую палитру специальных математических символов.

Фрагмент документа на рис. 6.7 иллюстрирует применение этих операторов. Заметим, что выражения с расширенными операторами возвращают вычисленные значения, поэтому их можно использовать в составе сложных математических выражений, представленных в наглядном виде. Поэтому расширенные операторы являются важным компонентом комплексной визуализации вычислений.

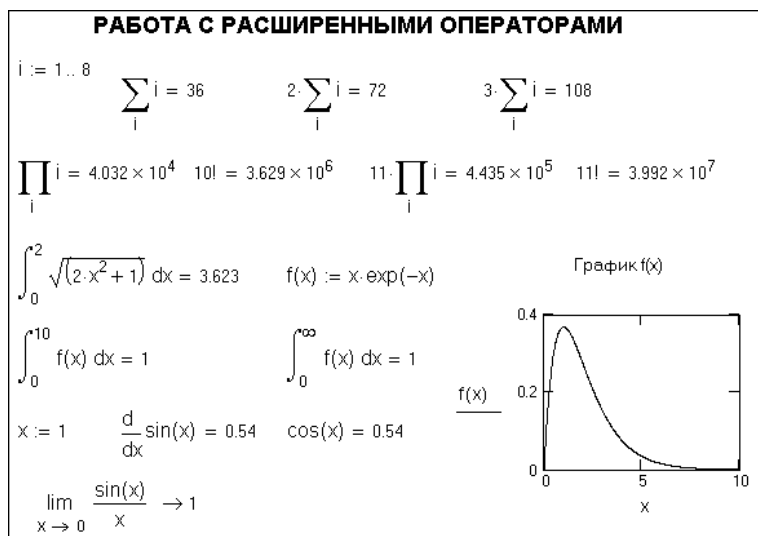


Рис. 6.7. Примеры применения расширенных операторов

Обратите внимание на весьма поучительный пример вычисления определенного интеграла с подынтегральной функцией $x \cdot \exp(-x)$. В версии Mathcad 7.0 этот интеграл при бесконечном верхнем пределе вычислялся некорректно — вместо точного значения 1 получалось значение 0. Причиной этого «парадокса» был используемый алгоритм численного интегрирования. Начиная с Mathcad версии 8.0 интеграл вычисляется точно. Это следствие применения существенно усовершенствованного адаптивного алгоритма численного интегрирования.

Можно, конечно, удивиться тому, что и при верхнем пределе, равном 10, также получается точная единица. В данном случае, помимо уже отмеченного обстоятельства (значения функции при больших значениях x быстро стремятся к нулю), сказывается конечная точность вывода результатов вычислений — учитываются только три значащие цифры, причем последующие нули по умолчанию не пишутся. Точность вывода результатов можно изменить с помощью контекстного меню, при этом точность расчетов определяется значением системной переменной TOL , изменение которой позволяет пользователю установить необходимую ему минимальную точность вычислений.

Применение расширенных операторов облегчает решение математических задач — раньше для выполняемых ими действий приходилось писать отдельные

программы. Достигаемая при использовании этих операторов прекрасная визуализация вычислений — важное достоинство системы Mathcad. Рассмотрим применение основных из этих операторов более подробно.

6.5.2. Вычисление сумм

Нередко нужно вычислять *суммы членов ряда* или значений некоторой функции (или некоторых функций) f_i при индексе i , меняющемся от начального значения m до конечного значения n :

$$\sum_{i=m}^n f_i = f_m + f_{m+1} + \dots + f_i + \dots + f_{n-1} + f_n$$

Здесь индекс i просто означает, что в общем случае каждая функция представлена отдельным членом ряда. Для вычисления сумм в Mathcad используются операторы и шаблоны сумм. Оператор $=$ дает вычисление в численном виде, а оператор \rightarrow для многих сумм (но не для всех) обеспечивает вычисления в символьном виде.

Пусть надо найти в численном виде суммы первых 10 членов последовательностей $f_i = i$, i^2 и $1/i$. Решение представлено ниже:

$$\sum_{i=1}^{10} i = 55, \quad \sum_{i=1}^{10} i^2 = 385, \quad \sum_{i=1}^{10} \frac{1}{i} = 2.929.$$

Далее, пусть надо вычислить две первые суммы для n членов из предшествующего примера в аналитическом виде. Имеем:

$$\sum_{i=1}^n i \rightarrow \frac{1}{2} \cdot (n+1)^2 - \frac{1}{2} \cdot n - \frac{1}{2}, \quad \sum_{i=1}^n i^2 \rightarrow \frac{1}{3} \cdot (n+1)^3 - \frac{1}{2} \cdot (n+1)^2 + \frac{1}{6} \cdot n + \frac{1}{6}$$

В Mathcad есть шаблон и для вычисления бесконечных сумм с указанием только индекса снизу. Многие из таких сумм сходятся, т. е. при бесконечном суммировании результат стремится к определенному значению, представленному числом или формулой. В ядре системы есть таблица таких сумм, используемая для их вычислений. Пример — вычислить в аналитической форме суммы $1/i$ для i от 1 до n и от 1 до ∞ . Имеем:

$$\sum_{i=1}^n \frac{1}{i} \rightarrow \text{Psi}(n+1) + \gamma, \quad \sum_i \frac{1}{i} \rightarrow \text{Psi}(i)$$

Обратите внимание на то, что эти суммы не выражаются через элементарные функции. Для их представления используется специальная математическая Psi-функция.

Часто при вычислении сумм полезно задать оптимизацию вычислений. Это показывает следующий пример:

$$\begin{aligned} \text{Sum}(n, z) &:= \sum_{i=1}^n \frac{1}{\left(\frac{i}{z}\right)} & \text{Sum}(10000, 20) = \blacksquare \\ \text{Sum}(n, z) &:= \sum_{i=1}^n \frac{1}{\left(\frac{i}{z}\right)^*} & \text{Sum}(10000, 20) = 3.847 \end{aligned}$$

Здесь в первом случае Mathcad отказался вычислять сумму с большим числом членов. Однако после установки режима оптимизации (он распознается по красной звездочке после выражения) сумма благополучно вычисляется. Если оптимизация не удалась, звездочка будет черного или синего цвета.

6.5.3. Вычисление произведений

Произведение членов ряда реализуется выражением:

$$\prod_{i=m}^n f_i = f_m \cdot f_{m+1} \cdot \dots \cdot f_i \cdot \dots \cdot f_{n-1} \cdot f_n.$$

Все сказанное о правилах применения операций суммирования относится и к операторам вычисления произведения. Для них используются соответствующие шаблоны.

Пусть надо вычислить произведение натуральных чисел от 1 до ∞ и их квадратов в аналитической форме:

$$\prod_i i \rightarrow \Gamma(i), \quad \prod_i i^2 \rightarrow \Gamma(i)^2.$$

Результат получен через специальную гамма-функцию.

Далее вычислим произведение натуральных чисел от 1 до 10 и их квадратов в численном виде:

$$i := 1..10 \quad \prod_i i = 3628800$$

$$\prod_{i=1}^{10} i = 3.629 \times 10^6 \quad \prod_{i=1}^{10} i^2 = 1.317 \times 10^{13}$$

Обратите внимание на два варианта задания пределов изменения i — в виде ранжированной переменной и в шаблоне произведения. Сказанное об оптимизации вычислений суммы относится и к вычислению произведения.

6.5.4. Вычисление пределов

Пределом функции $f(x)$ называют то ее значение b , к которому функция неограниченно приближается в точке $x=a$ (предел в точке) или слева или справа от нее. Предел обозначается как:

Предел в точке a	Предел слева от точки a	Предел справа от точки a
$\lim_{x \rightarrow a} f(x) = b$	$\lim_{x \rightarrow a^-} f(x) = b$	$\lim_{x \rightarrow a^+} f(x) = b$

При этом подразумевается, что функция $f(x)$ определена на некотором промежутке, включающем точку $x=a$ и во всех точках, близких к ней слева и справа. В последнем случае предел вычисляется для $x = a - h$ или $x = a + h$ при h , стремящемся к нулю. Пределом может быть число, математическое выражение, положительная или отрицательная бесконечность. В системе Mathcad вычисление предела возможно только при символьном выводе. Для этого используется соответствующий шаблон предела.

Рассмотрим следующий пример — надо вычислить предел выражения $\sin(x)/x$ в точке $x=0$, имеющей в этой точке устранимую неопределенность вида $0/0 \rightarrow 1$:

$$\lim_{x \rightarrow 0} \frac{\sin(x)}{x} \rightarrow 1$$

$$\lim_{x \rightarrow 2} \frac{1}{(x-2)} \rightarrow \text{undefined} \quad \lim_{x \rightarrow 2^+} \frac{1}{(x-2)} \rightarrow \infty \quad \lim_{x \rightarrow 2^-} \frac{1}{(x-2)} \rightarrow -\infty$$

Далее вычислим пределы выражения $1/(x - 2)$ в точке $x = 2$, слева и справа от нее:

Обратите внимание на то, что в первом случае Mathcad выдал сообщение *undefined* (не определен), которое и указывает на отсутствие предела. В других примерах пределом является бесконечность («положительная» и «отрицательная»). Можно также говорить о пределе ряда функций при $x \rightarrow \infty$ или при $x \rightarrow -\infty$. Например, функция e^{-x} имеет предел 0 при $x \rightarrow \infty$. Проверьте это сами.

6.5.5. Определение производной и полного дифференциала

Производная непрерывной функции $f(x)$ это предел, к которому стремится отношение бесконечно малого приращения функции к соответствующему бесконечно малому приращению аргумента Δx :

$$f'(x) = \frac{df(x)}{dx} = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

Если речь идет о вычислении численного значения производной, то оно производится в некоторой точке $x = x_0$. Как известно, значение производной *геометрически* характеризуется наклоном касательной к графику $f(x)$ в точке $x = x_0$. Производную можно рассматривать и как скорость изменения функции в заданной точке. В экстремумах функций производная равна нулю.

Помимо производной часто оперируют понятием дифференциала

$$df(x) = f'(x) \cdot \Delta x,$$

то есть произведения производной функции на приращение ее аргумента $\Delta x \rightarrow 0$.

Если функция имеет производную в точке x , то она в этой точке непрерывна. Разрывные функции в точках разрыва не имеют производных, хотя у них возможны производные слева и справа от точек разрыва. Непрерывность функции не является достаточным признаком того, что она имеет производную. Не все непрерывные функции имеют производные во всех точках. В принципе возможны даже непрерывные функции, вообще не имеющие производных. Примером могут быть самоподобные кривые — *фракталы*, вид которых сохраняется при уменьшении или увеличении размеров.

Производная от первой производной $f'(x)$, то есть функция $f''(x)$ называется *производной второго порядка*. Могут быть и *производные высшего порядка*. Шаблон Mathcad для вычисления производных высшего порядка можно использовать для вычисления таких производных до 5-го порядка включительно. Производные высокого порядка вычисляются и последовательным применением шаблона для вычисления первой производной. Это поясняется примерами, представленными на рис. 6.8.

Mathcad вычисляет производные в численном виде адаптивным методом Риддера. Он гарантирует 7–8 точных знаков только для первой производной. Вычисление высших производных порядка выше 5 может сопровождаться потерей точности вычислений и резким возрастанием времени вычислений, что заметно уже в приведенных примерах. Поэтому лучше в этом случае попытаться найти симво-

x := 1	
$\frac{d}{dx} \sin(x) = 0.5403023059$	
$\frac{d^2}{dx^2} \sin(x) = -0.8414709848$	$\frac{d}{dx} \frac{d}{dx} \sin(x) = -0.8414709848$
$\frac{d^5}{dx^5} \sin(x) = 0.540302284$	$\frac{d}{dx} \frac{d}{dx} \frac{d}{dx} \frac{d}{dx} \frac{d}{dx} \sin(x) = 0.5403023057$
$\frac{d}{dx} \frac{d}{dx} \frac{d}{dx} \frac{d}{dx} \frac{d}{dx} \frac{d}{dx} \sin(x) \rightarrow -\cos(1) = -0.5403023059$	

Рис. 6.8. Примеры вычисления производных

льное выражение для производной и затем уже численное — см. последний пример из приведенных на рис. 6.8.

Довольно часто встречаются функции ряда переменных, например $f(x, y, z, \dots)$. В этом случае может идти речь о *частных производных* по переменным x, y, z, \dots . Например, частной производной по переменной x будет выражение:

$$f'_x(x, y, z, \dots) = \frac{\partial f(x, y, z, \dots)}{\partial x} = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x, y, z, \dots) - f(x, y, z, \dots)}{\Delta x}.$$

Подобные выражения нетрудно составить и для частных производных по другим переменным. Можно считать, что при вычислении частной производной по какой-то переменной остальные переменные рассматриваются как константы. Это иллюстрируют следующие примеры:

$$\frac{d}{dx} \sin(x \cdot y) \rightarrow \cos(x \cdot y) \cdot y \qquad \frac{d}{dy} \sin(x \cdot y) \rightarrow \cos(x \cdot y) \cdot x$$

Если пометить значок производной синим уголком, то в меню правой клавиши мыши можно обнаружить команду View Derivative As, которая раскрывает меню обозначений частных производных:

- Default — обозначение по умолчанию;
- Derivative — обозначение как обычной производной: $\frac{d}{dx} f(x, y, \dots)$;
- Partial Derivative — обозначение как частной производной: $\frac{\partial}{\partial x} f(x, y, \dots)$.

Можно также говорить о частных дифференциалах. *Полный дифференциал* функции ряда переменных:

$$df = \frac{\partial f}{\partial x} dx + \frac{\partial f}{\partial y} dy + \frac{\partial f}{\partial z} dz + \dots$$

Перейдем к практике вычисления производных.

6.5.6. Вычисление и визуализация производных

Для выполнения дифференцирования в системе Mathcad можно использовать команды Variable Differentiate в позиции Symbolic меню. Для выполнения этой операции надо выделить мышкой переменную в дифференцируемом выражении и исполнить данную команду. Результат вычислений появляется снизу под исходным выражением, справа от него или вместо исходного выражения. Это, как и вывод комментария, можно задать в окне команды Evaluation Style... в позиции Symbolic меню.

Данный способ имеет существенные ограничения. Одно из них — невозможность вычисления производных для выражений, содержащих функции пользователя. Не вполне очевидна связь исходного выражения с результатом вычисления производной. Впрочем, этот недостаток устраняется выводом комментария.

Пусть надо вычислить производные выражений $\sin(x)$, $2 \cdot \exp(x)$ и $x/(x-1)$ в командном режиме. Рисунок 6.9 показывает вычисления.

$\sin(x)$	by differentiation, yields	$\cos(x)$
$2 \cdot \exp(x)$	by differentiation, yields	$2 \cdot \exp(x)$
$\frac{x}{x-1}$	by differentiation, yields	$\frac{1}{(x-1)} - \frac{x}{(x-1)^2}$

Рис. 6.9. Примеры вычисления производных в символьном виде в командном режиме

Другой способ — с применением шаблонов вычисления производных и оператора символьных вычислений \rightarrow обычно более удобен и более нагляден. Пусть надо вычислить производные для выражений предшествующего примера, задав последнее выражение как функцию пользователя $g(x)$ и вычислив дополнительно третью производную от $g(x)$. Эти вычисления представлены на рис. 6.10.

$\frac{d}{dx} \sin(x) \rightarrow \cos(x)$	$\frac{d}{dx} 2 \cdot \exp(x) \rightarrow 2 \cdot \exp(x)$	$\frac{d}{dx} \frac{x}{x-1} \rightarrow \frac{1}{(x-1)} - \frac{x}{(x-1)^2}$
$g(x) := \frac{x}{x-1}$	$\frac{d^3}{dx^3} g(x) \rightarrow \frac{6}{(x-1)^3} - 6 \cdot \frac{x}{(x-1)^4}$	$\frac{d}{dx} \frac{d}{dx} \frac{d}{dx} g(x) \rightarrow \frac{6}{(x-1)^3} - 6 \cdot \frac{x}{(x-1)^4}$
$x := .5$	$\frac{d^3}{dx^3} g(x) \rightarrow -96.000000000000000000$	$\frac{d}{dx} \frac{d}{dx} \frac{d}{dx} g(x) = -96$

Рис. 6.10. Примеры вычисления производных с применением оператора символьного вывода

Рисунок 6.11 дает наглядную геометрическую трактовку производной, значение которой определяется наклоном касательной к точке графика функции, в которой определяется производная. Кроме того, на этом рисунке дается определение и построение перпендикуляра к линии функции в заданной точке.



Рис. 6.11. Построение касательной и перпендикуляра в точке функции

6.5.7. Определение интегралов

Интегральное исчисление зародилось из практической необходимости вычисления площадей, объемов и центров тяжести различных фигур. Если есть некоторая функция $f(x)$, то *определенный интеграл* вида

$$\int_a^b f(x) dx$$

дает значение площади, ограниченной вертикалями a и b (именуемыми пределами интегрирования), кривой $f(x)$ и осью абсцисс.

Если $f(x)dx$ есть дифференциал функции $F(x)$, то

$$f(x)dx = dF(x).$$

Функцию $F(x)$ называют первообразной функции $f(x)$. Наиболее общий вид первообразной функции $f(x)$ называют неопределенным интегралом и обозначают как

$$\int f(x) dx.$$

В состав этого выражения включена некоторая постоянная интегрирования C , подчеркивающая, что для одной и той же $f(x)$ существует масса первообразных, описываемых одной и той же линией, но смещенных по вертикали на произвольную постоянную. Например, для $f(x) = \sin(x)$ имеем:

$$\int \sin(x) dx = -\sin(x) + C.$$

Существуют специальные (в том числе весьма обширные) таблицы интегралов, значительно облегчающие их нахождение. Но в последнее время для этого чаще используются СКМ.

6.5.8. Вычисление однократных интегралов

Mathcad реализует вычисление интегралов с помощью *операторов интегрирования*, вводимых с помощью соответствующих шаблонов. Рисунок 6.12 дает наглядное представление об определенном и неопределенном интегралах и технике интегрирования в системе Mathcad. Как видно из него, под площадью надо понимать ее алгебраическое значение, то есть разность между площадью над осью x и под ней. В этом случае ясно, что определенный интеграл может иметь как положительные (первый пример на рис. 6.12), так и отрицательные (второй пример на рис. 6.9) значения.

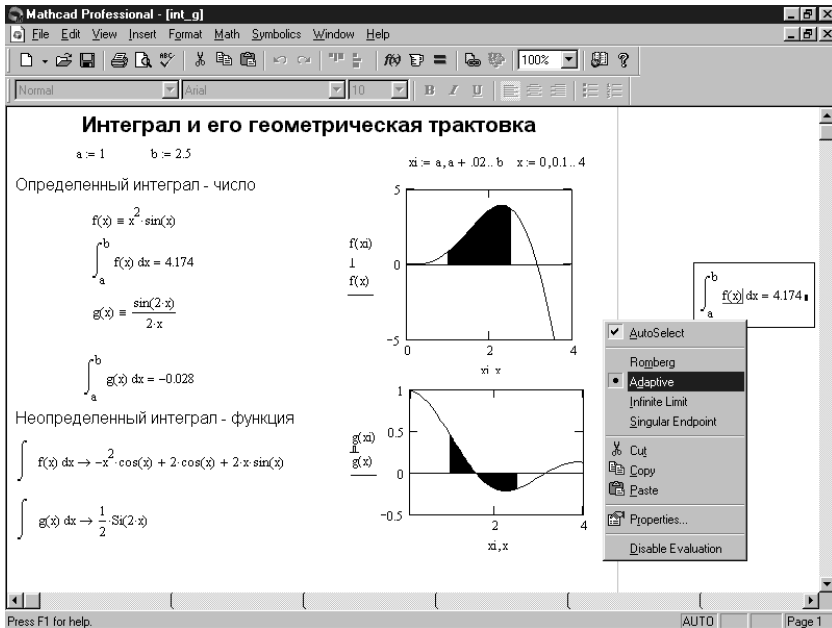


Рис. 6.12. Графическая иллюстрация интегрирования

Приведем еще один наглядный пример. Надо вычислить определенный интеграл от функции $\sin(x)$ в пределах от 0 до π , от π до 2π и от 0 до 2π :

$$\int_0^{\pi} \sin(x) dx = 2 \int_{\pi}^{2\pi} \sin(x) dx = -2 \int_0^{2\pi} \sin(x) dx = 0$$

Из него отчетливо видно, что площадь положительной полуволны синусоиды положительна, а отрицательной отрицательна. А потому значение интеграла от одного периода синусоиды равно 0.

Интегралы часто применяются для вычисления ряда геометрических понятий, например:

- площадь между двумя непрерывными кривыми $f_1(x)$ и $f_2(x)$ — A ,
- длина дуги кривой $f(x)$ между точками a и b — L ;
- объем фигуры, образованной вращением кривой $f(x)$ вокруг оси x , усеченной по оси x значениями a и b , — V ;
- площадь предшествующей фигуры — S .

На рис. 6.13 представлены примеры вычисления этих интегральных параметров.

$f1(x) := x^3$	$f2(x) := x^2$	$a := 2$	$b := 4$
$A := \int_a^b (f1(x) - f2(x)) dx$		$A = 41.333$	
$L := \int_a^b \sqrt{1 + \left(\frac{d}{dx} f1(x)\right)^2} dx$		$L = 56.042$	
$V := \int_a^b 2 \cdot \pi \cdot f2(x)^2 dx$		$V = 1.247 \times 10^3$	
$S := \int_a^b 2 \cdot \pi \cdot f2(x) \cdot \sqrt{1 + \left(\frac{d}{dx} f2(x)\right)^2} dx$		$S = 763.34$	

Рис. 6.13. Вычисление некоторых интегралов, относящихся к геометрическим понятиям

В новых версиях Mathcad введена возможность выбора метода вычисления определенных интегралов. Для этого достаточно указать курсором мыши на знак интеграла и нажать правую клавишу мыши — появится контекстно-зависимое меню этой клавиши (см. рис. 6.9 справа), в котором жирной точкой отмечен используемый метод — по умолчанию адаптивный. Можно установить любой из следующих методов:

- Romberg — известный метод Ромберга — деления интервала интегрирования пополам до тех пор, пока расхождение результатов не станет меньше заданной погрешности;
- Adaptive — адаптивный метод с изменением шага интегрирования в зависимости от особенностей функции с выдачей результата с заданной переменной TOL погрешностью;
- Infinite Limit — специальный метод, улучшающий интегрирование при бесконечных пределах (применяется автоматически всякий раз, когда хотя бы какой-то из пределов задается бесконечным);
- Singular Endpoint — специальный метод, учитывающий сингулярность (особенность) функции в конечной точке или в конечных точках промежутка интегрирования.

По умолчанию обычно задан адаптивный метод Adaptive, который чаще всего обеспечивает вычисление определенных интегралов с погрешностью не более значения TOL. Влияние значения TOL на результаты интегрирования представлено в приведенных на рис. 6.14 примерах (точное вычисление обеспечивается при использовании оператора символьного вывода \rightarrow).

Повысить точность вычислений интегралов можно и за счет применения оптимизации. Это иллюстрирует следующий пример:

$$f(x) := \int_0^x \sin(t)^3 \cdot e^{-t} dt \quad f(100) = 0.2999999999999997$$

$$g(x) := \int_0^x \sin(t)^3 \cdot e^{-t} dt * \quad g(100) = 0.3$$

	$\int_0^{\infty} \sin(x) \cdot e^{-x} dx \rightarrow \frac{1}{2}$
TOL := 0.1	$\int_0^{\infty} \sin(x) \cdot e^{-x} dx = 0.4999996452$
TOL := 0.01	$\int_0^{\infty} \sin(x) \cdot e^{-x} dx = 0.4999999601$
TOL := 0.0001	$\int_0^{\infty} \sin(x) \cdot e^{-x} dx = 0.5$

Рис. 6.14. Вычисление интеграла при разных значениях погрешности TOL

Признаком оптимизации является красная звездочка (*), располагаемая после оптимизируемого выражения.

6.5.9. Вычисление кратных интегралов

Вставив на место ввода функции шаблон интеграла можно получить шаблон двойного, тройного и так далее, словом кратного, интеграла. Примеры вычисления таких интегралов представлены ниже:

$$\int_0^2 \int_0^3 x \cdot y dx dy = 9$$

$$\int_1^2 \int_1^3 \int_1^4 x \cdot \frac{y}{z} dx dy dz = 20.794$$

Главным недостатком численных методов вычисления кратных интегралов, в том числе по реализованным в Mathcad операторам, является быстрое нарастание времени вычислений с ростом кратности интегралов. Это связано с тем, что численные методы реализуют разбижку пределов интегрирования по каждой переменной в определенное число раз (при адаптивных методах оно зависит от погрешности вычислений). При вычислении двойного интеграла это число возрастает в квадрате, при вычислении тройного интеграла в кубе и так далее. Позже мы рассмотрим возможности аналитического решения кратных интегралов, которое зачастую позволяет резко уменьшить время вычислений.

6.5.10. «Подводные камни» интегрирования

Интегрирование с первого взгляда кажется простым процессом. Особенно при использовании системы Mathcad — захотел вычислить интеграл в аналитическом виде, используй оператор символьного вывода \rightarrow , захотел вычислить определен-

ный интеграл в численном виде — используйте обычный оператор вывода =. И часто и впрямь верный результат получается почти мгновенно. Но далеко не все так просто!

При вычислении интегралов даже с простыми подынтегральными функциями могут возникать серьезные проблемы — от обеспечения нужной точности до полного отказа вычислить заданный интеграл.

Рассмотрим два интеграла с «простой» подынтегральной функцией $1/x$ (рис. 6.15). Первый интеграл вычисляться не желает — выражение с ним окрашивается в красный цвет, а если ввести маркер мыши в блок интеграла, то появляется всплывающая подсказка «Can't converge to a solution». Она говорит о том, что данный интеграл не сходится. Нетрудно догадаться почему — при нулевом нижнем пределе операция $1/x$ некорректна (деление на ноль запрещено). Теперь вполне понятно, что второй интеграл с пределами от 1 до 2 благополучно вычисляется.

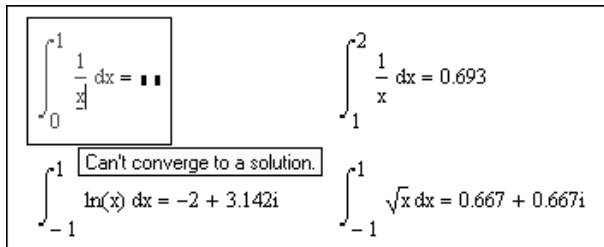


Рис. 6.15. Примеры интегрирования с особенностями

Может быть, надо вообще избегать недопустимых значений аргументов подынтегральной функции? Например, избегать вычисления логарифмов и квадратных корней от отрицательных чисел? Однако это далеко не так верно, как можно предположить. Введение комплексных чисел от такого рода ограничений нередко позволяет избавиться. Два последних примера на рис. 6.10 наглядно иллюстрируют сказанное.

Попробуйте вычислить определенный интеграл от функции $\tan(x)$ с пределами от -10 до 10 . Получится похожий на правду результат 0 . Но стоит увеличить верхний предел с 10 до 12 , как интеграл перестанет вычисляться. Причина таких казусов в том, что функция $\tan(x)$ является разрывной функцией и в точках разрыва ее значения устремляются к $+\infty$ и $-\infty$.

Вообще говоря, как отмечалось, алгоритм численного интегрирования в Mathcad адаптивный и он рассчитан на решение многих трудностей связанных с численным интегрированием. Например, интеграл

$$\int_{-2}^2 \frac{\sin(x)}{x} dx = 3.211 \quad \int_{-2}^2 \frac{\sin(x)}{x} dx \rightarrow 2 \cdot Si(2)$$

благополучно вычисляется, несмотря на деление на 0 в точке $x = 0$. Интересно, что он вычисляется как в численном, так и в аналитическом виде, причем вычисление в численном виде для большинства пользователей более понятно. Тогда как решение в символьном виде подобно «вопросу на вопрос» — специальная функция $Si(x)$ это просто неопределенный интеграл с подынтегральным выражением $\sin(x)/x$. Читатель, вероятно, уже догадался, что причиной благополучного решения в данном случае является устранимая неопределенность $0/0 \rightarrow 1$ в точке $x = 0$.

Вообще говоря, интегралы, имеющие стремящиеся к бесконечности пределы или значения подынтегральной функции, называются *несобственными интегралами*. С их вычислением возможна проблема даже в случае простых подынтегральных функций. Возьмем, к примеру, синусоидальную функцию $f(x) = A \sin(2\pi \cdot f \cdot t + \varphi)$. Это повсюду гладкая функция без каких-либо разрывов и повсюду дифференцируемая. К тому же это периодическая функция ($f(x) = f(x + 2\pi)$), имеющая постоянные параметры — амплитуду A , частоту f и фазовый сдвиг φ , т. е. эта функция времени является стационарной, поскольку ее параметры от времени не зависят. Добавьте к этому, что функция синуса определена от $-\infty$ до $+\infty$. Словом, сплошная гармония, недаром такую функцию и называют гармонической.

Но, увы, даже такая гармоническая функция иногда выдает «пенки» при ее применении в качестве подынтегральной функции. Об этом хорошо говорит рис. 6.16. Интегрирование синусоиды с конечными пределами не вызывает трудностей. Но как только мы задаем бесконечный один предел или оба бесконечных предела, интеграл перестает вычисляться. Причина тут в том, что хотя синусоида и определена в пределах от $-\infty$ до $+\infty$, вычисление ее площади становится неопределенным. И даже противоречащим «здравому смыслу» — вроде бы ясно, что поскольку площади положительной и отрицательной полуволн равны, то в пределе площадь их равна нулю. Но все дело в том, что это справедливо только для целого числа периодов!

Примеры дифференцирования и интегрирования и синусоидальной функции	
$\frac{d}{dx} \sin(x) \rightarrow \cos(x)$	$\frac{d^3}{dx^3} \sin(x) \rightarrow -\cos(x)$
$\int \sin(x) dx \rightarrow -\cos(x)$	$\int_a^b \sin(x) dx \rightarrow -\cos(b) + \cos(a)$
$\int_{-100}^{100} \sin(x) dx = 0$	$\int_0^{\infty} \sin(x) dx \rightarrow \text{undefined}$
$\int_{-\infty}^0 \sin(x) dx \rightarrow \text{undefined}$	$\int_{-\infty}^{\infty} \sin(x) dx \rightarrow \text{undefined}$

Рис. 6.16. Примеры интегрирования и дифференцирования синусоиды ($x = 2\pi f t$, $A = 1$)

Теперь рассмотрим простейшую нестационарную зависимость — затухающую «синусоиду» (рис. 6.17). Ее интегрирование как в конечных пределах, так и при бесконечном верхнем пределе трудностей уже не вызывает, ибо благодаря экспоненциальному затуханию сходимость интеграла в последнем случае обеспечена. Ибо при больших t функция вырождается в прямую, лежащую на оси x и имеющую нулевое значение.

А вот интеграл от синусоиды с экспоненциальным нарастанием сходить не желает (рис. 6.18). И нетрудно понять почему — при больших t эта функция превращается в обычную синусоиду, которая делает интеграл с бесконечным пределом не вычисляемым.

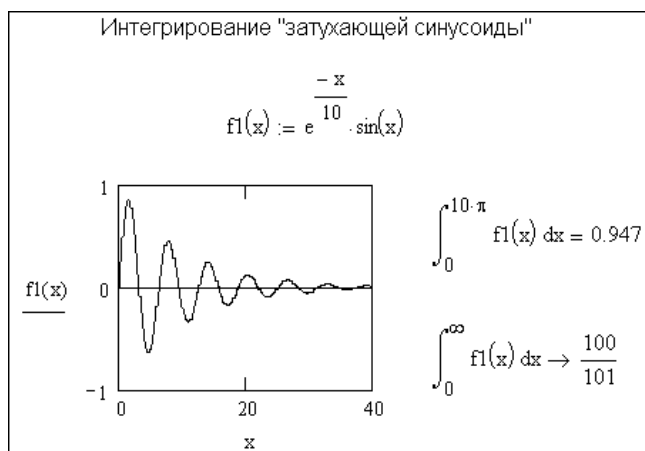


Рис. 6.17. Интегрирование затухающей синусоиды

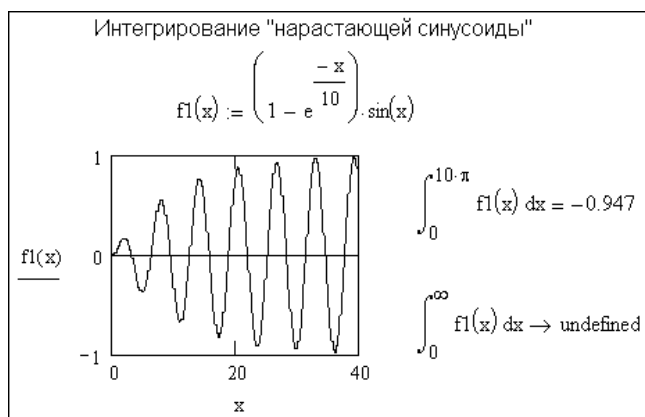


Рис. 6.18. Интегрирование нарастающей синусоиды

Мы остановились столь подробно на интегрировании синусоиды не случайно. Интегралы, содержащие синусы и косинусы с множителями — функциями, встречаются довольно часто. Примером являются коэффициенты Фурье. Трудности их вычисления привели к созданию целого ряда специальных методов интегрирования функций с осциллирующими значениями, например таких как БПФ (быстрое преобразование Фурье).

6.5.11. Задание и графическое представление градиента функции

Иногда необходимо вычислять градиент функции двух переменных. Задание и графическое представление градиента такой функции $f(x, y)$ представлено на рис. 6.14. Нетрудно заметить, что задание градиента сводится к дифференцированию функции по каждой независимой переменной.

Представление градиента на рис. 6.19 дано в виде диаграммы стрелок (рисунок слева), а представление самой функции в виде обычного графика поверхности (рисунок справа).



Рис. 6.19. Задание и графическое представление градиента функции

6.6. Вычисление элементарных функций

Mathcad содержит достаточно широкий набор встроенных элементарных функций [29, 32]. Функции задаются своим именем и значением аргумента, который записывается в круглых скобках. В ответ на обращение к ним функция возвращает вычисленное значение. Аргумент и значение функции могут быть действительными или комплексными числами. Ниже представлены эти функции, разбитые на группы. Полные названия распространенных и общеизвестных функций опущены.

6.6.1. Вычисление типовых элементарных функций

К элементарным функциям относятся следующие встроенные функции системы Mathcad:

- показательные и логарифмические функции: $\exp(z)$, $\ln(z)$, $\log(z, [b])$;
- тригонометрические функции: $\text{angle}(x, y)$, $\sin(z)$, $\cos(z)$, $\tan(z)$, $\sec(z)$, $\csc(z)$, $\cot(z)$;
- обратные тригонометрические функции: $\text{asin}(z)$, $\text{acos}(z)$, $\text{atan}(z)$, $\text{asec}(z)$, $\text{acsch}(z)$, $\text{acot}(z)$;
- гиперболические функции: $\sinh(z)$, $\cosh(z)$, $\tanh(z)$, $\text{sech}(z)$, $\text{csch}(z)$, $\text{coth}(z)$;
- обратные гиперболические функции: $\text{asinh}(z)$, $\text{acosh}(z)$, $\text{atanh}(z)$, $\text{asech}(z)$, $\text{acsch}(z)$, $\text{acoth}(z)$.

Большинство элементарных функций имеет один параметр, и их применение не вызывает никаких трудностей даже у начинающих пользователей. Особо стоит отметить две функции: $\log(z, b)$ — вычисляет логарифм z по основанию b (если b опущено, вычисляется логарифм по основанию 10), $\text{angle}(x, y)$ — возвращает угол (от 0 до 2π) радиус-вектора, имеющего начало в точке $(0,0)$ и конец в точке (x,y) .

6.6.2. Функции комплексного аргумента

Ряд функций определен для комплексного аргумента:

- $\arg(z)$ — вычисление аргумента комплексного числа z ;
- $\text{Im}(z)$ — выделение мнимой части комплексного числа z ;
- $\text{Re}(z)$ — выделение действительной части комплексного числа z ;
- $\text{csgn}(z)$ — возвращает либо 0, если $z = 0$, либо 1, если $\text{Re}(z) > 0$ или если $\text{Re}(z) = 0$ и $\text{Im}(z) > 0$, либо -1 в остальных случаях;
- $\text{signum}(z)$ — возвращает 0, если $z = 0$, и $z/|z|$ в ином случае.

На рис. 6.20 показан фрагмент документа, иллюстрирующий работу с элементарными функциями как при действительном, так и при комплексном аргументе z .



Рис. 6.20. Примеры вычисления значений элементарных функций

Набор элементарных функций в системе Mathcad функционально полон, точнее, он даже избыточен. Например, гиперболические функции можно вычислить через экспоненту, тангенс и котангенс легко выражаются через синус и косинус и т. д. Соответствующие формулы общеизвестны и потому здесь не приводятся. Полный перечень функций дан в приложении 4.

Элементарные функции характеризуются своими графиками. Поскольку построение их графиков с помощью средств системы Mathcad превратилось в тривиальную задачу, такие графики не приводятся. Напомним, что для построения графиков достаточно ввести запись функции, например $\sin(x)$, вывести шаблон двумерного графика (например, вводом символа @) и ввести имя независимой переменной — x . Щелкнув мышью в стороне от графика, можно наблюдать его построение.

6.6.3. Функции комбинаторики и теории чисел

Mathcad имеет небольшое число функций комбинаторики и теории чисел. Помимо оператора факториала $n!$ имеются следующие функции:

- `combin(n, k)` — возвращает число сочетаний k из n , где $n > k$;
- `lcm(v)` — целое положительное число, которое является наименьшим общим кратным для всех элементов вектора v , имеющего не менее двух элементов типа `real` или двух целых неотрицательных чисел;
- `gcd(v)` — целое число, которое является наибольшим общим делителем для всех элементов вектора v , содержащего не менее двух элементов типа `real` или двух целых неотрицательных чисел;
- `mod(x, y)` — остаток от деления x на y , записанный со знаком числа x ;
- `permut(n, k)` — возвращает число размещений из n элементов по k , причем n и k должны быть целыми неотрицательными числами.

Факториал целого числа n определен как $n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$, причем $0! = 1$. Значение факториала быстро нарастает с ростом n :

$10! \rightarrow 3628800$

$20! \rightarrow 2432902008176640000$

$50! \rightarrow 30414093201713378043612608166064768844377641568960512000000000000$

Факториал $n!$ задает число перестановок множества из n элементов. Числа сочетаний и размещений, вычисляемые функциями `combin` и `permut`, определяется как:

$$\text{combin}(n, k) = \frac{n!}{k!(n-k)!} \quad \text{permut}(n, k) = \frac{n!}{(n-k)!}$$

Введение этих функций в значительной степени устраняет проблемы, связанные с вычислениями факториалов больших чисел. На рис. 6.21 представлены примеры работы с функциями комбинаторики и теории чисел. В частности, построены логарифмические зависимости для факториала и других функций комбинаторики.

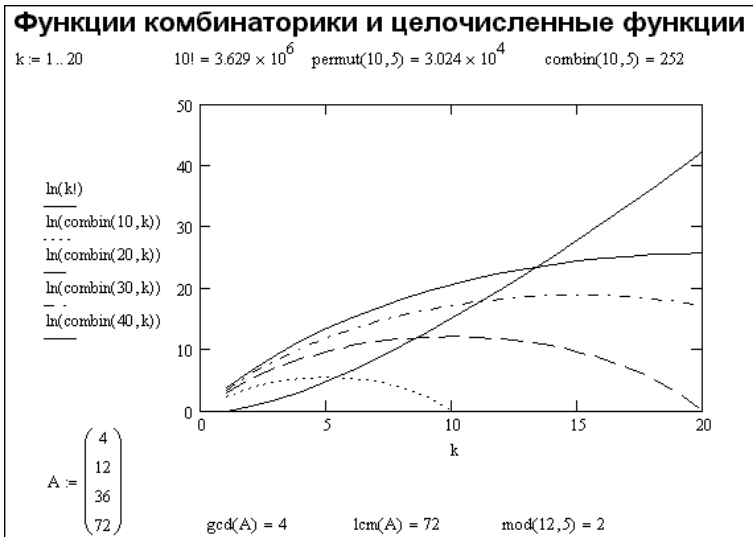


Рис. 6.21. Примеры вычисления функций комбинаторики и теории чисел

6.7. Вычисление специальных математических и иных функций

Наряду с элементарными функциями в системе Mathcad содержится ряд встроенных специальных математических функций [29]. Их применение расширяет возможности системы при решении сложных математических задач.

6.7.1. Вычисление функций Бесселя

К важнейшим из специальных математических функций принадлежат функции Бесселя, являющиеся решениями дифференциального уравнения второго порядка:

$$x^2 \frac{d^2 y}{dx^2} + x \frac{dy}{dx} + (x^2 - n^2)y = 0.$$

Здесь n — порядок функции (рассматриваются функции только целого порядка при действительном значении x). Функции Бесселя описывают колебательные процессы и широко используются в физике, электро- и радиотехнике. Приведенное уравнение имеет решения в виде функций Бесселя $J_n(x)$ первого рода и $Y_n(x)$ второго рода.

Существуют также *модифицированные* функции Бесселя, которые являются решениями следующего дифференциального уравнения:

$$x^2 \frac{d^2 y}{dx^2} + x \frac{dy}{dx} - (x^2 + n^2)y = 0.$$

Эти функции также представлены функциями первого рода $I_n(x)$ и второго рода $K_n(x)$.

Функции Бесселя используются с вещественным аргументом. Ниже приведен список функций Бесселя, вычисление которых реализовано в системе Mathcad:

- $J_0(x)$ — функция Бесселя первого рода нулевого порядка;
- $I_0(x)$ — модифицированная функция Бесселя первого рода нулевого порядка;
- $Y_0(x)$ — функция Бесселя второго рода нулевого порядка;
- $K_0(x)$ — модифицированная функция Бесселя второго рода нулевого порядка;
- $J_1(x)$ — функция Бесселя первого рода первого порядка;
- $I_1(x)$ — модифицированная функция Бесселя первого рода первого порядка;
- $Y_1(x)$ — функция Бесселя второго рода первого порядка;
- $K_1(x)$ — модифицированная функция Бесселя второго рода первого порядка;
- $J_n(n, x)$ — функция Бесселя первого рода n -го порядка;
- $I_n(n, x)$ — модифицированная функция Бесселя первого рода n -го порядка;
- $Y_n(n, x)$ — функция Бесселя второго рода n -го порядка;
- $K_n(n, x)$ — модифицированная функция Бесселя второго рода n -го порядка.

Эти функции есть во всех вариантах поставки Mathcad, начиная с версии Mathcad 8. Читателю рекомендуется построить графики функций Бесселя, если они его интересуют.

6.7.2. Вычисление гамма-функции

Другой широко распространенной специальной функцией, вычисление которой (причем как при действительном, так и при комплексном аргументе z) предусмотрено в системе Mathcad, является гамма-функция ($\Gamma(z)$). Она широко применяется в статистических расчетах, в которых используется также функция ошибок $\text{erf}(x)$, называемая еще интегралом вероятности.

На рис. 6.22 представлен фрагмент документа с примерами вычисления ряда специальных математических функций и графиками этих функций.

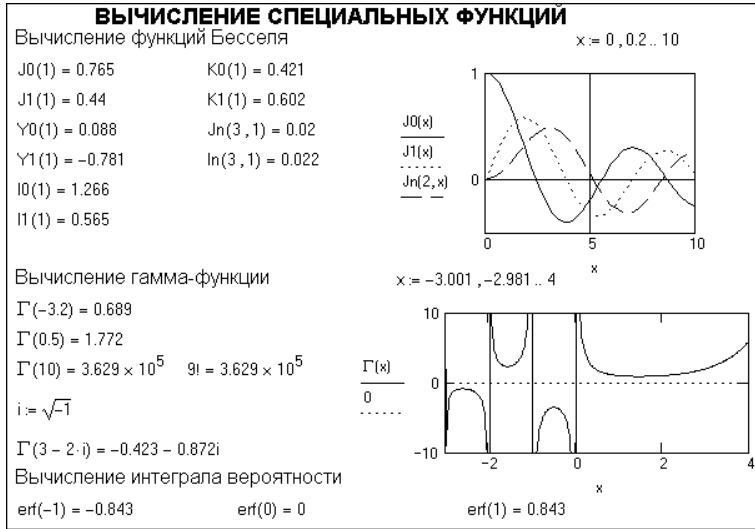


Рис. 6.22. Вычисление специальных математических функций

Наличие встроенных в систему наиболее распространенных математических функций расширяет ее возможности. Многие другие специальные математические функции могут быть определены через перечисленные встроенные функции или заданы своим интегральным либо дифференциальным представлением.

6.7.3. Дополнительные специальные функции

В систему Mathcad 8.0 PRO было введено около 50 новых функций. Среди них имеется ряд специальных математических функций.

- Дополнительные функции Бесселя:
 - $Ai(x)$ — функция Эйри первого рода;
 - $bei(n, x)$ — мнимая часть функции Бесселя—Кельвина порядка n ;
 - $ber(n, x)$ — действительная часть функции Бесселя—Кельвина порядка n ;
 - $Bi(x)$ — функция Эйри второго рода;
 - $js(n, x)$ — сферическая функция Бесселя первого рода целого порядка n ($n > 200$) в точке x ($x > 0$);
 - $ys(n, x)$ — сферическая функция Бесселя второго рода порядка n ($n > 200$) в точке x ($x > 0$).

- Гипергеометрические функции:
 - $\text{dhypergeom}(m, n, M, N)$ — гипергеометрическая функция;
 - $\text{fhyper}(a, b, c, x)$ — гипергеометрическая функция Гаусса в точке x с параметрами a, b и c ;
 - $\text{mhyper}(a, b, x)$ — конфлюэнтная гипергеометрическая функция в точке x с параметрами a и b .
- Ортогональные полиномы:
 - $\text{Her}(n, x)$ — полином Эрмита степени n с аргументом x ;
 - $\text{Jac}(n, a, b, x)$ — полином Якоби степени n в точке x с параметрами a и b ;
 - $\text{Lag}(n, x)$ — полином Лагерра степени n в точке x ;
 - $\text{Leg}(n, x)$ — полином Лежандра степени n в точке x ;
 - $\text{Tcheb}(n, x)$ — полином Чебышева первого рода степени n в точке x ;
 - $\text{Ucheb}(n, x)$ — полином Чебышева второго рода степени n в точке x .
- Другие функции:
 - $\text{ibeta}(a, x, y)$ — неполная бета-функция параметров a, x и y .

Появление этих функций делает систему Mathcad PRO более удобной в применении, поскольку исключает задание данных функций пользователем. Эти функции потому и названы специальными, что не имеют простых представлений через элементарные функции и требуют для своего вычисления применения специальных методов. Определения данных функций даны в справочной системе Mathcad. Все эти функции включены в состав систем Mathcad 2001i и Mathcad 11.

6.7.4. Альтернативные функции с новой нормировкой в Mathcad 11/12

В Mathcad 11 был добавлен ряд новых функций со специальной нормировкой. Эти функции относятся к классу функций Бесселя и их можно найти в списке функций Бесселя в окне, которое открывается активизацией кнопки $f(x)$ в панели инструментов. В Mathcad 12 к ним добавились также альтернативные функции Эйри из этой же группы. Имена альтернативных функций состоят из имени основной функции и расширения .sc. Например, если основная функция $Y_0(z)$, то альтернативная будет $Y_0.sc(z)$. В справке и в окне кнопки $f(x)$ указываются нормирующие множители для альтернативных функций. Читатель может легко построить графики основных и альтернативных функций и сравнить их между собой.

6.7.5. Дополнительные неактивные функции

При загрузке символьного процессора система распознает ряд дополнительных специальных функций:

- $\text{FresnelC}(x)$ — интеграл Френеля $C(x)$;
- $\text{FresnelS}(x)$ — интеграл Френеля $S(x)$;
- $\text{Ci}(x)$ — интегральный косинус;
- $\text{Si}(x)$ — интегральный синус;
- $\text{Ei}(x)$ — интегральная показательная функция;
- $\text{dilog}(x)$ — дилогарифм;

- $\text{Psi}(n, x)$ — n -я производная пси-функции и т. д.

К сожалению, статус этих функций необычен — они могут фигурировать в *результатах* символьных операций, но недоступны для обычных операций. Обращение к этим функциям не приводит к возвращению их значений. Эти функции, по существу, являются просто текстовыми записями. Например, невозможно вычислить значения функций при подстановке в них числового значения аргумента и применении знака = для вывода результата вычислений. Однако, используя известные (в основном интегральные) представления для этих функций, их можно определить как функции пользователя. Выражения с неактивными функциями Mathcad размещает в буфере обмена, и их можно извлечь оттуда для просмотра.

Здесь не приводятся представления для всех этих функций, поскольку их легко найти в справочной системе Mathcad.

6.7.6. Задание функций пользователя

Несмотря на довольно широкий набор встроенных функций, очень часто возникает необходимость расширить систему новыми функциями, представляющими интерес для пользователя. Функции пользователя вводятся с применением следующей конструкции:

Имя_функции(Список_аргументов) := Выражение

Имя функции задается как любой идентификатор, например имя переменной. В скобках указывается список аргументов функции — это перечень используемых в выражении переменных, записанных через запятую. Выражение — любое выражение, содержащее доступные системе операторы и функции с операндами и аргументами, указанными в списке параметров.

Примеры задания функций пользователя одной и двух переменных:

$\text{fun}(x) := 5(1 - \exp(x))$

$\text{module}(x, y) := \sqrt{x^2 + y^2}$

Следует отметить особый статус переменных, указанных в списке параметров функции пользователя. Эти переменные являются *локальными*, поэтому они могут не определяться до задания функции — фактически их указание в списке аргументов и является заданием определенного статуса этих переменных. Естественно, что локальные переменные могут использоваться только в выражении, описывающем функцию. Их имена могут совпадать с именами глобальных переменных, введенных ранее. Но при этом по выходе из блока задания функции значения этих переменных будут сохранять ранее заданные (для глобальных переменных) значения.

Следует с осторожностью относиться к применению *глобальных* переменных в теле функции пользователя, хотя это и возможно (но, конечно, только в том случае, когда глобальная переменная определена ранее). Например, вполне приемлемо задание следующей функции пользователя, поскольку переменная A определена до задания функции:

$A := 10$

$\text{fun}(x) := A(1 - \exp(x))$

Если убрать строку с определением переменной, задание функции $\text{fun}(x)$ станет невозможным, точнее, во введенном выражении переменная A будет выделена как несуществующая. Разумеется, эту переменную можно определить глобально и

после задания функции пользователя, используя для этого символ глобального присваивания \equiv :

```
fun(x) := A(1 - exp(x))
A ≡ 10
```

Такая конструкция уже верна, так как глобально заданная переменная A может быть определена в любом месте документа, в том числе в теле функции $fun(x)$.

Еще ряд примеров задания функций пользователя приведен на рис. 6.23.

ЗАДАНИЕ И ПРИМЕНЕНИЕ ФУНКЦИЙ ПОЛЬЗОВАТЕЛЯ	
$f(x) := x \cdot \sin(x)$	Функция одной переменной, заданная локально
$z := 0.5$	Переменной z присвоено значение 0.5
$f(z) = 0.24$	Пример вычисления функции $f(z)$
$zxy(0.5, 2) = 4.25$	Эта, заданная глобально, функция двух переменных вычисляется, хотя задана в конце документа
$zxy(z, 2) = 4.25$	
$\int_1^2 f(x) dx = 1.44$	Этот пример иллюстрирует применение функции пользователя в составе математического выражения
$\int f(x) dx \rightarrow \sin(x) - x \cdot \cos(x)$	Этот пример оказывает применение функции пользователя в символьных вычислениях
$zxy(x, y) \equiv x^2 + 2 \cdot x \cdot y + y$ Функция пользователя, заданная глобально	

Рис. 6.23. Примеры задания и применения функций пользователя

Заданные с применением оператора $:=$ функции являются *заданными локально*, поэтому они должны вводиться в документ до того, как будут использованы. Функцию пользователя можно задать и с применением оператора \equiv , тогда она будет определена как глобальная. В этом случае она может задаваться в любом месте документа, в том числе и в конце.

Например, в представленном на рисунке фрагменте документа следующая функция определяется глобально:

```
zxy(x, y) ≡ x + 2 · x · y + y
```

Выражения с этой функцией вполне корректно вычисляются, хотя и находятся в документе выше, чем определение функции.

Если же функция нигде и никак не определена, то применять ее, естественно, нельзя. Попытки такого применения будут сопровождаться сообщениями об ошибке — имя функции окажется выделенным красным цветом.

Функции пользователя — мощный аппарат расширения системы. Они облегчают адаптацию системы к решению специфических задач пользователя. Напоминаем еще раз, что, к сожалению, функции пользователя нельзя применять в составе выражений, над которыми осуществляются символьные операции с помощью команд меню (см. главу 9), однако преобразования с применением оператора \rightarrow системы SmartMath могут выполняться и без этого ограничения. Например, в по-

казанном на рисунке фрагменте документа одна из функций пользователя определяется следующим образом:

$$f(x) := x \cdot \sin(x)$$

Далее эта функция используется в символьных вычислениях:

$$\int f(x)dx \rightarrow \sin(x) - x \cdot \cos(x)$$

6.7.7. Числовые функции с условиями сравнения

Существует ряд встроенных функций, у которых возвращаемый результат зависит от знака или значения аргумента. При их вычислении производится сравнение аргумента с некоторыми числовыми константами, например, с нулем или какими-либо другими целыми числами. Ниже представлены такие функции:

- $\text{ceil}(x)$ — наименьшее целое, большее или равное x ;
- $\text{floor}(x)$ — наибольшее целое, меньшее или равное x ;
- $\text{round}(x, n)$ — округленное значение вещественного x с точностью до n знаков после десятичной точки (если $n < 0$, то x округляется слева от десятичной точки);
- $\Phi(x)$ — функция Хевисайда — возвращает 0 при $x < 0$ и 1 в противном случае;
- $\delta(m, n)$ — функция, именуемая символом Кронекера, возвращает 1 при $m=n$ и 0 в противном случае (в Mathcad 12 эта функция изъята).

Назначение этих функций довольно очевидно. К примеру, функция Хевисайда может использоваться для задания прямоугольного импульса с шириной w :

$$\text{pulse}(t, w) := \Phi(t) - \Phi(t-w)$$

Функцию Хевисайда можно также использовать для создания логических функций, которых нет в старых версиях системы Mathcad:

- $\text{not}(a) := \Phi(.5-a)$ — функция логического отрицания;
- $\text{and}(a, b) := \Phi(a \cdot b - .5)$ — функция логического перемножения;
- $\text{or}(a, b) := \Phi(a+b - .5)$ — функция логического сложения.

Эти функции используются с параметрами a и b , которые имеют логические значения 0 (FALSE) и 1 (TRUE). Функции возвращают результат (0 или 1) в соответствии с таблицами истинности логических функций NOT, AND и OR. Аналогичным образом можно задать и другие логические функции.

В Mathcad 2001i/11 введены следующие новые функции этого класса:

- $\text{sign}(x)$ — функция знака (возвращает либо 0, если $x = 0$, либо 1, если $x > 0$, либо -1 в остальных случаях);
- $\text{trunc}(x)$ — целая часть от действительного числа x .

Эти довольно очевидные функции призваны сократить работу пользователя по их заданию в тех задачах, где такие функции встречаются.

6.7.8. Функция условных выражений

Довольно широкие возможности дает функция if , предназначенная для создания условных выражений:

$$\text{if}(\text{Условие}, \text{Выражение 1}, \text{Выражение 2})$$

Если в этой функции Условие выполняется, то будет вычисляться и возвращаться Выражение 1, в противном случае — Выражение 2. Заметим, что эта функция широко распространена в языках программирования.

На рис. 6.24 приведен пример применения функции `if` для моделирования процесса однополупериодного выпрямления, который используется в электротехнике для преобразования переменного тока в пульсирующий однополярный ток. Этот процесс иллюстрируется наглядной графикой. В данном случае соответствующие сигналы моделируются следующим образом:

- синусоидальный сигнал:

$$f(x) := \sin(x)$$

- сигнал однополупериодного выпрямления:

$$y(x) := \text{if}(f(x) \geq 0, f(x), 0)$$

- сигнал двухполупериодного выпрямления:

$$z(x) := \text{if}(f(x) \geq 0, f(x), -f(x))$$

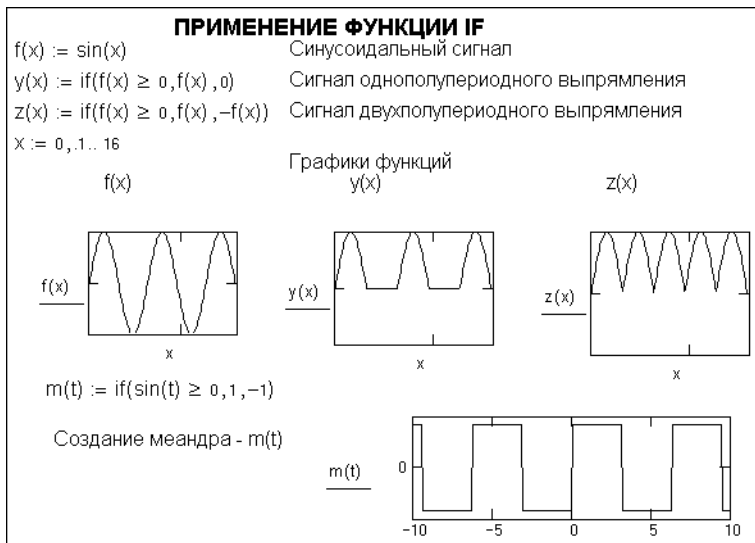


Рис. 6.24. Применение функции `if`

С помощью функций с элементами сравнения можно моделировать и другие процессы, например, логические или описывающие импульсные сигналы различного вида. В качестве примера на рисунке задана функция $m(t)$:

$$m(t) := \text{if}(\sin(t) \geq 0, 1, -1)$$

Эта функция имитирует прямоугольные периодические колебания, называемые меандром — они имеют равные по длительности полуволны с амплитудой 1 для положительной полуволны и -1 — для отрицательной.

6.7.9. Рекурсивные функции

Важно отметить, что Mathcad (до версии 12) допускает внутри тела функции пользователя задавать обращение к самой этой функции. Это означает возмож-

ность создания *рекурсивных функций*, порой заметно упрощающих реализацию различных алгоритмов. Функция `if` также позволяет создавать рекурсивные алгоритмы. Ниже приведен пример создания рекурсивного алгоритма для вычисления факториала:

```
fact(n) := if(n=0, 1, nfact(n-1))
fact(0) = 1
fact(5) = 120
```

Этот пример носит чисто познавательное значение, поскольку в Mathcad имеется оператор, позволяющий вычислять факториал — $n!$.

Примером полезности таких функций может служить моделирование периодических сигналов, широко используемых для испытания различных электронных устройств. Например, линейно падающее периодическое пилообразное колебание $P(x)$ с периодом T можно создать, используя следующие выражения:

```
F(x) := 1-x
P(x) := if(x<T, F(x), P(x-T))
```

Однако не следует увлекаться слишком глубокой рекурсией, так как это может привести к переполнению стека рекурсий системы Mathcad, что приведет к выводу сообщения об ошибке — `Stack overflow` (Стек переполнен). Хотя рекурсия — мощный математический прием, в большинстве случаев более эффективным является решение задач без ее применения. Учитывая это, в Mathcad 12 задание рекурсивных и рекуррентных функций запрещено.

6.7.10. Функции контроля типа переменных

Начиная с Mathcad 2000 появились четыре новые функции контроля типа переменных:

- `IsArray(x)` — возвращает 1, если x — матрица или вектор, иначе возвращает 0;
- `IsScalar(x)` — возвращает 1, если x — скаляр, иначе возвращает 0;
- `IsString(x)` — возвращает 1, если x — строка, иначе возвращает 0;
- `UnitsOf(x)` — возвращает размерность x , если x — размерная переменная, иначе возвращает 1.

Работа этих функций вполне очевидна, и мы не будем комментировать их более подробно. Читатель может легко проверить действие данных функций самостоятельно. В Mathcad 12 к этим функциям добавилась `IsNaN(x)`. Она возвращает значение логической 1, если x типа NaN, и логический 0 в ином случае.

6.7.11. Функции обработки строк

Впервые в версию Mathcad 7.0 PRO был включен новый тип функций — строковые функции. Есть он и в новых версиях Mathcad. Напоминаем, что строка S задается как цепочка символов, заключенная в кавычки. Имеются следующие функции для работы со строками:

- `concat(S1, S2)` — строковая переменная, полученная объединением строковых переменных или констант $S1$ и $S2$;
- `num2str(z)` — строковое представление числа z ;

- `str2num(S)` — преобразование строкового представления числа (в любой форме) в вещественное число;
- `str2vec(S)` — преобразование в вектор строки `S` с записями вещественных чисел в строковом формате;
- `strlen(S)` — количество знаков в строке `S`;
- `vec2str(v)` — строковое представление вектора `v`.

Рассмотрим ряд примеров применения строковых функций. Пусть заданы две строки (`A` и `B`) и два числа (`N` и `M`):

```
A := "Hello" B := "my friend!"
N := 0M := 4
```

При таких исходных данных вызов соответствующих функций приведет к следующим результатам:

- объединение строк `A` и `B`:

```
concat(A,B) = "Hello my friend!"
```

- число символов в строке `A`:

```
strlen(A) = 6
```

- число позиций перед символом "i" в строке `B`:

```
search(B,"i",N) = 5
```

- часть строки `B`, начиная с позиции `N` и кончая позицией `M`:

```
substr(B,N,M) = "my f"
```

- преобразование строки в число:

```
str2num("2 + 3i") = 2 + 3i
```

- преобразование числа в строку:

```
num2str(123·1051) = "1.23e + 053"
```

- преобразование вектора ASCII-кодов в строку:

$$\text{vec2str} \begin{pmatrix} 97 \\ 98 \\ 99 \end{pmatrix} = \text{"abc"}$$

- преобразование строки в вектор ASCII-кодов:

$$\text{str2vec}(\text{"Hello"}) = \begin{pmatrix} 72 \\ 101 \\ 108 \\ 108 \\ 111 \end{pmatrix}$$

При работе со строковыми функциями полезна системная переменная `CWD`. Она возвращает строку с полным путем к папке, в которой хранится текущий документ. Например:

```
CWD = "C:\BOOK\MCAD2000\The book\"
```


Как видно из приведенных примеров, строковые значения могут быть присвоены переменным, и над ними можно выполнять различные операции — объединять строки, выделять из них подстроки, преобразовать строки в числа и в ASCII-коды (и наоборот), вычислять длину строки и так далее. Строковые функции можно применять и в программных конструкциях.

В Mathcad 12 работа со строковыми функциями улучшена. В частности, переменная ORIGIN теперь позволяет указывать индекс массивов с кодами, начиная с которых ведется их обработка и преобразование.

6.7.12. Вычисление различных специальных математических функций

В Mathcad 2001 поддерживается приличное число специальных математических функций, которое к тому же растет от версии к версии. Однако многие специальные математические функции по-прежнему в систему не включены, и в этом нет особой нужды, поскольку они легко вычисляются через встроенные операторы и функции системы. Необходимые выражения описаны в литературе [29].

Например, дилогарифм можно вычислить через определенный интеграл, а си-нусный интеграл через сумму членов ряда:

$$di \log(x) := \int_1^x \frac{1}{1-t} \cdot \ln(t) dt$$

$$Si(x) := \sum_{n=0}^{50} \frac{(-1)^n \cdot x^{2n+1}}{(2 \cdot n + 1) \cdot 2 \cdot n + 1!}$$

Примеры вычисления этих и других выражений иллюстрируют фрагменты документа Mathcad, показанные на рис. 6.25 и 6.26.

Вычисление некоторых специальных функций средствами Mathcad - часть 1	
<p>Константа Каталана</p> $G = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2 \cdot n + 1)^2} \int_0^{\frac{\pi}{2}} \operatorname{asinh}(\sin(x)) dx = 0.915965594$	<p>Гиперболический косинусный интеграл</p> $\gamma := .5772156649015328606065120900824024310422 \quad N = 50$
<p>Косинусный интеграл</p> $Ci(x) = \gamma + \ln(x) + \sum_{n=1}^N \frac{(-1)^n \cdot x^{2n}}{(2 \cdot n) \cdot 2 \cdot n!} \quad Ci(2) = 0.422980829$	<p>Chi(x) := \gamma + \ln(x) + \sum_{n=1}^N \frac{x^{2n}}{(2 \cdot n) \cdot 2 \cdot n!} \quad Chi(2) = 2.452666923 </p>
<p>Дилогарифм</p> $di \log(x) := \int_1^x \frac{1}{1-t} \cdot \ln(t) dt \quad di \log(.5) = 0.582240526$	<p>Интегральная показательная функция</p> $N := 100 \quad Ei(x) := \gamma + \ln(x) + \sum_{n=1}^N \frac{x^n}{n \cdot n!} \quad Ei(2) = 4.954234356$
<p>Константа Эйлера</p> $\lim_{N \rightarrow \infty} \left(\sum_{k=1}^N \frac{1}{k} - \ln(N) \right) \rightarrow \gamma$	<p>Комплексная функция ошибок</p> $z \operatorname{erf}(z) := \frac{2}{\sqrt{\pi}} \sum_{k=0}^N \frac{(-1)^k \cdot z^{2 \cdot k + 1}}{k! \cdot 2 \cdot k + 1} := \sqrt{-1} \quad \operatorname{erf}(1) = 0.842700793$ $z \operatorname{erf}(1 + i) = 1.316151 + 0.190453i \quad z \operatorname{erf}(1) = 0.842700793$
<p>Интеграл Френеля косинусный</p> $\operatorname{FresnelC}(x) = \int_0^x \cos\left(\frac{\pi}{2} \cdot t^2\right) dt \quad \text{TOL} = 10^{-8} \quad \operatorname{FresnelC}(2) = 0.483253$	<p>Интеграл Френеля синусный</p> $\operatorname{FresnelS}(x) = \int_0^x \sin\left(\frac{\pi}{2} \cdot t^2\right) dt \quad \text{TOL} = 10^{-8} \quad \operatorname{FresnelS}(2) = 0.343416$
<p>Неполный эллиптический интеграл 1-го рода</p> $\operatorname{LegendreF}(x, k) := \int_0^x \frac{1}{\sqrt{(1-t^2) \cdot (1-k^2 \cdot t^2)}} dt$	<p>Полный эллиптический интеграл 1-го рода</p> $\operatorname{LegendreKc}(k) := \int_0^1 \frac{1}{\sqrt{(1-t^2) \cdot (1-k^2 \cdot t^2)}} dt$
$\operatorname{LegendreF}\left(\frac{1}{3}, \frac{1}{2}\right) = 0.3415$	$\operatorname{LegendreKc}(.5) = 1.68575$

Рис. 6.25. Примеры вычисления специальных функций (начало)

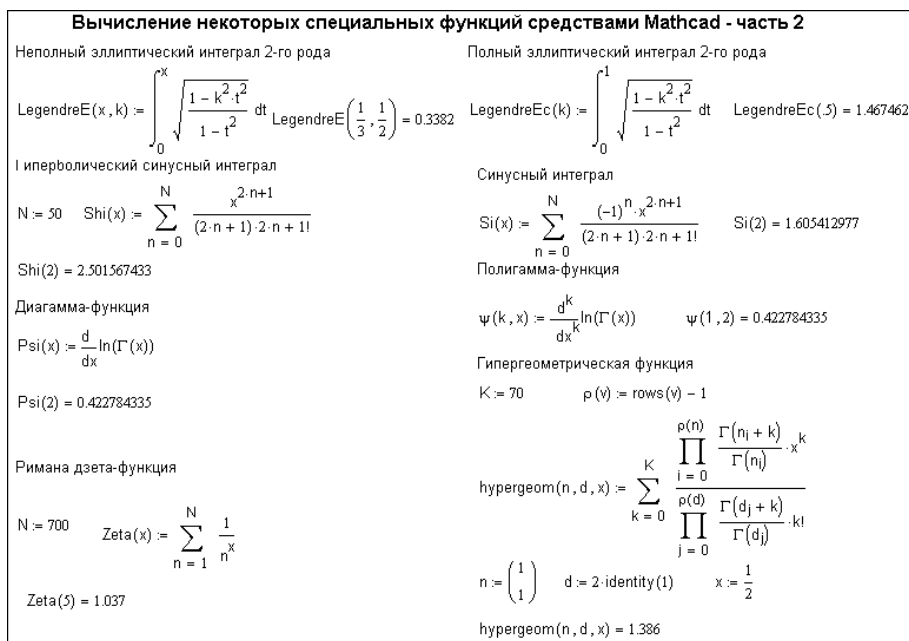


Рис. 6.26. Примеры вычисления специальных функций (окончание)

Из этих примеров хорошо видны основные методы вычисления специальных функций — используются их представления через суммы или интегралы. В подавляющем большинстве случаев вычисления реализуются численными методами и с погрешностью, вполне удовлетворяющей потребностям практики.

6.8. Решение задач линейной алгебры

Решение задач линейной алгебры является одной из важных сфер применения различных систем компьютерной математики [28]. Большие возможности в этом представляют и системы Mathcad, хотя лидерами в области решения задач линейной алгебры являются системы класса MATLAB [26, 27].

6.8.1. Массивы, векторы и матрицы

Центральным понятием линейной алгебры является понятие массива, т. е. обозначенной именем совокупности данных. К таким простейшим данным принадлежат векторы — одномерные массивы. Ранжированная переменная отличается от вектора тем, что невозможно использование ее отдельных значений, скажем, второго или пятого, — она существует «вся сразу», к отдельным ее значениям доступа нет. При необходимости иметь доступ к каждому значению переменной с несколькими компонентами она должна быть задана в виде *массива* — одномерного (вектора) или двумерного (матрицы). Доступ означает, что каждому элементу вектора или массива может быть присвоено значение и к каждому элементу можно обратиться для считывания этого значения. Так, элементами вектора, например V , являются индексированные переменные V_i . Здесь i — номер, или индекс, элемента массива V .

Массивы в общем случае могут содержать как числовые, так и символьные данные:

- $\begin{pmatrix} 2 \\ 4 \\ 8 \end{pmatrix}$ — вектор-столбец с числовыми данными;
- $(a \ b + c \ d)$ — вектор-строка с символьными данными;
- $\begin{pmatrix} \sin(1) & 1 & 0 \\ 1 & a + b & 1 \\ 0 & 1 & \cos(1) \end{pmatrix}$ — матрица с элементами различного типа.

Как уже отмечалось, в Mathcad 2001 векторы и матрицы отображаются не в квадратных скобках, как это было в прежних версиях Mathcad, а в длинных круглых скобках. Таким образом, исключается возможность отождествления векторов с таблицами значений ранжированных переменных.

ПРИМЕЧАНИЕ *Mathcad позволяет использовать одномерные и двумерные массивы — векторы и матрицы. Применение многомерных массивов не предусмотрено. Элементами матриц в ряде случаев могут быть векторы и матрицы, что по существу означает возможность имитации многомерных массивов. Однако надо учитывать, что в части работы с массивами большого размера (в том числе многомерными) значительными преимуществами обладает матричная система MATLAB. В Mathcad есть средства для совместной работы с этой системой.*

6.8.2. Использование индексированных переменных

В Mathcad массив, как и любая другая переменная, задается именем. Местоположение элемента задается одним индексом для вектора или двумя индексами для матрицы. Нижняя граница индексации определяется системной переменной `ORIGIN`, которая может принимать значение 0 или 1 (по умолчанию ее значение равно 0). Индексы могут быть только целыми положительными числами (и нулем).

Для ввода индекса используется клавиша `[` — прямая открывающаяся скобка. Не следует путать индексированные переменные со скалярными переменными, имеющими индекс в своем имени, например, ток I_1 , где нижний индекс имени означает просто «первый ток», или ток в ветви 1 некоторой электрической схемы. Подобные индексы — индексы в имени переменной — вводятся с помощью точки, причем синий уголок курсора ввода при этом охватывает все имя, а не только область ввода индекса. К сожалению, индексированные переменные и переменные с индексами в имени на экране дисплея плохо различимы: единственное отличие состоит в том, что индекс в имени скалярной переменной с индексом зрительно отделен от собственно имени пробелом, то есть он находится правее, чем индекс индексированной переменной. Выглядит это примерно так:

- x_1 — скалярная переменная с индексом;
- x_1 — индексированная переменная.

Элементы матриц также являются индексированными переменными, имена которых совпадают с именами матриц. Например, можно задать следующую матрицу:

$$M := \begin{pmatrix} a & 1 & 0 \\ 1 & a+b & 1 \\ 0 & 1 & b \end{pmatrix}$$

В этом случае указывают два индекса — один для номера строки, другой для номера столбца. Например, если `ORIGIN = 0`, средний элемент указанной матрицы `M` обозначается как `M1,1`, а правый нижний — как `M2,2`.

6.8.3. Ввод элементов векторов и матриц

Векторы и матрицы можно задавать путем ввода их элементов — индексированных переменных. Для указания подстрочных индексов после имени переменной нажимается клавиша символа открывающей квадратной скобки.

Клавиши	Отображаемое значение
<code>V3[2:</code>	<code>V3₂ :=</code>

Для элементов матрицы подстрочные индексы вводятся аналогично с разделением их запятыми.

Клавиши	Отображаемое значение
<code>V3[1,2:</code>	<code>V3_{1,2} :=</code>

Для задания векторов и матриц можно либо воспользоваться командой `Matrix` (Матрицы) меню `Insert` (Вставка), либо нажать комбинацию клавиш `Ctrl+V`, либо щелкнуть на кнопке с изображением шаблона матрицы. Любое из этих действий вызывает появление диалогового окна, в котором надо указать размер матрицы, то есть количество ее строк `m` и столбцов `n` (см. подраздел «Вставка шаблона матриц и векторов» в разделе «Вставка неграфических шаблонов» главы 4). Для векторов один из этих параметров должен быть равен 1. При `m = 1` получим вектор-строку, а при `n = 1` — вектор-столбец. Матрица является двумерным массивом с числом элементов `m × n`.

6.8.4. Задание векторов и матриц

В отношении индексированных переменных действуют те же правила присваивания и вывода, что и для обычных переменных. В частности, с помощью оператора присваивания можно создать массив (вектор или матрицу) заданного размера и заданного типа без ручного заполнения их шаблонов. Например, вектор можно задать прямым присваиванием значений некоторым элементам вектора:

$$V_0 := 1 \quad V_2 := 2 \quad V_3 := 3$$

В результате будет получен следующий вектор:

$$V = \begin{pmatrix} 1 \\ 2 \\ 0 \\ 3 \end{pmatrix}$$

Аналогично (то есть без заполнения шаблона) можно создать, например, нулевую матрицу, используя ранжированные переменные:

```
j := 0..2 i := 0..2
M0i,j := 0
```

В результате будет получена следующая матрица:

$$M0 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Здесь представлен вариант отображения массива в виде обычной матрицы. Возможен также вариант отображения в виде *электронной таблицы*, которая имеет полосы прокрутки, обеспечивая представление массивов больших размеров.

6.8.5. Векторные и матричные операции

Для работы с векторами и матрицами система Mathcad поддерживает ряд специальных операторов и функций. Вначале рассмотрим операторы, придерживаясь следующих обозначений:

- V — вектор;
- M — матрица;
- Z — скалярная величина.

Ниже представлены основные векторные и матричные операции, решающие простейшие задачи линейной алгебры — прежде всего выполнение арифметических операций над векторами и матрицами.

Запись	Клавиши	Описание
$V1+V2$	$V1 + V2$	Сложение двух векторов $V1$ и $V2$
$V1-V2$	$V1 - V2$	Вычитание двух векторов $V1$ и $V2$
$-V$	$-V$	Смена знака у элементов вектора V
$-M$	$-M$	Смена знака у элементов матрицы M
$V-Z$	$V - Z$	Вычитание скаляра Z из всех элементов вектора V
$Z \cdot V, V \cdot Z$	$Z * V, V * Z$	Скалярное умножение вектора V на скаляр Z
$Z \cdot M, M \cdot Z$	$Z * M, M * Z$	Скалярное умножение матрицы M на скаляр Z
$V1 \cdot V2$	$V1 * V2$	Скалярное умножение двух векторов $V1$ и $V2$
$M \cdot V$	$M * V$	Скалярное умножение матрицы M на вектор V
$M1 \cdot M2$	$M1 * M2$	Скалярное умножение двух матриц $M1$ и $M2$
$\frac{V}{Z}$	V / Z	Деление всех элементов вектора V на скаляр Z
$\frac{M}{Z}$	M / Z	Деление матрицы M на скаляр Z
M^{-1}	$M ^ - 1$	Обращение матрицы M

Запись	Клавиши	Описание
M^n	$M \wedge n$	Возведение матрицы M в степень n
$ V $	$ V$	Вычисление модуля вектора V
$ M $	$ M$	Вычисление определителя матрицы
V^T	$V \text{ Ctrl } !$	Транспонирование вектора V
M^T	$M \text{ Ctrl } !$	Транспонирование матрицы M
$V_1 \times V_2$	$V_1 \text{ Ctrl } * V_2$	Векторное умножение двух векторов V_1 и V_2
$\sum V$	$\text{Alt } \$ V$	Вычисление суммы элементов вектора V
\vec{V}	$V \text{ Ctrl } -$	Векторизация вектора V
\vec{M}	$M \text{ Ctrl } -$	Векторизация матрицы M
$M^{<n>}$	$M \text{ Ctrl } \wedge n$	Выделение n -го столбца матрицы M
V_n	$V [n$	Выделение n -го элемента вектора V
$M_{m,n}$	$M [(m,n)$	Выделение элемента (m, n) матрицы M
M	$M \text{ Ctrl } T$	Вставка рисунка, данные которого хранятся в матрице M
\bar{V}, \bar{M}	$V ", M "$	Получение комплексно сопряженного вектора или матрицы

Следует отметить, что в некоторых операторах для ввода используется клавиша Ctrl, тогда как в более ранних версиях системы Mathcad использовалась клавиша Alt (в последней версии клавиша Alt предназначена для активизации строки меню). Все представленные выше операторы (кроме последнего) могут вызываться из палитры матричных операций.

6.8.6. Операция векторизации

Большинство приведенных операторов достаточно известны из математического аппарата матричных вычислений. Под необычным для нашей математической литературы понятием «векторизация» подразумевается одновременное проведение некоторой скалярной операции над всеми элементами вектора или матрицы, помеченными операторами векторизации. Это можно понимать и как возможность записи (увы, не реального исполнения) параллельных вычислений.

Векторизация может изменить смысл математических выражений и даже превратить недопустимое в прежних версиях Mathcad выражение во вполне допустимое. Например, если V — вектор, то выражение $\cos(V)$ будет недопустимым, поскольку аргументом функции \cos может быть только скалярная переменная. Однако с оператором векторизации функция $\cos(V)$ возвращает вектор, каждый элемент которого есть косинус соответствующего элемента исходного вектора V .

Начиная с Mathcad 8 в системе класса Mathcad было введено очередное усовершенствование — в качестве аргумента функции можно задавать векторы и матрицы. Таким образом, выражение $\cos(V)$, где V — вектор, становится допустимым и без применения оператора векторизации. Система Mathcad 8/2000/2001

стала более «интеллектуальной» — в подобных случаях векторизация выполняется автоматически. Таким образом, в нашем примере будет возвращен вектор, каждый элемент которого равен косинусу соответствующего элемента вектора V .

Векторизация осуществляется помещением соответствующего выражения под знак длинной стрелки. Если, к примеру, A и B — векторы, то $A \cdot B$ дает скалярное произведение этих векторов. Но то же произведение под знаком векторизации создает новый вектор, каждый j -й элемент которого есть произведение j -х элементов векторов A и B .

Итак, векторизация позволяет использовать скалярные операторы и функции с массивами. Нередко это заметно упрощает запись математических алгоритмов, особенно для обеспечения параллельных вычислений. Впрочем, параллельность относится не к самим вычислениям, а лишь к их алгоритмической записи. Поэтому от нее вряд ли можно ожидать кардинального уменьшения времени выполнения операций. Тем не менее благодаря оптимизации алгоритма векторизации некоторое ускорение операций, по сравнению с обычным заданием этой операции (например в программном цикле), возможно.

6.8.7. Векторные и матричные функции

Mathcad поддерживает также ряд встроенных векторных и матричных функций, которые облегчают решение задач линейной алгебры и других сфер приложения векторов и матриц:

- $\text{length}(V)$ — возвращает число элементов вектора;
- $\text{last}(V)$ — возвращает номер последнего элемента;
- $\text{max}(V)$ — возвращает максимальный по значению элемент вектора (или матрицы);
- $\text{min}(V)$ — возвращает минимальный по значению элемент вектора (или матрицы);
- $\text{Re}(V)$ — возвращает вектор действительных частей вектора с комплексными элементами;
- $\text{Im}(V)$ — возвращает вектор мнимых частей вектора с комплексными элементами;
- $\varepsilon(i, j, k)$ — единичный полностью антисимметричный тензор третьего ранга, при этом i, j и k должны быть целыми числами от 0 до 2 (или от ORIGIN до $\text{ORIGIN}+2$, если $\text{ORIGIN} \neq 0$); результат равен 0, если любые два аргумента равны, 1 — если три аргумента являются четной перестановкой (0, 1, 2), и -1, если три аргумента являются нечетной перестановкой (0, 1, 2).

Только для работы с матрицами также существует ряд встроенных функций:

- $\text{augment}(M1, M2)$ — объединяет в одну две матрицы $M1$ и $M2$, имеющие одинаковое число строк (объединение идет «по горизонтали»);
- $\text{identity}(n)$ — создает единичную квадратную матрицу размером $n \times n$;
- $\text{stack}(M1, M2)$ — объединяет «по вертикали» две матрицы $M1$ и $M2$, имеющие одинаковое число столбцов;
- $\text{submatrix}(A, ir, jr, ic, jc)$ — возвращает подматрицу, состоящую из всех элементов, содержащихся в строках с ir по jr и столбцов с ic по jc ($ir \leq jr$ и $ic \leq jc$);
- $\text{diag}(V)$ — создает диагональную матрицу, элементы главной диагонали которой равны элементам вектора V ;

- $\text{matrix}(m, n, f)$ — создает матрицу, в которой (i, j) -й элемент равен $f(i, j)$, где $i=0, 1, \dots, m$ и $j=0, 1, \dots, n$; $f(i, j)$ — некоторая функция;
- $\text{Re}(M)$ — возвращает матрицу действительных частей матрицы M с комплексными элементами;
- $\text{Im}(M)$ — возвращает матрицу мнимых частей матрицы M с комплексными элементами.

6.8.8. Функции, возвращающие специальные характеристики матриц

Следующие функции возвращают специальные характеристики матриц:

- $\text{cols}(M)$ — возвращает число столбцов матрицы M ;
- $\text{rows}(M)$ — возвращает число строк матрицы M ;
- $\text{rank}(M)$ — возвращает ранг матрицы M ;
- $\text{tr}(M)$ — возвращает след (сумму диагональных элементов) квадратной матрицы M ;
- $\text{mean}(M)$ — возвращает среднее значение элементов массива M ;
- $\text{median}(M)$ — возвращает медиану элементов массива M ;
- $\text{cond1}(M)$ — возвращает число обусловленности матрицы, вычисленное в норме $L1$;
- $\text{cond2}(M)$ — возвращает число обусловленности матрицы, вычисленное в норме $L2$;
- $\text{conde}(M)$ — возвращает число обусловленности матрицы, вычисленное в норме евклидова пространства;
- $\text{condi}(M)$ — возвращает число обусловленности матрицы, основанное на бесконечной норме;
- $\text{norm1}(M)$ — возвращает норму $L1$ матрицы M ;
- $\text{norm2}(M)$ — возвращает норму $L2$ матрицы M ;
- $\text{norme}(M)$ — возвращает евклидову норму матрицы M ;
- $\text{normi}(M)$ — возвращает бесконечную норму матрицы M .

6.8.9. Примеры применения векторных и матричных операторов и функций

Рассмотрим примеры использования наиболее распространенных векторных операторов. Они представлены на рис. 6.27.

Mathcad делает работу с векторами и матрицами столь же простой, как и с обычными числами и переменными. Это, безусловно, способствует проникновению векторных и матричных методов математических вычислений в практику научно-технических и иных расчетов. На рис. 6.28 показано начало документа с примерами матричных операций, а на рис. 6.29 конец этого документа.

Теперь рассмотрим ряд примеров применения наиболее распространенных матричных функций. Примеры таких операций представлены на рис. 6.30 и 6.31.

РАБОТА С ВЕКТОРАМИ		
$V := \begin{pmatrix} 1+2 \\ 4 \end{pmatrix}$	$U := V \cdot 3$	$U = \begin{pmatrix} 9 \\ 12 \end{pmatrix}$ Умножение вектора на константу
$V1 := \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$	$V2 := \begin{pmatrix} 3 \\ 4 \\ 5 \end{pmatrix}$	$V3 := \begin{pmatrix} 6 \\ 7 \\ 8 \end{pmatrix}$
		$V1 + V2 - V3 = \begin{pmatrix} -2 \\ -1 \\ 0 \end{pmatrix}$ Задание и сложение трех векторов
$V3 := V1 \cdot V2$	$V3 = 26$	Умножение двух векторов
$V4 := V1 \times V2$		Кросс-умножение двух трехэлементных векторов
$V := \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$	$VSUM := \sum V$	$VSUM = 6$ Суммирование элементов вектора V
$U := V^T$	$U = (1 \ 2 \ 3)$	Транспонирование вектора V
$U := \ln(V)$	$U = \begin{pmatrix} 0 \\ 0.693 \\ 1.099 \end{pmatrix}$	$ V = 3.742$ Вычисление нормы вектора V
		$U_1 = 0.693$ Векторизация вектора V
$\text{length}(V) = 3$	$\text{last}(V) = 2$	Выделение элемента вектора U
$\text{max}(V) = 3$	$\text{min}(V) = 1$	Вычисление встроенных функций вектора V

Рис. 6.27. Примеры операций с векторами

ДЕЙСТВИЯ С МАТРИЦАМИ - 1		
$A := \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$	$\text{rows}(A) = 2$	$\text{cols}(A) = 3$ Задание матрицы A с размерностью 2x3
$B := A^T$	$B = \begin{pmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{pmatrix}$	Транспонирование матрицы A
$M := \text{identity}(2)$	$M = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	Создание единичной матрицы M
$\text{tr}(M) = 2$		Вычисление следа матрицы
$A := \begin{pmatrix} 7 & 8 \\ 4 & 5 \end{pmatrix}$	$B := A^{-1}$	$B = \begin{pmatrix} 1.667 & -2.667 \\ -1.333 & 2.333 \end{pmatrix}$ Задание и обращение матрицы A
$C := B^{-1}$	$C = \begin{pmatrix} 7 & 8 \\ 4 & 5 \end{pmatrix}$	Повторное обращение восстанавливает исходную матрицу (матрица C = A)
$A := \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$	$B := \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix}$	$A \cdot B = \begin{pmatrix} 19 & 22 \\ 43 & 50 \end{pmatrix}$ Умножение двух матриц
$A := \begin{pmatrix} 2 & 5 & -3 \\ 1 & 4 & -1 \\ 1 & 3 & 2 \end{pmatrix}$	$D := A $	$D = 10$ Задание квадратной матрицы A и вычисление ее детерминанта

Рис. 6.28. Примеры операций с матрицами (начало документа)

6.8.10. Дополнительные матричные функции

В Mathcad 2001i/11 включен ряд дополнительных матричных функций. Они перечислены ниже:

- $\text{eigenvals}(M)$ — возвращает вектор, содержащий собственные значения матрицы M;

ДЕЙСТВИЯ С МАТРИЦАМИ - 2		
$V := A^{(1)}$	$V = \begin{pmatrix} 5 \\ 4 \\ 3 \end{pmatrix}$	Выделение второго (отсчет с нуля) столбца матрицы A
$A_{1,1} = 4$ $A_{0,0} = 2$ $A_{2,1} = 3$		Выделение элементов матрицы A
$A := \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$	$-A = \begin{pmatrix} -1 & -2 & -3 \\ -4 & -5 & -6 \end{pmatrix}$	Задание матрицы A, смена знака у всех ее элементов, удвоение их и деление на два
$2 \cdot A = \begin{pmatrix} 2 & 4 & 6 \\ 8 & 10 & 12 \end{pmatrix}$	$\frac{A}{2} = \begin{pmatrix} 0.5 & 1 & 1.5 \\ 2 & 2.5 & 3 \end{pmatrix}$	
$A := \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ $B := \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix}$	$A + B = \begin{pmatrix} 6 & 8 \\ 10 & 12 \end{pmatrix}$	Суммирование двух матриц - A и B
$VV := \exp(V)$	$VV = \begin{pmatrix} 148.413 \\ 54.598 \\ 20.086 \end{pmatrix}$	Векторизация вектора V
$i := \sqrt{-1}$	$A := \begin{pmatrix} 1 + 2i & 0 \\ 0 & 4 - 5i \end{pmatrix}$	Задание матрицы A с комплексными элементами
$B := \bar{A}$	$B = \begin{pmatrix} 1 - 2i & 0 \\ 0 & 4 + 5i \end{pmatrix}$	Получение комплексно сопряженной матрицы B

Рис. 6.29. Примеры операций с матрицами (конец документа)

РАБОТА С МАТРИЧНЫМИ ФУНКЦИЯМИ - 1		
$M := \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$	Задание матрицы M с размерностью 2x3	
$rows(M) = 2$ $cols(M) = 3$ $ORIGIN := 1$	Вычисление матричных функций:	
$M1 := identity(3)$	$M1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	Задание и вывод единичной квадратной матрицы M1
$V := \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$	$augment(M1, V) = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \end{pmatrix}$	Задание вектора V и подключение его к ранее созданной матрице M1
$M2 := \begin{pmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{pmatrix}$	$augment(M1, M2) = \begin{pmatrix} 1 & 0 & 0 & 1 & 4 \\ 0 & 1 & 0 & 2 & 5 \\ 0 & 0 & 1 & 3 & 6 \end{pmatrix}$	Задание матрицы M2 и подключение ее к матрице M1
$tr(M1) = 3$	Вычисление следа матрицы M1: суммы диагональных элементов	

Рис. 6.30. Примеры применения матричных функций (начало документа)

- $eigenvec(M, Z)$ — для указанной матрицы M и заданного собственного значения Z возвращает соответствующий этому собственному значению вектор;
- $eigenvecs(M)$ — возвращает матрицу, столбцами которой являются собственные векторы матрицы M (порядок расположения собственных векторов соответствует порядку собственных значений, возвращаемых функцией $eigenvals$);

РАБОТА С МАТРИЧНЫМИ ФУНКЦИЯМИ - 2		
$i := \sqrt{-1}$	$CM := \begin{pmatrix} 1 + 2i & 2 + 3i & 4 + 4i \\ 5 + 6i & 6 + 7i & 7 + 8i \\ 8 + 9i & 9 + 10i & 11 \end{pmatrix}$	Задание матрицы CM с комплексными элементами
$R := \text{Re}(CM)$	$R = \begin{pmatrix} 1 & 2 & 4 \\ 5 & 6 & 7 \\ 8 & 9 & 11 \end{pmatrix}$	Выделение матрицы - действительной части матрицы CM
$J := \text{Im}(CM)$	$J = \begin{pmatrix} 2 & 3 & 4 \\ 6 & 7 & 8 \\ 9 & 10 & 0 \end{pmatrix}$	Выделение матрицы - мнимой части матрицы CM
$K := 1..3$		
$L := 1..3$	$MC_{K,L} := R_{K,L} + J_{K,L} \cdot i$	Восстановление матрицы с комплексными элементами по матрицам R и J
	$MC = \begin{pmatrix} 1 + 2i & 2 + 3i & 4 + 4i \\ 5 + 6i & 6 + 7i & 7 + 8i \\ 8 + 9i & 9 + 10i & 11 \end{pmatrix}$	Вывод восстановленной матрицы (MC=CM)

Рис. 6.31. Примеры применения матричных функций (конец документа)

- $\text{genvals}(M, N)$ — возвращает вектор обобщенных собственных значений v_i , соответствующий решению уравнения $M \cdot x = v_i \cdot N \cdot x$ (матрицы M и N должны быть вещественными);
- $\text{genvals}(M, N)$ — возвращает матрицу, столбцы которой содержат нормированные обобщенные собственные векторы;
- $\text{lu}(M)$ — выполняет треугольное разложение матрицы M: $P \cdot M = L \cdot U$, где L и U — соответственно нижняя и верхняя треугольные матрицы, P — матрица перестановки, все четыре матрицы квадратные, одного порядка;
- $\text{qr}(A)$ — дает разложение матрицы A: $A = Q \cdot R$, где Q — ортогональная матрица, а R — верхняя треугольная матрица;
- $\text{svd}(A)$ — дает сингулярное разложение матрицы A размером $n \times m$: $A = U \cdot S \cdot V^T$, где U и V — ортогональные матрицы размером $m \times m$ и $n \times n$ соответственно, S — диагональная матрица, на диагонали которой расположены сингулярные числа матрицы A;
- $\text{svds}(A)$ — возвращает вектор, содержащий сингулярные числа матрицы A размером $m \times n$, где $m \geq n$;
- $\text{geninv}(A)$ — левая обратная к матрице A: $L \cdot A = E$, где E — единичная матрица размером $n \times n$, L — прямоугольная матрица размером $n \times m$, A — прямоугольная матрица размером $m \times n$.

6.8.11. Функции сортировки для векторов и матриц

Начиная с версии 3.0 для Windows в системе Mathcad появились некоторые дополнительные функции сортировки — перестановка элементов векторов и матриц:

- $\text{sort}(V)$ — сортировка элементов вектора в порядке возрастания их значений;

- `csort(M,n)` — перестановка строк матрицы M таким образом, чтобы отсортированным оказался n -й столбец;
- `rsort(M,n)` — перестановка столбцов матрицы M таким образом, чтобы отсортированной оказалась n -я строка.

С этими функциями часто используется функция `reverse(V)`, изменяющая порядок расположения элементов вектора на противоположный (начиная с конца).

6.8.12. Примеры применения дополнительных векторных и матричных функций

Ниже представлены примеры, иллюстрирующие работу некоторых дополнительных векторных и матричных функций.

ДОПОЛНИТЕЛЬНЫЕ ВЕКТОРНЫЕ И МАТРИЧНЫЕ ФУНКЦИИ			
Исходный вектор	Прямая сортировка	Реверс после сортировки	
$V := \begin{pmatrix} 3 \\ 2 \\ 4 \\ 1 \end{pmatrix}$	$VS := \text{sort}(V)$	$VS = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix}$	$VR := \text{reverse}(VS) \quad VR = \begin{pmatrix} 4 \\ 3 \\ 2 \\ 1 \end{pmatrix}$
Исходная матрица			
$M := \begin{pmatrix} 1 & 12 & 4 \\ 3 & 15 & 27 \\ 17 & 2 & 13 \end{pmatrix}$	Сортировка по первому столбцу и по первой строке		
	$csort(M,1) = \begin{pmatrix} 17 & 2 & 13 \\ 1 & 12 & 4 \\ 3 & 15 & 27 \end{pmatrix}$	$rsort(M,1) = \begin{pmatrix} 1 & 12 & 4 \\ 3 & 15 & 27 \\ 17 & 2 & 13 \end{pmatrix}$	
Вычисление вектора VE собственных значений квадратной матрицы A			
$A := \begin{pmatrix} 10 & 5 & 6 \\ 5 & 20 & 4 \\ 6 & 4 & 30 \end{pmatrix}$	$VE := \text{eigenvals}(A)$	$VE = \begin{pmatrix} 7.142 \\ 19.149 \\ 33.709 \end{pmatrix}$	
Вычисление принадлежащего собственному значению 19.149 вектора			
$VE_1 = 19.149$	$VV := \text{eigenvec}(A, VE_1)$	$VV = \begin{pmatrix} -0.197 \\ -0.88 \\ 0.433 \end{pmatrix}$	

Рис. 6.32. Примеры применения дополнительных матричных функций

В целом видно, что запись матричных операций в системе Mathcad весьма наглядна, и это большое достоинство входного языка данной системы.

6.8.13. Решение систем линейных уравнений

Векторные и матричные операторы и функции системы Mathcad позволяют решать широкий круг задач линейной алгебры. К примеру, если заданы матрица A и вектор B для системы линейных уравнений в матричной форме $A \cdot X = B$, то вектор решения можно получить из очевидного выражения $X = A^{-1} \cdot B$. Ниже приведен пример решения системы линейных уравнений, который иллюстрирует фрагмент документа, показанный на рис. 6.33.

Поскольку решение систем линейных уравнений — довольно распространенная задача, в Mathcad, начиная с шестой версии, введена встроенная функция `lsolve(A,B)`, которая возвращает вектор X для системы линейных уравнений $A \cdot X = B$ при заданной матрице коэффициентов A и векторе свободных членов B .

РЕШЕНИЕ СИСТЕМЫ ЛИНЕЙНЫХ УРАВНЕНИЙ $A \cdot X = B$		
$i := \sqrt{-1}$		
$A := \begin{pmatrix} 4 + i & 0.24 & -0.08 \\ 0.09 & 3 & -0.15 \\ 0.04 & -0.08 & 4 + i \end{pmatrix}$	Матрица комплексных коэффициентов системы линейных уравнений	
$B := \begin{pmatrix} 8 \\ 9 \\ 20 \end{pmatrix}$	Вектор свободных членов	
$X := A^{(-1)} \cdot B$	Решение системы	
$X = \begin{pmatrix} 1.787 - 0.468i \\ 3.184 - 0.045i \\ 4.75 - 1.184i \end{pmatrix}$	Результаты решения	
$X1 := \text{Isolve}(A, B)$	$X1 = \begin{pmatrix} 1.787 - 0.468i \\ 3.184 - 0.045i \\ 4.75 - 1.184i \end{pmatrix}$	Решение с применением функции Isolve

Рис. 6.33. Примеры решения систем линейных уравнений

Если уравнений n , размер вектора B должен быть равен n , а матрицы A — $n \times n$. Пример применения этой функции дан в нижней части показанного на рисунке фрагмента документа Mathcad.

Глава 7. Численные методы

Мы уже использовали численные методы решения математических задач, например вычисление определенных интегралов, и матричные методы решения систем линейных уравнений. Продолжим эту важную тему. Многие уравнения, например трансцендентные, и системы из них вообще не имеют аналитических решений. Однако они могут решаться специальными численными методами [29—37] с заданной погрешностью (не более значения, заданного системной переменной `TOL`). Реализации специальных численных методов в системе Mathcad посвящается данная глава. Помимо решения нелинейных уравнений описаны методы осуществления интегральных преобразований Фурье и вейвлет-преобразований, а также методы численного решения дифференциальных уравнений и решения вариационных задач.

7.1. Решение нелинейных и алгебраических уравнений

7.1.1. Решение нелинейных уравнений вида $F(x) = 0$

Для простейших нелинейных уравнений вида $F(x) = 0$ решение находится с помощью функции

```
root (Выражение, Имя_переменной)
```

Эта функция возвращает с заданной точностью значение переменной, при котором `Выражение` равно 0. Функция реализует вычисления итерационным методом, причем перед ее применением надо задать *начальное* значение переменной. Это особенно полезно, если возможно несколько решений. Тогда выбор одного из них определяется начальным значением переменной. Технику применения функции `root` для вычисления корней кубического полинома иллюстрирует фрагмент документа Mathcad, показанный на рис. 7.1.

Как известно, кубическое уравнение обязательно имеет хотя бы один действительный корень x_1 . Он найден вначале функцией `root`. Два других корня могут оказаться как действительными, так и комплексными. Функция `root` может отыскивать любые корни. Для поиска второго корня x_2 первый исключается делением $F(x)$ на $(x-x_1)$. Соответственно для поиска третьего корня x_3 нужно повторить деление, при этом $F(x)$ делится еще на $(x-x_2)$. Эту процедуру можно распространить и на поиск корней полиномов более высокой степени, однако надо помнить, что найти корни полинома можно гораздо более изящным и простым способом — используя операцию символьных вычислений (см. главу 9).

Представленную форму записи функции `root` поддерживают все версии, начиная с Mathcad 8.0. Однако в последних версиях возможности этой функции расширены и она может записываться в виде:

```
root (Выражение, Имя_переменной, a, b)
```

Здесь a и b — границы численного интервала для поиска корней (интервала изоляции корня). Такое применение функции `root` позволяет избежать вывода корней, не представляющих интереса при решении задач — например физиче-

ПРИМЕНЕНИЕ ФУНКЦИЙ root И polyroots

$a_3 := 2$ $a_2 := -8$ $a_1 := 25$ $a_0 := -64$ Коэффициенты полинома

$F(x) := a_3 \cdot x^3 + a_2 \cdot x^2 + a_1 \cdot x + a_0$ Задание полинома

Вычисление действительного корня

$x := 0$ $x_1 := \text{root}(F(x), x)$ $x_1 = 3.211$

Вычисление двух других (возможно комплексных) корней

$i := \sqrt{-1}$ $x := 1 + 1 \cdot i$ $x_2 := \text{root}\left(\frac{F(x)}{x - x_1}, x\right)$ $x_2 = 0.395 + 3.132i$

$x_3 := \text{root}\left[\frac{F(x)}{(x - x_1) \cdot (x - x_2)}, x\right]$ $x_3 = 0.395 - 3.132i$

Как и следовало ожидать, комплексные корни кубического полинома являются взаимно сопряженными!

$V_0 := a_0$ $V_1 := a_1$ $V_2 := a_2$ $V_3 := a_3$

Пример применения функции polyroots: $\text{polyroots}(V) = \begin{pmatrix} 0.395 + 3.132i \\ 0.395 - 3.132i \\ 3.211 \end{pmatrix}$

Рис. 7.1. Вычисление корней кубического полинома

ских. При этом нет необходимости задавать начальное значение x , поскольку эта переменная определена в заданном интервале $[a, b]$. Пример:

$$\text{root}(x^2 - 9, x, 0, 5) = 3$$

ПРИМЕЧАНИЕ Выражение, корень которого находится функцией *root*, может быть представлено в виде $F(x)$ или даже $F_1(x) - F_2(x)$, если решается уравнение $F_1(x) = F_2(x)$.

7.1.2. Вычисление корней степенного многочлена

Рассмотрим обычный полином степени n :

$$p(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$$

Для поиска корней этого полинома Mathcad поддерживает очень удобную функцию *polyroots(V)*. Она возвращает вектор всех корней многочлена (полинома) степени n , коэффициенты которого находятся в векторе V , имеющем длину $n + 1$. Системная переменная *ORIGIN* должна иметь значение 0. Обычно оно задано таким по умолчанию.

Заметим, что корни полинома могут быть как вещественными, так и комплексными числами. Не рекомендуется пользоваться этой функцией, если степень полинома выше пятой-шестой, так как тогда трудно получить малую погрешность вычисления корней. В нижней части фрагмента документа, показанного на рис. 7.1, имеется пример использования функции *polyroots* для нахождения корней кубического полинома. Как видите, применение функций *root* и *polyroots* дает один и тот же результат, хотя пользоваться функцией *polyroots* несомненно удобнее.

7.1.3. Функция root в составе функции пользователя

Функцию *root* можно использовать и в составе функции пользователя, что нередко позволяет получать довольно простые решения нетривиальных задач. При-

мером может служить вычисление функции $G(a, x)$. Исходные данные и результат решения, представленный в виде вектора и графика, показаны на рис. 7.2.



Рис. 7.2. Вычисление функции $G(a, x)$ с помощью функции *root* в составе функции пользователя

Недостатком функции *root* является невозможность решения с ее помощью систем нелинейных уравнений. Эта функция предназначена только для решения одиночных уравнений.

7.1.4. Решение систем нелинейных уравнений с блоком *Given*

При решении систем нелинейных уравнений используется специальный вычислительный блок, открываемый директивой *Given* и имеющий следующую структуру:

Начальные условия

Given

Уравнения

Ограничительные условия

Выражения с функциями *Find*, *Minerr*, *Maximize* и *Minimize*

Начальные условия определяют начальные значения искомых переменных и задаются в виде переменная := значение, то есть обычным присваиванием переменным заданных значений. Если переменных несколько, то используется векторное представление для начальных условий. Уравнения задаются в виде левая часть выражения = левая часть выражения с применением жирного знака равенства = (вводимого клавишами **Ctrl + =**). Ограничительные условия обычно задаются в виде неравенств или равенств, которые должны удовлетворяться при решении системы уравнений.

Рекомендуется дополнить блок проверкой решения системы. Вначале число уравнений ограничивалось 50, а число ограничительных условий должно было точно соответствовать числу уравнений. Начиная с версий *Mathcad 2000/2001 PRO* это ограничение снято, а максимальное число уравнений в си-

стеме доведено до 200. Но, пожалуй, самым далеко идущим усовершенствованием стала возможность решения систем нелинейных уравнений при задании не только начальных условий, но и уравнений и ограничительных условий в *векторной форме*. При этом решение может искажаться и в символьной форме (естественно, если оно существует).

В блоке `Given` используется одна из следующих двух функций:

- `Find(v1, v2, ..., vn)` — возвращает значение одной или нескольких переменных для точного решения;
- `Minerr(v1, v2, ..., vn)` — возвращает значение одной или нескольких переменных для приближенного решения.

Между этими функциями существуют принципиальные различия. Первая функция используется, когда решение реально существует (хотя и не является аналитическим). Вторая функция пытается найти максимальное приближение даже к несуществующему решению, путем минимизации среднеквадратичной погрешности решения. Позже мы рассмотрим и другие функции, включающиеся в блок решения — `Maximize` и `Minimize`, введенные в `Mathcad` и в последующие версии (включая, разумеется, и `Mathcad 2001i/11/12`).

Часто системы нелинейных уравнений приходится решать при наличии определенных ограничений на значения тех или иных переменных или выражений с ними. Такие ограничения имеют вид равенств или неравенств. Для их задания используются логические операторы, представленные ниже.

Оператор	Клавиши	Значение оператора
$e1 > e2$	<code>e1 > e2</code>	$e1$ больше $e2$
$e1 < e2$	<code>e1 < e2</code>	$e1$ меньше $e2$
$e1 \geq e2$	<code>e1 Ctrl) e2</code>	$e1$ больше или равно $e2$
$e1 \leq e2$	<code>e1 Ctrl (e2</code>	$e1$ меньше или равно $e2$
$e1 \neq e2$	<code>e1 Ctrl # e2</code>	$e1$ не равно $e2$
$e1 = e2$	<code>e1 Ctrl = e2</code>	$e1$ равно $e2$

7.1.5. Примеры решения нелинейных уравнений

Как уже упоминалось, для решения одного или нескольких уравнений могут использоваться функции `Find` и `Minerr`. Рассмотрим процедуру решения уравнения $x^2 = 3$ с помощью функции `Find`.

Для решения уравнения в документ `Mathcad` требуется ввести следующие данные (в первую очередь задается начальное значение для поиска корня):

```
x := 10
Given
  x2 := 3
  x0 := Find(x)
```

Результат:

```
x0 = 1.732
```

Функция `Minerr` используется аналогично. Процедуры поиска решения с применением обеих функций иллюстрирует фрагмент документа Mathcad, представленный на рис. 7.3. В этом фрагменте также показано, что задание оператора присваивания вместо знака символического (логического) равенства в блоке `Given` некорректно и система реагирует на подобную попытку сообщением об ошибке.

<pre>x := 10 Given x^2 = 3 x0 := Find(x)</pre>	Точное задание $x=10$ делает некорректным решение уравнения $x^2=3$. Поэтому система отвечает на попытку решения сообщением об ошибке - значение x_0 выделено красным цветом.
<pre>x := 10 Given x^2 = 3 x0 := Find(x) x0 = 1.732</pre>	Однако задание уравнения в виде приближенного равенства позволяет найти решение
<pre>x := 10 Given x^2 = 3 x1 := Minerr(x) x1 = 1.732</pre>	Решение с применением функции <code>minerr</code> также позволяет найти решение

Рис. 7.3. Примеры решения нелинейного уравнения с использованием блока `Given`

Как видно из рисунка, обе функции дают одинаковое решение.

Рассмотрим пример решения другой задачи — поиска точек пересечения параболы и прямой с помощью функции `Find`. Таким образом, требуется найти решение следующей системы двух уравнений (одно из них нелинейное):

$$\begin{cases} y = x^2 \\ y = 8 + 3x \end{cases}$$

Кроме того, пусть имеются ограничительные условия, задающие область поиска корня ($x < 0$ для отрицательного корня и $x > 0$ для положительного).

Начальные значения для поиска корней, вычислительные блоки с ограничительными условиями и результаты представлены во фрагменте документа Mathcad, показанном на рис. 7.4.

Рассмотрим еще один пример решения системы нелинейных уравнений (на этот раз с помощью функции `Minerr`):

$$\begin{cases} (x^2 + 1)^2 + (y^2 + 1)^2 = 5.5 \\ x + y = 0.95 \end{cases}$$

Решение этой системы иллюстрирует фрагмент документа Mathcad, показанный на рис. 7.5.

При использовании функции `Minerr` для решения систем нелинейных уравнений надо проявлять известную осторожность и обязательно предусматривать проверку решений. Нередки случаи, когда решения могут оказаться ошибочными, чаще всего из-за того, что из нескольких корней система предлагает нереальный (или не представляющий интереса) корень. Полезно как можно точнее указывать начальные приближения к решению.



Рис. 7.4. Вычисление координат точек пересечения параболы и прямой

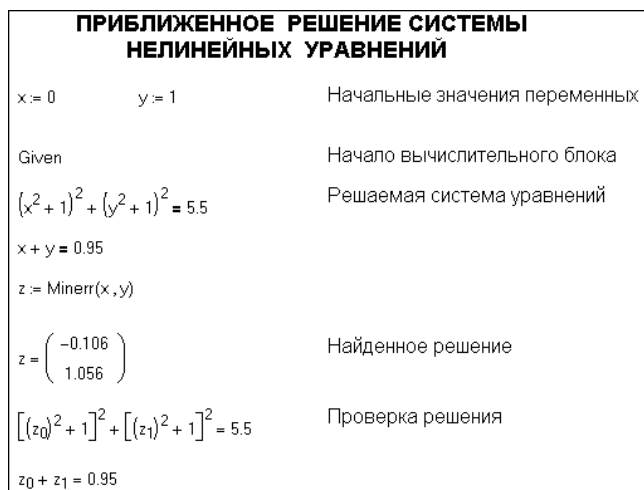


Рис. 7.5. Решение системы из двух уравнений с помощью функции Minerr

7.2. Реализация итерационных вычислений

Система Mathcad позволяет реализовать разного рода итерационные вычисления, в том числе осуществляемые по *рекуррентным соотношениям*. Рекуррентными называют такие соотношения, при которых значение некоторой функции находится по одному или нескольким предшествующим значениям этой функции.

7.2.1. Рекуррентное вычисление чисел Фибоначчи

Классическим примером рекуррентных вычислений является расчет чисел Фибоначчи. Как известно, два первых числа Фибоначчи определены как единицы, а каждое из последующих есть сумма двух предыдущих.

На рис. 7.6 показан фрагмент документа, который иллюстрирует процедуру создания в Mathcad вектора, содержащего 10 чисел Фибоначчи. Этот пример соответствует многократным вычислениям по рекуррентной формуле Фибоначчи.

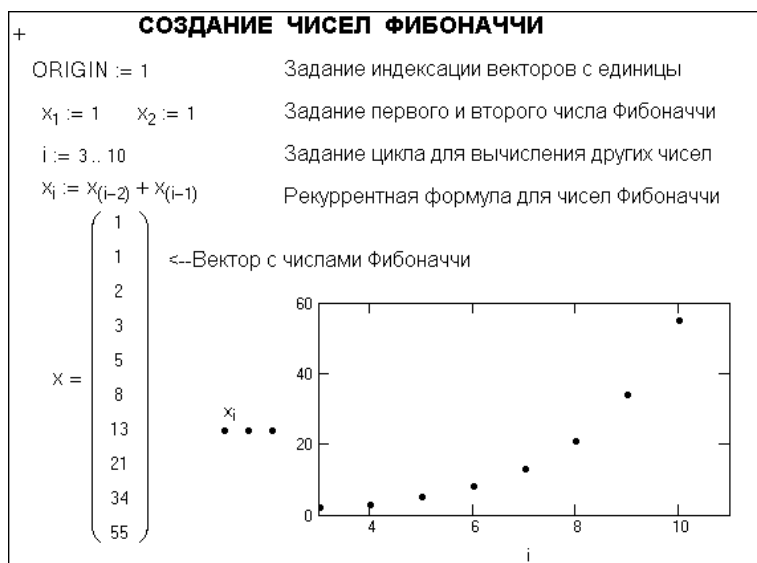


Рис. 7.6. Вычисление десяти чисел Фибоначчи

7.2.2. Функция until и ее применение

Более актуальна задача реализации вычислений, в том числе итерационных, с их окончанием по некоторому заданному условию. Для этого системы Mathcad ранее поддерживали специальную функцию:

`until(выражение_1, выражение_2)`

Эта функция позволяет повторять вычисления с возвратом значения выражение_2 до тех пор, пока выражение_1 больше или равно 0. В противном случае вычисления прекращаются. Начиная с версий Mathcad 8/2000 функция until не рекомендуется к применению. Тем не менее рассмотрим применение этой функции на классическом примере — вычислении квадратного корня из числа N итерационным методом Ньютона.

Фрагмент документа Mathcad, который иллюстрирует описанную процедуру, показан на рис. 7.7. Обратите внимание, что Mathcad прерывает итерации, как только погрешность вычислений становится меньше заданной (`err`).

Этот пример позволяет проследить динамику схождения результатов вычислений по итерационной формуле Ньютона к точному значению. Нетрудно заметить, что даже при столь простом алгоритме сходимость к решению не является монотонной. Более того, для ряда уравнений процесс решения может быть расходящимся. В Mathcad 2001i/11 этот пример уже не работает, что к достоинствам этих версий отнести нельзя — теряется совместимость с некоторыми старыми документами, содержащими функцию until. В Mathcad 12 эта функция «реабилитирована» и ее можно применять без ограничений.

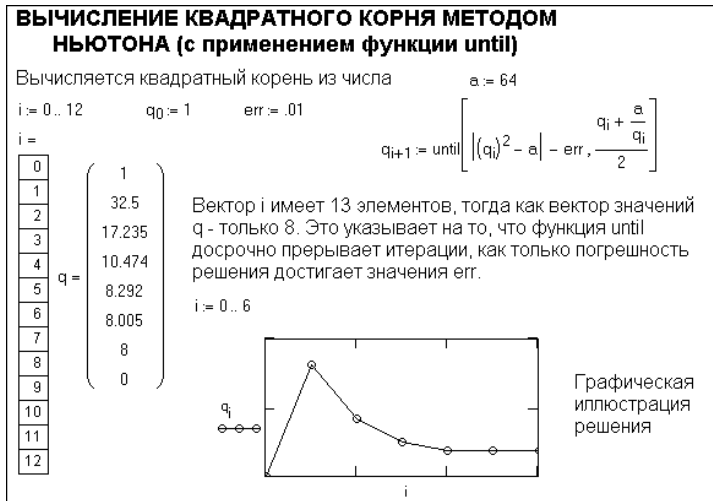


Рис. 7.7. Вычисление квадратного корня итерационным методом Ньютона

7.2.3. Пример решения системы рекуррентных уравнений

В общем случае рекуррентные вычисления могут осуществляться по нескольким формулам, причем в них возможно перекрестное применение переменных. Такого рода вычисления в системе Mathcad необходимо реализовывать в векторной форме. В этом случае уточнение значений переменных идет не по строкам, а по столбцам.

Рассмотрим решение подобной задачи — моделирование развития эпидемии. По существу в этом случае конечно-разностным методом решается система дифференциальных уравнений, описывающих развитие и угасание эпидемии. Начальные значения переменных (число здоровых и заболевших людей и т. д.) заданы вектором начальных условий. Параметр g характеризует меры профилактики, например, проведенные прививки, применение средств личной профилактики и т. д. Расчетные соотношения очевидны из записи системы конечно-разностных уравнений. Результат наглядно представлен в виде графиков (рис. 7.8).

Возможны и более сложные способы реализации рекуррентных и других подобных вычислений. Например, к ним сводится решение систем дифференциальных уравнений любыми разностными методами, например Эйлера, Рунге—Кутта и др.

7.3. Решение задач оптимизации

Решение задач оптимизации — одна из важнейших сфер применения численных математических методов. К задачам оптимизации сводятся задачи линейного программирования, поиск минимума и максимума функций нескольких переменных и многие другие. Тем не менее прежние версии системы Mathcad имели ограниченные возможности в решении данного класса задач, что требовало заметных усилий по поиску и реализации их алгоритмов имеющимися скромными средствами, например с применением функции `Minerr`. В новой версии с этими ограничениями покончено благодаря введению функций, реализующих решение задач данного класса.

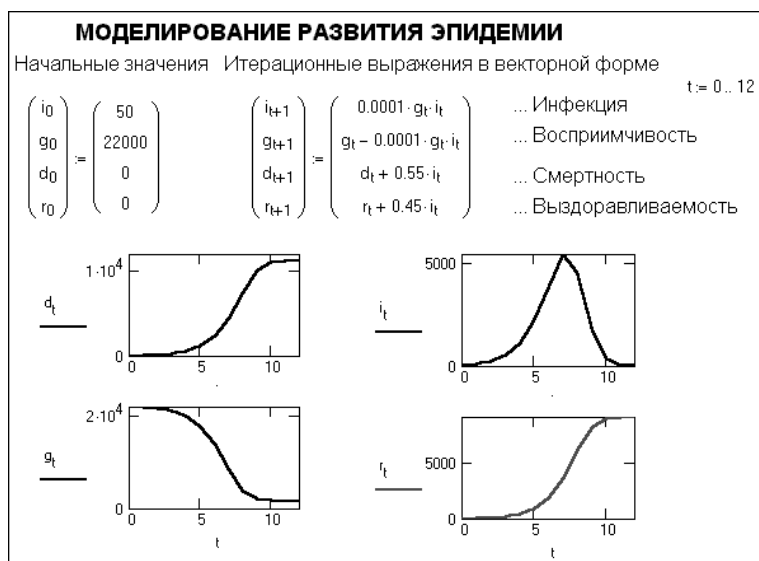


Рис. 7.8. Динамика протекания процесса эпидемии

7.3.1. Поиск минимума функции Розенброка с помощью функции Minerr

Прежде чем изучать новые функции, рассмотрим применение функции Minerr (она поддерживается и в предшествующих версиях Mathcad) для решения типовой оптимизационной задачи — поиска минимума функции Розенброка градиентным методом. Это типовая тестовая функция двух переменных x и y , график которой напоминает глубокий овраг, что сильно осложняет реализацию многих алгоритмов оптимизации.

Фрагмент документа Mathcad, который иллюстрирует процедуру поиска минимума функции Розенброка с помощью функции Minerr, показан на рис. 7.9.

Нетрудно заметить, что решение данной задачи указанным методом требует вычисления производных заданной функции по переменным x и y . В точке минимума эти производные равны нулю. На рис. 7.10 показан фрагмент документа Mathcad с графиком функции Розенброка, который весьма наглядно иллюстрирует сложность поиска минимума данной функции.

К чести разработчиков системы Mathcad, надо отметить, что описанный алгоритм блестяще справляется с непростой задачей поиска минимума функции Розенброка градиентным методом — минимум функции найден в точке (1,1) с максимально верным (в пределах точности отображения результатов вычислений) значением — 0.

7.3.2. Поиск экстремумов функции Розенброка с помощью функций Maximize и Minimize

Для поиска значений переменных x_1, x_2, \dots, x_n , при которых некоторая функция $f(x_1, x_2, \dots, x_n)$ имеет максимальное или минимальное значение, используются функции Maximize(f, x_1, x_2, \dots, x_n) и Minimize(f, x_1, x_2, \dots, x_n). Обе эти

ПОИСК МИНИМУМА ФУНКЦИИ РОЗЕНБРОКА				
$x := 1.2$	$y := .6$	Инициализация x и y		
$f(x,y) := 100 \cdot (y - x^2)^2 + (1 - x)^2$		Функция Розенброка		
Given		Начало вычислительного блока		
$\frac{d}{dx} f(x,y) = 0$	$\frac{d}{dy} f(x,y) = 0$	Условия минимума		
$\begin{pmatrix} x \\ y \end{pmatrix} := \text{Minerr}(x,y)$		Поиск минимума с помощью функции minerr		
$x = 1$	$y = 1$	$f(x,y) = 0$	Проверка решения	
Построение графика функции Розенброка				
$i := 0..100$	$j := 0..100$	$a_i := 0.01 \cdot i - 0.5$	$b_j := 0.01 \cdot j - 0.5$	$M_{i,j} := f(a_i, b_j)$

Рис. 7.9. Решение задачи на поиск минимума функции Розенброка

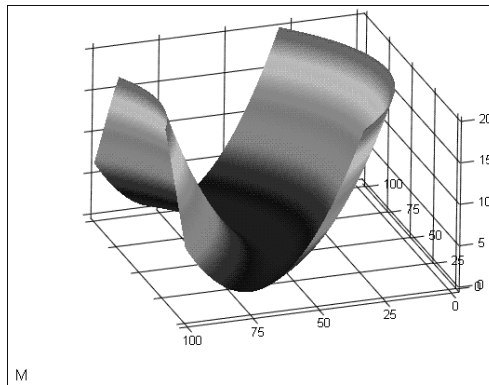


Рис. 7.10. График функции Розенброка

функции реализованы достаточно универсальными алгоритмами оптимизации, которые не требуют вычисления производных функции $f(x_1, x_2, \dots, x_n)$, что не только упрощает запись алгоритмов, но и позволяет решать задачи, в которых вычисление производных по тем или иным причинам невозможно.

Эти функции должны использоваться в составе блока Given. Они возвращают вектор неизвестных, при котором заданная функция имеет максимальное или минимальное значение соответственно. Внутри блока могут быть различные ограничительные условия в виде равенств или неравенств. Число условий ограничено только памятью ПК, то есть практически почти не ограничено.

Перед блоком решения надо задать начальные значения искоемых переменных. Чем они ближе к верному решению, тем быстрее будет получен правильный результат.

Первый фрагмент документа Mathcad, реализующего решение данной задачи, представлен на рис. 7.11. Здесь показано задание самого уравнения, построение графика функции и геометрическое представление ограничительных условий. Полезно особо отметить, что решение задачи оптимизации с ограничениями в принципе намного сложнее, чем без таковых.



Рис. 7.11. Начало документа с решением задачи поиска минимума и максимума функции Розенброка с учетом ограничений

Второй фрагмент документа Mathcad показан на рис. 7.12. Обратите внимание, что функция Minimize (или Maximize) может вычислять минимум (или максимум) как при отсутствии ограничений, так и при их наличии.



Рис. 7.12. Окончание документа с решением задачи поиска минимума и максимума функции Розенброка с учетом ограничений

Объективности ради надо заметить, что результаты решения сильно зависят от выбора начальных значений переменных и далеко не всегда имеют устраивающую пользователя погрешность.

7.3.3. Решение задач линейного программирования

Функции `Maximize` и `Minimize` могут успешно применяться при решении задач линейного программирования, которые широко используются в экономических и производственных расчетах. Рассмотрим такую задачу. Фрагмент документа `Mathcad`, дающий постановку задачи и реализующий ее решение, представлен на рис. 7.13.

Решение типовой задачи линейного программирования

Пусть цех малого предприятия должен изготовить 100 изделий трех типов. Каждого изделия нужно сделать не менее 20 штук. На изделия уходят соответственно 4, 3.4 и 2 кг металла при его общем запасе 340 кг, а также по 4.75, 11 и 2 кг пластмассы при ее общем запасе 700 кг. Сколько изделий каждого типа x_1 , x_2 и x_3 надо выпустить для получения максимального объема выпуска в денежном выражении, если цена изделий составляет по калькуляции 4, 3 и 2 рубля? Итак, задача сводится к вычислению максимума функции

$$f(x_1, x_2, x_3) = 4 \cdot x_1 + 3 \cdot x_2 + 2 \cdot x_3$$

Решение данной задачи имеет следующий вид

$x_1 := 1$	$x_2 := 1$	$x_3 := 1$	Произвольные начальные значения
Given			Блок решения Given
$x_1 \geq 20$	$x_2 \geq 20$	$x_3 \geq 20$	
$4 \cdot x_1 + 3.4 \cdot x_2 + 2 \cdot x_3 \leq 340$			Ограничивающие условия
$4.75 \cdot x_1 + 11 \cdot x_2 + 2 \cdot x_3 \leq 700$			
$x_1 + x_2 + x_3 = 100$			
$R := \text{Maximize}(f, x_1, x_2, x_3)$			
			$R = \begin{pmatrix} 56 \\ 20 \\ 24 \end{pmatrix}$ Полученное решение
			$x_1=56, x_2=20$ и $x_3=24$

Рис. 7.13. Пример решения типовой задачи линейного программирования

Очевидно, что введение в `Mathcad` (начиная с версии 8.0) новых функций оптимизации `Maximize` и `Minimize` расширяет круг решаемых системой задач при минимальных затратах времени на подготовку средств их решения. В главе 11 будет приведен ряд других задач, относящихся к линейному программированию.

Функция `Find` также может использоваться для решения задач линейного программирования, пример чему приведен ниже:

$$A := \begin{pmatrix} -1 & 2 & 3 \\ 4 & 8 & 1 \\ 7 & -4 & 0 \end{pmatrix} \quad b := \begin{pmatrix} 2 \\ -1 \\ 5 \end{pmatrix}$$

Given

$$A \cdot x = b \quad y^3 = 5 + 7 \cdot y$$

$$\text{Find}(x, y) = \begin{pmatrix} \begin{pmatrix} 0.437 \\ -0.485 \\ 1.136 \\ -0.783 \end{pmatrix} \end{pmatrix}$$

Здесь интересно отметить, что результат получен в форме массива (вектора), вложенного в другой массив. Это указывает на множественность (в нашем случае тройственность) решения.

7.4. Интегральные преобразования Фурье

К фундаментальным положениям математики относится возможность представления периодических (а при определенных условиях и непериодических) функций совокупностью их гармонических составляющих в виде так называемого ряда Фурье [34—38]. Эта возможность используется во множестве прикладных областей, достаточно отметить, что на ее основе реализуется передача через каналы связи практически любой информации, например речи или музыки.

7.4.1. Спектральный анализ и синтез на основе рядов Фурье

Рядом Фурье для интегрируемой на отрезке $[-\pi, \pi]$ функции $y(x)$, удовлетворяющей известным условиям Дирихле, называют следующий ряд:

$$y(x) \approx \frac{a_0}{2} \sum_{k=1}^{\infty} (a_k \cos(kx) + b_k \sin(kx)).$$

Коэффициенты Фурье этого ряда находятся по формулам Эйлера—Фурье:

$$a_k = \frac{1}{\pi} \int_{-\pi}^{\pi} y(x) \cos(kx) dx,$$

$$b_k = \frac{1}{\pi} \int_{-\pi}^{\pi} y(x) \sin(kx) dx.$$

Важными сферами применения рядов Фурье являются радиотехнические расчеты. В них периодические сигналы обычно представляют как функции времени $y(t)$ на отрезке $[0, T]$ с периодом $T = 1/f_1$, где f_1 — частота первой гармоники периодического сигнала. В этом случае ряд Фурье после несложных преобразований записывается в виде

$$y(t) \approx \frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k \cos(2\pi k f_1 t) + b_k \sin(2\pi k f_1 t)).$$

Здесь коэффициенты выглядят следующим образом:

$$a_k = \frac{2}{T} \int_0^T y(t) \cos(2\pi k f_1 t) dt,$$

$$b_k = \frac{2}{T} \int_0^T y(t) \sin(2\pi k f_1 t) dt.$$

В этом случае коэффициенты a_k и b_k описывают косинусную и синусную составляющие k -й гармоники сигнала с периодом T и частотой повторения $f_1 = 1/T$. Часто используется иная форма ряда Фурье, упрощающая его синтез:

$$y(t) \approx \frac{a_0}{2} \sum_{k=1}^{\infty} (A_k \cos(2\pi k f_1 t + \varphi_k)).$$

Здесь A_k — амплитуда k -й гармоники периодического сигнала, φ_k — фаза k -й гармоники. Они вычисляются по формулам:

$$A_k = \sqrt{a_k^2 + b_k^2},$$

$$\varphi_k = -\arctan(b_k/a_k).$$

Разложение функции на гармонические составляющие, то есть вычисление коэффициентов Фурье, принято называть *спектральным анализом*. А воссоздание приближения функции рядом Фурье, то есть получение ее тригонометрического представления, называют *спектральным синтезом*.

Гармонику с $k = 1$ называют *основной*, или *первой*, *гармоникой* сигнала. Она задает его частоту повторения f_1 . Остальные гармоники называют *высшими*, их частоты равны $f_k = k \cdot f_1$, где $k = 2, 3, \dots$. Таким образом, спектр периодических сигналов *дискретный* — он содержит набор фиксированных частот f_k , где $k = 1, 2, 3, \dots$. У непериодических сигналов спектр будет *сплошным* и вместо амплитуды гармоник он характеризуется *спектральной плотностью* сигнала.

7.4.2. Быстрые прямое и обратное преобразования Фурье

Переход от некоторой функции $f(t)$ к параметрам ее ряда Фурье (амплитудам и фазам гармоник) называется *прямым преобразованием Фурье*, а обратный переход — *обратным преобразованием Фурье*. К сожалению, эти переходы связаны с вычислением интегралов, подынтегральные функции в которых быстро осциллируют, что существенно затрудняет вычисление таких интегралов численными методами с заданной точностью и ведет к значительным затратам времени.

Если сигнал представлен в виде вектора дискретных значений, применяется *дискретное преобразование Фурье* (ДПФ), для которого, в свою очередь, существует алгоритм эффективной реализации вычислений, называемый *быстрым преобразованием Фурье* (БПФ, или FFT — Fast Fourier Transform). Увы, сложность этого процесса [36, 37] не позволяет детально описать его в настоящей книге. Ограничимся описанием функций, реализующих прямое и обратное БПФ в системе Mathcad, которая предоставляет возможность проводить указанные преобразования для данных в виде векторов с действительными и комплексными числами.

Функция $fft(v)$ системы Mathcad выполняет БПФ для данных, представленных действительными числами — значениями исходного вектора v . Однако при этом вектор v должен иметь 2^m составляющих, где m — целое число. В противном случае выводится сообщение об ошибке — неверном размере вектора. Если число элементов вектора v все же отличается от целой степени двух, то его можно дополнить нулями до этой величины.

Элементы вектора, возвращаемого функцией $fft(v)$, соответствуют формуле

$$C_j = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} v_k e^{2\pi i(j/n)k}.$$

Здесь n — число элементов вектора v , i — мнимая единица, k — индекс суммирования (от 0 до $n - 1$) и j — номер гармоники (от 0 до $n/2$). Эти элементы вектора соответствуют следующим частотам:

$$f_j = \frac{j}{n} \cdot f_s.$$

Здесь f_s — частота квантования сигнала, который подвергается БПФ. Элементы вектора, возвращаемого функцией $fft(v)$, — это в общем случае комплексные числа, даже если сигнал представлен вещественными отсчетами.

Функция $ifft(v)$ реализует обратное (инверсное) преобразование Фурье для вектора v с комплексными элементами. Вектор v здесь должен иметь $1 + 2^{m+1}$

элементов. В противном случае выдается сообщение об ошибке. Функция $\text{ifft}(v)$ вначале создает вектор w , комплексно-сопряженный с v , и затем присоединяет его к вектору v . После этого вычисляется вектор d с элементами, рассчитанными по следующей формуле:

$$d_j = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} w_k e^{-2\pi i(j/n)k}.$$

Функции $\text{fft}(v)$ и $\text{ifft}(v)$ дают точные (в пределах погрешности численных расчетов) обращения. При этом $\text{ifft}(\text{fft}(v)) = v$, что можно использовать для проверки преобразований.

Функция $\text{cfft}(A)$ аналогична предыдущей, но реализует прямое преобразование Фурье для вектора A с комплексными элементами. Если A — матрица, реализуется двумерное преобразование. Введение функции $\text{fft}(v)$ обусловлено тем, что преобразование для векторов с действительными элементами реализуется по более быстрому алгоритму (БПФ) и занимает меньше времени. Кроме того, в этом случае проще ввод исходных данных.

Функция $\text{icfft}(B)$ выполняет обратное преобразование Фурье по полному алгоритму, при котором как исходный, так и результирующий вектор или матрицы содержат элементы с комплексными значениями.

7.4.3. Примеры преобразований Фурье

Для проверки преобразований Фурье зададим некоторый вектор x из $2^3 = 8$ комплексных элементов и, проведя прямое преобразование с помощью функции cfft , получим новый вектор Y .

Исходные данные (в том числе выражение с функцией преобразования):

$$i := \sqrt{-1} \quad X := \begin{pmatrix} 1 + 8 \cdot i \\ 2 + 7 \cdot i \\ 3 + 6 \cdot i \\ 4 + 5 \cdot i \\ 5 + 4 \cdot i \\ 6 + 3 \cdot i \\ 7 + 2 \cdot i \\ 8 + 1 \cdot i \end{pmatrix} \quad Y := \text{cfft}(X)$$

Результат:

$$Y = \begin{pmatrix} 12.728 + 12.728i \\ -4.828 - 2i \\ -2.828 \\ -2 + 0.828i \\ -1.414 + 1.414i \\ -0.828 + 2i \\ 2.828i \\ 2 + 4.828i \end{pmatrix}$$

Далее проведем над полученным вектором Y обратное преобразование, используя функцию `icfft`.

Исходные данные (помимо полученного ранее вектора Y):

$$XI := icfft(Y)$$

Результат:

$$XI = \begin{pmatrix} 1 + 8i \\ 2 + 7i \\ 3 + 6i \\ 4 + 5i \\ 5 + 4i \\ 6 + 3i \\ 7 + 2i \\ 8 + 1i \end{pmatrix}$$

Нетрудно заметить, что полученный таким двукратным преобразованием вектор XI полностью совпадает с исходным.

Прямое преобразование Фурье по существу означает перевод временной зависимости в ее частотный спектр. А обратное преобразование Фурье переводит частотный спектр вновь во временную зависимость. Технику проведения БПФ на примере разложения прямоугольного импульса и последующего его синтеза с помощью ряда Фурье с ограниченным ($k = 10$) числом гармоник иллюстрирует фрагмент документа Mathcad, показанный на рис. 7.14. Здесь исходный вектор задан элементарии действительного типа, поэтому используются функции `fft` и `ifft`.

БПФ широко применяется при решении задач фильтрации сигналов или аппроксимации функций. При ограниченном числе гармоник приближение функ-

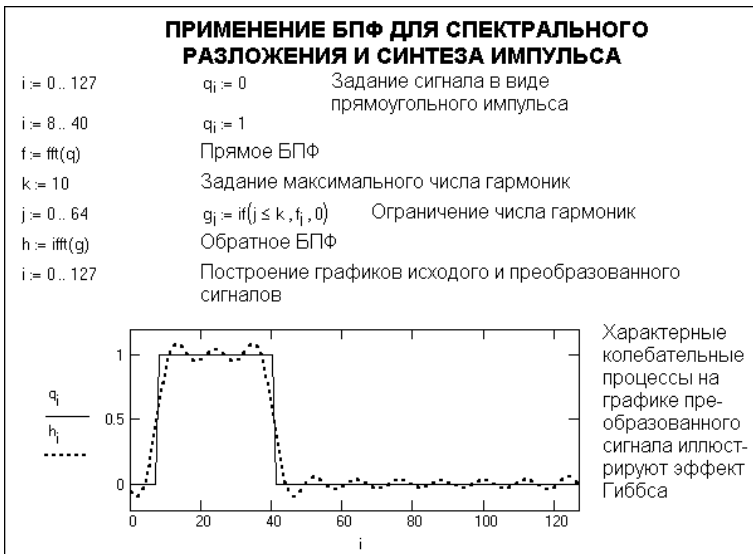


Рис. 7.14. Применение БПФ для спектрального разложения и синтеза прямоугольного импульса

ции тригонометрическим рядом Фурье обеспечивает наименьшую среднеквадратичную погрешность, если при этом используется БПФ.

Фрагмент документа Mathcad, показанный на рис. 7.15, иллюстрирует построение АЧХ и ФЧХ спектра прямоугольного импульса (см. рис. 7.14) для случая, когда прямое БПФ реализовано функцией `fft`. Обратите внимание на то, что для вычисления амплитуд гармоник используется оператор вычисления модуля, а для вычисления фаз гармоник — функция `arg`.

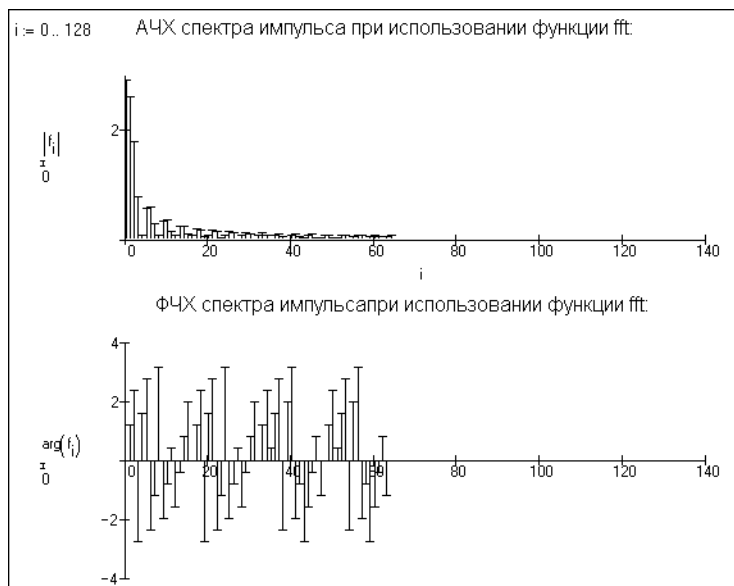


Рис. 7.15. АЧХ (сверху) и ФЧХ (снизу) спектра прямоугольного импульса, показанного на рис. 7.14, при использовании функции `fft` для прямого БПФ

При вычислении АЧХ и ФЧХ использован особый вид графика с представлением зон «ошибок». Такой график строится форматированием графиков двух линий, причем в качестве типа линий (см. подраздел «Форматирование линий графиков» раздела «Форматирование двумерных графиков в прямоугольной системе координат» в главе 5) выбирается вариант `Error` (Ошибка). В данном случае за вторую линию взята линия, значения ординат которой равны 0. Поэтому получается график в виде отсчетов первой (основной) линии, представляющий точечные значения амплитуд и фаз спектра гармоник. Для функции `fft` спектр ограничен числом гармоник, вдвое меньшим, чем число отсчетов сигнала — в нашем случае прямоугольного импульса. Это ограничение хорошо видно на рисунке. АЧХ и ФЧХ спектра имеют обычный вид.

На рис. 7.16 показаны АЧХ и ФЧХ того же импульса, но полученные с помощью функции `cfft`. На этот раз можно заметить два принципиально важных отличия:

- выходной вектор имеет то же число отсчетов, что и входной, то есть вдвое больше, чем при использовании функции `fft`;
- АЧХ и ФЧХ спектра имеют двойную длину и представлены как обычным своим видом в левой части графика, так и их зеркальным отражением в его правой части.

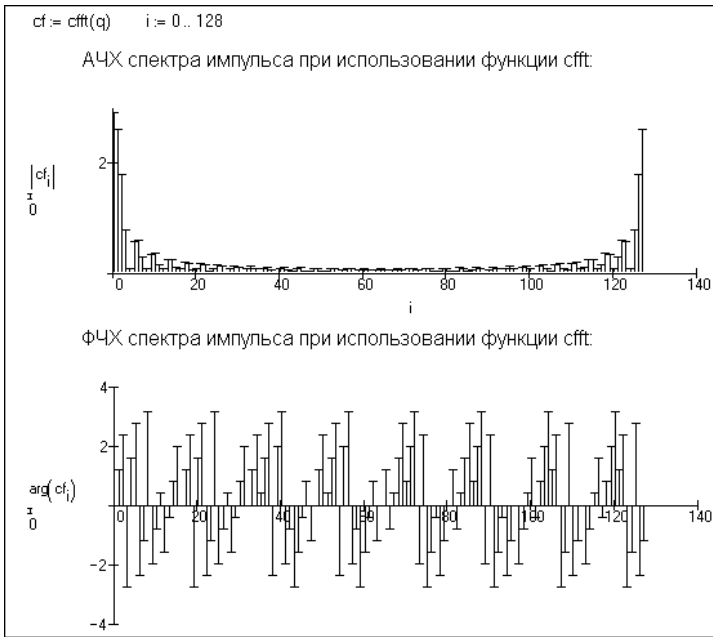


Рис. 7.16. АЧХ (сверху) и ФЧХ (снизу) спектра прямоугольного импульса, показанного на рис. 7.15, при использовании функции cfft для прямого БПФ

На рис. 7.17 показан еще один фрагмент документа Mathcad с примером прямого и обратного преобразований Фурье. Здесь для преобразования использована сложная трехкомпонентная функция, содержащая два синусоидальных компонента и компонент в виде случайных чисел.

На левом графике показанного на рисунке фрагмента документа Mathcad производится сравнение исходной функции с результатом ее прямого и обратного



Рис. 7.17. Пример БПФ для сложной функции с построением ее спектрограммы

преобразований Фурье, а на правом графике дается построение спектрограммы функции (на этот раз линией, представляющей собой огибающую спектра). Читателю рекомендуется после прямого БПФ задать искажения гармоник, характерные для работы какого-либо фильтра, — при этом можно наблюдать степень искажения сигнала, прошедшего через фильтр.

7.4.4. Альтернативные преобразования Фурье

Рассмотренные выше функции основаны на обычных формулах преобразований Фурье. Однако существуют и *альтернативные формы* такого преобразования, две из которых показаны ниже:

$$F(v) = \frac{1}{n} \sum_{\tau=1}^n f(\tau) e^{-2\pi i v (\tau/n)},$$

$$f(\tau) = \sum_{v=1}^n F(v) e^{-2\pi i v (\tau/n)}.$$

Вместо множителя $1/\sqrt{n}$ перед обоими выражениями перед первым выражением стоит множитель $1/n$, а перед вторым — 1. Знак «минус» перед показателем степени имеется только в первой формуле (его нет во второй).

Альтернативные формулы преобразований Фурье используются в функциях `FFFT(v)`, `IFFFT(v)`, `CFFT(v)` и `ICFFT(v)`. В остальном использование этих функций не отличается от аналогичных функций `fft(v)`, `ifft(v)`, `cfft(v)` и `icfft(v)`. Надо лишь помнить о том, что нормировка функций БПФ в литературе может быть различной. Это нередко создает путаницу при проведении операций спектрального анализа и синтеза и приводит к разным результатам.

7.4.5. Двумерное преобразование Фурье

Если в функциях преобразования Фурье в качестве входного параметра задана матрица, то реализуется двумерное обратное преобразование Фурье. Заметим, что в ранних версиях эта возможность отсутствовала, но могла быть реализована (хотя и более сложным путем) с помощью одномерного БПФ.

Фрагмент документа Mathcad с примером двумерного прямого преобразования Фурье для функции двух переменных $f(i, j)$ представлен на рис. 7.18.

В этом примере использована функция альтернативного прямого комплексного БПФ `CFFT`. Рекомендуется опробовать этот пример с функцией `cfft` и проанализировать изменения в графике, представленном на рис. 7.18 справа.

7.5. Дискретные волновые (вейвлет) преобразования

В последние годы широкое распространение получила техника так называемых дискретных волновых преобразований, которые еще называют вейвлет-, или wavelet-преобразованиями [38—41].

7.5.1. Вейвлеты как новый базис представления сигналов

Из теории сигналов известно, что произвольный сигнал $s(t)$ можно представить в виде взвешенной суммы простых составляющих — базисных функций $\Psi_k(t)$, помноженных на коэффициенты C_k :

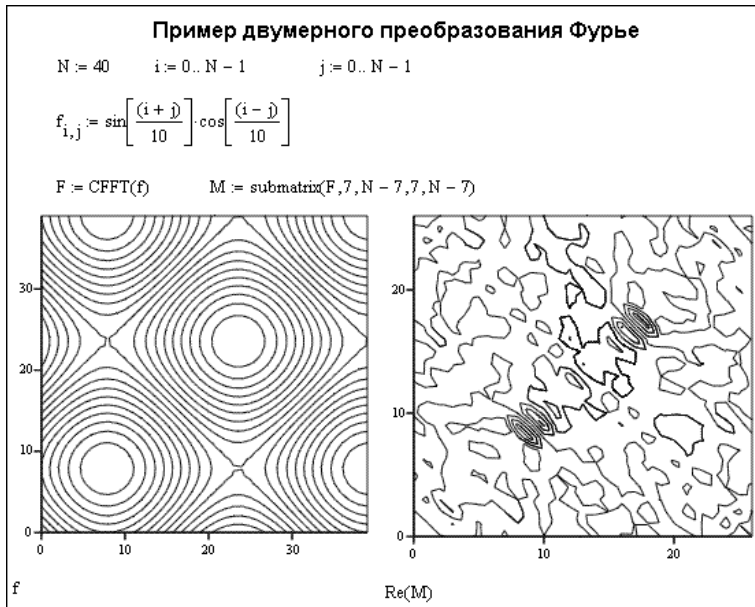


Рис. 7.18. Пример двумерного прямого БПФ для функции двух переменных

$$s(t) = \sum_k C_k \psi_k(t). \quad (7.1)$$

Так как базисные функции $\psi_k(t)$ предполагаются заданными как функции вполне определенного вида, то только коэффициенты C_k содержат информацию о конкретном сигнале. Таким образом, можно говорить о возможности представления произвольных сигналов на основе рядов общего вида (7.1) с различными базисными функциями. Так, хорошо известные синусоидальные функции легли в основу рядов Фурье, рассмотренных выше.

Синусоиды-гармоники периодических сигналов (иные мы, простоты ради, не рассматриваем) предельно локализованы в частотной области, вырождаясь на спектрограммах в вертикальные линии, но не локализованы вообще во временной области — они определены в интервале времен $-\infty$ до $+\infty$. Подобное определение является теоретической абстракцией. Поэтому содержащие их ряды Фурье плохо пригодны для представления коротких локальных особенностей сигналов и функций, таких как перепады и скачки. В таких местах зарождается эффект Гиббса.

Более того, ряды Фурье в классическом виде принципиально непригодны для представления нестационарных сигналов. Представьте мысленно, что некоторый сигнал содержит N синусоидальных компонент с кратными частотами, действующими все время. А другой сигнал содержит эти компоненты, но действующие поодиночке на отрезках времени T/N , где T — период первой компоненты. Такие два сигнала имеют совершенно разную форму временной зависимости, но спектр Фурье их качественно абсолютно идентичен — это N линий с частотами компонент.

В связи с этим многие годы ученые искали иные базисы для разложения сигналов. В начале 90-х годов было всерьез доказано, что такие достаточно универсальные базисы и впрямь существуют и получили название вейвлетов. Термин *вейвлет*, в переводе с английского *wavelet*, означает «короткая или маленькая волна»

или «волночка». На основе совокупности таких волн, перемещаемых и масштабируемых, и зародилась техника вейвлет-преобразований.

Оказалось, что простейший вейвлет Хаара (однократная волна прямоугольной формы в виде меандра) был известен еще в 1910 г., но тогда никто не догадывался, что он является новым базисом декомпозиции произвольных функций и сигналов с возможностью их абсолютно точного восстановления. Затем были открыты десятки новых и старых вейвлет-функций, причем реализация большинства из них возможна только итерационными и программными методами.

Дадим наглядную трактовку применения вейвлетов Хаара. Пусть имеется сигнал, представленный целочисленными компонентами вектора [9 7 3 5]. Это могут быть, например, значения пикселей некоторой подстроки изображения. Разрешение в этом случае равно 4. Перейдем к более грубому (вдвое меньшему) разрешению 2, для чего вычислим среднее из каждой пары компонентов сигнала. Получим вектор [8 4] с двумя детализирующими коэффициентами [1 -1]. Одни представляют половину от приращений уровня относительно среднего значения, т. е. $(9 - 7)/2 = 1$ и $(3 - 5)/2 = -1$.

Прибавив и отняв первый коэффициент от первого компонента вектора огрубленного сигнала — числа 8 получим компоненты 9 и 7. Аналогично, прибавив и отняв -1 от второго компонента вектора огрубленного сигнала 4, получим 3 и 5, т. е. вторую пару компонентов исходного вектора.

Продолжим огрублять сигнал вдвое и перейдем к разрешению 1. Наш вектор превратится в [6] с детализирующим коэффициентом 2. Его прибавление и отнимание дадут вектор [8 4]. Итак, для декомпозиции (разложения) исходного сигнала имеем:

Разрешение	Аппроксимирующие коэффициенты	Детализирующие коэффициенты
4	[9 7 3 5]	
2	[8 4]	[1 -1]
1	[6]	[2]

Таким образом, для представления сигнала достаточно хранить его грубое значение 6 и детализирующие коэффициенты 2, 1 и -1. Операции с ними задаются видом вейвлета Хаара. Например, на уровне разрешения 1 он представляется двумя функциями — аппроксимирующей с уровнем 1 и детализирующей с уровнем +1 на первой половине периода и -1 на второй половине периода (именно это задает вначале сложение, а затем вычитание детализирующего коэффициента). В итоге, осуществляя композицию сигнала, мы точно восстанавливаем его значение, используя последний (самый грубый) аппроксимирующий коэффициент и ряд детализирующих коэффициентов.

Процедуры изменения разрешения вдвое в ходе композиции и декомпозиции реализуют так называемый *диадический* метод. Он является разновидностью более общего *кратномасштабного* метода и лежит в основе устранения избыточности, свойственной непрерывным вейвлет-преобразованиям (см. ниже).

Казалось бы, какой прок в таком представлении, коли число компонентов вектора осталось неизменным? Оказывается, прок есть, и весьма существенный. Прежде всего, мы перешли от представления независимых значений сигнала к его приращениям. Коэффициенты вейвлет-представления реальных сигналов часто существенно меньше числа, чем представления отсчетов сигналов. Для реальных сигналов многие коэффициенты по уровню оказываются настолько малыми, что их можно отбросить. Это означает возможность значительного сокращения объе-

ма информации о сигнале, выполнения его компрессии и очистки от шумов. Добавьте к этому, что сейчас есть множество куда более ценных и интересных вейвлетов, что дает обширный выбор базисных функций как для точного, так и приближенного представления любых сигналов.

Правда, точное представление могут давать только так называемые *ортогональные вейвлеты*. Объем этого раздела не позволяет остановиться на строгом определении *ортогональности*, как и многих иных свойств вейвлетов. Они даны в литературе по ним [34—37]. Мы же будем исходить из того, что такие вейвлеты есть. Кроме вейвлета Хаара, к ним относятся хорошо известные вейвлеты Добеши, для прямого и обратного преобразований которых Mathcad имеет встроенные в ядро функции.

7.5.2. Особенности вейвлетов

Как и в случае преобразований Фурье, возможны два вида вейвлет-преобразований — прямое и обратное. Прямое означает вейвлет-декомпозицию сигнала, т. е. его разложение по базису вейвлетов. В основе непрерывного вейвлет-преобразования НВП (или CWT — Continue Wavelet Transform) лежит использование двух непрерывных и интегрируемых по всей оси t (или x) функций:

- вейвлет-функция $\psi(t)$ с нулевым значением интеграла $(\int_{-\infty}^{\infty} \psi(t) dt = 0)$, определяющая детали сигнала и порождающая детализирующие коэффициенты;
- масштабирующая или скейлинг-функция $\phi(t)$ с единичным значением интеграла $(\int_{-\infty}^{\infty} \phi(t) dt = 1)$, определяющая грубое приближение (аппроксимацию) сигнала и порождающая коэффициенты аппроксимации.

ϕ -функции $\phi(t)$ присущи только тем вейвлетам, которые относятся к ортогональным. ψ -функция $\psi(t)$ создается на основе той или иной *базисной функции* $\psi_0(t)$, которая, как и $\psi(t)$, определяет тип вейвлета. Базисная функция должна удовлетворять всем тем требованиям, которые были отмечены для ψ -функции $\psi(t)$. Она должна обеспечивать выполнение двух основных операций:

- смещение по оси времени $t - \psi_0(t - b)$ при $b \in \mathbf{R}$;
- масштабирование $- a^{-1/2} \psi_0\left(\frac{t}{a}\right)$ при $a > 0$ и $a \in \mathbf{R}^+ - \{0\}$.

Параметр a задает ширину этого пакета, а b — его положение, \mathbf{R} — задает область определения параметров, в общем случае бесконечную (верхний индекс $+$ означает область положительную, а обозначение $-\{0\}$ исключение нулевой точки). В ряде литературных источников вместо явного указания времени t используется аргумент x , а вместо параметров a и b используются имеющие тот же смысл иные обозначения. Нетрудно убедиться в том, что следующее выражение задает сразу два этих свойства функции $\psi(t)$:

$$\psi(t) \equiv \psi(a, b, t) = a^{-1/2} \psi_0\left(\frac{t - b}{a}\right).$$

Итак, для заданных a и b функция $\psi(t)$ и есть *вейвлет*. Вейвлеты являются вещественными функциями времени t и колеблются вокруг оси t (или x). Параметр b задает положение вейвлетов, а параметр a — их масштаб. О вейвлетах, четко локализованных в пространстве (или во времени), говорят, что они имеют *компактный носитель*.

7.5.3. Вейвлеты типа «мексиканская шляпа»

Вейвлеты характеризуются своим временным и частотным образами. Временной образ определяется некоторой *psi-функцией* времени $\psi(t)$. А частотный образ определяется ее *фурье-образом* $\hat{\psi}(\omega) = F(\omega)$, который задает огибающую спектра вейвлета. Фурье-образ определяется выражением:

$$F(\omega) = \int_{-\infty}^{\infty} \hat{\psi}(t) e^{-i\omega t} dt.$$

Для демонстрации особенностей вейвлетов часто используется вейвлет типа «мексиканская шляпа». Его временной образ, в отличие от подавляющего большинства других вейвлетов, описывается аналитическим выражением:

$$mhat(t) = (1 - t^2)e^{-t^2/2}.$$

Это выражение является второй производной гауссиана, в которой опущен не имеющий принципиального значения числовой множитель. С помощью системы Mathcad, имеющей оператор символьного преобразования \rightarrow и средства построения графиков функций, нетрудно получить и представить графически временной и частотный образ этого вейвлета (рис. 7.19). Иногда предпочитают перевернутый вейвлет «мексиканская шляпа», у которого главный пик обращен вниз.

Хорошо видно, что данный вейвлет напоминает сильно затухающее синусоидальное колебание с некоторой «средней частотой» и площадью (над осью времени

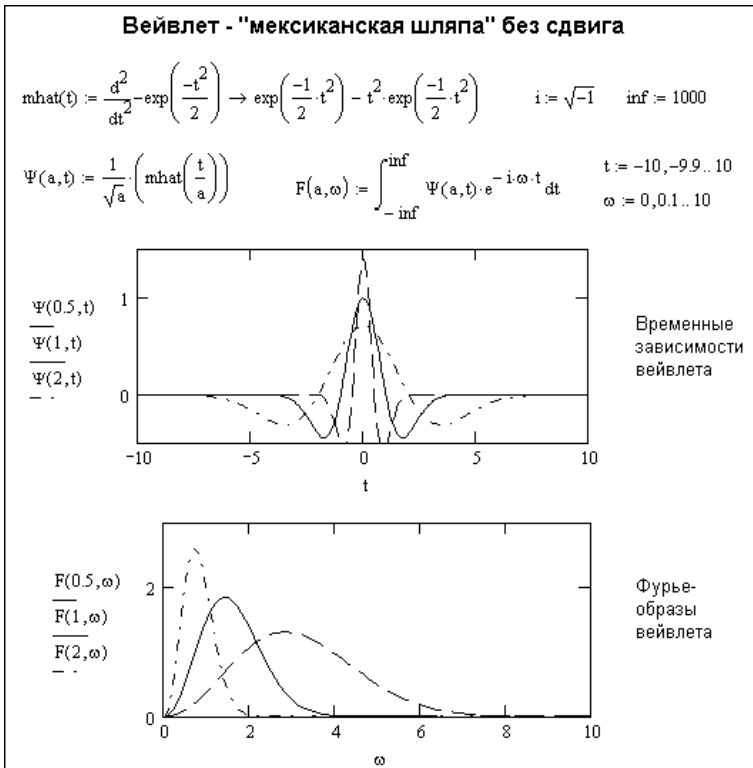


Рис. 7.19. Временной и частотный образы вейвлета «мексиканская шляпа»

и под ней), близкой к нулю. На самом деле точное нулевое значение площади (над и под осью времени) — одно из важнейших условий, позволяющих отнести временную зависимость $\hat{m}(t)$ к вейвлетам. Если вейвлет в пространстве сужается, его «средняя частота» повышается, спектр вейвлета перемещается в область более высоких частот и расширяется. Этот процесс можно считать линейным — если вейвлет сужается вдвое, то его средняя частота и ширина спектра возрастают также вдвое.

На рис. 7.20 показано еще одно построение вейвлета «мексиканская шляпа». На этот раз «шляпа» представлена перевернутой сменой знака у исходной временной функции.

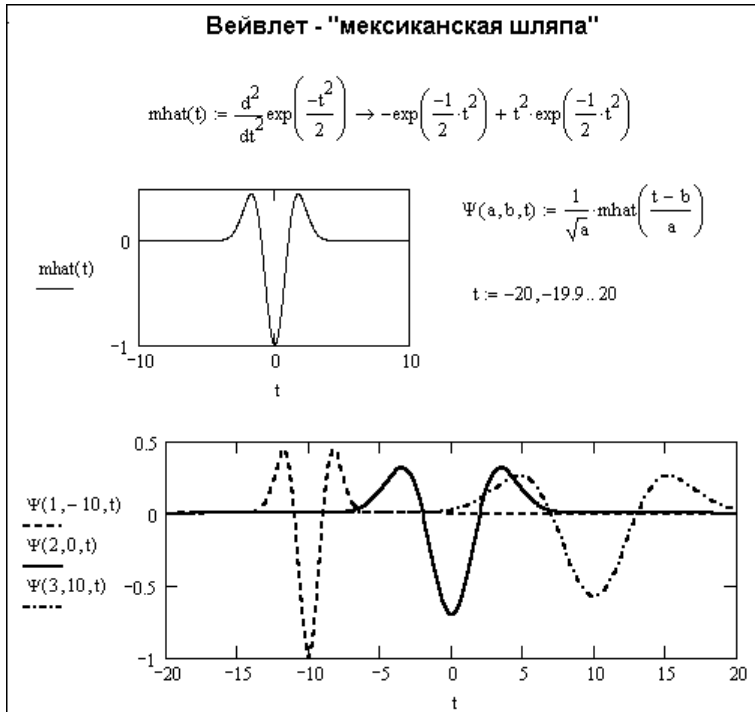


Рис. 7.20. Иллюстрация к сдвигу и масштабированию вейвлета типа «мексиканская шляпа»

Здесь представлена базисная функция данного вейвлета и функция $\Psi(a, b, t)$ для разных a и b , что иллюстрирует сдвиг вейвлета и его масштабирование. В Mathcad для обеспечения изменений a и b функцию $\Psi(a, b, t)$ приходится задавать в более полном виде, чем $\Psi(t)$, в котором зависимость Ψ от a и b лишь подразумевается.

В частотной области малые значения a соответствуют высоким частотам, а большие — низким частотам. Таким образом, операция задания окна, используемая в оконном преобразовании Фурье, как бы заложена в самой базисной функции вейвлетов. Это создает предпосылки их приспособления (адаптации) к сигналам, которые могут быть представлены совокупностью вейвлетов.

Тут важно отметить, что невозможна хорошая локализация вейвлетов одновременно во временной и частотной областях. Однако было бы неверно оценивать это обстоятельство как недостаток вейвлетов. Напротив, в сравнении с оконным

преобразованием Фурье, именно хорошая временная локализация вейвлетов дает нужное средство для повышения частоты тех спектральных составляющих вейвлетов, которые и нужны для представления кратковременных локальных особенностей сигналов.

7.5.4. Прямое и обратное непрерывные вейвлет-преобразования

Прямое непрерывное вейвлет-преобразование (ПНВП) сигнала $s(t)$ задается, по формальной аналогии с преобразованием Фурье, путем вычисления вейвлет-коэффициентов по формуле:

$$C(a, b) = \int_{-\infty}^{\infty} s(t) a^{-1/2} \psi\left(\frac{t-b}{a}\right) dt.$$

Если сигнал ограничен во времени $a, b \in \mathbf{R}$, $a \neq 0$, то интегрировать можно в конечных пределах:

$$C(a, b) = \int_R s(t) a^{-1/2} \psi\left(\frac{t-b}{a}\right) dt.$$

Обратное вейвлет-преобразование задается выражением

$$s(t) = \frac{1}{K_\psi} \int_{R^+} \int_R C(a, b) a^{-1/2} \psi\left(\frac{t-b}{a}\right) \frac{dadb}{a^2}$$

и обеспечивает восстановление сигнала, если используемые вейвлеты ортогональны. К сожалению, вейвлет типа «мексиканская шляпа» к таковым не относится.

Как уже отмечалось, на практике непрерывное изменение параметров a и b вызывает избыточность вейвлет-представления сигналов. Поэтому, как это описывалось в примере с вейвлетом Хаара, используется кратномасштабный (в частности диадический) метод.

7.5.5. Примеры непрерывных вейвлет-преобразований и построение их спектрограмм

Рисунок 7.21 демонстрирует документ системы Mathcad, обеспечивающий создание вейвлета «мексиканская шляпа» и проведение прямого вейвлет-преобразования для функции, которая имеет вид симметричных прямоугольных импульсов — меандра. Они создаются с помощью выражения $s(t) := \text{sign}(\sin(0.04 * t))$, где sign — сигнум-функция $\text{sign}(x) = 1$ при $x > 0$, 0 при $x = 0$ и -1 при $x < 0$.

В представленном документе системы Mathcad вычисление вейвлет-коэффициентов $C(a, b)$ выполняется стандартным оператором интегрирования, что придает документу, с одной стороны, прекрасную наглядность, а с другой — обрекает его на медлительность исполнения.

На другом рисунке (рис. 7.22) представлено прямое вейвлет-преобразование для периодической функции синуса в кубе. Такая функция имеет ряд особенностей, в частности переходы производной через нуль в моменты экстремумов и явные перегибы в моменты перехода функции через нуль.

Итак, эти примеры столь же наглядны, сколь и малоэффективны — даже на ПК автора этой книги с процессором Pentium III 600 МГц и оперативной памятью в 320 Мбайт время вычислений составляет до нескольких минут. Последнее связано с тем, что вычисление определенных интегралов в СКМ Mathcad реализо-

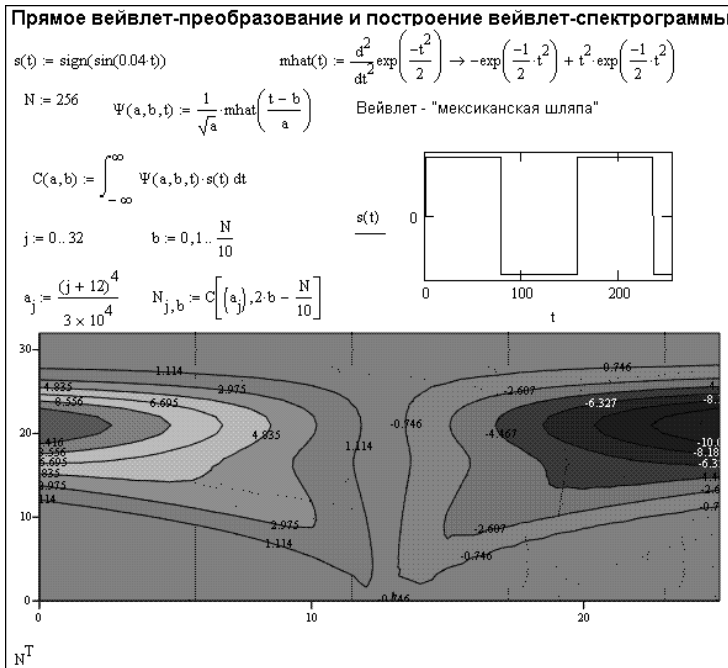


Рис. 7.21. Прямое вейвлет-преобразование для периодических прямоугольных импульсов — меандра

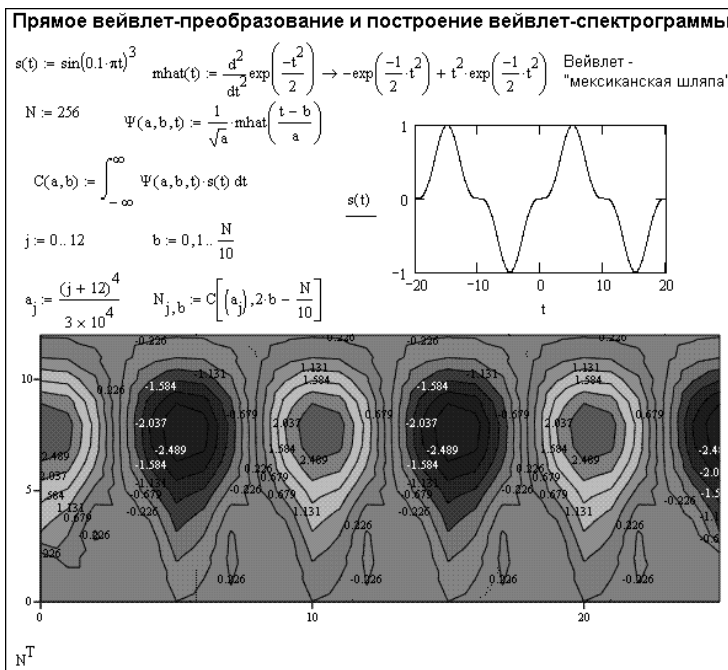


Рис. 7.22. Прямое вейвлет-преобразование для искаженной синусоидальной функции

вано довольно сложным адаптивным методом интегрирования. На практике для вейвлетов с компактным носителем нет никакой необходимости вычислять интегралы с бесконечными пределами. Можно учесть, что области изменения параметров a и b ограничены.

И еще одно важное обстоятельство. Практически невозможно вести вычисления с представленными непрерывными вейвлетами. На практике, так или иначе, приходится задавать дискретные значения a и b , например для графического представления результатов вычислений, что и сделано в конце документов, представленных на рис. 7.21 и 7.22. Единственной специфической возможностью непрерывного преобразования в этих примерах, иногда ошибочно именуемого «дискретным», является практически произвольный (в пределах общих теоретических допущений) выбор значений a и b . Практически он сделан таким, чтобы представить вейвлет-спектрограммы в этих примерах в наиболее наглядном (а точнее детальном) виде.

Непрерывное прямое вейвлет-преобразование обладает большой избыточностью, что и ведет к неоправданно большим затратам машинного времени на его вычисления. Однако, как уже отмечалось, это преобразование, даже с вейвлетами, не допускающими реконструкции сигналов, может использоваться для построения вейвлет-спектрограмм в плоскости «коэффициенты-время» (более точно, пожалуй, назвать эту плоскость «коэффициенты-сдвиг»). Такие спектрограммы и представлены в Mathcad-документах рис. 7.21 и 7.22. В данном случае спектрограммы строятся с применением контурных графиков с функциональной закраской областей.

Избыточность непрерывного вейвлет-преобразования в некотором смысле дискредитирует его на практике. К примеру, вейвлет-спектрограммы в зависимости от их конкретной реализации, заданной соответствующими программными средствами, порой могут давать самую разнообразную окраску тех или иных локальных особенностей сигнала. Это может приводить к ошибочной интерпретации рентгеновских снимков или снимков томографа в медицине, лжеоткрытиям туманностей в астрофизике или к «обнаружению» полезных ископаемых по данным геофизической разведки, на самом деле просто несуществующих. Тем не менее вейвлет-спектрограммы являются важнейшим продуктом вейвлет-анализа сигналов и прекрасным дополнением к обычным спектрограммам на основе преобразований Фурье.

Различия между обычными спектрограммами Фурье-анализа и вейвлет-спектрограммами носят принципиальный характер. Обычные спектрограммы удобны для обнаружения периодического стационарного компонента сигнала. Если такой компонент гармонический сигнал, то для его представления достаточно всего одна гармоника ряда Фурье. В то же время особенности, например разрывы, сигналов спектрограммы Фурье выявляют очень плохо. Например, разрывы порождают множество гармоник малой амплитуды, которые «расплываются» на спектрограмме. О месте разрыва судить вообще нельзя.

При вейвлет-анализе даже гармонического сигнала нам придется использовать множество вейвлетов, перекрывающих всю ось времени. Бесспорно, это может показаться совершенно неоправданным усложнением для представленного весьма частного случая — когда сигнал априорно известен как простая синусоида. В этом случае преобразование Фурье просто означает представление синусоиды синусоидой! Однако если сигнал имеет локальные особенности, если он не периодичен и не стационарен, то вейвлет-анализ и полученные по его результатам спектрограм-

мы становятся именно тем инструментом исследования, к которым фурье-анализ в классическом виде неприменим.

Большие уровни при вейвлет-анализе характерны только для коэффициентов тех вейвлетов, которые располагаются вблизи той или иной локальной особенности функции. Это позволяет легко определять местоположение локальных особенностей, что практически невозможно при спектральном фурье-анализе. Особенно хорошо фиксируются разрывы функции и ее производных. Таким образом, вейвлет-спектрограммы дают представление в амплитудно-временной области (если мы рассматриваем сигнал в виде временной функции).

ВНИМАНИЕ *Вейвлет-анализ сигналов открывает принципиально новые возможности в детальном анализе тонких особенностей сигналов. Это особенно важно для звуковых сигналов и сигналов изображения, где именно такие особенности подчас определяют качество их воспроизведения. Биология, картография, медицина, акустика, астрономия и космос — все это именно те области, где применение вейвлетов способно привести к новым открытиям, путем выявления характерных особенностей сигналов и изображений, мало заметных на временных зависимостях сигналов и на их спектрах Фурье. Однако пользоваться формально построенными спектрограммами, без тщательного изучения причин возникновения тех или иных их особенностей, недопустимо, ибо может привести к «лжеоткрытиям».*

7.5.6. Встроенные в Mathcad функции вейвлет-преобразований

Начиная с версии Mathcad 2000 в систему Mathcad были включены две простые функции дискретных волновых преобразований:

- $\text{wave}(V)$ — дискретное волновое преобразование действительных чисел с использованием 4-коэффициентного волнового базиса Добеши. Вектор V должен содержать $2n$ действительных значений, где n — целое число;
- $\text{iwave}(V)$ — обратное преобразование относительно преобразования $\text{wave}(V)$ (V — вектор с числом элементов $2n$).

Эти функции реализуют пирамидальный алгоритм быстрого вейвлет-преобразования. Вейвлет-преобразования в книге [11] посвящена отдельная глава. В ней подробно описаны особенности вейвлетов и техника их расширенного применения. Здесь же мы рассмотрим только простые примеры применения описанных выше двух функций.

7.5.7. Простые примеры дискретных вейвлет-преобразований

Несмотря на весьма скромные средства вейвлет-преобразований в ядре системы Mathcad (сейчас известны сотни вариантов вейвлетов) они позволяют выполнять многие важные вычисления и преобразования, которые возможны с применением вейвлетов.

Фрагмент документа Mathcad с примером выполнения дискретных волновых преобразований показан на рис. 7.23.

В этом примере над сложным нестационарным сигналом, представленным 512 точками, осуществляются прямое и обратное вейвлет-преобразования, в результате чего исходная зависимость практически восстанавливается (с погрешностью порядка погрешности машинных расчетов — менее 10^{-14}). Совпадение исходного

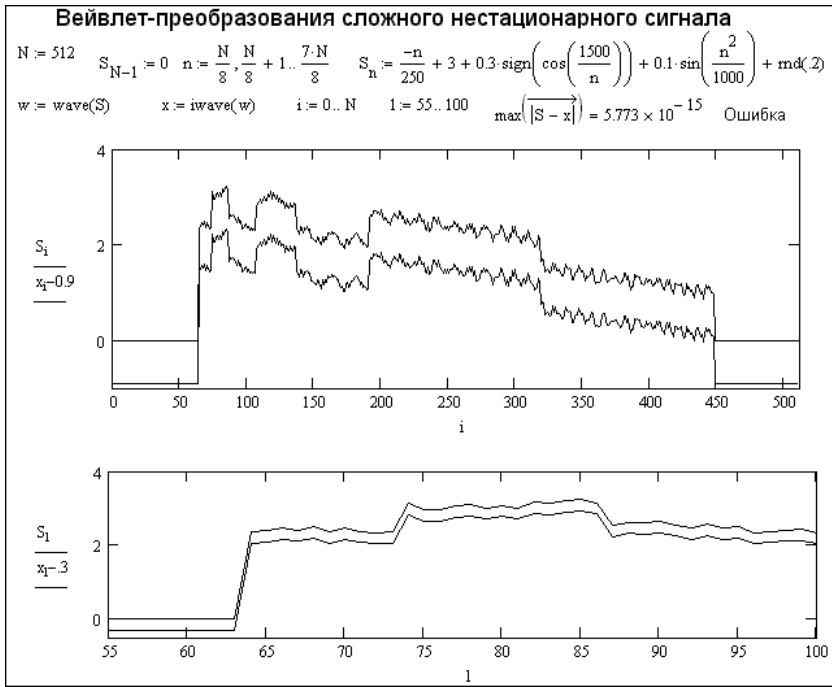


Рис. 7.23. Пример выполнения дискретных волновых преобразований

сигнала с синтезированными вейвлетами наблюдается как для всего сигнала, так и его малой части снизу документа. Однако такой результат характерен только для случая, когда используются практически все вейвлет-коэффициенты. В этом случае ортогональные вейвлеты обеспечивают точное восстановление произвольных сигналов (зависимостей) после проведения прямого и обратного вейвлет-преобразований.

Простейшую технику вейвлет-преобразований иллюстрирует фрагмент документа Mathcad, представленный на рис. 7.24. Здесь задается прямоугольный импульс и выполняется прямое вейвлет-преобразование. В его результате получается набор коэффициентов, причем для трех из них построены графики, из которых ясно, что вейвлет-коэффициенты являются функциями времени. В этом документе вычисляется максимальный уровень декомпозиции Nlevels, выше которого имеет место точное восстановление сигнала в ходе вейвлет-декомпозиции и композиции.

Следующий фрагмент документа Mathcad (рис. 7.25) иллюстрирует обратное вейвлет-преобразование, для которого используются только пять первых коэффициентов. Это аналогично эффекту ограничения числа гармоник при обычных преобразованиях Фурье и может рассматриваться как пример элементарной вейвлет-фильтрации исходного импульса. Естественно, что при этом возникают искажения формы восстановленного импульса, также показанные на рисунке. Не стоит отождествлять процесс вейвлет-фильтрации с фильтрацией с помощью фурье-преобразований путем ограничения числа гармоник ряда Фурье. Но тем не менее эффект, подобный эффекту Гиббса, присутствует при ограничении уровней диадического вейвлет-преобразования (прямого и обратного).

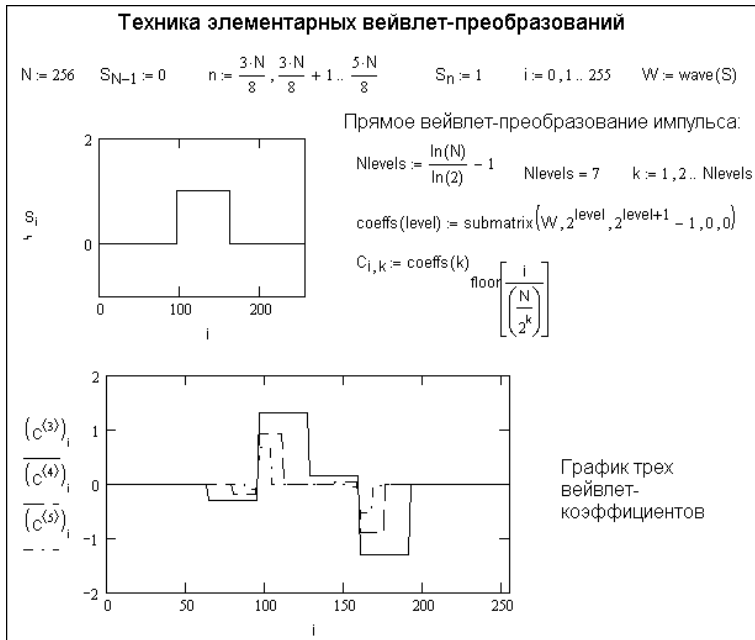


Рис. 7.24. Пример прямого вейвлет-преобразования прямоугольного импульса

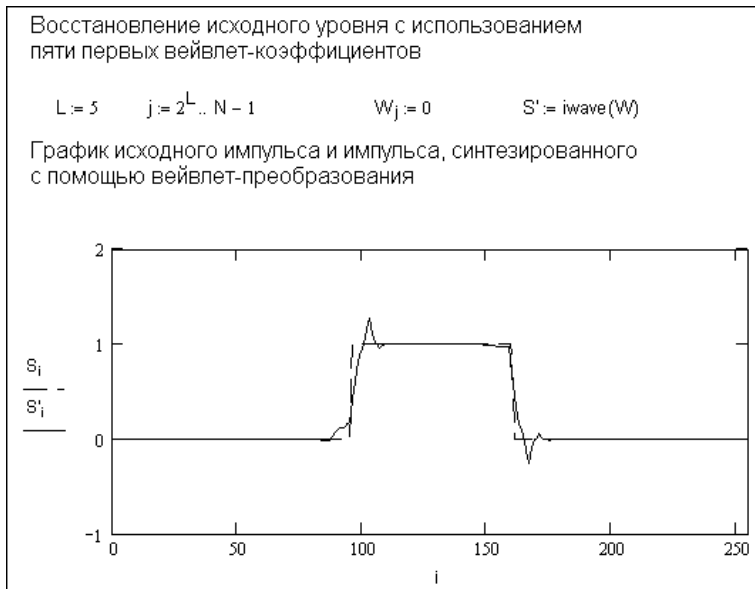


Рис. 7.25. Обратное вейвлет-преобразование прямоугольного импульса и его восстановление

В настоящее время дискретные волновые преобразования стали важным направлением в технике обработки информации. По ним издано множество книг [34–37]. В Mathcad есть специальное расширение Wavelet Extension Pack для ра-

боты с дискретными волновыми преобразованиями. В нем содержится свыше сотни новых функций, расширяющих возможности систем Mathcad в этой новой и перспективной области обработки сигналов и изображений. Это расширение описано в главе 15.

7.6. Решение обыкновенных дифференциальных уравнений

7.6.1. Представление дифференциальных уравнений

Многие серьезные физические и научно-технические задачи (особенно относящиеся к анализу динамических систем и к их математическому моделированию) базируются на решении систем дифференциальных уравнений (ДУ). Нелинейные ДУ и системы с такими уравнениями, как правило, не имеют аналитических методов решения, и здесь особенно важна возможность их решения численными методами. В большинстве случаев желательно представление решений в графическом виде. Эти возможности системы Mathcad и рассматриваются ниже.

Системы из обыкновенных дифференциальных уравнений (ОДУ) для их решения в среде Mathcad должны быть представлены в форме Коши:

$$\begin{cases} y_1(x_0) = y_{0,1} \\ y_2(x_0) = y_{0,2} \\ \dots \\ y_n(x_0) = y_{0,n} \end{cases},$$

$$\begin{cases} y_1' = f_1(x, y_1, y_2, \dots, y_n) \\ y_2' = f_2(x, y_1, y_2, \dots, y_n) \\ \dots \\ y_n' = f_n(x, y_1, y_2, \dots, y_n) \end{cases}.$$

Здесь первая система задает начальные условия, а вторая представляет систему ОДУ. Нетрудно заметить, что эти системы можно представить в векторном виде:

$$Y(x_0) = Y_0,$$

$$Y' = F(x, Y).$$

Отсюда следует важный вывод — решение системы ОДУ в форме Коши осуществляется аналогично решению одиночного ДУ, но должно быть организовано в *векторной форме*. При этом добавление очередного уравнения не увеличивает число уравнений в векторной их записи.

Дифференциальные уравнения n -го порядка:

$$y^{(n)} = f(x, y, y', y'', \dots, y^{(n-1)}),$$

где $y(x_0) = y_0$, $y'(x_0) = y_{0,1}$, $y''(x_0) = y_{0,2}$, ..., $y^{(n-1)}(x_0) = y_{0,n-1}$, можно свести к системе ДУ первого порядка. Для этого введем следующие обозначения:

$$y_1(x) = y(x), \quad y_2(x) = y'(x), \quad \dots, \quad y_n(x) = y^{(n-1)}(x),$$

$$y_{0,0} = y(x_0), \quad y_{0,1} = y'(x_0), \quad \dots, \quad y_{0,n-1} = y^{(n-1)}(x_0).$$

Теперь можно записать:

$$\begin{cases} y_1(x_0) = y_{0,0} \\ y_2(x_0) = y_{0,1} \\ \dots \\ y_n(x_0) = y_{0,n-1} \end{cases},$$

$$\begin{cases} y'_1 = y_2 \\ y'_2 = y_3 \\ \dots \\ y'_{n-1} = y_n \\ y'_n = f(x, y_1, y_2, \dots, y_{n-1}) \end{cases}.$$

В таком виде ДУ n -го порядка может решаться стандартными средствами решения систем ОДУ, входящими в большинство математических систем.

7.6.2. Решение систем обыкновенных дифференциальных уравнений

Для решения задач такого класса в Mathcad введен ряд функций. Вначале остановимся на функциях, дающих решения для систем обыкновенных дифференциальных уравнений, представленных в обычной форме Коши:

- `rkadapt(y, x1, x2, acc, n, F, k, s)` — возвращает матрицу, содержащую таблицу значений решения задачи Коши на интервале от x_1 до x_2 для системы обыкновенных дифференциальных уравнений, вычисленную методом Рунге—Кутта с переменным шагом и начальными условиями в векторе y (правые части системы записаны в векторе F , n — число шагов, k — максимальное число промежуточных точек решения и s — минимально допустимый интервал между точками, он же шаг интегрирования);
- `Rkadapt(y, x1, x2, n, F)` — возвращает матрицу решений методом Рунге—Кутта с переменным шагом для системы обыкновенных дифференциальных уравнений с начальными условиями в векторе y , правые части которых записаны в символьном векторе F на интервале от x_1 до x_2 при фиксированном числе шагов n ;
- `rkfixed(y, x1, x2, n, F)` — возвращает матрицу решений методом Рунге—Кутта системы обыкновенных дифференциальных уравнений с начальными условиями в векторе y , правые части которых записаны в символьном векторе F на интервале от x_1 до x_2 при фиксированном числе шагов n .

Создаваемая этими функциями матрица содержит ряд столбцов, число которых на 1 больше числа уравнений. Первый столбец содержит значения переменной x на равных интервалах решения, а другие столбцы — значения искоемых переменных. Если в процессе решения ищутся временные зависимости, то параметр x означает время t , т. е. $x = t$.

Фрагмент документа Mathcad, представленный на рис. 7.26, иллюстрирует технику решения системы из двух дифференциальных уравнений Ван—дер—Поля и представление решения в виде фазового портрета колебаний, которые описывает рассматриваемая система уравнений, а также временных зависимостей решения.

Система уравнений Ван—дер—Поля описывает в обобщенном виде широкий класс систем и устройств, относящихся к автогенераторам (впервые эта система была предложена для описания автогенераторов на электронных лампах, но вполне применима и к автогенераторам на современных полевых транзисторах [41]).

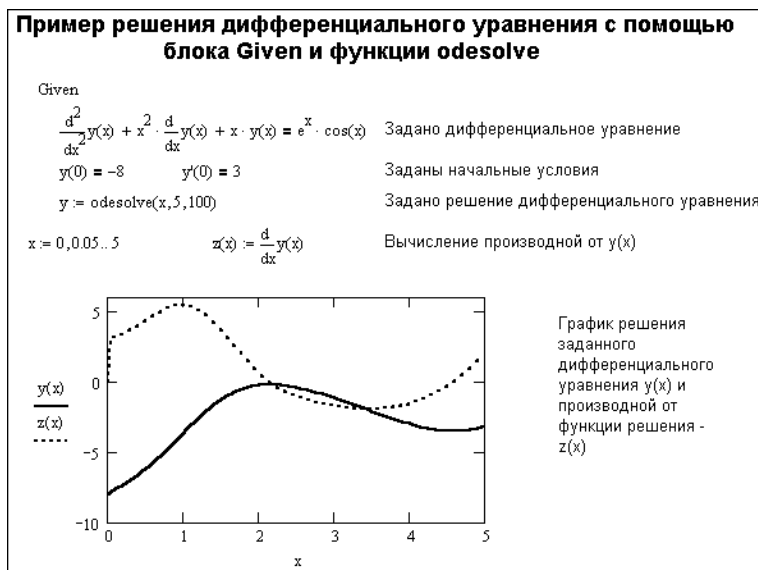


Рис. 7.26. Решение системы дифференциальных уравнений с применением функции *rkfixed*

Фрагмент документа Mathcad, представленный на рис. 7.27, иллюстрирует решение той же системы с применением функции *Rkadapt*. Эта функция благодаря автоматическому изменению шага решения дает более точный результат. Естественно, по скорости вычислений она проигрывает функции *rkfixed*, хотя и не всегда — если решение меняется медленно, это может привести к заметному уменьшению числа шагов. Таким образом, функция *Rkadapt* более привлекательна для решения систем дифференциальных уравнений, имеющих решения как с медленными, так и быстрыми участками изменения. Несмотря на автоматическое изменение шага вывод решения дается при равномерном расположении точек решения.

В последнем примере, как и в предыдущем, решение представлено в виде временных зависимостей. Много других примеров решения дифференциальных уравнений можно найти в электронных книгах системы.

Если решение системы дифференциальных уравнений имеет вид гладких функций, то вместо описанной ранее функции *rkfixed* целесообразно применять новую функцию *Bulstoer* ($y, x1, x2, n, F$). Она возвращает матрицу решения системы обыкновенных дифференциальных уравнений, правая часть которых (в виде первых производных неизвестных функций) записана в векторе $F(x, y)$ при заданных в векторе y начальных условиях и при решении на интервале от $x1$ до $x2$ для n точек решения, не считая начальной точки. Вы можете заменить функцию *Rkadapt* во фрагменте документа Mathcad, показанном на рис. 7.27, на функцию *Bulstoer* и опробовать ее в работе. Она реализует метод Булирша—Штера (*Bulirsch-Stoer*).



*Рис. 7.28. Решение дифференциального уравнения второго порядка с помощью функции *odesolve**

описывающих работу электронных цепей с сильно различающимися постоянными времени и при решении резко нелинейных дифференциальных уравнений.

Решение таких систем характерно резко различной скоростью изменения значений переменных и требует очень малого шага, выбираемого исходя из наивысшей скорости изменения значений переменных. Оно подчас просто невозможно указанными выше явными методами. Для решения жестких дифференциальных уравнений в Mathcad введен ряд функций:

- $\text{bulstoer}(y, x1, x2, \text{acc}, n, F, k, s)$ — возвращает матрицу решения системы обыкновенных дифференциальных уравнений на интервале от $x1$ до $x2$, правая часть которых записана в символьном векторе F с заданными в векторе y начальными условиями (используется метод решения Булирша—Штера с переменным шагом, параметры k и s задают максимальное число промежуточных точек, на которых ищется решение, и минимально допустимый интервал между ними);
- $\text{Stiffb}(y, x1, x2, n, F, J)$ — возвращает матрицу решений жесткого дифференциального уравнения, записанного в векторе F и функции Якобиана J , y — вектор начальных значений на интервале $[x1, x2]$ (для решения используется метод Булирша—Штера);
- $\text{stiffb}(y, x1, x2, \text{acc}, n, F, J, k, s)$ — возвращает матрицу решений только в конечной точке жесткого дифференциального уравнения, записанного в векторе F и функции Якобиана J , y — вектор начальных значений на интервале $[x1, x2]$ (для решения используется метод Булирша—Штера с переменным шагом);
- $\text{StiffR}(y, x1, x2, n, F, J)$ — возвращает матрицу решений дифференциального уравнения, записанного в векторе F и функции Якобиана J , y — вектор начальных значений на интервале $[x1, x2]$ (для решения используется метод Розенброка);

- `stiff(r(y, x1, x2, acc, n, F, J, k, s))` — матрица решений только в конечной форме жесткого дифференциального уравнения, записанного в векторе F и функции Якобиана J , y — вектор начальных значений на интервале $[x1, x2]$ (для решения используется метод Розенброка с переменным шагом).

В приведенных функциях: acc — погрешность решения (рекомендуется порядка 0.001), k — максимальное число промежуточных точек и s — минимально допустимый интервал между точками, в которых ищется решение (шаг интегрирования). Обратите внимание, что функции, начинающиеся с малой буквы, дают решения только для конечной точки. Различаются функции также и методом решения. Если решение расходится, что может привести к переполнению разрядной сетки чисел, то прежде всего надо попытаться уменьшить шаг интегрирования.

Матрица-функция Якоби J , фигурирующая в этих функциях, имеет размер $n \times (n+1)$ и представляется в виде:

$$J(x, y) = \begin{bmatrix} \frac{\partial f_1(x, y)}{\partial x} & \frac{\partial f_1(x, y)}{\partial y_1} & \dots & \frac{\partial f_1(x, y)}{\partial y_n} \\ \frac{\partial f_2(x, y)}{\partial x} & \frac{\partial f_2(x, y)}{\partial y_1} & \dots & \frac{\partial f_2(x, y)}{\partial y_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial f_n(\ddot{x}, y)}{\partial x} & \frac{\partial f_n(\ddot{x}, y)}{\partial y_1} & \dots & \frac{\partial f_n(\ddot{x}, y)}{\partial y_n} \end{bmatrix}$$

Чаще всего эту матрицу несложно вычислить вручную или с помощью символического дифференцирования.

Фрагмент документа Mathcad с простым примером решения жесткой системы из двух дифференциальных уравнений показан на рис. 7.29. Решение системы характерно наличием двух резко отличающихся стадий — крутой и пологой (или быстрой и медленной, если под независимой переменной подразумевать время).

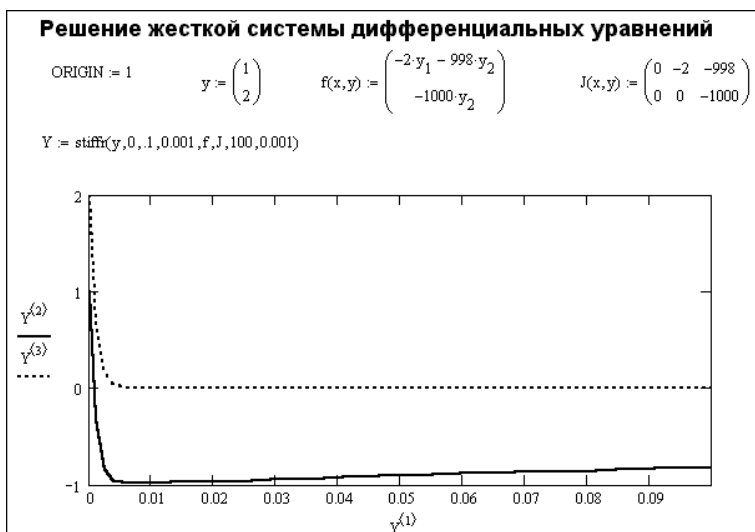


Рис. 7.29. Пример решения жесткой системы дифференциальных уравнений

В данном примере использована функция `stiffrr`, которая реализует метод Розенброка. В справочной системе Mathcad можно найти и другие примеры решения жестких систем дифференциальных уравнений.

Теперь рассмотрим типичную задачу химической кинетики — изменение во времени концентрации трех веществ при их смешении. Опуская подробное описание этих превращений, хорошо известное из курса химии, зададим систему дифференциальных уравнений в виде функции $F(t,y)$ и начальных условий y_0 (это начальные концентрации фракций смеси). Далее, составим матрицу Якоби и найдем решение с помощью функции `Stiffrr`, реализующей метод Розенброка (вы можете убедиться в том, что другая функция `Stiffb`, реализующая метод Булирша—Штера, дает тот же результат). Подготовленный в соответствии с этим описанием документ представлен на рис. 7.30.



Рис. 7.30. Пример решения жесткой системы дифференциальных уравнений химической кинетики

При построении графика решения решения используется выделение столбцов из матрицы решения D . При этом столбец $D^{(0)}$ дает вектор значений времени t , а столбцы $D^{(1)}$, $D^{(2)}$ и $D^{(3)}$ дают векторы значений переменных (концентрации компонентов смеси).

7.6.5. Новая функция `Radau` в Mathcad 2001i/11/12

Безусловно, необходимость вычисления матрицы Якоби (см. примеры на рис. 7.29 и 7.30) не всегда воспринимается с восторгом. В Mathcad 2001i/11/12 введена новая функция для решения жестких систем дифференциальных уравнений `Radau` ($y, x1, x2, n, F$), которая реализует новый метод RADAU5, предложенный в 1996 г. и не требующий подготовки матрицы Якоби. Относящиеся к этой функции параметры уже обсуждались.

Воспользовавшись примером на рис. 7.30, дополним его примером решения системы дифференциальных уравнений химической кинетики с помощью новой функции `Radau`. Пример этого представлен на рис. 7.31 (это конец документа, представленного на рис. 7.30).

К сожалению, с применением функции `Radau` не все обстоит благополучно. При расширении диапазона времени шаг решения может превысить допустимую

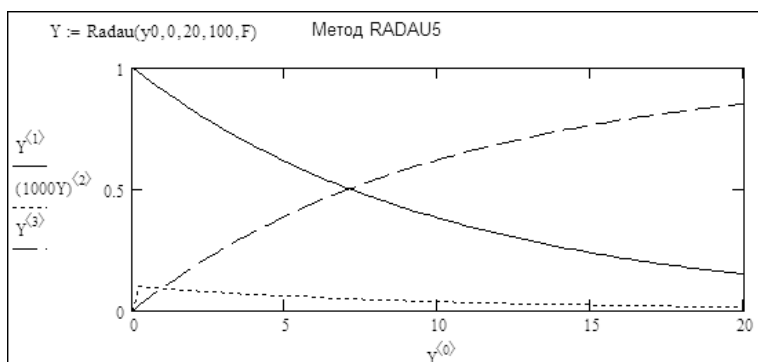


Рис. 7.31. Пример решения жесткой системы дифференциальных уравнений химической кинетики с помощью функции Radau

величину и решение не будет получено. К примеру, заменив конечное время $t = 20$ на $t = 50$ можно наблюдать такую ситуацию. При этом выражение с функцией окрашивается в красный цвет и появляется всплывающее сообщение «Can't converge a solution. Encountered too many integrator steps».

Еще один вариант этой функции `radau(y, x1, x2, acc, F, k, s)` служит для получения решения в конечной точке с заданной погрешностью `acc`. Параметры `k` и `s` задают максимальное число промежуточных точек, на которых ищется решение, и минимально допустимый интервал между ними).

7.6.6. Решение двухточечных краевых задач

Рассмотренные до сих пор методы численного решения дифференциальных уравнений исходили из задания начальных условий в одной точке — слева. Однако нередко надо найти такое решение дифференциального уравнения, при котором оно завершается также в заданной точке. Наглядным примером этого является решение задачи о попадании снаряда, выпущенного из пушки, точно в цель. При стрельбе полет снаряда, выпущенного из орудия с заданной скоростью, описывается системой дифференциальных уравнений. Неизвестным является угол, под которым надо выстрелить снаряд, чтобы он попал в цель. Задачи такого рода называются двухточечными краевыми задачами, а один из методов их решения именуется методом стрельбы или пристрелки (shooting method).

Mathcad позволяет решать задачи данного типа, у которых часть начальных условий задана в начальной точке интервала решения, а остальная часть в его конечной точке. Возможно также решение задач с граничными условиями в некоторой точке в середине интервала решения. Некоторые задачи имеют решения для определенных значений некоторого числового параметра — это задачи на собственные значения.

Для решения двухточечных краевых задач в Mathcad предназначены следующие функции:

- `bvalfit(v1, v2, x1, x2, xf, D, load1, load2, score)` — возвращает вектор недостающих начальных условий для краевой задачи, заданной в векторах `D`, `v1` и `v2` на интервале от `x1` до `x2`, где решение известно в некоторой промежуточной точке `xf`;
- `sbval(y, x1, x2, D, load, score)` — возвращает вектор недостающих `L` начальных условий на левой границе интервала решений для краевой задачи,

определенной в символьном векторе D , вектор y определяет начальные условия на интервале $[x_1, x_2]$, прочие параметры определены ниже.

В этих функциях векторы v, v_1, v_2 задают начальные условия, а x, x_1, x_2 — граничные точки интервала решений. $D(x, y)$ — функция, возвращающая N -компонентный вектор с первыми производными неизвестных функций. $load(x_1, v)$, $load1(x_1, v_1)$ и $load2(x_2, v_2)$ — векторзначные функции, возвращающие значения начальных условий в точках x_1 (x_2). $score(xf, y)$ — векторзначная функция, возвращающая n -элементный вектор соответствия. Он указывает, насколько значения решений, начинающихся из точек x_1 и x_2 , должны соответствовать xf . Например, если нужно совпадение решений, то $score(xf, y) := y$.

Число элементов векторов D и $load$ равно количеству уравнений N , а векторов z и результата выполнения функции $sbval$ — количеству правых граничных условий L . Соответственно, число левых граничных условий должно быть $(N-L)$.

Фрагмент документа Mathcad с двумя примерами решения краевой задачи показан на рис. 7.32. Оба примера достаточно очевидны и не нуждаются в особом описании, тем более что представленные графические зависимости хорошо проясняют суть решаемых задач.

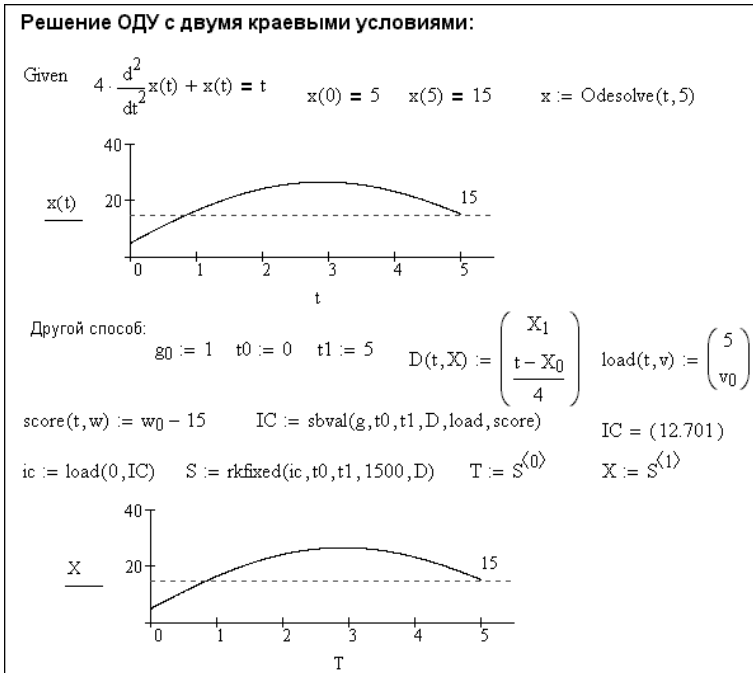


Рис. 7.32. Пример решения двухточечной краевой задачи

В приведенных примерах первые производные функций решения непрерывны. В этом случае можно применять функции $Odesolve$ и $sbval$. Если это не так, то можно применять функцию $bvalfit$. Ограничимся приведенным ниже примером:

$$F(x, y) := \begin{bmatrix} y_1 \\ (x < 0) \cdot y_0 + (x \geq 0) \cdot -y_0 \end{bmatrix}$$

7.6.8. Новые функции Mathcad 11/12 для решения ОДУ в частных производных

Иногда возникает необходимость в решении дифференциальных уравнений в частных производных. Классический пример решения такой задачи — вычисление колебаний струны, зажатой с двух сторон, или колебание мембраны, впрысванной в прямоугольное отверстие. Далеко не все системы компьютерной математики имеют средства для решения таких задач. Но в новых версиях Mathcad 11/12 такие средства впервые появились. Они реализованы двумя новыми функциями, предназначенными для решения гиперболических и параболических дифференциальных уравнений в частных производных.

Функция

```
Pdesolve(u, x, xrange, t, trange [,xpts] [, tpts])
```

возвращает функцию или вектор функций и t для решения систем дифференциальных уравнений как без ограничений, так и с ограничениями, заданными в виде алгебраических уравнений. В этой функции используются следующие параметры: u — вектор из имен заданных функций, x — пространственная переменная, $xrange$ — двухэлементный вектор, задающий граничные значения переменной x , t — время, $trange$ — двухэлементный вектор, задающий действительные граничные значения времени, $xpts$ и $tpts$ — целые числа, задающие дискретность изменения x и t .

Другая функция

```
numol(x_endpts, xpts, t_endpts, tpts, num_pde, num_pae, pde_func, pinit, bc_func)
```

возвращает матрицу решений размером $xpts$ на $tpts$ для одномерного дифференциального уравнения в частных производных. Здесь x_endpts и t_endpts — двухэлементные векторы-столбцы, определяющие область интегрирования, $xpts$ и $tpts$ — целые числа, определяющие число точек в области интегрирования, num_pde и num_pae — целые числа, определяющие число дифференциальных уравнений и алгебраических уравнений, pde_func — вектор, задающий функцию от переменных x , t , u , u_x и u_{xx} с длиной $(num_pde + num_pae)$, $pinit$ — вектор начальных условий с длиной $(num_pde + num_pae)$, а bc_func — матрица размера $num_pde \times 3$, строки которой задаются в форме:

- $(bc_left(t) \ bc_right(t) \ "D")$ — для граничных условий Дирихле;
- $(bc_left(t) \ bc_right(t) \ "N")$ — для граничных условий Неймана.

7.6.9. Анализ колебаний струны в одномерном случае

Рассмотрим реализацию классической задачи колебаний струны в плоскости (одномерный случай), если струна наглухо закреплена с обеих сторон. Волновое уравнение колебания струны и его решение с помощью функции `Pdesolve` представлено на рис. 7.34.

На этом рисунке показан также график решения данного уравнения при заданных параметрах.

7.6.10. Анализ колебаний поверхности

Несколько более сложной является задача о колебаниях упругой поверхности (мембраны), размещенной в отверстии квадратной формы массивной плиты. Ре-

шение этой задачи с помощью функции командой строки `numol` представлено на рис. 7.35.



Рис. 7.34. Решение задачи на колебание струны



Рис. 7.35. Решение задачи на колебание поверхности

Представляет интерес форма разреза колеблющейся поверхности. Для некоторых заданных параметров она представлена рисунками. Разрезы выделяются формированием субматрицы из общей матрицы решения. А на рис. 7.36 показано сравнение поверхности, построенной с помощью функции CreateMesh (она дает идеализированное решение), и поверхности, полученной решением с помощью функции numol.

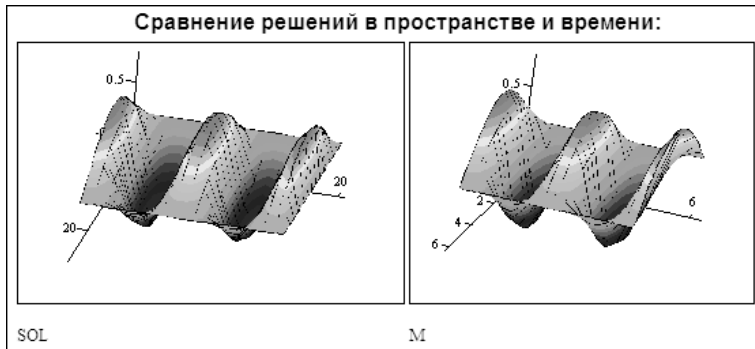


Рис. 7.36. Сравнение решений задачи на колебание поверхности

7.6.11. Анимация колебания поверхности

Наглядное представление о колебаниях поверхности дает анимация — представление об изменении во времени разреза поверхности. Для этого достаточно представить график разреза с выборкой его столбцов с помощью целочисленной переменной FRAME. На рис. 7.37 показана предварительная подготовка к созда-

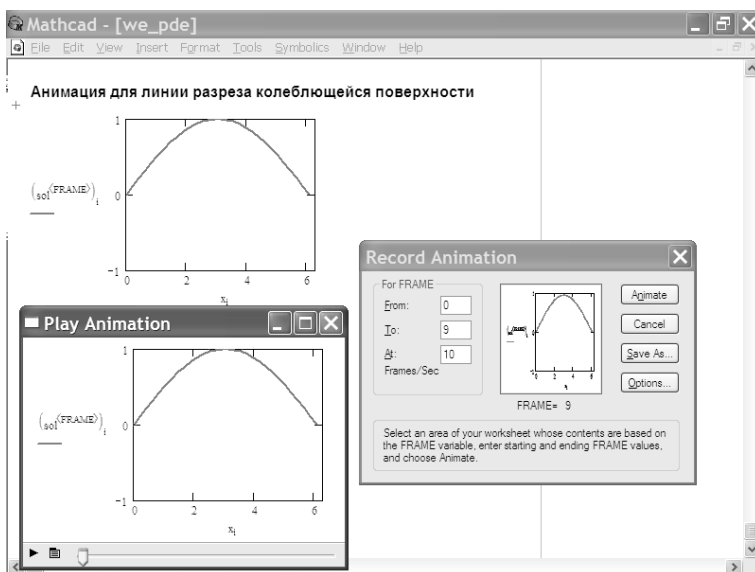


Рис. 7.37. Подготовка к анимации графика, описывающего колебание разреза поверхности

нию анимационного клипа — задан шаблон графика и выполнена команда Record... в подменю Animation позиции Tools меню Mathcad 11/12.

После выполнения указанных команд появляется окно Record Animation. В нем надо задать диапазон изменения переменной Frame и скорость ее изменения. Затем надо выделить график, анимацию которого мы решили наблюдать, и нажать кнопку Animation. В окне будут видны последовательно кадры анимации, после чего появится проигрыватель анимационных клипов — он показан на рис. 7.37 под графиком. В окне проигрывателя виден первый кадр анимации. Остальные кадры можно последовательно просмотреть, нажав кнопку пуска (черный треугольник) проигрывателя.

На рис. 7.38 показан некоторый промежуточный кадр анимации выделенного графика — с малым прогибом поверхности по сравнению с показанным в окне проигрывателя на рис. 7.37.

Наконец, на рис. 7.39 показан еще один кадр анимации с прогибом поверхности вниз. Заметим, что «вручную» наблюдать кадры анимации можно, просто перемещая ползунок проигрывателя слева направо.

Заинтересованный читатель может попытаться создать анимацию всей поверхности в целом. Это даст более красочную картину, но для детального анализа процесса колебаний наблюдать колебания точек поверхности ее разреза более удобно.

7.6.12. Решение дифференциальных уравнений с комплексными параметрами

Изредка может понадобиться решение дифференциальных уравнений с комплексными параметрами, включая воздействие, записанное в комплексной форме. Фрагмент документа Mathcad с примером решения такой задачи представлен на рис. 7.40.

Обратите внимание на то, что в данном случае для решения использован адаптивный метод Рунге—Кутты, реализованный функцией $Rkadapt$.

Множество примеров решения дифференциальных уравнений вы найдете в главе 12, посвященной математическому моделированию, а также в [11], в которой описан ряд новых функций для решения дифференциальных уравнений и систем с ними с помощью пакета расширения Numeric Recipes.

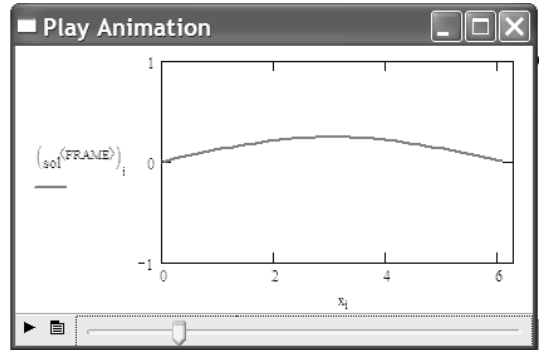


Рис. 7.38. Промежуточный кадр анимации с небольшим прогибом поверхности вверх

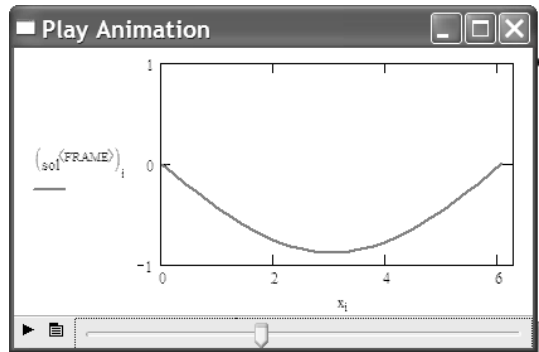


Рис. 7.39. Кадр анимации при прогибе поверхности вниз

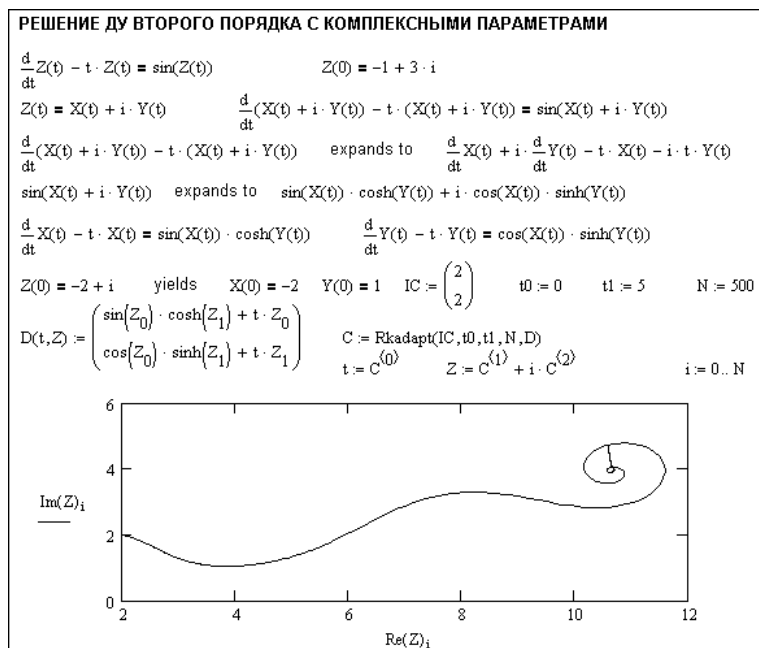


Рис. 7.40. Пример решения ДУ с комплексными параметрами

7.7. Реализация некоторых специальных численных методов

Mathcad имеет реализацию многих численных методов в виде описанных выше встроенных функций. Зачастую это довольно сложные и адаптивные методы, использующие по существу различные численные методы и выбирающие оптимальный метод решения в зависимости от условий решаемой задачи. Это не всегда удобно — нередко пользователь хочет точно знать, какими методами решается его задача. Это нужно также при изучении численных методов, когда понимание их сути стоит на первом месте, а эффективность метода на втором. В таких случаях можно отказаться от встроенных функций и перейти к реализации нужных численных методов напрямую. В этом разделе представлена достаточно большая подборка таких задач.

7.7.1. Вычисление ортогональных полиномов по рекуррентным формулам

Особый класс полиномов представляют ортогональные полиномы. Они описаны достаточно подробно в математической литературе [24, 25], так что мы не будем обсуждать их суть и особенности. Отметим только, что такие полиномы легко вычисляются по рекуррентным соотношениям. Расчет по ним представлен на рис. 7.41 и 7.42. Он хорошо иллюстрирует технику таких вычислений.

Наряду с вычислениями по рекуррентным формулам в документе рис. 7.41 и 7.42 даны примеры вычислений ортогональных полиномов по встроенным форму-

лам для системы Mathcad 2001i/11. Достоинством рекуррентных вычислений является возможность вычисления вектора всех значений полиномов разного порядка, вплоть до заданного n , с минимальным временем вычислений.

**ВЫЧИСЛЕНИЕ ОРТОГОНАЛЬНЫХ ПОЛИНОМОВ
ПО РЕКУРРЕНТНЫМ ФОРМУЛАМ**

1. Вычисление полинома Чебышева $T_n(x)$ по аналитической формуле

$$T(n, x) := \cos(n \cdot \arccos(x)) \quad T(3, 0.1) = -0.296 \quad T(10, 1) = 1$$

2. Вычисление полинома Чебышева $T_n(x)$ по рекуррентной формуле

$$x := 0.1 \quad n := 3 \quad T_0 := 1 \quad T_1 := x$$

$$i := 1 .. n - 1 \quad T_{i+1} := 2 \cdot x \cdot T_i - T_{i-1} \quad T = \begin{pmatrix} 1 \\ 0.1 \\ -0.98 \\ -0.296 \end{pmatrix}$$

$$T_{\text{cheb}}(n, x) = -0.296$$

3. Вычисление полинома Чебышева $U_n(x)$ по рекуррентной формуле

$$x := 0.1 \quad n := 3 \quad U_0 := 1 \quad U_1 := 2 \cdot x$$

$$i := 1 .. n - 1 \quad U_{i+1} := 2 \cdot x \cdot U_i - U_{i-1} \quad U = \begin{pmatrix} 1 \\ 0.2 \\ -0.96 \\ -0.392 \end{pmatrix}$$

$$U_{\text{cheb}}(n, x) = -0.392$$

Рис. 7.41. Начало документа с вычислениями ортогональных полиномов

4. Вычисление полинома Лежандра $P_n(x)$ по рекуррентной формуле

$$x := 2 \quad n := 4 \quad P_0 := 1 \quad P_1 := x$$

$$i := 1 .. n - 1 \quad P_{i+1} := \frac{(2 \cdot i + 1) \cdot x \cdot P_i - i \cdot P_{i-1}}{i + 1} \quad P = \begin{pmatrix} 1 \\ 2 \\ 5.5 \\ 17 \\ 55.375 \end{pmatrix}$$

$$\text{Leg}(n, x) = 55.375$$

5. Вычисление полинома Лагерра $L_n(x)$ по рекуррентной формуле

$$x := 3 \quad n := 4 \quad L_0 := 1 \quad L_1 := 1 - x$$

$$i := 1 .. n - 1 \quad L_{i+1} := \frac{(2 \cdot i + 1 - x) \cdot L_i - i \cdot L_{i-1}}{i + 1} \quad L = \begin{pmatrix} 1 \\ -2 \\ -0.5 \\ 1 \\ 1.375 \end{pmatrix}$$

$$\text{Lag}(n, x) = 1.375$$

6. Вычисление полинома Эрмита $H_n(x)$ по рекуррентной формуле

$$x := 3 \quad n := 4 \quad H_0 := 1 \quad H_1 := 2 \cdot x$$

$$i := 1 .. n - 1 \quad H_{i+1} := 2 \cdot x \cdot H_i - 2 \cdot i \cdot H_{i-1} \quad H = \begin{pmatrix} 1 \\ 6 \\ 34 \\ 180 \\ 876 \end{pmatrix}$$

$$\text{Her}(n, x) = 876$$

Рис. 7.42. Конец документа с вычислениями ортогональных полиномов

7.7.2. Численное дифференцирование

В математической литературе многие функции табулированы и представлены в таблицах. Примером может служить получивший большую известность справочник по специальным математическим функциям [25], в котором приведены многочисленные таблицы с достаточно точными значениями специальных математических функций.

Часто возникает необходимость в вычислении производной таблично заданной функции. Обычно эта задача решается применением полиномиальной аппроксимации (или интерполяции) функции с последующим вычислением производной полученного полинома. Так получают формулы численного дифференцирования, приведенные в упомянутом справочнике по специальным функциям. Такие формулы есть для разного числа узлов и разного выбора центрального узла. Обычно формулы записываются в виде, обеспечивающем интерполяцию, т. е. нахождение производной не только для узловых точек, но и в середине между ними.

На рис. 7.43 показано начало документа, в котором представлены формулы для численного дифференцирования с интерполяцией по трем и четырем точкам. Формулы представлены в виде функций пользователя $y'(p)$, где p — параметр интерполяции, характеризующий относительное отклонение значения независимой переменной x от центрального узла.

На другом рисунке (рис. 7.44) представлены формулы численного дифференцирования с интерполяцией для пяти точек. В конце документа представлена функция, чьи узловые точки были использованы для контрольных примеров на численное дифференцирование.

Нетрудно заметить, что число верных цифр результата на единицу больше числа узловых точек, для которых вычисляется приближенное значение производной. Такая точность обычно вполне удовлетворяет потребностям практики.

Быстрое дифференцирование аналитических функций

По формулам 3 ординат:

$y1s := 0.3555325$	$y0 := 0.3520653$	$y1h := 0.3484925$
$h := 0.02$	$x0 := 0.5$	$x := 0.52$

$$p := \frac{x - x0}{h}$$

$$y'(p) := \frac{(p - 0.5) \cdot y1s - 2 \cdot p \cdot y0 + (p + 0.5) \cdot y1h}{h}$$

$p = 1$ $y'(p) = -0.18128$ 4 верных цифры

По формулам 4 ординат:

$y1s := 0.3555325$	$y0 := 0.3520653$	$y1h := 0.3484925$	$y2h := 0.344818$
--------------------	-------------------	--------------------	-------------------

$$y1'(p) := -(3 \cdot p^2 - 6 \cdot p + 2) \cdot \frac{y1s}{6} + (3 \cdot p^2 - 4 \cdot p - 1) \cdot \frac{y0}{2}$$

$$y'(x) := \frac{y1'(x) - (3 \cdot p^2 - 2 \cdot p - 2) \cdot \frac{y1h}{2} + (3 \cdot p^2 - 1) \cdot \frac{y2h}{6}}{h}$$

$y'(p) = -0.181215$ 5 верных цифр

Рис. 7.43. Численное дифференцирование с интерполяцией по 3 и 4 точкам

По формулам 5 ординат:

$$y_{2s} := 0.35889029 \quad y_{1s} := 0.35553253 \quad y_0 := 0.35206533$$

$$y_{1h} := 0.34849251 \quad y_{2h} := 0.344818$$

$$y_1'(p) := (2 \cdot p^3 - 3 \cdot p^2 - p + 1) \cdot \frac{y_{2s}}{12} - (4 \cdot p^3 - 3 \cdot p^2 - 8 \cdot p + 4) \cdot \frac{y_{1s}}{6}$$

$$y_2'(p) := y_1'(p) + (2 \cdot p^3 - 5 \cdot p) \cdot \frac{y_0}{2} - (4 \cdot p^3 + 3 \cdot p^2 - 8 \cdot p - 4) \cdot \frac{y_{1h}}{6}$$

$$y'(p) := \frac{y_2'(p) + (2 \cdot p^3 + 3 \cdot p^2 - p - 1) \cdot \frac{y_{2h}}{12}}{h}$$

$$y'(p) = -0.1812164583 \quad \text{6 верных цифр}$$

Пример дан для функции:

$$y(x) := \frac{\exp\left(\frac{-x^2}{2}\right)}{2.506628275}$$

$$\text{TOL} := 1 \cdot 10^{-10} \quad \frac{d}{dx} y(x) = -0.1812161066$$

Рис. 7.44. Численное дифференцирование с интерполяцией по 5 точкам

7.7.3. Численное вычисление трех производных аналитической функции

Описанный выше подход может использоваться для вычисления и производных более высокого порядка. Это иллюстрирует рис. 7.45, на котором представлено задание функций пользователя, вычисляющих три производные аналитически заданной функции. На этот раз ради упрощения формул интерполяция не предусмотрена. Зато предусмотрено изменение шага h , задающего сдвиг между узловыми точками.

В конце рис. 7.45 представлены результаты контрольного вычисления значения производной как по представленным формулам, так и с помощью встроенного оператора дифференцирования систем Mathcad. Нетрудно заметить, что погрешность вычислений заметно ухудшается по мере увеличения порядка вычисляемой производной.

Нужны ли вычисления по приближенным формулам дифференцирования, коль система Mathcad имеет встроенный оператор дифференцирования? К тому же реализующий адаптивный метод вычисления производных, гарантирующий минимально возможную погрешность вычислений. Ответ на этот вопрос не вполне однозначен. Во-первых, документ рис. 7.45 для того и создан, чтобы продемонстрировать возможности формул численного дифференцирования, которые вполне применимы и для вычислений в иных системах — в том числе в среде обычных языков программирования. Во-вторых, в отличие от адаптивного метода численного дифференцирования в системе Mathcad эти формулы позволяют значительно уменьшить время вычислений, например в случае построения графиков производных заданной функции — рис. 7.46.

Вычисление трех производных функции численным методом

Дифференцируемая функция: $y(x) := \frac{\exp\left(\frac{-x^2}{2}\right)}{2.506628275}$

Формулы дифференцирования для 5 ординат:

$$y'(x, h) := \frac{-y(x + 2 \cdot h) + 8 \cdot y(x + h) - 8 \cdot y(x - h) + y(x - 2 \cdot h)}{12 \cdot h}$$

$$y''(x, h) := \frac{-y(x + 2 \cdot h) + 16 \cdot y(x + h) - 30 \cdot y(x) + 16 \cdot y(x - h) - y(x - 2 \cdot h)}{12 \cdot h^2}$$

$$y'''(x, h) := \frac{y(x + 2 \cdot h) - 2 \cdot y(x + h) + 2 \cdot y(x - h) - y(x - 2 \cdot h)}{2 \cdot h^3}$$

Примеры вычислений и их проверка: $x := 0.5$

$y'(0.5, 0.01) = -0.1760326626$	$\frac{d}{dx}y(x) = -0.1760326634$
$y''(0.5, 0.01) = -0.2640489949$	$\frac{d}{dx}\frac{d}{dx}y(x) = -0.264048995$
$y'''(0.5, 0.01) = 0.4840345425$	$\frac{d}{dx}\frac{d}{dx}\frac{d}{dx}y(x) = 0.4840898243$

Рис. 7.45. Начало документа с вычислением трех производных по формулам численного дифференцирования

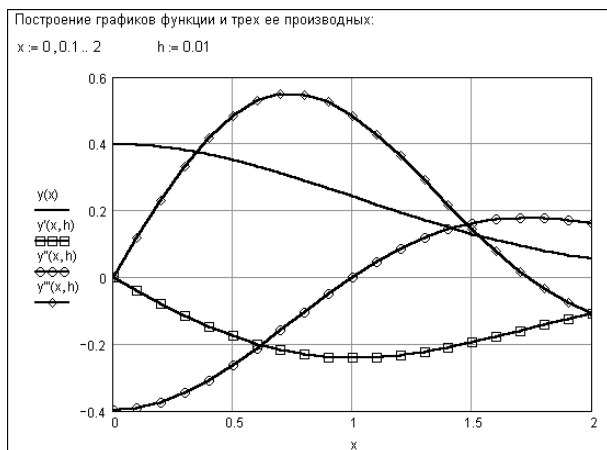


Рис. 7.46. Конец документа с вычислением трех производных по формулам численного дифференцирования

Если речь идет о численном дифференцировании в среде Mathcad, то преимущество стоит отдать все же применению встроенного оператора дифференцирования. Более того, применение оптимизации вычислений позволяет резко повысить скорость вычислений производных и получить аналитические выражения для них. Так что приведенные примеры имеют лишь познавательную и учебную ценность. Они, безусловно, полезны при постановке работ по численным методам.

7.7.4. Численное дифференцирование зашумленной функции

При наличии погрешности в задании точек функции или шумов численное дифференцирование становится неустойчивым процессом. И даже при использовании аппарата сглаживания функции трудно надеяться на хорошие результаты. Поясним это документом, начало которого представлено на рис. 7.47. Здесь показана аналитическая функция, которая представлена точками с компонентами шума, имеющего нормальное распределение.

```

Сглаживание данных с численным дифференцированием
Зададим исходную зависимость:  $f(x) := \cos(4x^2)^3$ 
Сформируем массив исходных данных:  $\text{imax} := 100$   $i := 0, 1.. \text{imax}$   $X_i := \frac{i}{100}$   $Y0_i := f(X_i)$ 
Введем в данные шум:  $\text{eps} := 0.15$   $\text{znak}_i := \text{if}(\text{md}(1) < 0.5, 1, -1)$   $Y_i := Y0_i + \text{znak}_i \cdot \text{md}(\text{eps})$ 
Создадим массив сглаженных данных по методу линейного сглаживания по 5 точкам:


$$Y_{s_i} := \begin{cases} \frac{3 \cdot Y_0 + 2 \cdot Y_1 + Y_2 - Y_3}{5} & \text{if } i = 0 \\ \frac{4 \cdot Y_0 + 3 \cdot Y_1 + 2 \cdot Y_2 + Y_3}{10} & \text{if } i = 1 \\ \frac{Y_{\text{imax}-3} + 2 \cdot Y_{\text{imax}-2} + 3 \cdot Y_{\text{imax}-1} + 4 \cdot Y_{\text{imax}}}{10} & \text{if } i = \text{imax} - 1 \\ \frac{-Y_{\text{imax}-3} + Y_{\text{imax}-2} + 2 \cdot Y_{\text{imax}-1} + 3 \cdot Y_{\text{imax}}}{5} & \text{if } i = \text{imax} \\ \frac{Y_{i-2} + Y_{i-1} + Y_i + Y_{i+1} + Y_{i+2}}{5} & \text{otherwise} \end{cases}$$

 $x := 0, 0.02.. 1$ 
    
```

Рис. 7.47. Начало документа, иллюстрирующего численное дифференцирование зашумленной функции

Снизу рис. 7.47 показан простой программный модуль, реализующий линейное сглаживание по пяти точкам. Рисунок 7.48 показывает график точек зашумленной функции, график самой функции без шумов (тонкая линия) и график сглаженной функции (жирная линия). Эффективность сглаживания назвать высокой трудно.

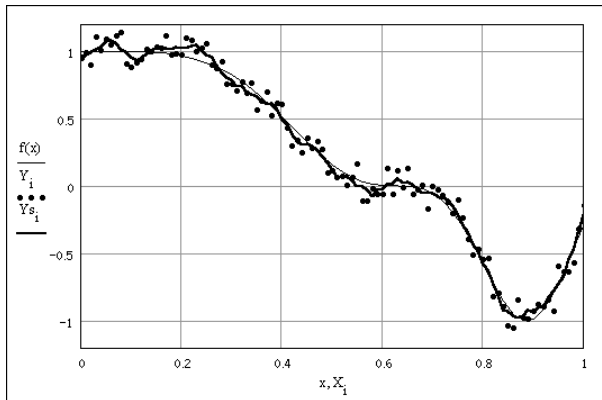


Рис. 7.48. График точек зашумленной функции, самой функции без шумов (тонкая линия) и сглаженной функции (жирная линия)

Рисунок 7.49 показывает результаты дифференцирования функции без шумов, численного дифференцирования массива точек зашумленной функции и численного дифференцирования сглаженного массива точек.

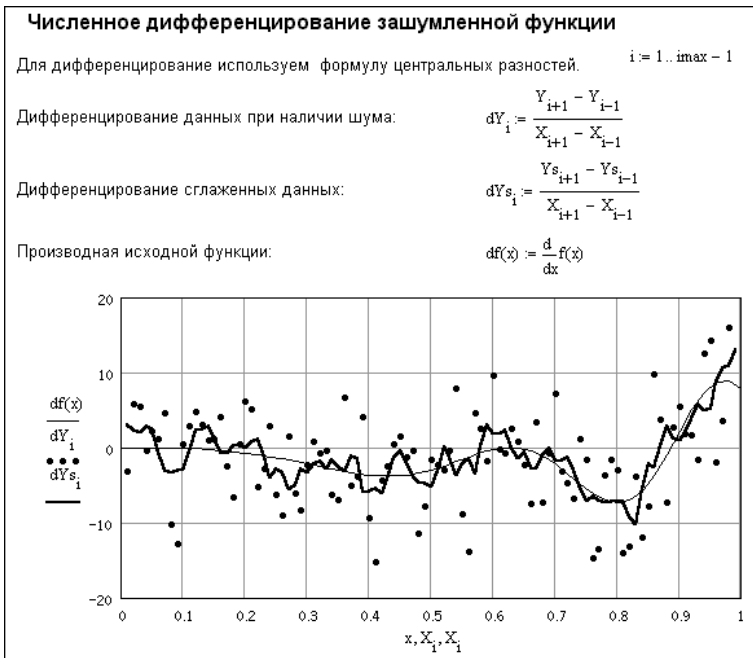


Рис. 7.49. Результаты численного дифференцирования

Безусловно, сглаживание заметно уменьшает погрешность численного дифференцирования зашумленной функции, но результат такого дифференцирования остается не слишком удовлетворительный. Численное дифференцирование приемлемо в случае дифференцирования функций без больших погрешностей в задании отсчетов и при отсутствии шумов.

7.7.5. Интегрирование таблично заданных функций

Часто возникает необходимость в вычислении определенного интеграла для таблично заданной функции. Тогда прямое применение встроенного в систему оператора вычисления интеграла оказывается невозможным, так как он предполагает задание подынтегральной функции в аналитическом виде.

Документ рис. 7.50 иллюстрирует ряд способов вычисления определенного интеграла при табличном задании подынтегральной функции. Первые два способа это хорошо известные методы трапеций и Симпсона. При этом таблично заданная функция интерполируется линейной зависимостью или набором сплайновых функций (полиномов третьей степени).

Далее даны еще три способа численного вычисления определенного интеграла — с применением линейной и сплайновой интерполяции функции, а также с использованием встроенного оператора численного интегрирования. В качестве исходных данных взяты ординаты квадратичной параболы, что позволяет вычислить интеграл без этих ухищрений. Такое вычисление представлено для контроля

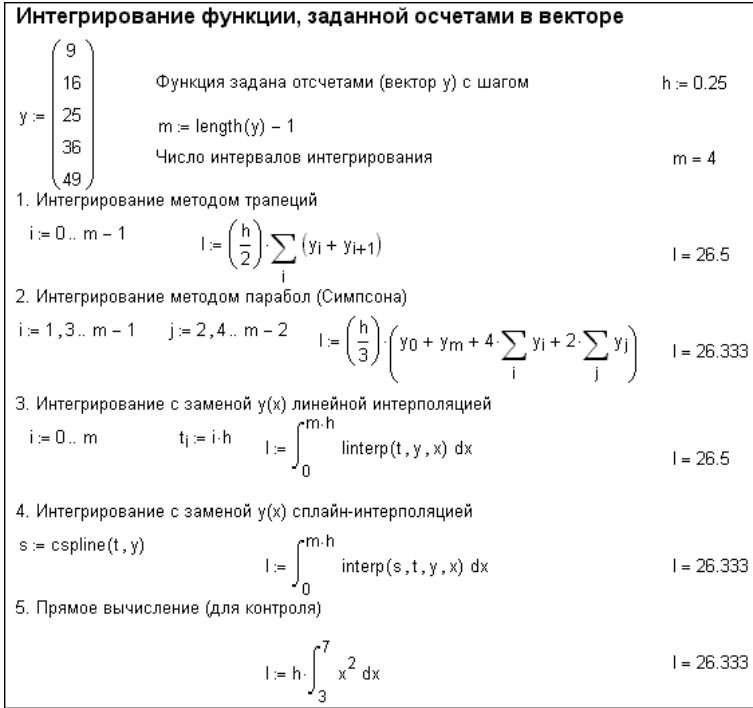


Рис. 7.50. Интегрирование табличных данных (начало документа)

в конце документа. Нетрудно заметить, что лишь интегрирование методом Симпсона и интегрирование со сплайн-интерполяцией дают полное совпадение с прямым интегрированием (не стоит забывать, что установленный формат цифровых данных выводит результат только с шестью значащими цифрами после десятичной точки).

7.7.6. Вычисление площади под таблично заданной кривой

Одним из основных применений численных методов интегрирования является вычисление площади под кривой, функция которой задана в виде таблицы или матрицы с двумя столбцами — один для координат x , а другой для ординат y . Если узлы функции заданы неравномерно, то, пожалуй, самым приемлемым будет способ решения данной задачи, основанный на интерполяции функции сплайнами. Это и реализует документ, показанный на рис. 7.51.

Надо иметь в виду, что точки кривой должны быть упорядочены, т. е. значения координат x в векторе координат должны идти в возрастающем порядке.

7.7.7. Быстрое вычисление интегралов по формулам Уэддля и Ньютона—Котесса

Некоторые формулы численного интегрирования, например 6-точечные формулы Уэддля и Ньютона—Котесса, обеспечивают настолько высокую точность, что зачастую могут применяться без разбивки области интегрирования на отдельные участки. Рисунок 7.52 демонстрирует эту возможность на практике. Это обеспечивает наиболее высокую скорость вычислений.



Рис. 7.51. Вычисление площади под кривой, заданной точками



Рис. 7.52. Быстрое интегрирование на основе формул Уэддлля и Ньютона—Котесса

Нетрудно заметить, что в данном примере получено 6 верных знаков результата после десятичной точки.

7.7.8. Интерполяция по частным формулам Лагранжа

Многие таблицы, например всемирно известные таблицы специальных функций Адамовича и Стигана [26], специально рассчитаны на интерполяцию по *частным формулам Лагранжа*, позволяющим выполнять интерполяцию по ряду точек, начиная с двух (линейная интерполяция). Документ, представленный на рис. 14.32, содержит эти формулы, повторяющие приведенные в [32], и позволяет вычислять значения табличных данных при постоянном шаге интерполяции h и задании параметра $p = (x - x_c)/h$, где x — значение x для вычисления $y(x)$ и x_c — координата центрального узла.

Интерполяция по частным формулам Лагранжа

$$f1(p, y) := (1 - p) \cdot y_0 + p \cdot y_1 \quad f2(p, y) := \left(p \cdot \frac{p-1}{2}\right) \cdot y_0 + (1 - p^2) \cdot y_1 + \left(p \cdot \frac{p+1}{2}\right) \cdot y_2$$

$$ff3(p, y) := \left[p \cdot (p-1) \cdot \frac{p-2}{6}\right] \cdot y_0 + \left[\left(p^2 - 1\right) \cdot \frac{p-2}{2}\right] \cdot y_1$$

$$f3(p, y) := ff3(p, y) + \left[p \cdot (p+1) \cdot \frac{p-2}{2}\right] \cdot y_2 + \left[p \cdot \frac{p^2-1}{2}\right] \cdot y_3$$

$$ff4(p, y) := \left[\left(p^2 - 1\right) \cdot p \cdot \frac{p-2}{24}\right] \cdot y_0 - \left[(p-1) \cdot p \cdot \frac{p^2-4}{6}\right] \cdot y_1$$

$$fff4(p, y) := ff4(p, y) + \left[\left(p^2 - 1\right) \cdot \frac{p^2-4}{4}\right] \cdot y_2 - \left[(p+1) \cdot p \cdot \frac{p^2-4}{6}\right] \cdot y_3$$

$$f4(p, y) := fff4(p, y) + \left[\left(p^2 - 1\right) \cdot p \cdot \frac{p+2}{24}\right] \cdot y_4$$

$$ff5(p, y) := -\left[p \cdot \left(p^2 - 1\right) \cdot (p-2) \cdot \frac{p-3}{120}\right] \cdot y_0 + \left[p \cdot (p-1) \cdot \left(p^2 - 4\right) \cdot \frac{p-3}{24}\right] \cdot y_1$$

$$fff5(p, y) := ff5(p, y) - \left[\left(p^2 - 1\right) \cdot \left(p^2 - 4\right) \cdot \frac{p-3}{12}\right] \cdot y_2$$

$$ffff5(p, y) := fff5(p, y) + \left[p \cdot (p+1) \cdot \left(p^2 - 4\right) \cdot \frac{p-3}{12}\right] \cdot y_3$$

$$fffff5(p, y) := fffff5(p, y) - p \cdot \left(p^2 - 1\right) \cdot (p+2) \cdot \frac{p-3}{24} \cdot y_4$$

$$f5(p, y) := fffff5(p, y) + \left[p \cdot \left(p^2 - 1\right) \cdot \frac{p^2-4}{120}\right] \cdot y_5$$

Рис. 7.53. Задание функции для вычислений по частным формулам Лагранжа

На рис. 7.54 показан конец документа, начало которого дано на рис. 7.53. В нем даны примеры применения заданных формул и таблица с указанием центрального узла x_c .

Несмотря на громоздкость формул, вычисления по ним выполняются быстрее, чем по другим формулам, например, по общей формуле Лагранжа. Это связано с тем, что частные формулы гораздо проще развернутой общей формулы интерполяции Лагранжа и содержат меньше вычислительных операций. Данное свойство может сделать их полезными в некоторых специальных случаях.

7.7.9. Поиск глобального максимума многоэкстремальной функции

Иногда необходимо отыскать глобальный максимум (или минимум) многоэкстремальной функции. Ограничимся рассмотрением решения этой задачи для функции одной переменной. Рисунок 7.55 показывает примеры такого решения.

Примеры интерполяции, $p = (x-x_c)/h$, где h - шаг расположения узлов и x_c - отклонение от центральной точки x_c :

$y := \begin{pmatrix} 1 \\ 4 \end{pmatrix}$	$f1(0, y) = 1$ $f1(0.5, y) = 2.5$	$y := \begin{pmatrix} 1 \\ 4 \\ 9 \end{pmatrix}$	$f2(0, y) = 4$ $f2(0.5, y) = 6.25$										
$y := \begin{pmatrix} 1 \\ 4 \\ 9 \\ 16 \end{pmatrix}$	$f3(0, y) = 4$ $f3(1.5, y) = 5.25$	$y := \begin{pmatrix} 1 \\ 4 \\ 9 \\ 16 \\ 25 \end{pmatrix}$	$f4(0, y) = 9$ $f4(1.5, y) = 20.25$										
$y := \begin{pmatrix} 1 \\ 4 \\ 9 \\ 16 \\ 25 \\ 36 \end{pmatrix}$	$f5(0, y) = 9$ $f5(2.5, y) = 30.25$	Центральный узел для <table border="0"> <tr><td>$f1$</td><td>$f2$</td><td>$f3$</td><td>$f3$</td><td>$f5$</td></tr> <tr><td>1</td><td>2</td><td>2</td><td>3</td><td>3</td></tr> </table>		$f1$	$f2$	$f3$	$f3$	$f5$	1	2	2	3	3
$f1$	$f2$	$f3$	$f3$	$f5$									
1	2	2	3	3									

Рис. 7.54. Конец документа для вычислений по частным формулам Лагранжа

Поиск глобального максимума функции

$y(x) := x \cdot e^{-x} \cdot \sin(5 \cdot x)$ Задана многоэкстремальная функция $y(x)$ $x := 0, 0.1 \dots 5$



График функции показывает, что при $0 < x < 5$ функция имеет 8 экстремумов, из них 4 минимума и 4 - максимума. Видно, что глобальный максимум лежит в области, где примерно $x=1.6$. Его и надо найти автоматически (т.е. без подсказки примерного значения x).

$x := .9$ Given $\frac{d}{dx} y(x) = 0$ $x > 0$ Этот пример показывает, что при неточном задании начального $x=0.9$ функция `minerr` обнаруживает локальный минимум, а не глобальный максимум.

Minerr(x) = 0.945

Первый способ (Монте-Карло) $i := 0 \dots 10000$ $Y_i := y(\text{rnd}(2.5))$ $\max(Y) = 0.327$

Создав вектор из 101 значения $y(x)$ найдем в нем максимальное значение:

$i := 0 \dots 100$ $x_i := \frac{i}{20}$ $y_i := y(x_i)$ $y_m := \max(y_i)$ $y_m = 0.327$

Теперь найдем x_m при котором получено это значение:

$x := 0$ Given $y(x) = y_m$ $x_m := \text{Find}(x)$ $x_m = 1.55$

Второй способ поиска глобального максимума:

$x := x_m$ Given $\frac{d}{dx} y(x) = 0$ Minerr(x) = 1.557 $y(1.577) = 0.326$

Третий способ способ поиска глобального максимума:

Given $y(x) = 10$ Minerr(x) = 1.557

Рис. 7.55. Примеры поиска глобального максимума многоэкстремальной функции

Вначале на нем дан пример того, как функция `Minerr` не справляется с этой задачей и отыскивает локальный минимум вместо глобального максимума.

Одним из способов поиска глобального максимума является применение метода Монте—Карло — генерация множества случайных значений x и выбор из них просто максимального значения. Этот первый способ, представленный на

рис. 7.55, имеет слишком много недостатков, чтобы его можно было всерьез рекомендовать на практике. Во-первых, надо генерировать слишком много случайных чисел, чтобы найти глобальный максимум с достаточной точностью. Во-вторых, он ведет к большим затратам памяти для хранения массива значений функции. В-третьих, он не позволяет явно найти значение x , при котором $y(x)$ максимальна (впрочем, сделать это достаточно легко — см. прием, описанный ниже).

Более рациональным является просто просчет некоторого множества значений $y(x)$ при определенном числе равноотстоящих значений x . Найдя значение x для приближенного максимума, можно уточнить x , решая уравнение $y(x) = ym$ или $y(x) - ym = 0$, и затем уточненное значение x использовать в качестве начального приближения для поиска максимума.

Наконец, третий способ (самый простой) заключается в свойстве функции Minerr отыскивать ближайшее решение. Применение функции Minerr для поиска экстремумов не документируемый, а потому довольно оригинальный прием. Дело в том, что эта функция в блоке решения Given (он был описан выше) старается дать приближение к решению задачи с наименьшей среднеквадратической погрешностью. Другими словами, блок решения с этой функцией может решать некорректные задачи. Если вблизи некоторого значения x_0 есть максимум функции $f(x_{max}) = max$, то его можно найти, попытавшись решить некорректное уравнение $f(x_{max}) = c > max$. Аналогично для нахождения минимума можно найти $f(x_{min}) = c < min$.

Задав уравнение $y(x) = y0$, где заведомо $y0 > ym$, можно найти глобальный максимум, ибо только при нем реализуется минимальная среднеквадратическая ошибка.

7.7.10. Поиск локальных экстремумов в массиве

Если некоторая зависимость задана массивом своих точек (x_i, y_i) , то на основании сравнения их значений нетрудно выявить точки локальных максимумов. Для этого достаточно «просмотреть» массив и всякий раз, когда очередное значение точки массива становится меньше предыдущего значения, зафиксировать предыдущее значение как максимум. Это делает программный блок, представленный на рис. 7.56. Реализованная в нем функция MaxM возвращает матрицу из двух столбцов. Первый содержит координаты точек максимумов, а второй — их ординаты. Для поиска минимумов достаточно поставить знак «-» у массива точек и у результата, выдаваемого функцией MaxM. Это и делает функция MinM для поиска локальных минимумов в массиве M.

Примеры даны для массива Data. Построенный снизу рис. 7.56 график позволяет убедиться в том, что найденные значения локальных максимумов и минимумов и впрямь существуют.

Если разбор этого программного модуля вызывает у вас трудности, стоит познакомиться с основами программирования в системе Mathcad, описанными в главе 10.

7.7.11. Анализ сложной функции

График сложной функции нередко обнаруживает различные особенности ее применения, например экстремумы (максимумы и минимумы), а также нули функции, на графике отображаемые как точки пересечения графика функции $F(x)$ с осью абсцисс. Рисунок 7.57 показывает начало документа с анализом нулей функ-

ции и нахождением экстремумов этой функции. Нетрудно заметить, что представленная на нем функция имеет 5 нулей и 4 экстремума.

Для поиска нулей функции достаточно последовательно применить функцию root, указав приблизительные значения x в точках, где $F(x) = 0$. Конец документа

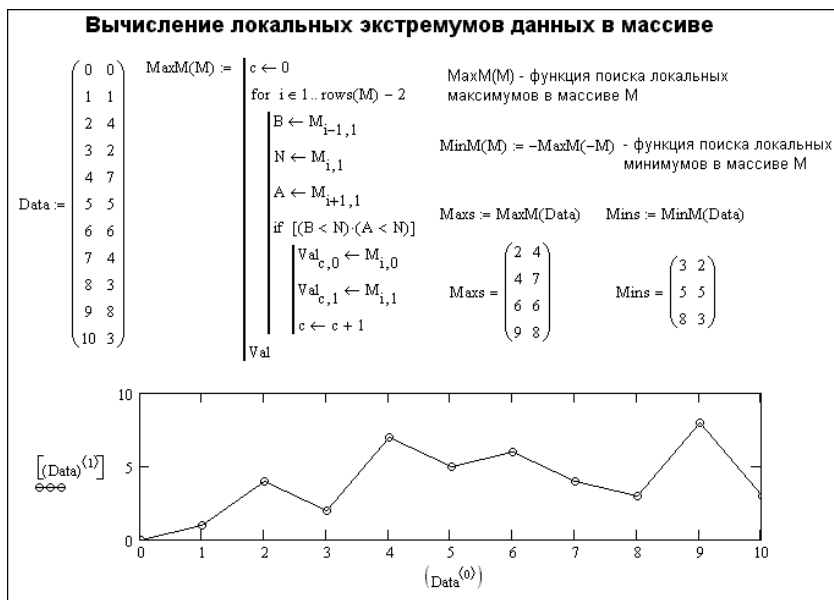


Рис. 7.56. Поиск локальных максимумов в массиве

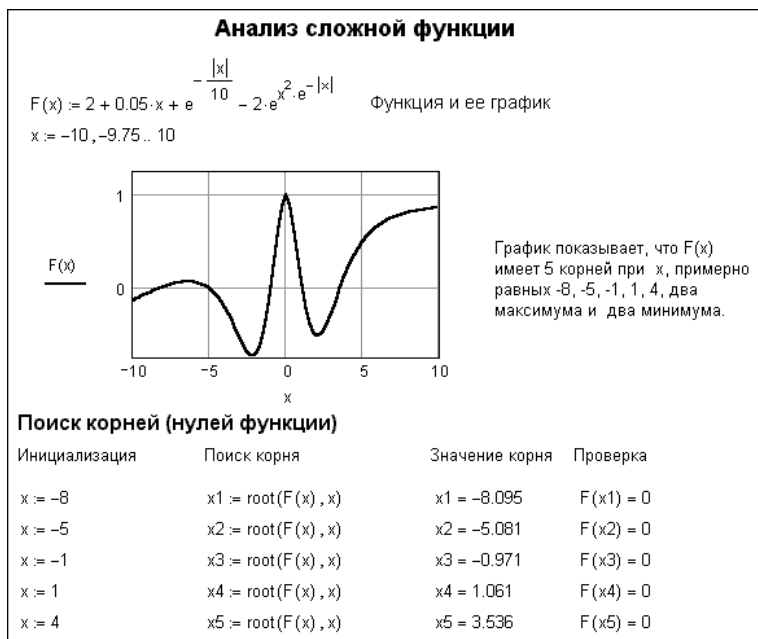


Рис. 7.57. Начало документа Mathcad с анализом сложной функции $F(x)$

Поиск экстремумов (максимумов и минимумов функции)				
Вычисление главного максимума $F(x)$		Given		
$F(x) = 2$	$x = 0$	$x6 := \text{Minerr}(x)$	$x6 = 0$	$F(x6) = 1$
Контроль	$F(x6 + 0.1) = 0.977$	$F(x6 - 0.1) = 0.967$		
Вычисление другого максимума $F(x)$		Given		
$F(x) = 1$	$x < -5.5$	$x7 := \text{Minerr}(x)$	$x7 = -6.359$	$F(x7) = 0.066$
Контроль	$F(x7 + 0.2) = 0.065$	$F(x7 - 0.2) = 0.065$		
Вычисление первого минимума $F(x)$		Given		
$F(x) = -0.8$	$x < -1$	$x8 := \text{Minerr}(x)$	$x8 = -2.152$	$F(x8) = -0.728$
Контроль:	$F(x8 + 0.1) = -0.724$	$F(x8 - 0.1) = -0.724$		
Вычисление второго минимума $F(x)$		Given		
$F(x) = -0.6$	$x > 1$	$x9 := \text{Minerr}(x)$	$x9 = 2.035$	$F(x9) = -0.518$
Контроль:	$F(x9 + 0.1) = -0.514$	$F(x9 - 0.1) = -0.514$		

Рис. 7.58. Конец документа Mathcad с анализом сложной функции $F(x)$

(рис. 7.58) показывает поиск экстремумов с помощью функции `Minerr`. Здесь также надо аккуратно подсказать системе Mathcad условия применения этой функции. Особенно это касается пологих максимумов и минимумов, где алгоритм работы функции `Minerr` может давать сбои.

Разумеется, в новых версиях Mathcad можно использовать встроенные функции для поиска минимумов и максимумов функций вида $f(x)$ — `maximize` и `minimize`. Такое применение, в принципе, не дает ничего нового.

7.7.12. Расчет и построение асимптот функции $f(x)$

Многие функции при устремлении x в бесконечность (положительную или отрицательную) обнаруживают свойство постепенного приближения к некоторой прямой $k \cdot x + b$, называемой асимптотой. Рисунок 7.59 показывает определение основной наклонной асимптоты функции $f(x)$ и ее построение на графике функции. Для определения k и b приходится вычислять пределы функции.

В местах разрыва функции часто существуют вертикальные асимптоты. Условия их существования также представлены на рис. 7.59. Для данной функции вертикальная асимптота соответствует значению $x = -2$. Обратите внимание на способ построения этой асимптоты — по оси ординат указывается x , а по оси абсцисс -2 . График рис. 7.59 дает наглядное представление об асимптотах функции $f(x)$.

7.8. Численное решение ОДУ по классическим формулам

Со стороны некоторых преподавателей и даже ученых иногда раздается критика в адрес системы Mathcad. Говорят, что эта система отучает учащихся мыслить математическими категориями и делает непрозрачными вычисления. Такого рода

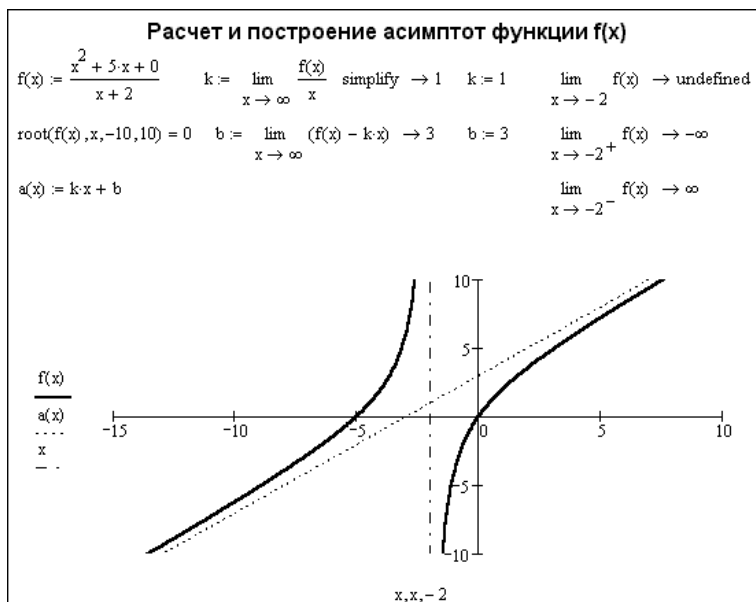


Рис. 7.59. Построение графика функции с ее асимптотами

критика исходит, как правило, от людей, просто не удосужившихся поработать с системой Mathcad. Она напоминает анекдот о даме, которая, впервые увидев поезд, заявила, что «этого быть не может хотя бы потому, что этого не может быть». Приведенные выше многочисленные примеры показывают, что на самом деле Mathcad лишь помогает учащимся познавать премудрости математики. И чтобы эта система приносила максимальную пользу в познании математических понятий, надо просто правильно методически оформлять задания на вычисления с помощью системы Mathcad. А это уже дело педагога. В этом разделе мы завершаем рассмотрение приемов реализации классических методов вычислений с помощью системы Mathcad на примерах реализации численных методов решения дифференциальных уравнений по классическим формулам.

7.8.1. Решение ДУ модифицированным методом Эйлера

Простейшим методом решения одиночного дифференциального уравнения вида $y' = f(x, y)$ является простой метод Эйлера. Он реализуется следующей рекуррентной формулой:

$$Y_i = Y_{i-1} + h \cdot f(x_{i-1}, Y_{i-1}) .$$

Здесь h — шаг решения. Погрешность этого метода значительна (порядка h), поэтому он на практике почти не применяется.

В документе рис. 7.60 дана реализация так называемого модифицированного метода Эйлера, погрешность которого близка к h^2 (т. е. порядка 1% при $h = 0.1$), что нередко уже приемлемо для приближенного решения дифференциальных уравнений первого порядка.

Улучшение точности вычислений при использовании этого метода фактически достигнуто за счет интегрирования методом трапеций вместо метода прямоуголь-

Решение дифференциального уравнения первого порядка $y'=f(x,y)$ модифицированным методом Эйлера

$h = 0.1$ $N := 10$ Шаг h и число точек N решения

$f(x, y) := x \cdot y$ Функция $f(x, y)$

$\begin{pmatrix} x_0 \\ y_0 \end{pmatrix} := \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ Начальные значения x и y

$i := 1 .. N$ Начало цикла вычислений

$\begin{pmatrix} x_i \\ y_i \end{pmatrix} := \begin{pmatrix} x_{i-1} + h \\ y_{i-1} + h \cdot f\left(x_{i-1} + \frac{h}{2}, y_{i-1} + h \cdot \frac{f(x_{i-1}, y_{i-1})}{2}\right) \end{pmatrix}$ Рекуррентные формулы модифицированного метода Эйлера в векторной форме

$y_{ij} = \exp\left[\frac{(x_i)^2}{2}\right]$ Точное решение (для сравнения)

Рис. 7.60. Решение одиночного ДУ модифицированным методом Эйлера (начало документа)

ников, характерного для реализации простого метода Эйлера. В конце рис. 7.60 показано точное решение ДУ, взятого для этого примера.

На рис. 7.61 представлена вторая часть данного документа, на которой даны графики точного решения (сплошная линия) и приближенного численного решения модифицированным методом Эйлера. Нетрудно заметить, что точки (ввиду невысокой точности оценки вычислений по графикам) укладываются на кривую точного решения.

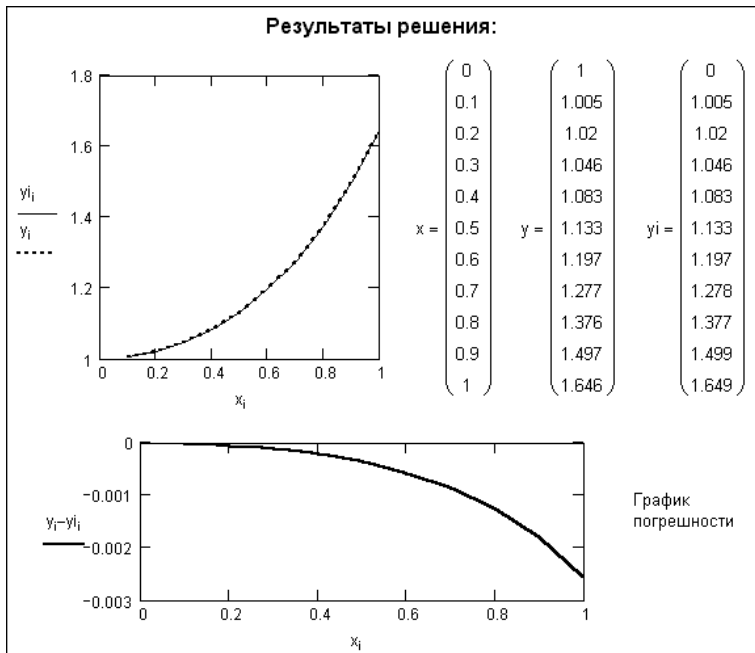


Рис. 7.61. Решение одиночного ДУ модифицированным методом Эйлера (конец документа)

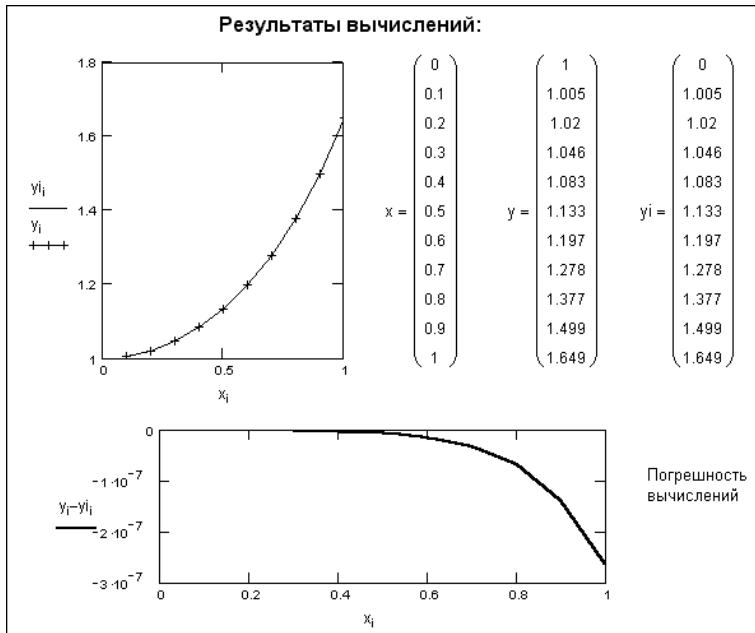


Рис. 7.63. Конец документа, иллюстрирующего решение дифференциального уравнения методом Рунге—Кутта

7.8.3. Решение ДУ вида $y''=F(x,y,z)$ методом Рунге—Кутта

ДУ вида $y'' = F(x,y,z)$, где $z = y'$ называют дифференциальными уравнениями второго порядка. Это очень важный класс ДУ, имеющих обширные сферы применения, например, при анализе автоколебательных систем в физике и в радиотехнике. Поэтому полезно ознакомиться с реализацией алгоритмов их решения.

Рисунок 7.64 показывает реализацию решения ДУ вида $y'' = F(x,y,z)$ методом Рунге—Кутта. Это фактически верхняя часть документа. Нетрудно заметить, что от реализации данного метода для ДУ первого порядка документ отличается только конкретным видом формул для коэффициентов K_i и введением помимо функции $K(x,y,z)$ еще одной функции $S(x,y,z)$, нужной для создания сетки значений $z = y'$. Поэтому в состав системы рекуррентных уравнений в векторной форме добавлено одно уравнение, задающее сетку значений z .

Конец этого документа представлен на рис. 7.65. На график выведены решения $y(x)$ и $z(x)$.

Для получения решения при новой функции $F(x,y,z)$ достаточно задать ее в начале документа, задать шаг решения h и число точек решения N .

7.8.4. Решение ДУ вида $y''(t) + Ay'(t) + By(t) = C(t)$

Предшествующий пример носил скорее иллюстративный, чем практически полезный характер. А вот решение ДУ вида $y''(t) + Ay'(t) + By(t) = C(t)$ представляет явный практический интерес, поскольку к нему сводится решение многих физических и иных задач. Рис. 7.66 показывает видимую левую часть документа, содержащего решение этого уравнения в заданном интервале времени t от начального $startt$ и до конечного $endt$ и при заданном числе шагов n .

Решение дифференциального уравнения второго порядка $y''=F(x,y,z)$, где $z=y'$ методом Рунге-Кутта

$F(x,y,z) = \left(\frac{3}{4}\right) \cdot \left(x \cdot \frac{z}{\sqrt{y}} + 1\right)$ Функция $F(x,y,z)$

$h = 0.05$ $N = 5$ Шаг h и число точек N решения

$K1(x,y,z) := h \cdot F(x,y,z)$

$K2(x,y,z) := h \cdot F\left(x + \frac{h}{2}, y + h \cdot \frac{z}{2} + h \cdot \frac{K1(x,y,z)}{8}, z + \frac{K1(x,y,z)}{2}\right)$

$K3(x,y,z) := h \cdot F\left(x + \frac{h}{2}, y + h \cdot \frac{z}{2} + h \cdot \frac{K1(x,y,z)}{8}, z + \frac{K2(x,y,z)}{2}\right)$

$K4(x,y,z) := h \cdot F\left(x + h, y + h \cdot z + h \cdot \frac{K3(x,y,z)}{2}, z + K3(x,y,z)\right)$

$K(x,y,z) = \frac{K1(x,y,z) + K2(x,y,z) + K3(x,y,z)}{6}$

$S(x,y,z) = \frac{K1(x,y,z) + 2 \cdot K2(x,y,z) + 2 \cdot K3(x,y,z) + K4(x,y,z)}{6}$

$\begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} := \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$ Задание начальных значений x, y и z

$i := 1 .. N$ Задание цикла вычислений

$\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} := \begin{bmatrix} x_{i-1} + h \\ y_{i-1} + h \cdot (z_{i-1} + K(x_{i-1}, y_{i-1}, z_{i-1})) \\ z_{i-1} + S(x_{i-1}, y_{i-1}, z_{i-1}) \end{bmatrix}$ Рекуррентные формулы метода Рунге-Кутта в векторной форме

Формулы решения методом Рунге-Кутта ДУ второго порядка

Рис. 7.64. Начало документа с алгоритмом решения ДУ вида $y'' = F(x,y,z)$



Рис. 7.65. Конец документа с алгоритмом решения ДУ вида $y'' = F(x,y,z)$

Детали вычислений по этому алгоритму, также реализующему метод Рунге—Кутта, представлены на рис. 7.67. Специфика реализации метода заключается в отдельном формировании сетки значений времени t , однако основные уравнения метода по прежнему заданы в векторной форме. Все определения заданы с помощью оператора глобального присваивания, что позволяет размещать блок решения в любой части документа. В данном случае он расположен в правой части документа, обычно невидимой. Разумеется, можно ввести этот блок в закрываемую секцию, что позволяют новые версии системы Mathcad.



Рис. 7.66. Левая часть документа с решением ДУ вида $y''(t) + Ay'(t) + By(t) = C(t)$

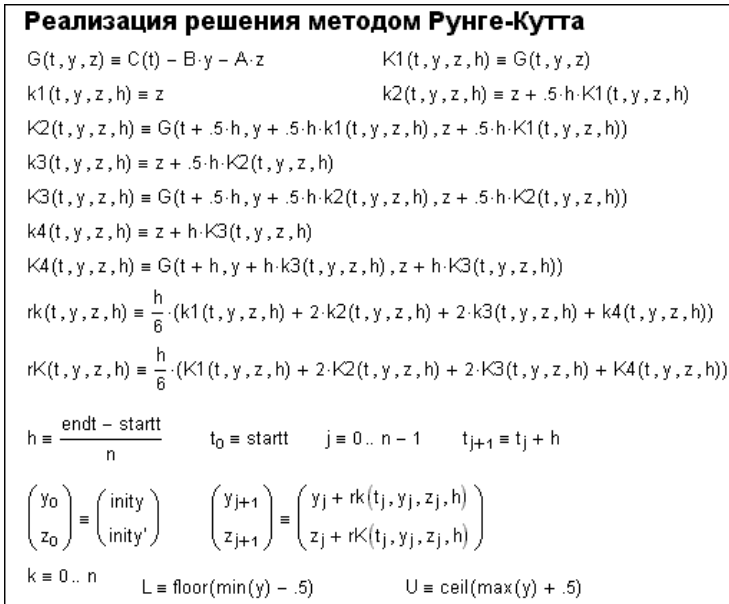


Рис. 7.67. Правая часть документа с решением ДУ вида $y''(t) + Ay'(t) + By(t) = C(t)$

В целом из приведенных примеров видно, что реализация методов решения ДУ усложняется по мере увеличения их порядка.

7.8.5. Решение системы из двух ДУ методом Рунге—Куты

При переходе от решения одиночных ДУ к решению систем ДУ сложность решения быстро нарастает. В этом нетрудно убедиться уже из реализации решения системы из двух ДУ, представленной на рис. 7.68.



Рис. 7.68. Решение системы из двух ДУ методом Рунге—Кутта (левая часть документа)

Собственно самого решения на рис. 7.68 не видно, поскольку оно скрыто в невидимой правой части документа (как и в предшествующем примере). Она представлена на рис. 7.69. Усложнение решения проявляется в усложнении функций пользователя, определяющих коэффициенты k_3 – k_4 , используемые в формулах Рунге—Кутта (k_1 отдельно не вычисляется, его значение подставлено в формулу для k в открытом виде).

Данный пример наглядно показывает, что при решении систем ДУ «в лоб» этот путь оказывается тупиковым из-за быстрого нарастания сложности уравнений и их частного характера. Можно, разумеется, составить универсальную программу на Mathcad для решения систем дифференциальных уравнений первого порядка с любым их числом. Но на практике в этом нет уж явно никакой необходимости, поскольку эта возможность профессионально выполнена с помощью встроенных функций решения дифференциальных уравнений, которые мы уже описали в этой главе.

7.8.6. Решение дифференциального уравнения $dy/dx = f(x,y)$ методом Рунге—Кутта—Фельберга

Дальнейшим усовершенствованием метода Рунге—Кутта стал метод Рунге—Кутта—Фельберга. Его суть для решения дифференциального уравнения вида $dy/dx = f(x,y)$ поясняет рис. 7.70.

Этот метод имеет два варианта решения — для погрешности пропорциональной шагу h в четвертой и в пятой степени. Пример решения дан на рис. 7.71. На этом рисунке представлены также графики погрешности решения для обоих вариантов метода.

Как и в случае других примеров этого раздела, данный пример носит только познавательный и учебный характер и наглядно показывает, насколько сложны формулы метода даже при решении одного дифференциального уравнения. Метод мож-

Реализация метода Рунге-Кутта

$$hd \equiv .5 \cdot h \qquad hk \equiv \frac{h}{6}$$

$$k2(t, x, y) \equiv F(t + hd, x + hd \cdot F(t, x, y), y + hd \cdot G(t, x, y))$$

$$K2(t, x, y) \equiv G(t + hd, x + hd \cdot F(t, x, y), y + hd \cdot G(t, x, y))$$

$$k3(t, x, y) \equiv F(t + hd, x + hd \cdot k2(t, x, y), y + hd \cdot K2(t, x, y))$$

$$K3(t, x, y) \equiv G(t + hd, x + hd \cdot k2(t, x, y), y + hd \cdot K2(t, x, y))$$

$$k4(t, x, y) \equiv F(t + h, x + h \cdot k3(t, x, y), y + h \cdot K3(t, x, y))$$

$$K4(t, x, y) \equiv G(t + h, x + h \cdot k3(t, x, y), y + h \cdot K3(t, x, y))$$

$$rk(t, x, y) \equiv hk \cdot [F(t, x, y) + 2 \cdot (k2(t, x, y) + k3(t, x, y)) + k4(t, x, y)]$$

$$rK(t, x, y) \equiv hk \cdot [G(t, x, y) + 2 \cdot (K2(t, x, y) + K3(t, x, y)) + K4(t, x, y)]$$

$$\text{endt} \equiv \text{startt} + n \cdot h \qquad j \equiv 0 \dots n - 1 \qquad t_0 \equiv \text{startt} \qquad t_{j+1} \equiv t_j + h$$

$$\begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \equiv \begin{pmatrix} \text{initx} \\ \text{inity} \end{pmatrix} \qquad \begin{pmatrix} x_{j+1} \\ y_{j+1} \end{pmatrix} \equiv \begin{pmatrix} x_j + rk(t_j, x_j, y_j) \\ y_j + rK(t_j, x_j, y_j) \end{pmatrix}$$

$$L \equiv \text{floor}(\text{if}(\min(x) < \min(y), \min(x), \min(y)) - .5)$$

$$U \equiv \text{ceil}(\text{if}(\max(x) > \max(y), \max(x), \max(y)) + .5) \qquad k \equiv 0 \dots n$$

Рис. 7.69. Решение системы из двух ДУ методом Рунге–Кутта (правая часть документа)

Метод Рунге-Кутта-Фельберга для решения уравнения $dy/dx=f(x,y)$

$$f(x, y) \equiv x \cdot y \qquad x_0 \equiv 0 \qquad x_{\text{end}} \equiv 2 \qquad y_{1,0} \equiv 1 \qquad y_{2,0} \equiv 1 \qquad n \equiv 10 \qquad h \equiv \frac{x_{\text{end}} - x_0}{n}$$
Реализация метода Рунге-Кутта-Фелберга

$$k1(f, x, y, h) \equiv h \cdot f(x, y) \qquad k2(f, x, y, h) \equiv h \cdot f(x + .25 \cdot h, y + .25 \cdot h \cdot k1(f, x, y, h))$$

$$k3(f, x, y, h) \equiv h \cdot f\left(x + \frac{3 \cdot h}{8}, y + \frac{3}{32} \cdot k1(f, x, y, h) + \frac{9}{32} \cdot k2(f, x, y, h)\right)$$

$$k4(f, x, y, h) \equiv h \cdot f\left(x + \frac{12 \cdot h}{13}, y + \frac{1932}{2197} \cdot k1(f, x, y, h) - \frac{7200}{2197} \cdot k2(f, x, y, h) + \frac{7296}{2197} \cdot k3(f, x, y, h)\right)$$

$$k5(f, x, y, h) \equiv h \cdot f\left(x + h, y + \frac{439}{216} \cdot k1(f, x, y, h) - 8 \cdot k2(f, x, y, h) + \frac{3680}{513} \cdot k3(f, x, y, h) - \frac{845}{4104} \cdot k4(f, x, y, h)\right)$$

$$k6(f, x, y, h) \equiv h \cdot f\left(x + \frac{h}{2}, y - \frac{8}{27} \cdot k1(f, x, y, h) + 2 \cdot k2(f, x, y, h) - \frac{3544}{2565} \cdot k3(f, x, y, h) + \frac{1859}{4104} \cdot k4(f, x, y, h) - \frac{11}{40} \cdot k5(f, x, y, h)\right)$$

$$rk1(f, x, y, h) \equiv \frac{25}{216} \cdot k1(f, x, y, h) + \frac{1408}{2565} \cdot k3(f, x, y, h) + \frac{2197}{4104} \cdot k4(f, x, y, h) - \frac{k5(f, x, y, h)}{5}$$

$$rk2(f, x, y, h) \equiv \frac{16}{135} \cdot k1(f, x, y, h) + \frac{6656}{12825} \cdot k3(f, x, y, h) + \frac{28561}{56430} \cdot k4(f, x, y, h) - \frac{9}{50} \cdot k5(f, x, y, h) + \frac{2}{55} \cdot k6(f, x, y, h)$$

$$j \equiv 1 \dots n \qquad x_j \equiv x_0 + j \cdot h \qquad y_{1,j} \equiv y_{1,j-1} + rk1(f, x_{j-1}, y_{1,j-1}, h) \qquad \text{Формула для ошибки } O(h^4)$$

$$y_{2,j} \equiv y_{2,j-1} + rk2(f, x_{j-1}, y_{2,j-1}, h) \qquad \text{Формула для ошибки } O(h^5)$$

$$y(x) := \exp\left(\frac{x^2}{2}\right) \qquad \text{Аналитическое решение}$$

Рис. 7.70. Решение дифференциального уравнения методом Рунге–Кутта–Фельберга (начало документа)

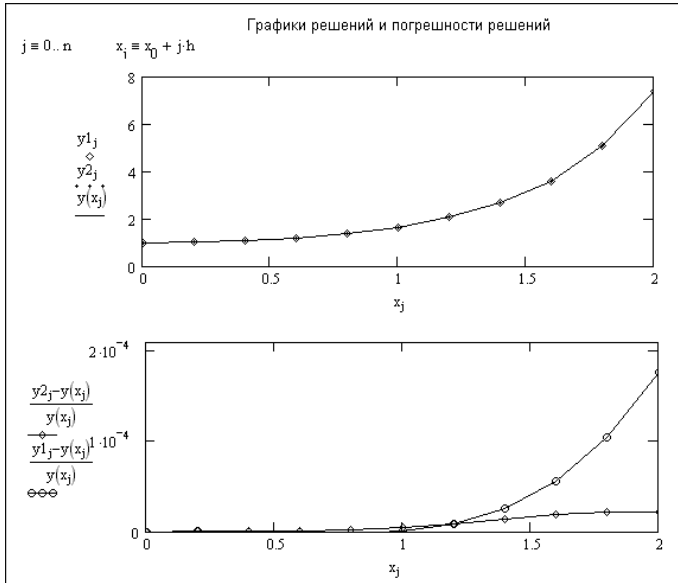


Рис. 7.71. Решение дифференциального уравнения методом Рунге—Кутты—Фельберга (конец документа)

но распространить и на решение системы ДУ, но необходимости в этом также нет, поскольку удобно использовать для этого встроены функции решения систем ДУ.

7.9. О реализации в Mathcad вариационных методов

7.9.1. Особенности решения задач механики вариационными методами

Основные проблемы механики могут решаться наряду с решениями дифференциальных уравнений с помощью *вариационных методов*. Например, положение равновесия механической системы соответствует минимуму ее потенциальной энергии. Поэтому проблема решения краевой задачи для дифференциального уравнения, описывающего механическую систему, эквивалентна проблеме отыскания функции, для которой интеграл, выражающий потенциальную энергию системы, принимает наименьшее значение. Рассмотрим одну из типовых задач реализации вариационных методов (задача предложена доц. Р. Е. Кристаллинским).

7.9.2. Решение задачи на прогиб струны

Пусть потенциальная энергия струны, закрепленной в точках $x = 0$ и $x = l$, под действием внешней нагрузки $f(x)$ определяется равенством

$$I = \frac{1}{2} \int_0^l \left(\frac{\mu}{2} (y')^2 + f \cdot y \right) dx, \quad (7.2)$$

где $y(x)$ — уравнение струны, μ — некоторая постоянная, значение которой зависит от материала, из которого изготовлена струна. Таким образом, требуется най-

ти функцию $y(x)$ удовлетворяющую краевым условиям $y(0) = y(l) = 0$, для которой приведенный интеграл принимает наименьшее значение.

К нахождению минимума некоторого интеграла сводится и решение краевых задач для дифференциальных уравнений в частных производных. Так, задача о нахождении решения уравнения Пуассона

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y)$$

в заданной области D удовлетворяющего на контуре области Γ краевому условию

$$u = \varphi(s),$$

сводится к задаче о нахождении минимума интеграла

$$I = \iint_D \left[\left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial u}{\partial y} \right)^2 + 2f(x, y)u \right] dx dy.$$

В 1908 году Вальтер Ритц предложил следующий метод приближенного решения рассматриваемого класса задач. Сущность этого метода состоит в следующем. Искомая функция представляется в виде

$$u = \phi_0 + c_1 \phi_1 + \dots + c_n \phi_n,$$

где функции $\phi_0, \phi_1, \dots, \phi_n$ линейно независимы и при всех значениях параметров c_1, c_2, \dots, c_n выполняются заданные краевые условия. Эта функция подставляется в интеграл, минимум которого мы находим. После преобразований мы получаем функцию от переменных c_1, c_2, \dots, c_n . Затем находится минимум этой функции.

При традиционном подходе задача нахождения минимума сводится к решению системы линейных уравнений. Однако если мы попытаемся увеличить точность решения за счет увеличения, полученная система становится плохо обусловленной, надежность полученных результатов при этом существенно уменьшается.

7.9.3. Решение задачи на прогиб струны в среде Mathcad

Система Mathcad позволяет непосредственно найти минимум полученной функции, что способствует повышению точности решения рассматриваемой задачи. Интегралы, которые получаются в процессе решения, целесообразно находить сначала в символьном виде.

Рассмотрим в качестве примера решение задачи о нахождении минимума интеграла (7.2). Предположим, что искомая функция обращается в 0 на концах отрезка $[0, 1]$, $\mu = 0.3$, $f(x) = x(1 - x)$. Решение будем искать в виде

$$y = \sum_{i=1}^5 c_i \sin(i\pi x).$$

Тогда

$$\int_0^1 [y''(x)]^2 dx = \sum_{i=1}^5 \sum_{j=1}^5 c_i c_j i^2 j^2 \pi^4 \int_0^1 \sin(i\pi x) \sin(j\pi x) dx = \frac{1}{2} \sum_{i=1}^5 c_i^2 i^4 \pi^4.$$

Положим

$$a_i = \frac{i^4 \pi^4}{2}.$$

Далее положим

$$b_i = \int_0^1 x(1-x) \sin(i\pi x) dx = \frac{2}{i^3 \pi^3} (1 - \cos(\pi i)).$$

Функция, минимум которой нам надлежит определить, будет иметь вид:

$$I(c) = \frac{\mu}{2} \sum_{i=1}^5 a_i c_i^2 + \sum_{i=1}^5 b_i c_i.$$



Рис. 7.72. Решение вариационной задачи в среде системы Mathcad

Дальнейшие вычисления производим в системе Mathcad (рис. 7.72). В результате решения получен профиль струны и найден его минимум.

Глава 8. Обработка данных, статистика и финансовые расчеты

При выполнении физических и иных экспериментов данные обычно представляются с той или иной погрешностью, поэтому их обработка требует применения соответствующих статистических методов. Наиболее распространенные из них и предлагает Mathcad. Кроме того, для представления физических закономерностей и при проведении научно-технических расчетов неизбежно возникает задача приближенного вычисления значений функций в промежутках между узловыми точками и за их пределами. Еще одной интересной областью применения Mathcad в обработке данных являются финансово-экономические расчеты. Именно подобного рода расчетам посвящена эта глава.

8.1. Интерполяция, экстраполяция и аппроксимация

Для представления физических закономерностей, а также при проведении научно-технических расчетов часто используются функциональные зависимости вида $y(x)$, причем число заданных точек этих зависимостей ограничено. Нередко возникает задача приближенного вычисления значений функций в промежутках между узловыми точками (*интерполяция*) и за их пределами (*экстраполяция*). Интерполяция заключается в нахождении того или иного аппарата, позволяющего вычислять значения $y(x)$ в промежутках между точками.

Другим понятием является аппроксимация. Обычно это подмена исходной зависимости какой-либо достаточно простой функцией или легко вычисляемой функцией. Система Mathcad предоставляет возможность аппроксимации двумя важными типами функций: кусочно-линейной и сплайновой. Впрочем, довольно легко можно организовать и другие методы интерполяции, экстраполяции и аппроксимации.

8.1.1. Одномерная линейная интерполяция и экстраполяция

При *кусочно-линейной интерполяции* вычисления дополнительных точек выполняются по линейной зависимости. Графически это означает просто соединение узловых точек отрезками прямых, для чего используется следующая функция:

```
linterp (VX, VY, x)
```

Для заданных векторов абсцисс VX и ординат VY узловых точек и заданного аргумента x функция `linterp` возвращает значение функции при ее кусочно-линейной аппроксимации (интерполяции). При экстраполяции используются отрезки прямых, проведенных через две крайние точки.

8.1.2. Одномерная сплайновая интерполяция и экстраполяция

При небольшом числе узловых точек (менее 10) линейная интерполяция оказывается довольно грубой. При ней даже первая производная функции аппроксимации испытывает резкие скачки в узловых точках. Для целей экстраполяции

функция `linterp` не предназначена и за пределами области определения может вести себя непредсказуемо.

Гораздо лучшие результаты дает *сплайновая интерполяция*. При ней исходная функция заменяется отрезками кубических полиномов, проходящих через три смежные узловые точки. Триады точек перемещаются по оси абсцисс, что создает набор полиномов. Коэффициенты полиномов рассчитываются так, чтобы непрерывными были первая и вторая производные. Линия, которую описывает сплайн-функция, напоминает по форме гибкую линейку, закрепленную в узловых точках. К сожалению, вывод уравнений для отдельных отрезков сплайновой функции не предусмотрен, а потому полноценная аппроксимация зависимостей затруднена.

Для осуществления сплайновой интерполяции и экстраполяции Mathcad предлагает четыре встроенные функции. Три из них служат для получения векторов вторых производных сплайн-функций при различных видах продолжения (экстраполяции):

- `cspline(VX, VY)` — возвращает вектор **VS** вторых производных с продолжением по кубическому полиному;
- `pspline(VX, VY)` — возвращает вектор **VS** вторых производных с продолжением по параболической кривой;
- `lspline(VX, VY)` — возвращает вектор **VS** вторых производных с линейным продолжением.

Наконец, четвертая функция `interp(VS, VX, VY, x)` возвращает значение $y(x)$ для заданных векторов **VS**, **VX**, **VY** и заданного значения x .

Таким образом, сплайн-интерполяция проводится в два этапа. На первом с помощью функции `cspline`, `pspline` или `lspline` отыскивается вектор вторых производных функции $y(x)$, заданной векторами VX и VY ее значений (абсцисс и ординат). Затем на втором этапе для каждой искомой точки вычисляется значение $y(x)$ с помощью функции `interp`.

8.1.3. Примеры линейной и сплайновой интерполяции

На рис. 8.1 показан фрагмент документа Mathcad, иллюстрирующий применение описанных функций для линейной и сплайновой интерполяции при трех видах сплайнов — функции f_1 , f_2 и f_3 . Они соответствуют сплайнам с линейным, квадратическим и кубическим продолжением.

Нетрудно заметить, что при линейной интерполяции график оказывается слишком грубым — отчетливо видны точки перегибов, совпадающие с узловыми точками. В то же время сплайновая интерполяция, несмотря на малое число точек в этом примере (их всего 6), дает прекрасные результаты: график функции оказывается плавным и точки его перегиба вообще незаметны. Впрочем, стоит отметить, что при неудачном расположении узлов выбег сплайновой интерполяции может оказаться весьма значительным.

Как отмечалось, сплайновая интерполяция может использоваться для экстраполяции с тремя видами продолжения функций — линейным, параболическим и кубическим. Такая экстраполяция представлена на рис. 8.1 отрезками кривых в интервале x от 5 до 6. Нетрудно заметить, что качество экстраполяции довольно низкое и сильно зависит от того, какие сплайны выбраны: 1 — с линейным продолжением, 2 — с параболическим продолжением и 3 — с кубическим продолжением (эти цифры относятся соответственно к переменным S_n и функциям f_n). Низкое качество экстраполяции оправдывает название «продолжение функций».

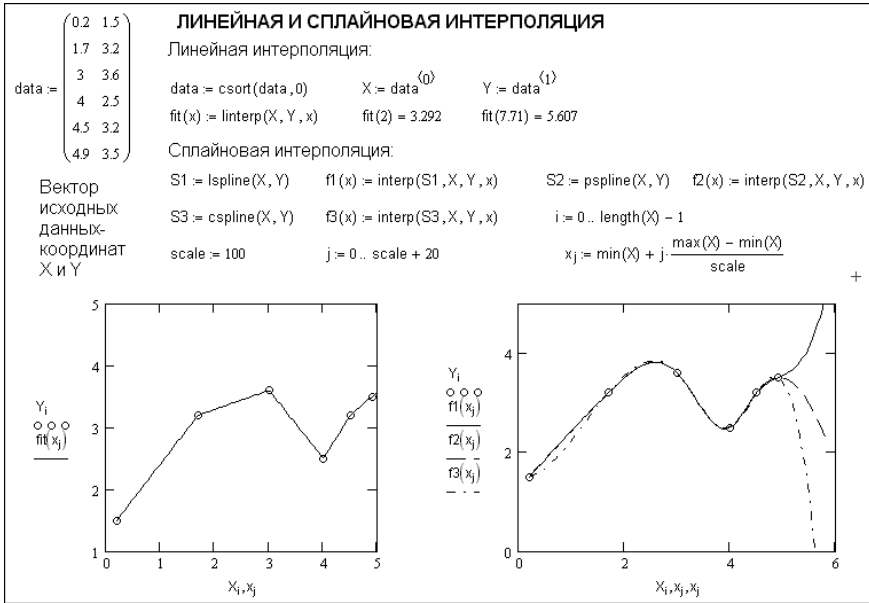


Рис. 8.1. Линейная и сплайновая интерполяция

Может показаться, что в связи с низким качеством экстраполяции само ее существование практически бесполезно. Но это далеко не так! При использовании сплайновой интерполяции для приближения нелинейных зависимостей в задачах моделирования приближение функций за пределами области узловых точек позволяет избежать разрывов функции, которые способны резко исказить ход моделирования. Другое дело, что продолжение едва ли уместно использовать для решения серьезных задач предсказания.

8.1.4. Одномерная В-сплайновая интерполяция и экстраполяция

В Mathcad введена и функция аппроксимации *В-сплайнами* `bspline(VX, VY, U, n)`, где U — вектор координат точек сшивки, $n = 1, 2$ или 3 — степень полиномов. Размер вектора U на $(n-1)$ меньше размера векторов x и y . *В-сплайновая* аппроксимация имеет продолжение и позволяет задавать сшивку сплайн-функций в точках, отличных от узловых. Крайняя левая точка сшивки должна быть левее, а правая — правее конечных узловых точек. На рис. 8.2 представлены примеры интерполяции и продолжения с применением сплайн-функций с линейным ($n = 1$), квадратичным ($n = 2$) и кубическим ($n = 3$) продолжением.

Разумеется, качество *В-сплайновой* интерполяции сильно зависит от правильного положения точек сшивки и, разумеется, от набора исходных узловых точек. Как видно из рис. 8.2, эта интерполяция может использоваться и для экстраполяции. Однако хорошей назвать экстраполяцию и в этом случае трудно.

8.1.5. Двумерная линейная и сплайновая интерполяция

Для повышения качества построения трехмерных графиков имеется возможность выполнения двумерной сплайновой интерполяции. Это позволяет существенно повысить представительность сложных графиков функций, в том числе контурных (рис. 8.3).

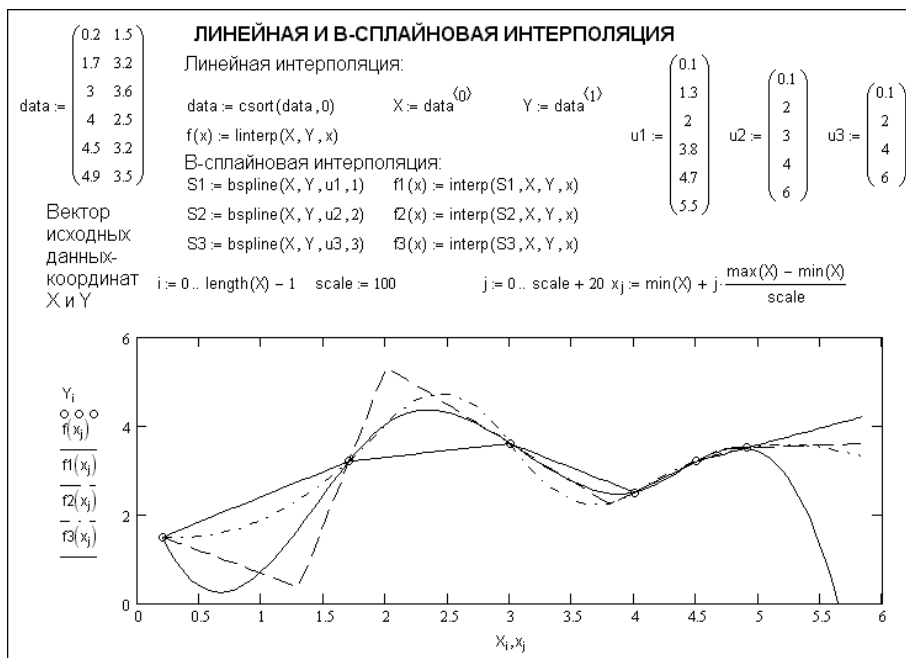


Рис. 8.2. Линейная и В-сплайновая интерполяция и экстраполяция

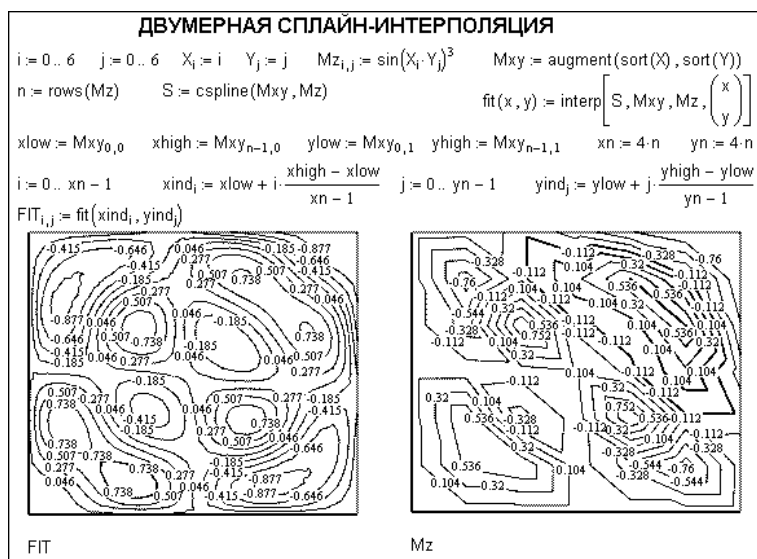


Рис. 8.3. Пример применения двумерной сплайновой интерполяции для построения контурного графика сложной поверхности

На этом рисунке слева показан контурный график после двумерной сплайновой интерполяции, а справа — без нее (с применением линейной интерполяции). Достоинства двумерной сплайновой интерполяции очевидны и в том случае, если представить графики рис. 8.3 в виде поверхностей, просто переформатировав их.

8.2. Аппроксимация функций и данных

8.2.1. Приближение функций рядом Тейлора

Помимо рассмотренной выше задачи интерполяции данных не меньшее место в практике научно-технических и математических расчетов занимает задача *аппроксимации* функций и данных — т. е. замена вычисления их на вычисления по какому-то аппроксимирующему выражению. Желательно, чтобы это выражение было проще исходной функции, допускало быстрое вычисление и обеспечивало приемлемую для практических целей погрешность. Также весьма желательно, чтобы аппроксимирующее выражение обеспечивало достаточное единообразие в представлении различных данных.

Одним из первых методов единообразного представления функций было разложение их в *ряд Тейлора*. Если $f(x)$ дифференцируемая функция, то ее разложение в ограниченный ряд Тейлора в окрестности точки x_0 задается приближенным выражением:

$$f(x) \cong f(x_0) + \frac{f'(x_0)}{1!}(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \dots + \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n.$$

Ряд Тейлора перестает быть ограниченным при $n \rightarrow \infty$. Если разложение выполняется относительно точки $x_0 = 0$, его принято называть *рядом Маклорена*:

$$f(x) \cong f(x_0) + \frac{f'(x_0)}{1!}x + \frac{f''(x_0)}{2!}x^2 + \dots + \frac{f^{(n)}(x_0)}{n!}x^n.$$

Ряд Тейлора можно представить в следующем виде:

$$f(x) \cong f(x_0) + \sum_{i=1}^{\infty} f^{(i)} \frac{(x - x_0)^i}{i!}. \quad (8.1)$$

К сожалению, представление произвольной функции рядом (8.1), особенно при ограничении числа членов, имеет множество недостатков. Даже для непрерывных функций этот ряд чаще всего сходится очень медленно. Как известно, есть и непрерывные не дифференцируемые функции. И вообще операция дифференцирования является далеко не простой и точной. Для разрывных и периодически повторяющихся функций разложение (8.1) практически неприемлемо.

Тем не менее операции разложения в ряд Тейлора и Маклорена настолько распространены, что включены в набор встроенных функций практически всех математических систем. Например, в системах Mathcad для этого используется опция *series* в составе оператора символьного вывода \rightarrow . Пример разложения функции $\sin(x)$ в ряд Маклорена представлен на рис. 8.4. В местах ввода после оператора \rightarrow указывается имя независимой переменной и порядок ряда. Следует отметить, что ряды многих функций могут не содержать члены с нечетной или четной степенью, поэтому порядок реального ряда может быть ниже заданного.

Этот рисунок прекрасно иллюстрирует и основные проблемы разложения функции $\sin(x)$ в ряд Маклорена:

- погрешность разложения мала только в малой окрестности точки x_0 ;
- при заметном удалении от точки x_0 отклонение кривой ряда от кривой исходной точки x_0 резко нарастает и становится практически непредсказуемым;
- никаких признаков периодичности функции $\sin(x)$ в ее разложении в ряд не обнаруживается.

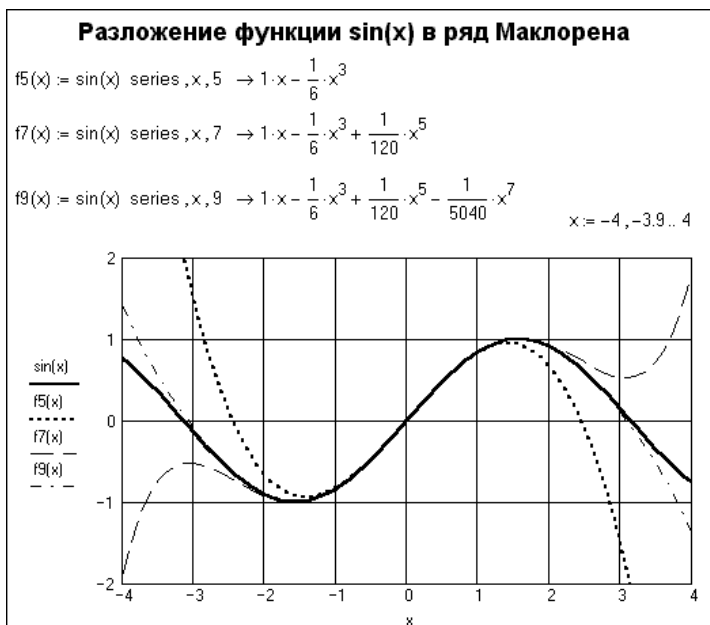


Рис. 8.4. Разложение функции $\sin(x)$ в ряд Маклорена

На рис. 8.5 показано разложение функции синуса в ряд Тейлора окрестности точки $x_0 = 1$. Хорошо видно, что в этом случае появляются члены ряда с четными степенями.

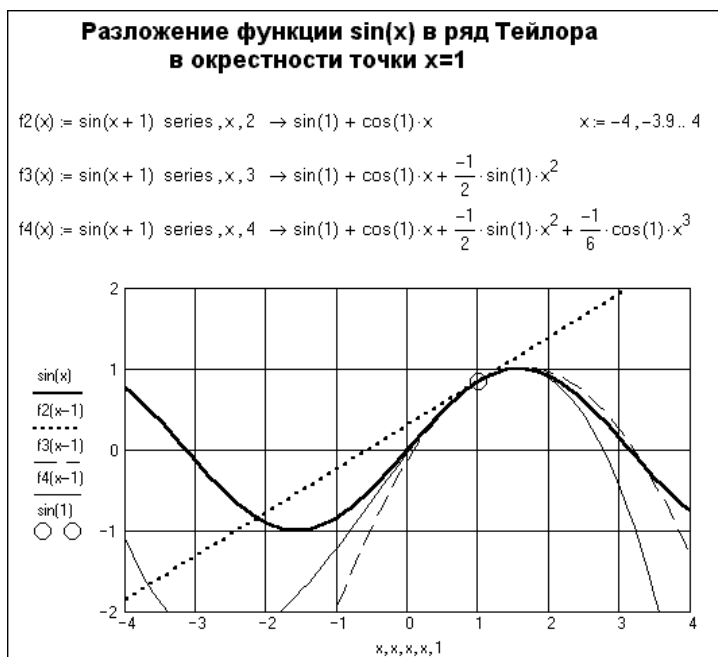


Рис. 8.5. Разложение функции $\sin(x)$ в ряд Тейлора в окрестности точки $x = 1$

Суммируя сказанное, можно со всей определенностью сказать, что ряды Тейлора и Маклорена плохо приближают даже гладкую и легко дифференцируемую синусоидальную функцию. Впрочем, как и подавляющее большинство других функций. Исключением являются лишь функции, которые могут быть представлены полиномами, но заданы в завуалированном виде. Но в этом случае, скорее всего, само разложение в ряд Тейлора или Маклорена является излишним и требуется просто грамотное преобразование функции в ее полиномиальное представление.

8.2.2. Интерполяция и экстраполяция функций по Лагранжу

Можно попытаться приблизить таблично заданную функцию другим способом, не привязываясь к конкретной одной точке, а используя все узловые точки. Пусть некоторая функция $f(x)$ определена рядом своих узловых точек (x_i, y_i) на некотором отрезке $[a, b]$. Под *интерполяцией* мы будем подразумевать вычисление значений $f(x)$ в любом промежутке $[x_i, x_{i+1}]$ в пределах отрезка $[a, b]$. Соответственно, любое вычисление $f(x)$ вне отрезка $[a, b]$ является *экстраполяцией*.

Широко применяемые в практике обработки таблиц линейная и квадратичная интерполяции, представленные выражениями

$$f(x)_{\text{лин}} = a_1x + a_0 \text{ и } f(x)_{\text{кв}} = a_2x^2 + a_1x + a_0,$$

являются частным видом *полиномиальной интерполяции* (и аппроксимации), при которой значения $f(x)$ вычисляются с помощью аппроксимирующего полинома:

$$f_a(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 = \sum_{i=0}^n a_i x^i. \quad (8.2)$$

Реализация полиномиальной аппроксимации сводится к вычислению коэффициентов полинома $a_n, a_{n-1}, \dots, a_1, a_0$ так, чтобы точки $f_a(x_i)$ точно совпадали с узловыми точками $f(x_i)$. Нетрудно показать, что эта задача решается довольно легко. В самом деле можно составить систему линейных относительно $a_n, a_{n-1}, \dots, a_1, a_0$ уравнений вида (8.2) так, что в каждом уравнении $x = x_i$ и $f(x_i) = f_a(x_i)$. Из решения такой системы можно получить требуемые коэффициенты $a_n, a_{n-1}, \dots, a_1, a_0$ многочлена. При этом значения полинома в узловых точках точно совпадают со значениями ординат в этих точках.

Есть и более эффективные способы полиномиальной аппроксимации и ряд удобных форм полинома — в частности форма, известная как формула интерполяции по Лагранжу. Ее удобство в том, что в ней фигурируют лишь координаты узловых точек (рис. 8.5). Эта формула использует только данные об узловых точках функции. Замечательно, что в таблице узлы могут быть в произвольном порядке!

Недостатками обобщенной формулы Лагранжа является ее сложность и отсутствие явного выражения для аппроксимирующего полинома. Тем не менее это единая аппроксимирующая формула и в ряде случаев ее применение удобно и полезно, например, если число узловых точек приходится часто менять. Иногда при вычислениях по этой формуле может происходить переполнение разрядной сетки чисел и тогда вычисления будут сопровождаться ошибкой.

8.2.3. Полиномиальная аппроксимация

Несложно выполнить и *полиномиальную аппроксимацию* таблично заданной функции в общем случае (для произвольно заданной степени полинома), то есть получить аппроксимирующий полином (8.2) в явном виде. Пример этого представлен на рис. 8.7. В данном случае гарантировано, что кривая полинома точно (или, точнее, с погрешностью машинных расчетов) пройдет через все узловые



Рис. 8.6. Пример интерполяции табличных данных с помощью обобщенной формулы Лагранжа

точки. Но, увы, гарантии высокой точности при интерполяции и тем более при экстраполяции нет. Тем не менее такая аппроксимация неплохо подходит для многих научно-технических расчетов — например, на рис. 8.7 она аппроксимирует N-образную вольт-амперную характеристику (ВАХ) туннельного диода. После этого можно выполнять вычисления ВАХ как в узловых точках, так и между ними и даже немного за пределами области задания узловых точек (экстраполяция).

Особенностью рассмотренной только что глобальной полиномиальной аппроксимации является однозначное соответствие между числом узловых точек N аппроксимируемой функции и степенью полинома $n = N - 1$. На практике можно нередко задать функцию множеством точек, но тогда степень полинома станет очень большой, его вычисления займут много времени, а главное, при этом точность вычислений может резко ухудшиться из-за осцилляций полиномов высоких степеней. Практически максимальная степень полинома не превышает 8—10 и лишь в СКМ с точными вычислениями может быть существенно повышена.

К сожалению, для более сложных видов полиномиальной аппроксимации (рациональной или Паде, Чебышевской, минимаксной и др.) специальных средств в обычных версиях Mathcad 2001i/11/12 нет. Но их можно найти в расширении Numeric Recipes или реализовать программно.

8.3. Статистическая обработка данных

8.3.1. Эксперименты, события и другие понятия статистики

Создание некоторой системы условий называют *испытанием* или *экспериментом*. Если до осуществления эксперимента его результаты нельзя точно предсказать, то эксперимент называют *вероятностным*, *случайным* или *стохастическим*. В ходе эксперимента происходят *факты* или *события* A , наступление которых



Рис. 8.7. Полиномиальная аппроксимация

можно наблюдать. Изучением законов, которым подчиняются случайные события, занимается *теория вероятности*.

События могут быть достоверными Ω , невозможными \emptyset и случайными. В последнем случае события могут наступать или не наступать. Пара событий может быть несовместной, если наступление одного события исключает другое (например, падение монеты на ту или иную сторону). События могут быть взаимно противоположными, если они несовместны и одно из них наступает. Возможны объединения (суммы) и пересечения событий.

Случайные события характеризуются *вероятностью события* $P(A)$, которую оценивают числом от 0 (событие не наступает) до 1 (при 1 событие непременно наступит). Если число элементарных исходов некоторого эксперимента равно n , а событию A благоприятствует m исходов, то *классическая вероятность* события A будет $P(A) = m/n$. Пусть на тарелке лежит 10 белых и $m = 5$ красных черешен. Значит $n = 10 + 5 = 15$. Какова вероятность, что мы возьмем наугад красную черешню? Она равна $P(A) = m/n = 5/15 = 1/3$.

Классическое определение вероятности неприемлемо, если события не являются равновероятными. Например, игральный кубик со скошенными некоторыми гранями не имеет равновероятные варианты выпадения. В таких случаях пользуются *статистической вероятностью* событий. Пусть при n экспериментов событие A наступило m раз. Это число называют абсолютной частотой события A , а $P^*(A) = m/n$ называют относительной частотой события. Вероятностью события A называют число $P(A)$, около которого группируются значения относительной частоты события A при большом числе экспериментов (испытаний).

Математическая статистика это наука о методах систематизации и использования статистических данных для получения научных и практических выводов. Она решает следующие типовые задачи:

- получение данных о различных перестановках и сочетаниях объектов (задачи комбинаторики);
- получение усредняющих данных об объектах, например успеваемости студентов или их росте и весе;
- сравнение эффективности разных технологий и процессов;
- решение задач приближения и аппроксимации экспериментальных зависимостей;
- сглаживание данных с большими случайными ошибками (шумами);
- прогноз некоторых событий (например курса валют и др.).

Есть и множество других, в том числе сложных задач статистики, но в рамках данной книги мы ограничимся этими задачами. При этом мы будем рассматривать некоторую совокупность данных, называемую *генеральной совокупностью*, а также выборки данных из нее, именуемые *выборочными совокупностями*. Как правило, данные мы будем представлять в виде *вариационного ряда*, при котором данные используются в порядке возрастания результатов наблюдения.

8.3.2. Решение задач комбинаторики

К числу элементарных задач статистики относятся задачи *комбинаторики*. Рассмотрим основные из них.

Перестановкой n объектов называют их расположение в определенном порядке. Число перестановок задается как значение факториала $P_n = n \cdot (n - 1) \cdot \dots \cdot 2 \cdot 1 = n!$, для вычисления которого в Mathcad есть оператор $!$. Например, число перестановок для 10 предметов есть $10! = 3628800$. Значения факториала быстро растут с ростом n .

Размещением некоторой части m из множества n элементов называется их расположение в определенном порядке. Число размещений обозначают как

$$A_n^m = n \cdot (n - 1) \cdot \dots \cdot (n - m) \cdot (n - m + 1) = \frac{n!}{(n - m)!}.$$

Пример: сколько вариантов набора двух разных цифр возможно на диске телефона, имеющем 10 цифр? Ответ $A_{10}^2 = 10 \cdot 9 = 90$.

Сочетанием m элементов из множества n элементов называют любую часть элементов (подмножества) этого множества

$$C_n^m = \frac{A_n^m}{m!}.$$

Пример: сколько способов выбора делегации из $m = 3$ человек возможны группы, насчитывающей $n = 10$ человек. Имеем

$$C_{10}^3 = \frac{10 \cdot 9 \cdot 8}{1 \cdot 2 \cdot 3} = 120.$$

В системе Mathcad имеются следующие основные функции комбинаторики (см. также раздел 6.6.3):

- `permit(n, m)` — вычисление числа размещений;
- `combin(m, n)` — вычисление числа сочетаний.

Примеры применения этих функций:

$$\text{permit}(10, 2) = 90 \quad \text{combin}(10, 3) = 120.$$

8.3.3. Дискретные и непрерывные случайные величины

В теории вероятностей *случайной величиной* называют переменную величину, которая, в зависимости от исхода испытания, случайно принимает какое-либо одно значение из множества возможных значений. Случайные величины могут быть *дискретными* (отличающимися на определенную и постоянную величину) и *непрерывными*. Примерами дискретных случайных величин является последняя цифра номера телефона или число студентов в группе. Примерами непрерывных величин является вес людей или температура воздуха. Эти параметры имеют непрерывно изменяющиеся значения, порою отличающиеся очень незначительно.

Дискретные случайные величины задаются своими значениями и их вероятностями, например в виде следующей таблицы:

X	x_1	x_2	x_3	...	x_{n-1}	x_n
P	p_1	p_2	p_3	...	p_{n-1}	p_n

В сумме вероятности дискретной случайной величины равны 1. Математическим ожиданием дискретной случайной величины называют значение $M(X) = x_1 p_1 + x_2 p_2 + \dots + x_n \cdot p_n$. Если дискретных случайных величин достаточно много, то математическое ожидание их приближенно равно среднему значению $M(X) \approx \bar{x} = (x_1 + x_2 + \dots + x_n)/n$.

Мерой «рассеивания» дискретных случайных величин могло бы служить отклонение случайных величин от их математического ожидания. Но, имея разные знаки, отклонения часто взаимно компенсируются. Поэтому мерой «рассеивания» принято считать *квадрат отклонений* случайной величины X (вы, вероятно, отметили, что большими буквами обозначаются случайные величинами, а малыми из значения).

Дисперсией $D(X)$ дискретной случайной величины X называется математическое ожидание квадрата отклонения случайной величины X от ее математического ожидания, т. е. $D(X) = M[X - M(X)]^2$. А средним квадратичным отклонением называют корень квадратный из дисперсии

$$\sigma(X) = \sqrt{D(X)}.$$

Непрерывные случайные величины могут принимать любые значения на том или ином отрезке. Поэтому их закон распределения нельзя описать в виде таблицы. *Функцией распределения* (или интегральной функцией распределения) непрерывных случайных величин называется функция $F(x)$, равная вероятности того, что случайная величина X приняла значение, меньшее x : $P(X) = \{X < x\}$. Функция $F(x)$ всегда монотонно растущая и ее значения лежат в пределах от 0 до 1. Если значения x лежат в пределах от $-\infty$ до $+\infty$, то $F(-\infty) = 0$ и $F(+\infty) = 1$.

Плотностью вероятности (или дифференциальной функцией распределения) случайной величины называют функцию $f(x) = F'(x)$. Вероятность попадания непрерывной случайной величины в интервал $(a; b)$ равна интегралу от $f(x)$ в пределах от a до b :

$$P(a < x < b) = \int_a^b f(x) dx = F(b) - F(a).$$

Математическим ожиданием $M(X)$ непрерывной случайной величины X с плотностью вероятности $f(x)$ называют величину несобственного интеграла

$$M(X) = \int_{-\infty}^{\infty} x \cdot f(x) dx,$$

а дисперсией случайной величины с математическим ожиданием a именуют величину:

$$D(X) = \int_{-\infty}^{\infty} (x - a)^2 f(x) dx.$$

Как и в случае дискретных величин, дисперсия является мерой отклонения случайных величин от их математического ожидания.

8.3.4. Законы распределения и статистические функции Mathcad

В статистической обработке данных генеральной или выборочной совокупностей используются различные законы распределения непрерывных случайных величин. Один из самых простых законов — равномерный. Он соответствует постоянному значению. $f(x) = C$ на отрезке $[a, b]$ с единичной площадью зависимости $f(x)$. Отсюда следует, что $C = 1/(b - a)$. Mathcad имеет функцию $\text{rnd}(x)$ для генерации случайных чисел с равномерным распределением на отрезке $[0, x]$.

Функция $\text{hist}(\text{int}, A)$ возвращает вектор частот попадания данных массива A в заданные вектором int интервалы. Вектор int должен содержать значения границ, число попаданий данных из вектора M должно подсчитываться. Если строится гистограмма из N элементов, то вектор int должен содержать $N + 1$ элементов. Функция возвращает вектор из N элементов, числовые значения которых можно использовать для графического построения гистограмм.

Приведем следующий пример: надо создать вектор из 1000 случайных чисел с равномерным распределением на отрезке $[0, 10]$, построить график точек, представляющих эти числа, и гистограмму из 10 столбцов. Документ, осуществляющий эти операции, представлен на рис. 8.8. Равномерность заполнения точками



Рис. 8.8. Работа со случайными числами и оценка их параметров

прямоугольника рисунка и примерно одинаковая высота всех столбиков гистограммы указывают, что и впрямь распределение чисел близко к равномерному.

Одним из самых распространенных является нормальный закон распределения:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{(x-a)^2}{2\sigma^2}},$$

где a — математическое ожидание и σ — среднее квадратичное отклонение. Интегральная функция распределения для него

$$F(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{(x-a)^2}{2\sigma^2}}.$$

Русским математиком А. М. Ляпуновым было доказано (центральная предельная теорема), что закон распределения суммы достаточно независимых величин весьма близок к нормальному закону.

В Mathcad имеется ряд наборов функций, относящихся к наиболее распространенным законам распределения. Характер функции для каждого закона распределения задается первой буквой их имени:

- d (Density) — плотность вероятности $f(x)$;
- p (Probability) — функция распределения $F(x)$;
- q (Quantil) — инверсная функция распределения — *квантиль*;
- r (Random) — вектор случайных чисел.

Квантили функций распределения случайных величин позволяют по заданной вероятности вычислить такое значение x , при котором вероятность равна или меньше заданного значения p . А функции, начинающиеся с буквы r, служат для генерации случайных чисел с заданным законом распределения.

В качестве примера приведем функции нормального и экспоненциального распределений:

- dnorm(x, μ, σ) pnorm(x, μ, σ) qnorm(p, μ, σ) rnorm(m, μ, σ) — нормальное распределение ($\mu = a$ — среднее значение, $\sigma > 0$ — среднеквадратичное отклонение);
- dexp(x, r) pexp(x, r) qexp(p, r) rexp(m, r) — экспоненциальное распределение ($r, x > 0$).

Статистические расчеты — весьма обширная область математики. К статистическим функциям общего характера в системе Mathcad относятся следующие функции скалярного аргумента x :

- cnorm(x) — кумулятивная нормальная функция — подобна функции pnorm($x, 0, 1$), описанной выше;
- erf(x) — функция ошибок (или интеграл вероятности) $erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$;
- cerf(x) — дополнительная функция ошибок ($1 - erf(x)$).

Приведем несколько примеров применения статистических функций, представленных выше. Примеры представлены на рис. 8.9.

Следующая группа функций относится к вычислению основных статистических параметров одного массива данных (матрицы размера $m \times n$ или вектора):



Рис. 8.9. Примеры применения статистических функций

- $\text{mean}(A)$ — возвращает среднее значение элементов массива A :

$$\text{mean} = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} A_{i,j};$$

- $\text{gmean}(A)$ — возвращает гармоническое среднее значение элементов массива A :

$$\text{gmean} = \left(\prod_{i=0}^{m-1} \prod_{j=0}^{n-1} A_{i,j} \right)^{1/(mn)};$$

- $\text{hmean}(A)$ — возвращает геометрическое среднее значение элементов массива A :

$$\text{hmean} = \left(\frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \frac{1}{A_{i,j}} \right)^{-1};$$

- $\text{median}(A)$ — возвращает медиану (средний элемент) массива A ;
- $\text{mode}(A)$ — возвращает значение наиболее часто повторяющегося элемента массива, если он явно есть, в противном случае дает сигнал ошибки;
- $\text{stdev}(A)$ — задает стандартное отклонение элементов массива A — $\sqrt{\text{var}(A)}$;
- $\text{Stdev}(A)$ — задает выборочное стандартное отклонение элементов массива A — $\sqrt{\text{Var}(A)}$;
- $\text{var}(A)$ — возвращает так называемую смещенную оценку дисперсии (вариацию) для элементов массива A :

$$\text{var} = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |A_{i,j} - \text{mean}(A)|^2;$$

- $\text{var}(A)$ — возвращает несмещенную оценку дисперсии для элементов массива A с иной, чем у функции $\text{var}(A)$ нормировкой:

$$\text{var} = \frac{1}{mn-1} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |A_{i,j} - \text{mean}(A)|^2;$$

- $\text{kurt}(A)$ — возвращает значение эксцесса (остроты кривой распределения):

$$\text{kurt} = \frac{mn(mn+1)}{(mn-1)(mn-2)(mn-3)} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \left(\frac{A_{i,j} - \text{mean}(A)}{\text{Stdev}(A)} \right)^4 - \frac{3(mn-1)^2}{(mn-2)(mn-2)}.$$

Эти функции часто используются для подсчета частоты используемых в речи звуков, букв и слов, данных о людях и др.

Функция mode нуждается в некоторых пояснениях. Ее действие очевидно из приведенных ниже примеров:

Массив	mode
1, 2, 3, 4, 5	нет
1, 2, 3, 3, 4, 5, 6	3
1, 2, 2, 3, 4, 5, 5, 5, 6, 7, 8	2, 5

Когда речь идет о некоторых зависимостях, данные обычно представляются двумя и более массивами. Степенью связи зависимостей является *коэффициент корреляции* $\text{corr}(A, B)$ — коэффициент корреляции Пирсона двух массивов A и B ($m \times n$ элементов). Если коэффициент корреляции близок к 0, то данные (зависимости) не согласованы, а при близости его к 1 данные согласованы.

8.3.5. Функции вычисления плотности распределения вероятности

Функции вычисления плотности вероятности распределения в Mathcad представлены следующим набором:

- $\text{dbeta}(x, s1, s2)$ — бета-распределение ($s1, s2 > 0$ — параметры формы, $0 < x < 1$);
- $\text{dbinom}(k, n, p)$ — биномиальное распределение (возвращает значение вероятности $P(x = k)$, где n и k целые числа, причем $0 \leq k \leq n$ и $0 \leq p \leq 1$);
- $\text{dcauchy}(x, l, s)$ — распределение Коши (l — параметр разложения, $s > 0$ — параметр масштаба);
- $\text{dchisq}(x, d)$ — хи-квадрат-распределение ($x, d > 0$, причем d — число степеней свободы);
- $\text{dexp}(x, r)$ — экспоненциальное распределение ($r, x > 0$);
- $\text{dF}(x, d1, d2)$ — распределение Фишера ($d1, d2 > 0$ — числа степеней свободы, $x > 0$);
- $\text{dgamma}(x, s)$ — гамма-распределение ($s > 0$ — параметр формы, $x \geq 0$);
- $\text{dgeom}(k, p)$ — геометрическое распределение ($0 < p \leq 1$ — вероятность успеха в отдельном испытании, k — целое неотрицательное число);
- $\text{dlnorm}(x, \mu, \sigma)$ — логарифмическое нормальное распределение (μ — натуральный логарифм среднего значения, $\sigma > 0$ — натуральный логарифм среднеквадратичного отклонения, $x > 0$);

- `dlogis(x, l, s)` — логистическое распределение (l — параметр разложения, $s > 0$ — параметр масштаба);
- `dnbinom(k, n, p)` — отрицательное биномиальное распределение ($n > 0$ и $k > 0$ — целые числа, $0 < p \leq 1$);
- `dnorm(x, μ , σ)` — нормальное распределение (μ — среднее значение, $\sigma > 0$ — среднеквадратичное отклонение);
- `dpois(k, λ)` — распределение Пуассона ($\lambda > 0$, k — целое неотрицательное число);
- `dt(x, d)` — распределение Стьюдента ($d > 0$ — число степеней свободы, x — вещественное число);
- `dunif(x, a, b)` — равномерное распределение (a и b — граничные точки интервала, причем $a < b$ и $a \leq x \leq b$);
- `dweibull(x, s)` — распределение Вейбулла ($s > 0$ — параметр формы).

8.3.6. Функции распределения

Функции распределения дают вероятность того, что случайная величина будет иметь значения, меньшие или равные определенной величине. Они представлены ниже (смысл и значения параметров указаны ранее):

- `pbeta(x, s1, s2)` — значение в точке x функции бета-распределения;
- `pbinom(k, n, p)` — значение функции распределения биномиального закона для k успехов в серии из n испытаний;
- `prcauchy(x, l, s)` — значение в точке x функции распределения Коши со шкалой параметров l и s ;
- `pchisq(x, d)` — значение в точке x кумулятивного хи-квадрат-распределения, в котором d — степень свободы;
- `rexp(x, r)` — значение в точке x функции экспоненциального распределения;
- `pF(x, d1, d2)` — значение в точке x функции распределения Фишера;
- `pgamma(x, s)` — значение в точке x функции гамма-распределения;
- `pgeom(k, p)` — значение в точке x функции геометрического распределения;
- `plnorm(x, μ , σ)` — значение в точке x функции логарифмического нормального распределения;
- `plogis(x, l, s)` — значение в точке x функции логистического распределения;
- `plnorm(x, μ , σ)` — значение в точке x функции нормального распределения;
- `pnbinom(k, n, p)` — значение в точке x функции отрицательного биномиального распределения;
- `ppois(k, λ)` — значение для k функции распределения Пуассона;
- `pt(x, d)` — значение в точке x функции распределения Стьюдента;
- `punif(x, a, b)` — значение в точке x функции равномерного распределения;
- `pweibull(x, s)` — значение в точке x функции распределения Вейбулла.

8.3.7. Квантили распределения

Следующая группа задает обращения (квантили) функций распределения случайных величин. Они позволяют по заданной вероятности вычислить такое значение x , при котором вероятность равна или меньше заданного значения p :

- `qbeta(p, s1, s2)` — квантили обратного бета-распределения с параметрами формы s_1 и s_2 ;
- `qbinom(p, n, q)` — количество успешных определений при решении уравнения Бернулли, если число испытаний равно n , вероятность этого количества успешных определений равна p , а q — вероятность успеха при однократном испытании ($0 \leq q \leq 1$ и $0 \leq p \leq 1$);
- `qcauchy(p, l, q)` — квантили обратного распределения Коши со шкалой параметров l и s ($s > 0$ и $0 < p < 1$);
- `qchisq(p, d)` — квантили обратного хи-квадрат-распределения;
- `qexp(p, r)` — квантили обратного экспоненциального распределения, при котором параметр $r > 0$ определяет частоту ($0 \leq p < 1$);
- `qF(p, d1, d2)` — квантили обратного распределения Фишера, в котором d_1 и d_2 — степени свободы;
- `qgamma(p, s)` — квантили обратного гамма-распределения;
- `qgeom(p, q)` — квантили обратного геометрического распределения;
- `qlnorm(p, μ , σ)` — квантили обратного логарифмического нормального распределения;
- `qlogis(p, l, s)` — квантили обратного логистического распределения;
- `qnbinom(p, n, q)` — квантили обратного отрицательного биномиального распределения с размером n и вероятностью ошибки q ;
- `qnorm(p, μ , σ)` — квантили обратного нормального распределения со средним значением μ и стандартным отклонением σ ;
- `qpois(p, λ)` — квантили обратного распределения Пуассона;
- `qt(p, d)` — квантили обратного распределения Стьюдента (d определяет степени свободы, $d > 0$ и $0 < p < 1$);
- `qunif(p, a, b)` — квантили обратного равномерного распределения;
- `qweibull(p, s)` — квантили обратного распределения Вейбулла.

8.3.8. Функции создания случайных чисел с различными законами распределения

Последняя группа статистических функций служит для создания векторов со случайными числами с определенными законами их распределения:

- `rbeta(m, s1, s2)` — бета-распределение;
- `rbinom(m, n, p)` — биномиальное распределение;
- `rcauchy(m, l, s)` — распределение Коши;
- `rchisq(m, d)` — хи-квадрат-распределение;
- `rexp(m, r)` — экспоненциальное распределение;
- `rF(m, d1, d2)` — распределение Фишера;
- `rgamma(m, s)` — гамма-распределение;
- `rgeom(m, p)` — геометрическое распределение;
- `rlnorm(m, μ , σ)` — логарифмическое нормальное распределение;
- `rlogis(m, l, s)` — логистическое распределение;
- `rnbinom(m, n, p)` — отрицательное биномиальное распределение;
- `rnorm(m, μ , σ)` — нормальное распределение;
- `rpois(m, λ)` — распределение Пуассона;

- $rt(m, d)$ — распределение Стьюдента;
- $runif(m, a, b)$ — равномерное распределение;
- $rweibull(m, s)$ — распределение Вейбулла.

На рис. 8.9 показан фрагмент документа Mathcad с примерами построения графиков различных статистических функций и задания наборов чисел с различным распределением.

Обилие статистических функций, включенных в систему Mathcad, позволяет ей выполнять достаточно сложные статистические расчеты. Однако ограниченный объем материала данного раздела не позволит рассмотреть все статистические задачи. В частности, мы отказываемся от рассмотрения задач сравнения, предполагающих проверку статистических гипотез. Множество примеров решения вполне тривиальных задач статистики можно найти в книгах по системе Mathcad [14—16, 19]. Примеры на решение типовых задач статистики можно найти в справке, а также в электронной книге Applied Statistics. Основное внимание сосредоточим на задачах приближения и прогноза различных зависимостей. При этом стоит отметить, что существуют более мощные специализированные пакеты для выполнения статистических расчетов, например Statistica или StatGraphics, которые заметно превосходят Mathcad в части многовариантности статистических вычислений.

8.3.9. Новые функции корреляции в Mathcad 12

Состав статистических функций в Mathcad 12 обогатился двумя новыми функциями для вычисления корреляции. Одна из них $correl(x, y)$ возвращает вектор корреляции для векторов x и y , вычисляемый по формуле одномерной корреляции:

$$correl(x, y)_i = \sum_k (x_k \cdot y_{i+k}).$$

Пример применения этой функции представлен ниже:

```
x:=(1 2 3 4)
y:=(2 3 4 5)
correl(x, y)=(2 5 12 20)
```

Другая функция $correl2d(A, K)$ вычисляет матрицу корреляции для матриц прототипа A и окна K (размер этой матрицы должен быть меньше, чем размер матрицы A). Пример ее применения:

$$A := \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

$$K := \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$correl2d(A, K) = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 6 & 8 \\ 7 & 12 & 14 \end{pmatrix}$$

Обе функции ориентированы на обработку сигналов. Более подробные данные об этих функциях можно найти в справке по ним.

8.4. Регрессия

Широко распространенной задачей обработки данных является представление их совокупности некоторой функцией $y(x)$. Задача *регрессии* чаще всего заключается в получении параметров этой функции такими, чтобы функция приближала «облако» исходных точек (заданных векторами $\forall x$ и $\forall y$) с наименьшей среднеквадратичной погрешностью. В этом случае говорят о регрессии *методом наименьших квадратов*.

8.4.1. Линейная регрессия

Чаще всего используется *линейная регрессия*, при которой функция $y(x)$ описывает отрезок прямой и имеет вид:

$$y(x) = a + b \cdot x.$$

К линейной регрессии можно свести многие виды нелинейной регрессии при зависимостях вида $y(x)$.

Математически постановка задачи регрессии сводится к следующему. Пусть есть набор точно определенных значений x_i и соответствующих им не точных значений y_i . Допустим, мы предполагаем, что существует некоторая зависимость $f(x, a_0, a_1, \dots, a_k)$, которая может рассматриваться как приближение к зависимости $y(x)$, чьи точки представлены как $y_i(x_i)$. Таким образом, мы вправе записать:

$$y_i = f(x_i, a_0, a_1, \dots, a_k) + \xi_i.$$

Здесь ξ_i — независимые случайные величины с нормальным законом распределения, определяющие погрешность задания y_i . Обычно их считают следствием ошибок эксперимента. Задача регрессии заключается в том, чтобы найти параметры a_0, a_1, \dots, a_k такими, при которых представление $y(x)$ нашей функцией $f(x)$ имело наименьшую среднеквадратичную погрешность. Для этого нужно минимизировать функцию:

$$\Phi(a_0, a_1, \dots, a_k) = \sum (f(x_i, a_0, a_1, \dots, a_k) - y_i)^2.$$

К примеру, для наиболее распространенной линейной регрессии, когда $f(x) = a_0 + a_1 \cdot x$ (часто обозначают $a_0 = a$ и $a_1 = b$) надо минимизировать следующее выражение:

$$\Phi(a_0, a_1) = \sum (a_0 + a_1 x_i - y_i)^2.$$

Если приравнять $\partial\Phi/\partial a_0$ и $\partial\Phi/\partial a_1$ к нулю, то для линейной регрессии можно найти ее параметры a_0 и a_1 в явной форме

$$a_0 = \frac{\sum_{i=0}^n y_i \sum_{i=0}^n x_i^2 - \sum_{i=0}^n x_i \sum_{i=0}^n x_i y_i}{n \sum_{i=0}^n x_i^2 - \left(\sum_{i=0}^n x_i \right)^2}, \quad a_1 = \frac{n \sum_{i=0}^n x_i y_i - \sum_{i=0}^n x_i \sum_{i=0}^n y_i}{n \sum_{i=0}^n x_i^2 - \left(\sum_{i=0}^n x_i \right)^2}.$$

Аналогичным образом можно получить выражения и для других видов регрессии: полиномиальной, экспоненциальной, логарифмической и так далее. Ввиду сложности соответствующих выражений они не приводятся — некоторые из формул можно найти в литературе. Многие задачи нелинейной регрессии можно свести к рассмотренной выше линейной, используя соответствующие преобразова-

ния. Но Mathcad для многих видов регрессии задает нужные формулы явно, что делает преобразования данных не нужными.

Для проведения линейной регрессии система Mathcad имеет следующие функции:

- $\text{corr}(VX, VY)$ — возвращает скаляр — коэффициент корреляции Пирсона;
- $\text{intercrpt}(VX, VY)$ — возвращает значение параметра a (смещение линии регрессии по вертикали);
- $\text{slope}(VX, VY)$ — возвращает значение параметра b (угловой коэффициент линии регрессии).

На рис. 8.10 показан фрагмент документа Mathcad с примером проведения линейной регрессии для данных, представленных значениями элементов в векторах VX и VY .

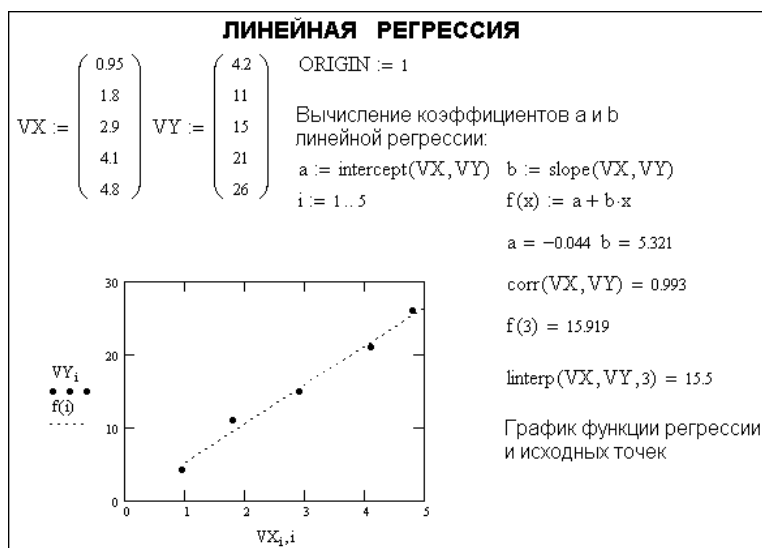


Рис. 8.10. Линейная регрессия

Как видно из рисунка, линия регрессии проходит в «облаке» исходных точек с максимальным среднеквадратичным приближением к ним. Чем ближе коэффициент корреляции к 1, тем точнее представленная исходными точками зависимость приближается к линейной.

8.4.2. Реализация линейной регрессии общего вида

В Mathcad реализована возможность выполнения линейной регрессии общего вида. При ней заданная совокупность точек приближается к функции вида:

$$F(x, K1, K2, \dots, Kn) = K1F1(x) + K2F2(x) + \dots + KnFn(x).$$

Таким образом, функция регрессии является линейной комбинацией функций $F1(x)$, $F2(x)$, ..., $Fn(x)$, причем сами эти функции могут быть нелинейными, что резко расширяет возможности такой аппроксимации и распространяет ее на многие нелинейные функции.

Для реализации линейной регрессии общего вида используется функция $\text{linfit}(VX, VY, F)$, возвращающая вектор K коэффициентов линейной регрессии об-

шего вида, при котором среднеквадратичная погрешность приближения «облака» исходных точек, координаты которых хранятся в векторах VX и VY , оказывается минимальной. Вектор F должен содержать функции $F_1(x)$, $F_2(x)$, ..., $F_n(x)$, записанные в символьном виде.

Рис. 8.11 поясняет проведение линейной регрессии общего вида с применением функции `linfit`. Процедура проведения вычислений настолько проста, что не нуждается в особых комментариях.

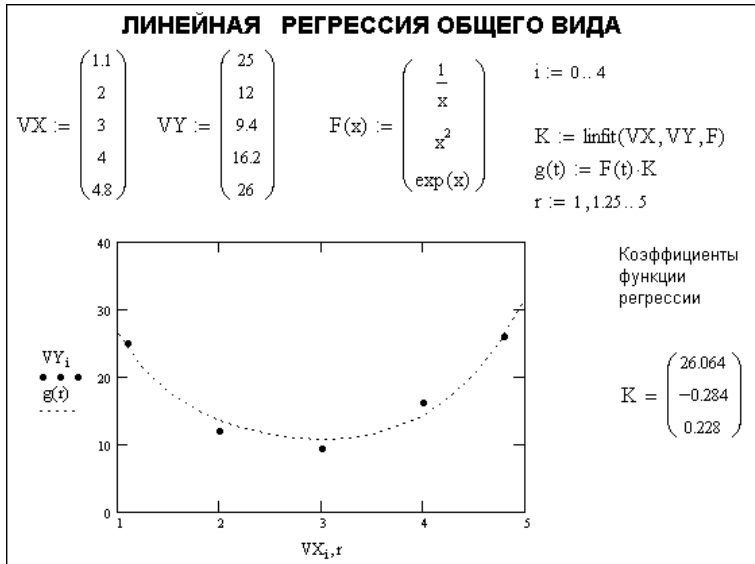


Рис. 8.11. Пример проведения линейной регрессии общего вида

Расположение абсцисс точек исходного массива в векторе VX может быть любым, но они должны идти в порядке возрастания. Вектор ординат VY должен содержать ординаты точек с абсциссами, указанными в векторе VX .

8.4.3. Реализация одномерной полиномиальной регрессии

В Mathcad введена и функция для обеспечения полиномиальной регрессии при произвольной степени полинома регрессии. Регрессия осуществляется функцией `regress(VX, VY, n)`, которая возвращает вектор VS , запрашиваемый функцией `interp(VS, VX, VY, x)` и содержащий коэффициенты многочлена n -й степени, который наилучшим образом приближает «облако» точек с координатами, хранящимися в векторах VX и VY .

На рис. 8.12 показан фрагмент документа Mathcad с примером выполнения полиномиальной регрессии. Для вычисления коэффициентов полинома регрессии используется функция `submatrix`.

На практике не рекомендуется делать степень аппроксимирующего полинома выше 4–6, поскольку погрешности реализации регрессии сильно возрастают.

Функция `regress` создает единственный приближающий полином, коэффициенты которого вычисляются по всей совокупности заданных точек. Иногда полезна другая функция полиномиальной регрессии, дающая локальные приближения отрезками полиномов второй степени `loess(VX, VY, span)`, которая возвра-



Рис. 8.12. Полиномиальная регрессия

шает вектор VS, используемый функцией $\text{interp}(VS, VX, VY, x)$ для наилучшего приближения данных векторов VX и VY отрезками полиномов второй степени. Аргумент $\text{span} > 0$ указывает размер локальной области приближаемых данных (рекомендуемое начальное значение — 0.75). Чем больше span, тем сильнее сказывается сглаживание данных. При больших значениях span эта функция приближается к функции $\text{regress}(VX, VY, 2)$.

На рис. 8.13 показан фрагмент документа Mathcad с примером приближения сложной функции со случайным разбросом ее значений с помощью совокупности отрезков полиномов второй степени (функция loess) для двух значений параметра span. Пример дан для Mathcad 2001.



Рис. 8.13. Пример регрессии отрезками полиномов второй степени

Из рисунка нетрудно заметить, что при значении $\text{span} = 0.05$ отслеживаются характерные случайные колебания значений функции, тогда как уже при $\text{span} = 0.5$ кривая регрессии становится практически гладкой. К сожалению, из-за отсутствия простого описания аппроксимирующей функции в виде отрезков полиномов этот вид регрессии широкого применения не нашел.

8.4.4. Проведение многомерной регрессии

Mathcad позволяет выполнять также многомерную регрессию. Самый типичный случай ее использования — приближение поверхностей в трехмерном пространстве. Их можно описать, задав массив значений высот z , соответствующих двумерному массиву M_{xy} координат точек (x, y) на горизонтальной плоскости.

Новых функций для этого не задано. Используются уже описанные ранее функции, но в несколько иной форме:

- `regress(Mxy, Vz, n)` — возвращает вектор, запрашиваемый функцией `interp(VS, Mxy, Vz, V)` для вычисления многочлена n -й степени, который наилучшим образом приближает точки множества M_{xy} и Vz (M_{xy} — матрица размера $m \times 2$, содержащая координаты x и y , Vz — m -мерный вектор, содержащий z -координаты, соответствующие m точкам, указанным в M_{xy});
- `loess(Mxy, Vz, span)` — аналогична `loess(VX, VY, span)`, но в многомерном случае;
- `interp(VS, Mxy, Vz, V)` — возвращает значение z по заданным векторам VS (создается функциями `regress` или `loess`) и M_{xy} , Vz и V (вектор координат x и y заданной точки, для которой находится z).

Пример многомерной интерполяции был приведен выше. В целом многомерная регрессия применяется сравнительно редко из-за сложности сбора исходных данных.

8.4.5. Проведение нелинейной регрессии общего вида

Под нелинейной регрессией общего вида подразумевается нахождение вектора K параметров произвольной функции $F(x, K_1, K_2, \dots, K_n)$, при котором обеспечивается минимальная среднеквадратичная погрешность приближения «облака» исходных точек.

Для проведения нелинейной регрессии общего вида используется функция `genfit(VX, VY, VS, F)`, которая возвращает вектор K параметров функции F , дающий минимальную среднеквадратичную погрешность приближения функцией $F(x, K_1, K_2, \dots, K_n)$ исходных данных.

Вектор F должен быть вектором с символьными элементами, причем они должны содержать аналитические выражения для исходной функции и ее производных по всем параметрам. Вектор VS должен содержать начальные значения элементов вектора K , необходимые для решения системы нелинейных уравнений регрессии итерационным методом.

На рис. 8.14 показан фрагмент документа Mathcad с примером выполнения нелинейной регрессии общего вида с помощью нелинейной функции $F(x, a, b) = a \cdot \exp(-b \cdot x) + a \cdot b$.

При решении этой задачи возникают две проблемы. Во-первых, надо вычислить значения производных по переменным a и b . Как показано на рисунке, это сделано средствами символьных операций (первая строка документа после заголовка), что наглядно показывает пользу от таких операций. Более подробно они рассматриваются в главе 9.

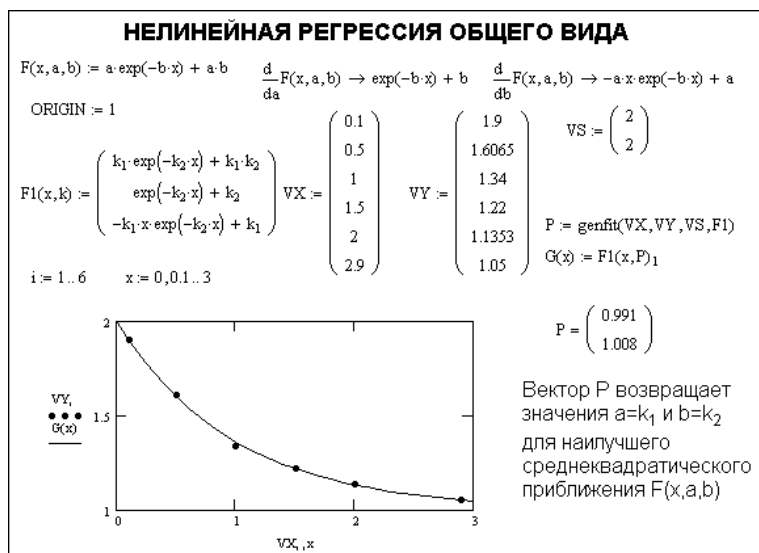


Рис. 8.14. Пример выполнения нелинейной регрессии общего вида

Вторая проблема связана с необходимостью применения функции `genfit` в ее стандартном виде. Поэтому пришлось заменить параметр a на k_1 , а параметр b на k_2 . Остальные операции в примере достаточно очевидны.

8.4.6. Новые функции для проведения регрессии в Mathcad 2000/2001

В последние версии Mathcad был введен ряд новых функций регрессии:

- `expfit(vx, vy, vg)` — возвращает вектор, содержащий коэффициенты (a , b и c) аппроксимирующего выражения вида $a \cdot e^{(bx)} + c$, график которого лучшим образом приближается к точкам, координаты которых хранятся в векторах vx и vy (вектор vg содержит первое приближение к решению);
- `lgsfit(vx, vy, vg)` — то же, но для выражения $a / (1 + b \cdot e^{(-cx)})$;
- `logfit(vx, vy)` — то же, но для выражения $a \cdot \ln(x + b) + c$ (начального приближения не требуется);
- `medfit(vx, vy)` — то же, но для выражения $a + bx$ (начального приближения не требуется);
- `pwrfit(vx, vy, vg)` — то же, но для выражения $a \cdot x^b + c$ (вектор vg содержит первое приближение к решению);
- `sinfit(vx, vy, vg)` — то же, но для выражения $a \cdot \sin(x + b) + c$.

Целесообразность введения этих функций вызывает легкое сомнение. С помощью алгоритма реализации нелинейной регрессии общего вида (см. рис. 8.14) можно легко осуществить регрессию любого частного вида, в том числе и всех тех видов, которые реализуются приведенными выше функциями. Нетрудно это сделать и с помощью функции `Minerror` (см. главу 7). Тем не менее, если пользователь нуждается в данных видах регрессии, то к его услугам теперь есть ряд функций, которые можно применить без «долгих раздумий» (эти слова взяты в кавычки, ибо любые вычисления без обдумывания лучше не выполнять).

8.4.7. Пример выполнения экспоненциальной регрессии

Поскольку все частные виды регрессии с помощью приведенных выше функций Mathcad выполняются по одной схеме, ограничимся двумя примерами ее проведения. Фрагмент документа Mathcad с примером на экспоненциальную регрессию приведен на рис. 8.15.

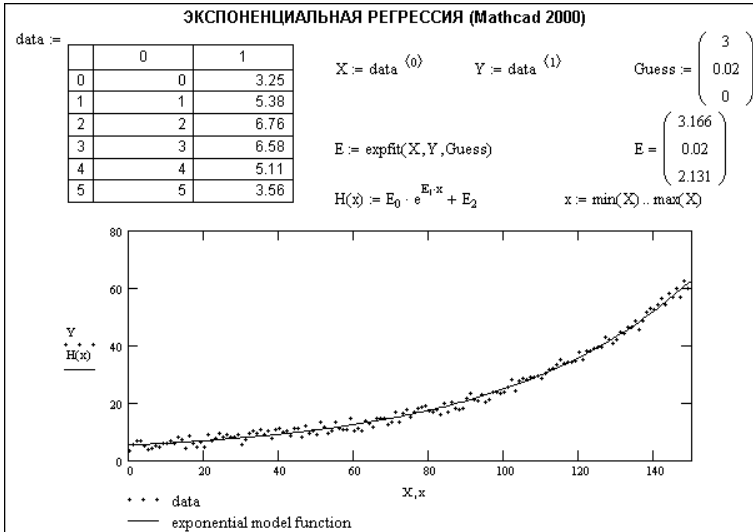


Рис. 8.15. Пример экспоненциальной регрессии

Исходные данные представлены матрицей *data*, имеющей два столбца. Нулевой столбец содержит координаты точек исходных данных, а первый — их абсциссы. Перед вычислением задаются начальные приближения параметров экспоненциальной регрессии — вектор *Guess*. Сама регрессия реализуется одним выражением с функцией *expfit*. На рисунке представлены график функции регрессии и исходные точки.

8.4.8. Пример выполнения синусоидальной регрессии

Еще один вид регрессии — синусоидальной — представляет фрагмент документа Mathcad, показанный на рис. 8.16. Здесь исходные векторы данных формируются путем добавления к значениям синусоидальной функции случайных чисел. Регрессия реализуется функцией *sinfit*.

Рекомендуется разобраться в небольших отличиях при реализации регрессии в приведенных выше примерах.

8.5. Сглаживание данных

8.5.1. Функции сглаживания данных

Данные большинства экспериментов имеют случайные составляющие, поэтому часто возникает необходимость статистического сглаживания данных. Ряд функ-

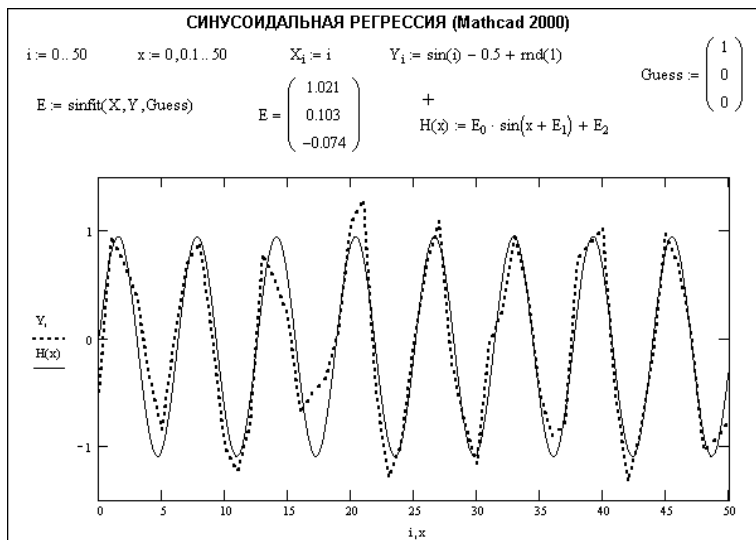


Рис. 8.16. Пример синусоидальной регрессии

ций Mathcad предназначен для выполнения операций сглаживания данных различными методами. В их названии имеется слово *smooth* (гладкий):

- *medsmooth* (VY, n) – для вектора с m действительными числами возвращает m -мерный вектор сглаженных данных по методу скользящей медианы, параметр n задает ширину окна сглаживания (n должно быть нечетным числом, меньшим m);
- *ksmooth* (VX, VY, b) – возвращает n -мерный вектор сглаженных данных VY , вычисленных на основе распределения Гаусса. VX и VY – n -мерные векторы действительных чисел. Параметр b (полоса пропускания) задает ширину окна сглаживания (b должно в несколько раз превышать интервал между точками по оси x);
- *supsmooth* (VX, VY) – возвращает n -мерный вектор сглаженных данных VY , вычисленных на основе использования процедуры линейного сглаживания методом наименьших квадратов по правилу k -ближайших соседей с адаптивным выбором k . VX и VY – n -мерные векторы действительных чисел. Элементы вектора VX должны идти в порядке возрастания.

На рис. 8.17 показан фрагмент документа Mathcad с примерами применения функций сглаживания, указанных выше. Из этих функций наиболее сильное сглаживание обеспечивает функция *supsmooth*. Однако при ней есть риск потери быстрых деталей сглаживаемой зависимости (в приведенных примерах для сглаживания используется синусоидальная функция с шумами, созданными генератором случайных чисел). Все функции сглаживания возвращают вектор сглаженных значений VY , размер которого соответствует размеру вектора VY .

В показанном на рисунке фрагменте документа для получения исходного массива точек зашумленной синусоиды используется генератор случайных чисел с равномерным распределением. При этом как расположение точек, так и вид кривых сглаживания могут отличаться от приведенных при иной настройке генератора случайных чисел. Но в любом случае будет хорошо видно, что кривые сглаживания намного более гладкие, чем кусочно-линейная функция, соединяющая точ-

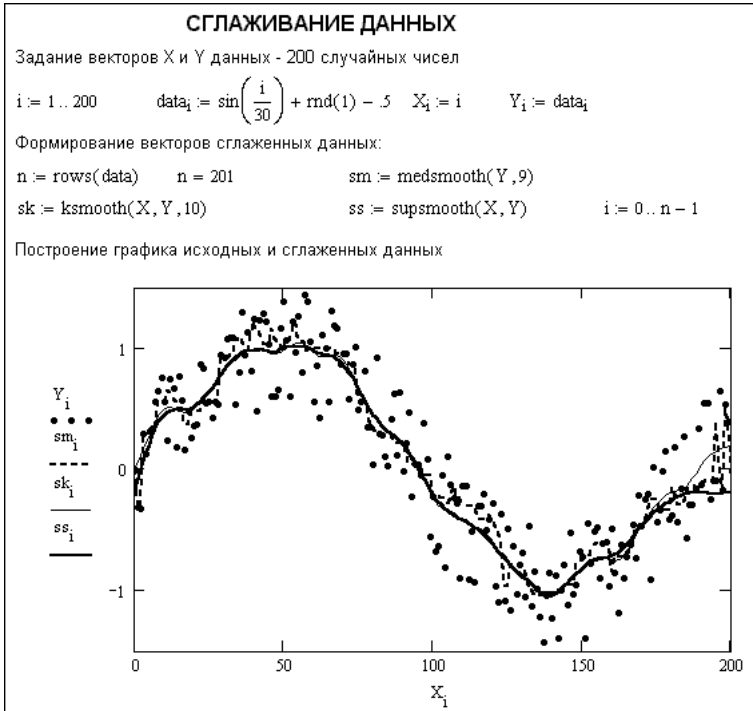


Рис. 8.17. Сглаживание данных с помощью функций сглаживания

ки друг с другом в последовательном порядке. При сглаживании бывает полезно применение функции `sort(Y)`, сортирующей данные векторов, что иногда уменьшает погрешности численного алгоритма сглаживания. Рекомендуется просмотреть примеры применения функций сглаживания в справке Mathcad.

8.5.2. Линейное сглаживание по пяти точкам

Иногда полезно знать, по каким формулам конкретно выполняется сглаживание. Рисунок 8.18 иллюстрирует решение классической задачи линейного сглаживания по пяти точкам. Термин «линейное» означает, что при сглаживании используется линейная функция. Приведены формулы сглаживания с учетом обработки краевых точек.

Линейное сглаживание по пяти точкам является частным случаем сглаживания по методу скользящей медианы — при нем каждая точка данных обрабатывается как среднее значение ее и ближайших точек. При этом происходит перемещение ансамбля обрабатываемых точек от начала вектора с ними к его концу.

8.5.3. Нелинейное сглаживание по семи точкам

Нелинейное сглаживание по семи точкам реализует документ, показанный на рис. 8.19. Увеличение числа обрабатываемых точек позволяет повысить степень сглаживания.

Результат линейного сглаживания зашумленной синусоиды показан на рис. 8.19. Сравнение его с рис. 8.18 показывает, что нелинейное сглаживание несколько лучше, чем линейное. Но чуда нет!



Рис. 8.18. Линейное сглаживание по пяти точкам

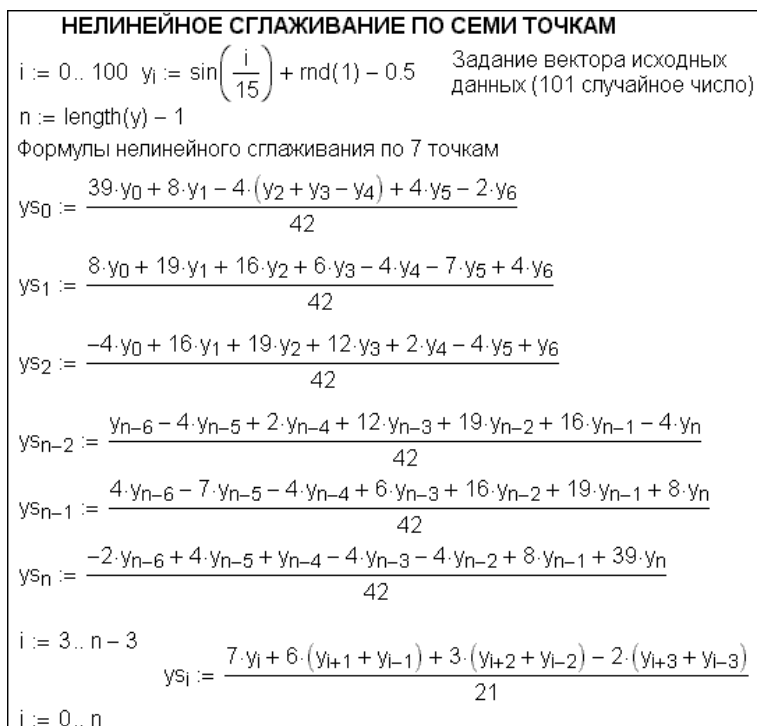


Рис. 8.19. Нелинейное сглаживание по семи точкам (алгоритм)

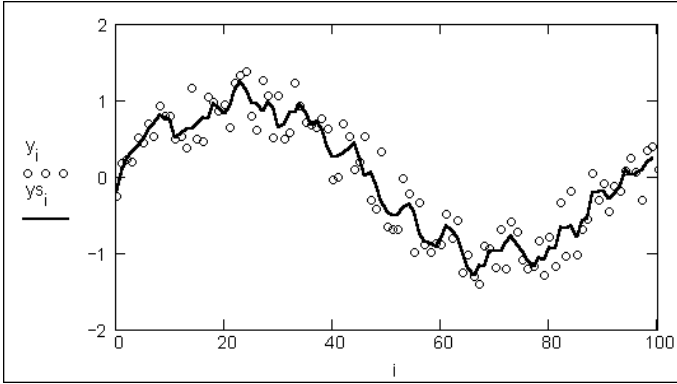


Рис. 8.20. Пример нелинейного сглаживания по семи точкам

8.5.4. Вейвлет- и вейвлет-медианное сглаживание

Сглаживание данных можно выполнить и с помощью новейших и пока не слишком известных вейвлет-преобразований. Здесь мы рассмотрим эту возможность, используя только встроенные в ядро Mathcad функции *wave* и *iwave*. Они реализуют работу с вейвлетами Добеши четвертного порядка. Это отнюдь не гладкие вейвлеты и сглаживаемые с их помощью функции имеют признаки фрактальных структур, что, конечно, не способствует наилучшему сглаживанию. Поэтому наряду со сглаживанием вейвлетами разумно применять дополнительное сглаживание каким-либо из описанных выше методов.

Рисунок 8.21 показывает применение вейвлет- и вейвлет-медианного сглаживания на примере сглаживания нестационарного сигнала в виде сильно зашумленной синусоиды с падающей во времени частотой.

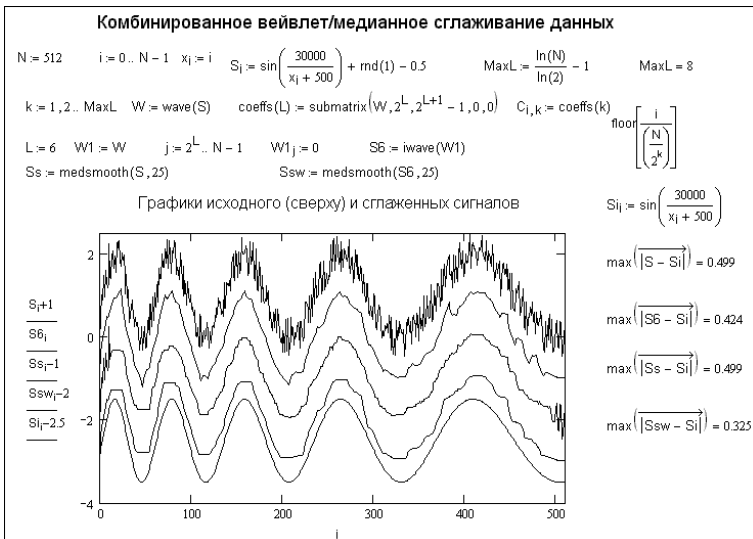


Рис. 8.21. Пример вейвлет- и вейвлет-медианного сглаживания

Само по себе прямое и обратное вейвлет-преобразования свойствами сглаживания не обладают, поскольку идеально восстанавливают исходный произвольный сигнал независимо от его формы. Для осуществления вейвлет-сглаживания надо уменьшить число используемых при синтезе сигнала вейвлет-коэффициентов. В нашем случае это достигается снижением уровня преобразований с максимального уровня 8 до 6.

На рис. 8.21 показаны следующие графики (сверху вниз):

- исходный зашумленный сигнал;
- сглаженный вейвлетами сигнал с уровнем преобразования 6;
- сигнал, сглаженный только методом скользящей медианы с помощью функции `medsmooth`;
- сигнал, сглаженный комбинированным методом;
- исходный сигнал без шума.

Можно заметить, что как вейвлет-, так и медианное сглаживания сами по себе достаточно эффективны, но не избавляют полностью полученный сигнал от высокочастотных компонент как шума, так и самих вейвлет-коэффициентов. Комбинированный метод сглаживания дает лучшие результаты — высокочастотные «пички» при нем сглажены. Объективную оценку сглаживания может дать только оценка соответствия результатов сглаживания с результатами представления исходной зависимости. В правой части рис. 8.21 представлены результаты вычисления суммарной (по всем точкам) погрешности между сглаженной и идеальной кривыми сигнала. Хорошо видно, что минимальную погрешность обеспечивает комбинированный вейвлет-медианный метод сглаживания.

8.5.5. Комбинированное сглаживание вейвлетами и фильтром «скользящее среднее»

Рассмотрим еще один пример комбинированного сглаживания, который представлен на рис. 8.22. Тут задан нестационарный сигнал еще более сложной зависимости — в виде сильно зашумленной синусоиды с растущими во времени амплитудой и частотой. Вначале сигнал обрабатывается с помощью вейвлет-преобразований с уровнем 7 (при максимальном уровне 9), а затем явно реализованным фильтром типа «скользящее среднее».

Высокая степень сглаживания в данном примере вполне очевидна. Можно отметить, что сглаживание является одновременно и эффективным средством очистки сигнала от шума.

Примененная реализация фильтра не учитывает необходимость смены выражений для сглаживания в конечных точках, что ведет к росту погрешности вблизи от них. Впрочем, при использовании этого метода при поточной информации данный недостаток не сказывается.

8.5.6. Вейвлет-сплайновое сглаживание

Сплайны также могут использоваться наряду с вейвлетами для сглаживания сложных зависимостей. На рис. 8.23 представлен документ, в котором сглаживанию подвергается нестационарный сигнал в виде искаженной (возведением в кубическую степень) синусоиды с нарастающим периодом и амплитудой, имеющей шумовую компоненту. Такой сигнал имеет характерную уплощенную область в моменты перехода синусоиды через ноль.

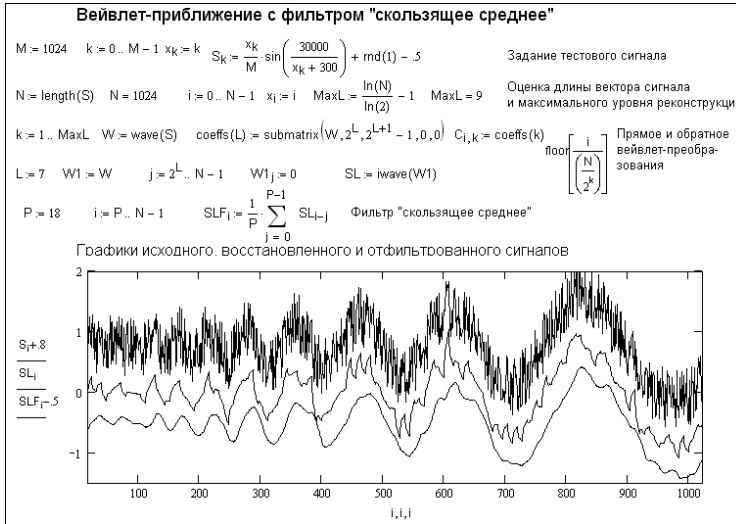


Рис. 8.22. Пример сглаживания нестационарного сигнала вейвлетами и фильтром «скользящее среднее»

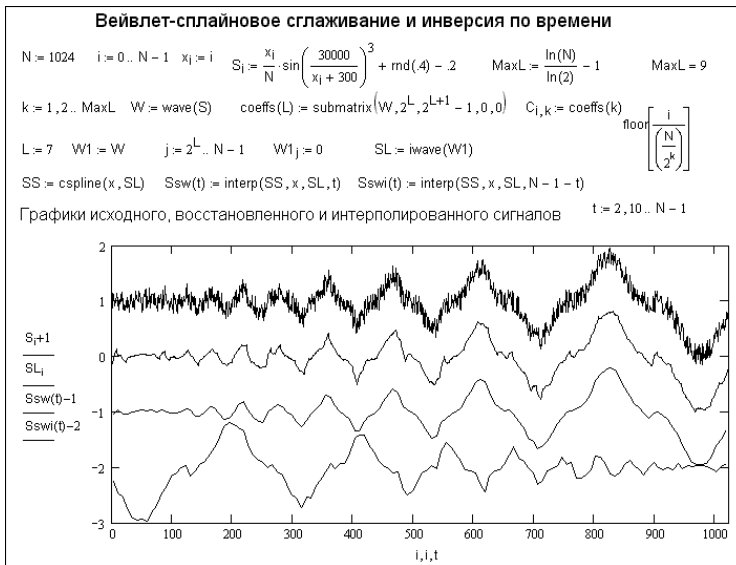


Рис. 8.23. Пример вейвлет-сплайнового сглаживания нестационарного сигнала

Нетрудно заметить, что эта особенность прослеживается и после сглаживания. Для эффективного сглаживания сплайнами нужно правильно подобрать начальную точку отсчета и шаг вывода кривой. Рисунок 8.23 показывает еще одну иногда полезную возможность — инверсию сглаженного сигнала во времени.

В заключение еще раз отметим, что в данных примерах использовались довольно грубые вейвлеты, с явно выраженной фрактальной структурой. В главе 12 мы продолжим рассмотрение методов сглаживания, основанных на применении «лучших», а точнее, более гладких вейвлетов.

8.6. Предсказание зависимостей

8.6.1. Предсказание методом Бурга «чистых» зависимостей

Весьма интересной является включенная в систему Mathcad (начиная с шестой версии) функция предсказания (экстраполяции) $\text{predict}(\text{data}, k, N)$, где data — вектор данных, k — число точек с конца вектора, используемых для предсказания, и N — число точек предсказания. Она по ряду заданных равномерно расположенных точек позволяет рассчитать некоторое число N последующих точек, то есть, по существу, осуществляет экстраполяцию произвольной (но достаточно гладкой и предсказуемой) зависимости. Эта функция в новых версиях Mathcad реализует новый и современный авторегрессионный метод Бурга, впервые описанный в 1996 году. Улучшенные функции предсказания вы найдете в описании пакета расширения Numeric Recipes и пакета Signal Processing [11].

На рис. 8.24 показан фрагмент документа Mathcad с примером применения функции предсказания (экстраполяции) для «чистой» (т. е. не засоренной шумами и погрешностями) аналитической зависимости. Исходная функция (синусоида с экспоненциальным нарастанием) задана своими 100 точками (тонкая линия). Область предсказания определена еще на 400 точек (жирные линии для 19 и 37 последних точек). Кроме того, построена линия для положения вершущек экспоненциально нарастающего сигнала. Видно, что при $k = 19$ предсказание превосходное даже для всех 400 точек, тогда как при $k = 37$ предсказание хорошо для 300 точек, а затем наступает «разболтка» и точки предсказания дают заметно большую амплитуду предсказанных значений, чем это есть на самом деле у предсказываемой функции.

Функция предсказания обеспечивает высокую точность при монотонных исходных функциях или исходных функциях, представляемых полиномом невысокой степени, и при достаточно большом числе исходных точек. Показанный на ри-

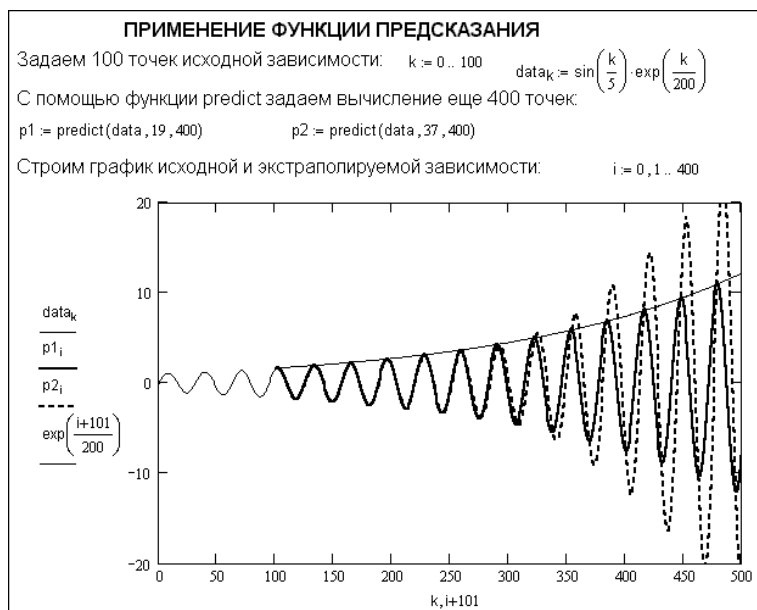


Рис. 8.24. Применение функции предсказания (экстраполяции)

сунке фрагмент документа иллюстрирует, что функция предсказания неплохо предсказывает функцию, содержащую колебательную затухающую (или нарастающую) синусоидальную составляющую. В общем случае предсказание выглядит очень неплохо для аналитических зависимостей, содержащих гладкие компоненты, синусоиды и косинусоиды, экспоненты и так далее. Порою, например при 19 точках, предсказание выглядит как сенсационное — многие исследователи в области предсказания зависимостей рады получить прогноз на несколько точек вперед.

Тем не менее слишком обольщаться результатами прогноза в данном случае не стоит. Для хорошего прогноза надо тщательно подбирать число k , иначе качество прогноза может сильно ухудшиться. При этом кривая прогноза может как резко затухать, так и давать неожиданные всплески.

8.6.2. Предсказание зашумленных зависимостей

Большой интерес представляет применение функции предсказания в случае зашумленных зависимостей. В этом случае точность предсказания неизбежно резко ухудшается. Некоторые специалисты считают, что линейные авторегрессионные методы сглаживания вообще неприменимы для зашумленных зависимостей. Но это не так.

Есть достаточно эффективный способ повысить точность предсказания и в этом случае — для этого надо предварительно сгладить зашумленную зависимость. Пример такого подхода дан на рис. 8.25. Тут исходная зависимость в виде экспоненциально затухающей синусоиды зашумлена шумом с равномерным распределением, создаваемым генератором случайных чисел (функция `rnd`). Для сглаживания используется функция `medsmooth`, реализующая сглаживание по методу скользящей медианы.

Еще один пример прогноза по зашумленной сложной зависимости представлен на рис. 8.26. Здесь шумовая компонента имеет нормальный закон распределения шумовой компоненты. Исходная функция представлена синусоидой, возведенной

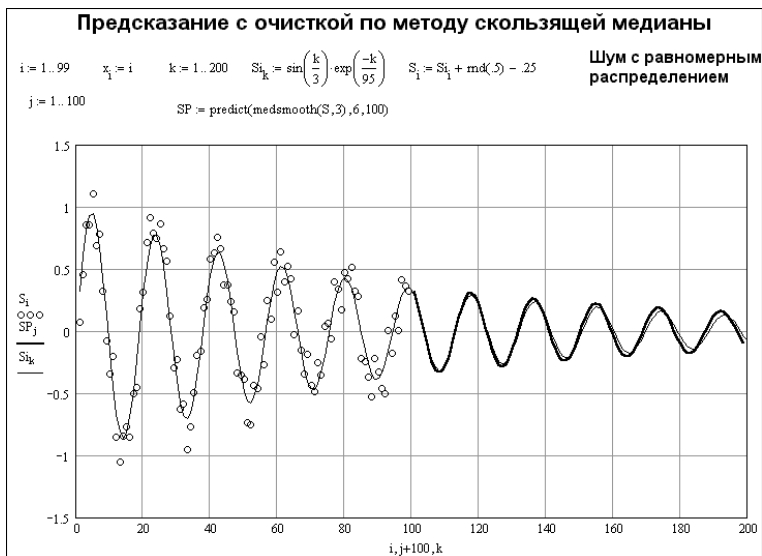


Рис. 8.25. Применение функции предсказания после сглаживания данных

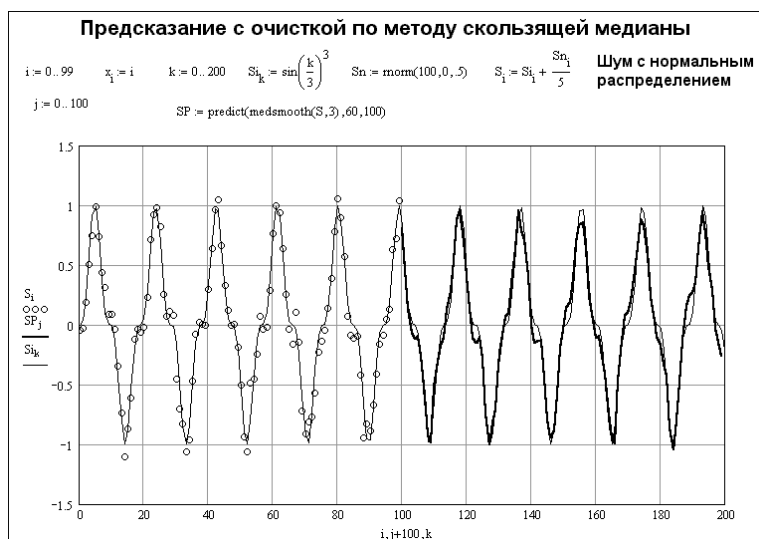


Рис. 8.26. Применение функции предсказания после сглаживания данных

в кубическую степень. Интересно, что особенность формы такой функции (горизонтальные «ступеньки» при переходе через 0) явно заметна и в кривой прогноза.

В разделе 11.6.2 будет рассмотрен еще один интересный пример применения функции предсказания для моделирования закона Мура с использованием предварительно сплайновой интерполяции данных. Достаточно подробные данные по работе функции предсказания можно найти в справке и в «быстрых шпаргалках» системы Mathcad 12.

8.7. Аппроксимация, сглаживание и экстраполяция рядами Фурье

8.7.1. Приближение данных рядом Фурье

Помимо полиномов существует и такой единообразный метод представления самых различных зависимостей, как ряд Фурье [34, 35]. Когда коэффициенты Фурье вычисляются методом прямоугольников, то ряд Фурье фактически приближает исходную зависимость (облако ее узловых точек) с наименьшей среднеквадратической погрешностью. Другими словами, он является вариантом регрессии.

На рис. 8.27 представлено начало документа, в котором реализовано приближение зашумленных данных рядом Фурье. Показано задание исходной зависимости $f(x)$, вектора ее значений Y_0 и вектора зашумленных данных Y .

Далее на рис. 8.28 представлено вычисление коэффициентов Фурье методом прямоугольников (программные модули) и построение ограниченного числом гармоник $n_{max} = 10$ ряда Фурье. Нетрудно заметить, что за исключением концевых точек ряд Фурье неплохо приближает облако точек исходной зависимости. Поскольку мы имеем функцию приближения в явном виде, можно говорить о полноценной аппроксимации, равно как и о довольно эффективном сглаживании данных. Из рис. 8.28 ясно, что спектр Фурье быстро затухает, что позволяет отбросить вышние гармоники и тем самым обеспечить эффективное сглаживание данных.



Рис. 8.27. Начало документа с примером приближения данных рядом Фурье

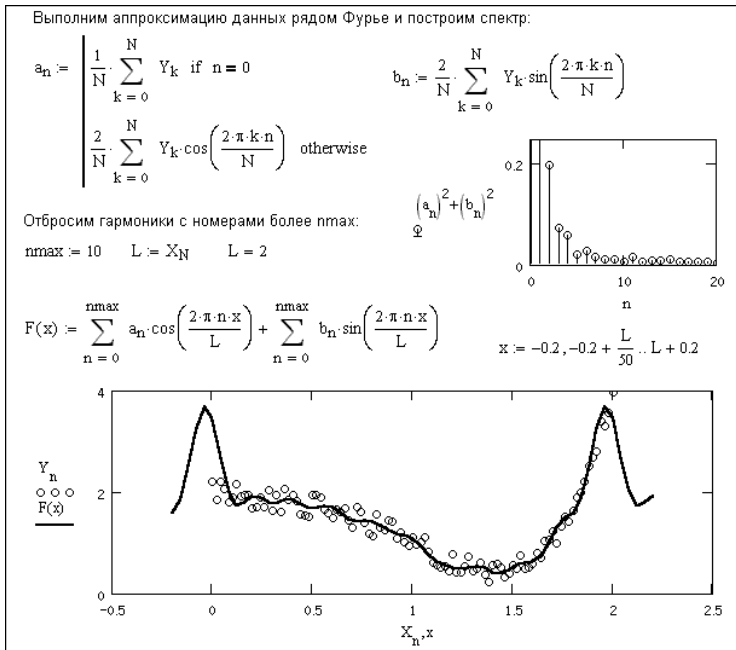


Рис. 8.28. Приближение данных рядом Фурье

Большой «ложкой дегтя» в таком приближении оказывается злополучный эффект Гибса. Он проявляется как плохое сходжение ряда Фурье на концах отрезка приближения (в нашем случае в точках с $x = 0$ и $x = 2$). При этом наблюдается совершенно недопустимый выброс кривой аппроксимации и волнообразный ее ход даже в центре графика.

8.7.2. Улучшение сходимости приближения рядом Фурье

Одним из главных способов улучшения сходимости рядов Фурье является устранение разрывов исходных функций в начале и в конце интервала приближе-

ния. Если данные представлены отдельными точками (узлами), то для этого достаточно использовать итерационную формулу, представленную в начале рис. 8.29.

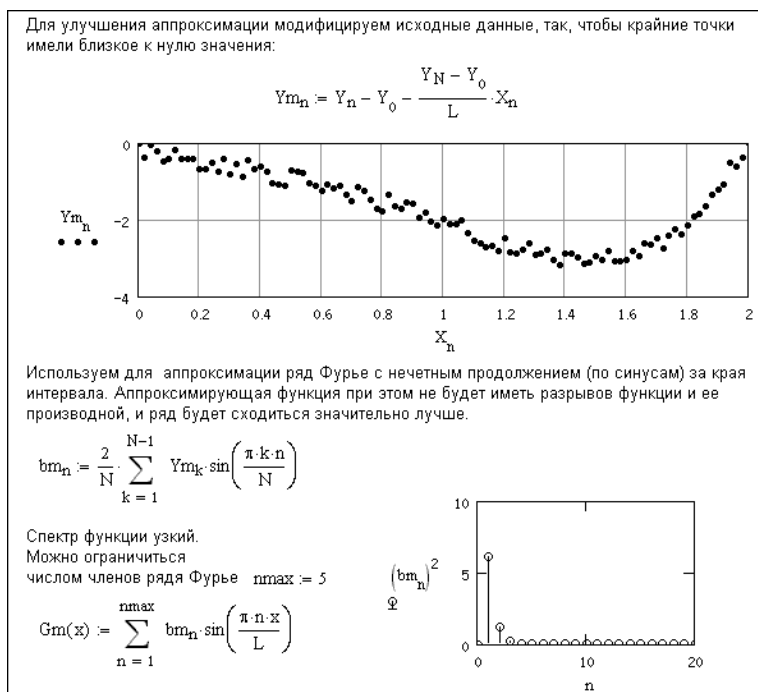


Рис. 8.29. Часть документа с примером приближения данных рядом Фурье с устранением разрывов в начале и конце отрезка приближения

В результате пересчета точек исходной зависимости получается новая зависимость с нулями в конце интервала приближения. Такую зависимость можно представить рядом Фурье с только синусными членами, что дает сразу несколько важных достоинств:

- заметно сужается спектр Фурье, что позволяет использовать меньшее число гармоник;
- ряд Фурье быстро сходится;
- проявление эффекта Гиббса резко уменьшается;
- появляется возможность экстраполяции (продолжения по синусам).

Так, в нашем случае оказалось достаточно ограничить число гармоник ряда Фурье значением 5 (на основании приведенной спектрограммы).

8.7.3. Эффективное приближение данных рядом Фурье

Теперь после преобразования данных можно получить весьма эффективное приближение их рядом Фурье с малым числом гармоник — рис. 8.30 (верхний график).

Используя далее обратную формулу преобразования, можно вернуть данным исходные значения и получить кривую регрессии, показанную на рис. 8.30 снизу. Нетрудно заметить, что полученная аппроксимирующая функция вполне удовлет-

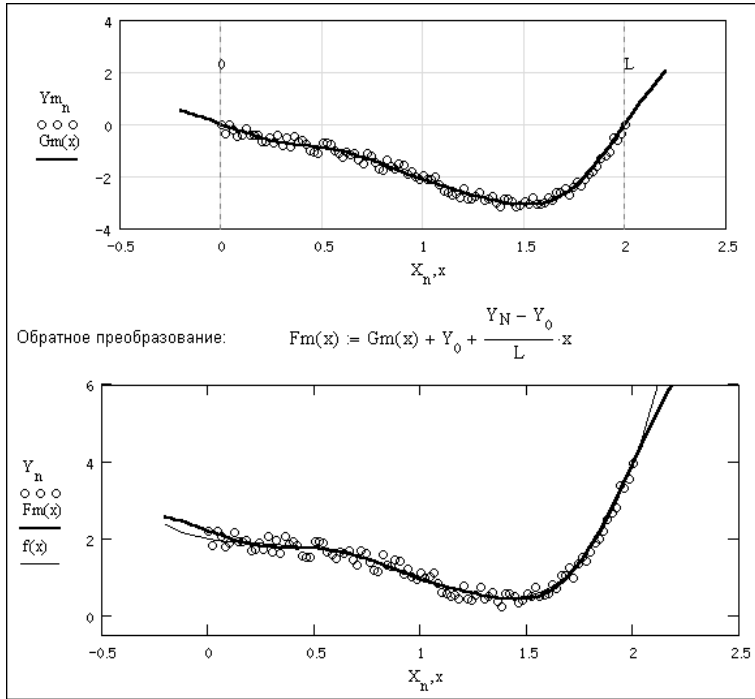


Рис. 8.30. Окончание документа с примером приближения данных рядом Фурье (сверху показано приближение преобразованной, а снизу восстановленной исходной зависимости)

ворительно справляется с экстраполяцией. Для демонстрации возможностей приближения на рис. 8.30 показаны облако точек исходной зависимости, кривая приближения (жирная) и чистая (без шумов) исходная зависимость.

8.8. Финансово-экономические расчеты

8.8.1. Функции финансово-экономических расчетов

В последние версии системы Mathcad введен пакет функций для выполнения финансово-экономических расчетов. В этом разделе мы оценим их возможности.

Любая из финансово-экономических функций предназначена для вычисления значения одного из параметров финансово-экономических расчетов по заданным значениям других параметров. К основным параметрам относятся:

- pv — сумма вклада в настоящее время;
- fv — сумма вклада в будущем;
- $rate$ — процент начислений;
- $crate$ — фиксированный процент начислений на период;
- $nper$ — число периодов;
- $cnper$ — число составных периодов;
- $start$ — начало периода;
- end — конец периода.

Ниже дан список финансово-экономических функций системы Mathcad 2001 (необязательные параметры указаны в квадратных скобках):

- $\text{cnper}(\text{rate}, \text{pv}, \text{fv})$ — число составных периодов для получения будущего значения вклада при заданных текущем значении вклада и проценте начислений;
- $\text{crate}(\text{nper}, \text{pv}, \text{fv})$ — фиксированный процент начислений на период, необходимый для прироста вклада от настоящего значения до будущего значения при заданном числе составных периодов;
- $\text{cumint}(\text{rate}, \text{nper}, \text{pv}, \text{start}, \text{end}, [\text{type}])$ — совокупный процент, который платится по займу;
- $\text{cumprn}(\text{rate}, \text{nper}, \text{pv}, \text{stall}, \text{end}, [\text{type}])$ — совокупная сумма, которая платится по займу;
- $\text{eff}(\text{rate}, \text{nper})$ — эффективная ежегодная сумма, которая платится по займу;
- $\text{fv}(\text{rate}, \text{nper}, \text{pmt}, [[\text{pv}], [\text{type}]])$ — будущее значение вклада;
- $\text{fvadj}(\text{prin}, v)$ — будущее значение вклада при ряде составных процентных ставок;
- $\text{fvc}(\text{rate}, v)$ — будущее значение вклада при фиксированных процентных ставках;
- $\text{ipmt}(\text{rate}, \text{per}, \text{nper}, \text{pv}, [[\text{fv}], [\text{type}]])$ — процентная ставка на период;
- $\text{irr}(v, [\text{guess}])$ — внутренний процент возврата для ряда денежных вливаний;
- $\text{mirr}(v, \text{fin_rate}, \text{rein_rate})$ — модифицированная внутренняя ставка возврата для ряда денежных взносов;
- $\text{nom}(\text{rate}, \text{nper})$ — номинальная процентная ставка;
- $\text{nper}(\text{rate}, \text{pmt}, \text{pv}, [[\text{fv}], [\text{type}]])$ — число периодов;
- $\text{npv}(\text{rate}, v)$ — нынешнее значение вклада;
- $\text{pmt}(\text{rate}, \text{nper}, \text{pv}, [[\text{fv}], [\text{type}]])$ — плата за период займа;
- $\text{ppmt}(\text{rate}, \text{per}, \text{nper}, \text{pv}, [[\text{fv}], [\text{type}]])$ — плата на общую сумму данного периода;
- $\text{pv}(\text{rate}, \text{nper}, \text{pmt}, [[\text{fv}], [\text{type}]])$ — вклад, который должен быть внесен в настоящее время при устойчивой плате и фиксированном проценте ставки;
- $\text{rate}(\text{nper}, \text{pmt}, \text{pv}, [[\text{fv}], [\text{type}], [\text{guess}]])$ — процентная ставка за период ренты.

К сожалению, определения финансово-экономических величин и расчетные соотношения между ними заметно различаются у нас и за рубежом, что чревато серьезными ошибками.

8.8.2. Примеры финансово-экономических расчетов

На рис. 8.31—8.33 представлены фрагменты документа Mathcad с практически примерами выполнения таких расчетов.

Обратите внимание, что получаемые суммы положительны, а отдаваемые отрицательны. Приведенные расчеты можно использовать на практике только после тщательной проверки. Зачастую гораздо проще составить документы, в которых такие расчеты выполняются по принятым у нас формулам и алгоритмам.

Финансово-экономические расчеты по встроенным функциям

Сколько пройдет годовых периодов чтобы при проценте годовых 10% начальный вклад в 1000\$ возрос до конечного значения в 2000\$?

$$\text{rate}_- := 10 \cdot \% \quad \text{pv}_- := 1000 \quad \text{fv}_- := 2000 \quad \text{cnpv}(\text{rate}_-, \text{pv}_-, \text{fv}_-) = 7.273$$

Какой процент годовых нужен, чтобы через 7 годовых периодов вклад в 1000\$ вырос до 2000\$?

$$\text{nper}_- := 7 \quad \text{pv}_- := 1000 \quad \text{fv}_- := 2000 \quad \text{crate}(\text{nper}_-, \text{pv}_-, \text{fv}_-) = 0.104$$

Некто приобрел в рассрочку дом стоимостью $\text{pv}=150000$ \$ и намерен выкупить его за 30 лет. Сколько надо ему внести в первый год при проценте годовых $\text{rate}=8\%$, чтобы с учетом накоплений вклада дом был выкуплен.

$$\text{rate}_- := 8.3 \cdot \% \quad \text{pv}_- := 150000 \quad \text{nper}_- := 30 \cdot 12 \quad \text{start} := 1 \quad \text{end} := 12$$

$$\text{cumint}\left(\frac{\text{rate}_-}{12}, \text{nper}_-, \text{pv}_-, \text{start}, \text{end}\right) = -12405.768$$

При тех же условиях каковы должны быть ежемесячные отчисления для выкупа дома?

$$\text{cumprn}\left(\frac{\text{rate}_-}{12}, \text{nper}_-, \text{pv}_-, \text{start}, \text{end}\right) = -1180.354$$

Процент накопления $\text{rate}=10\%$, а число периодов накопления $\text{nper}=24$. Чему равен эффективный процент накопления?

$$\text{rate}_- := 10 \cdot \% \quad \text{nper}_- := 24 \quad \frac{\text{eff}(\text{rate}_-, \text{nper}_-)}{\%} = 10.494$$

Рис. 8.31. Примеры финансово-экономических расчетов (набор 1)

Вкладчик решил вкладывать в банк ежемесячно по 100 \$. Сколько долларов окажется на вкладе через год, если процент годовых равен 6%?

$$\text{fv}\left(\frac{0.06}{12}, 12, -100\right) = 1233.556$$

Сколько денег будет на вкладе при тех же условиях, если сделан начальный вклад в 1000\$?

$$\text{fv}\left(\frac{0.06}{12}, 12, -100, -1000\right) = 2295.234$$

Вкладчик решил сделать в течении трех лет вклады в 1000, 1200 и 1500\$. При проценте годовых 8% каково будет будущее значение вклада?

$$\text{rate}_- := 8 \cdot \% \quad \text{V} := \begin{pmatrix} 1000 \\ 1200 \\ 1500 \end{pmatrix} \quad \text{fvc}(\text{rate}_-, \text{V}) = 3962.4$$

Вкладчик положил в банк начальную сумму $\text{prin}=1000$ \$ на три года. Если проценты годовых равны 6, 7 и 8%, то какая сумма будет на вкладе через три года?

$$\text{prin} := 1000 \quad \text{sheed} := \begin{pmatrix} 0.06 \\ 0.07 \\ 0.08 \end{pmatrix} \quad \text{fvadj}(\text{prin}, \text{sheed}) = 1224.936$$

Рис. 8.32. Примеры финансово-экономических расчетов (набор 2)

ВНИМАНИЕ Особую осторожность при проведении финансово-экономических расчетов надо проявлять в условиях экономической нестабильности. Такие события, как «черный вторник», «августовский кризис», внезапная деинфляция денежных знаков или просто резкие колебания курса, не учитываются в расчетах и способны свести их практическую ценность на нет.

Вкладчик внес в конце первого периода сумму в 10000\$, а затем снимал в конце последующих периодов суммы в 3000, 4200 и 6800\$. Если процент равен 10%, то какая сумма окажется на данный момент?

$$\text{prate} := 10 \cdot \% \quad \text{CF} := (0 \quad -10000 \quad 3000 \quad 4200 \quad 6800)^T \quad \text{npv}(\text{prate}, \text{CF}) = 1080.403$$

Вкладчик хочет получать $\text{pmt}=200\$$ в конце каждого месяца при проценте годовых 7,5% на протяжении 15 лет. Какой начальный вклад pv для этого нужно внести?

$$\text{pv}\left(\frac{0.075}{12}, 12 \cdot 15, 200\right) = -21574.685$$

Вкладчик внес вклад 10000\$ и намерен погашать его на протяжении 48 периодов суммами в размере 250\$. Какой процент начислений за каждый период должен обеспечить банк, если в конце всех периодов на счету должна остаться сумма в 1000\$?

$$\begin{aligned} \text{pv}_- &:= 10000 & \text{nper}_- &:= 48 & \text{fv}_- &:= 1000 & \text{pmt}_- &:= -250 \\ r &:= \text{rate}(\text{nper}_-, \text{pmt}_-, \text{pv}_-, \text{fv}_-) & r &= 0.004 & r_y &:= r \cdot 12 & r_y &= 0.052 \end{aligned}$$

Вкладчик имеет в банке сумму $\text{pv}=15000\$$. Процент годовых начислений $\text{rate1}=8.75$. В конце каждого из $\text{nper}=10$ периодов он намерен снимать некоторую сумму до полной ликвидации вклада - $\text{fv}=0$. Какую сумму он должен снимать?

$$\begin{aligned} \text{nper}_- &:= 10 & \text{rate1}_- &:= 8.75 \cdot \% & \text{pv}_- &:= 15000 & \text{fv}_- &:= 0 \\ \text{pmt}(\text{rate1}_-, \text{nper}_-, \text{pv}_-, \text{fv}_-) & & & & & & & = -2311.64 \end{aligned}$$

Рис. 8.33. Примеры финансово-экономических расчетов (набор 3)

Глава 10. Программирование и интеграция

Как быть, если той или иной системе, в нашем случае Mathcad, недостает некоторых средств, желательных для решения тех или иных специфических задач? Первое, что приходит в голову, — заняться программированием этих задач или применить какие-то более подходящие для их решения программные средства.

Ранние версии Mathcad вообще не имели средств обычного программирования. Система имела средства *визуально-ориентированного программирования* в виде шаблонов математических операций, из которых составлялись математические выражения. Между тем в ряде случаев, когда требуется скомпоновать ряд разнохарактерных операций в единый программный блок, они просто необходимы, поскольку, например, ранжированные переменные Mathcad не обеспечивают полноценной замены циклов. В Mathcad 8—12 минимально необходимые средства программирования включены в систему.

10.1. Создание простейших программных конструкций

10.1.1. Задание операторов пользователя

Начнем обсуждение программных средств Mathcad с функций пользователя, которые являются первым признаком программирования. Однако Mathcad поддерживает еще одну интересную возможность — задание новых *операторов* пользователя. Такой оператор задается практически так же, как функция пользователя, но вместо имени выбирается какой-либо подходящий знак. Например, можно задать оператор деления в виде:

$$\div(A, B) := \frac{A}{B}$$

После этого новым оператором можно пользоваться:

- $\div(6, 2) = 3$ — пример применения новой *функции* деления;
- $6\div 2 = 3$ — пример применения нового *оператора* деления.

При кажущейся простоте такого задания здесь есть проблемы. Встроенные в систему операторы нельзя переопределять, поэтому набор доступных знаков для обозначения новых операторов ограничен. К примеру, нельзя задать новый оператор деления знаком / (он уже использован), но можно взять знак \div , поскольку этот символ как знак деления системой не используется.

Вторая проблема связана с вводом символа нового оператора. Скорее всего, напрямую ввести его будет нельзя. Придется воспользоваться типовыми приемами ввода новых символов в документы Windows, например, используя приложение, выдающее таблицу символов и предоставляющее возможность экспорта символа из этой таблицы в документ другого приложения (в нашем случае — в документ Mathcad).

Можно также воспользоваться подходящим знаком из набора Extra Math Symbols (Дополнительные математические символы), имеющегося в составе быстрых «шпаргалок» (QuickSheets), доступ к которым предоставляет Центр ресурсов (см. главу 11). Для переноса знака в документ можно скопировать его в буфер об-

мена командой Copy (Копировать), а затем вставить в документ командой Paste (Вставить). Знак можно дополнить. Например, если вы хотите создать оператор с именем °C, то возьмите за основу знак ° и добавьте C. Для получения оператора °F придется проделать все заново, начав с ввода набора Extra Math Symbols.

Для работы с операторами пользователя служит панель математических знаков Evaluation (см. рис. 1.4 справа). В ней для этого имеется четыре кнопки:

- fx — использование унарного оператора пользователя;
- xf — использование инфиксного оператора пользователя;
- xfy — использование бинарного оператора пользователя;
- x^fy — использование бинарного оператора пользователя в форме «дерева».

Итак, после того как оператор задан, его можно использовать и как функцию, и как оператор. При использовании нового оператора надо вывести его шаблон с помощью панели математических знаков Evaluation. В нашем примере следует щелкнуть на кнопке xfy этой панели, которая выводит особый шаблон с тремя местами ввода. Введите операнды, например 6 и 2, в крайние места ввода, а символ оператора — в среднее. Поставив после этой конструкции знак равенства, увидите результат — число 3.

Ниже дан пример, когда используется бинарный оператор типа «дерево»:

$$\begin{array}{c} \div = 3 \\ \swarrow \quad \searrow \\ 6 \quad 2 \end{array}$$

Разумеется, можно задавать и другие операторы, в том числе для работы с одним операндом. Так, например, можно следующим образом определить операторы пересчета температуры по шкале Цельсия в температуру по шкале Фаренгейта:

$$^{\circ}\text{C}(x) := \frac{9}{5} \left| x + 32 \right. \quad ^{\circ}\text{F} := 1$$

Используя затем кнопку xf на панели символов отношения, можно выполнять операцию пересчета в виде:

$$37^{\circ}\text{C} = 98.6 \cdot ^{\circ}\text{F}$$

10.1.2. Задание и применение функций пользователя

Функции пользователя в Mathcad задаются предельно просто и естественно:

Имя_функции (Список_параметров) := Тело_функции

Имя функции составляется по правилам задания имен переменных. Список параметров содержит перечень переменных, значения которых под именем переменных передаются в тело функции. Примеры задания функций пользователя:

$$\begin{aligned} \sin 3(x) &:= \sin(x)^3 \\ \text{dist}(x, y) &:= \sqrt{x^2 + y^2} \end{aligned}$$

Переменные в списке параметров функции пользователя являются локальными переменными. На их место могут быть записаны константы или выражения, которые передаются в тело функции. Но за пределами функции пользователя эти переменные могут иметь иные значения или быть не определенными.

10.2. Модульное программирование

10.2.1. Задание программных модулей

Вплоть до появления последних версий системы Mathcad возможности программирования в них были крайне ограниченными. Фактически система Mathcad допускала построение лишь линейных программ, реализуя функциональное программирование, в основе которого лежит понятие функции. Функция `if` и ранжированные переменные в отдельных случаях могли заменить условные выражения и циклы, но с существенными ограничениями. Отсутствовала возможность создания завершенных программных модулей.

Возможность задания *программных блоков (модулей)* появилась в версии Mathcad PLUS 6.0 и в расширенном варианте поддерживается в последующих версиях Mathcad 8–12. Средства программирования сосредоточены в палитре программных элементов, показанной на рис. 10.1 на фоне фрагмента документа Mathcad.

ЗАДАНИЕ ПРОГРАММНЫХ БЛОКОВ

<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> Programming </div> <div style="border: 1px solid black; padding: 5px;"> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Add Line ←</td> <td></td> </tr> <tr> <td>if</td> <td>otherwise</td> </tr> <tr> <td>for</td> <td>while</td> </tr> <tr> <td>break</td> <td>continue</td> </tr> <tr> <td>return</td> <td>on error</td> </tr> </table> </div> <p>Панель программных инструкций</p> <p>Пример задания функции обычным способом:</p> <p>Пример задания функции в виде программного блока:</p>	Add Line ←		if	otherwise	for	while	break	continue	return	on error	$x := 25$ $\sqrt{x} = 5$ $x \leftarrow 12 = 3.464$ \sqrt{x} $x = 25$	<p>Задано значение x</p> <p>Вычислен квадратный корень из x</p> <p>Локально задано $x=12$ и вычислен квадратный корень из $x=12$</p> <p>За пределами программного блока x сохранило значение 25</p> $F(x, y, z) := \frac{x}{x+y+z} + \frac{y}{x+y+z} + \frac{z}{x+y+z}$ <p>$F(2, 3, 5) = 0.588$ $F(1, 5, 3) = 0.563$</p> $FP(x, y, z) := \left \begin{array}{l} a \leftarrow x + y \cdot z \\ \frac{x + y + z}{a} \end{array} \right.$ <p>$FP(2, 3, 5) = 0.588$ $FP(1, 5, 3) = 0.563$</p>
Add Line ←												
if	otherwise											
for	while											
break	continue											
return	on error											

Рис. 10.1. Задание программных модулей

Как видно из рисунка, программный блок в системе Mathcad превратился в самостоятельный модуль, выделяемый в тексте документа жирной вертикальной чертой. Модуль может вести себя как безымянная функция без параметров, но возвращающая результат (см. первый пример — вычисление квадратного корня из числа 12). Программный модуль может выполнять и роль тела функции пользователя с именем и параметрами (см. пример в нижней части рисунка).

Нетрудно заметить (см. палитру программных элементов на рис. 10.1), что набор инструкций для создания программных модулей весьма ограничен и содержит следующие элементы:

- Add Line — создает и при необходимости удлиняет жирную вертикальную линию, справа от которой в местах ввода производится запись программного блока;

- `←` — символ локального (в теле модуля) присваивания;
- `if` — условная инструкция;
- `otherwise` — инструкция иного выбора (обычно применяется с `if`);
- `for` — инструкция задания цикла с фиксированным числом повторений;
- `while` — инструкция задания цикла, действующего до тех пор, пока выполняется некоторое условие;
- `break` — инструкция прерывания;
- `continue` — инструкция продолжения;
- `return` — инструкция возврата;
- `on error` — инструкция обработки ошибок.

10.2.2. Инструкция добавления линий в модуль Add Line

Инструкция `Add Line` выполняет функции расширения программного блока. Расширение фиксируется удлинением вертикальной черты программных блоков или их древовидным расширением. Благодаря этому, в принципе, можно создавать сколь угодно большие программы.

10.2.3. Оператор внутреннего присваивания

Оператор `←` выполняет функции внутреннего (локального) присваивания. Например, выражение `x ← 123` присваивает переменной `x` значение 123. Локальный характер присваивания означает, что такое значение переменной `x` хранится только в теле программного модуля. За пределами тела программы значение переменной `x` может быть неопределенным либо равным значению, которое задается вне программного блока операторами локального (`:=`) или глобального (`≡`) присваивания.

***ПРИМЕЧАНИЕ** Не стоит путать оператор внутреннего присваивания `←` с оператором символического вывода `→`, у которого стрелка направлена в другую сторону. Эти операторы решают совершенно разные задачи.*

10.2.4. Условная инструкция if

Инструкция `if` позволяет строить условные выражения. Она задается в виде:

Выражение `if` Условие

Если `Условие` выполняется, то возвращается значение `Выражения`. Совместно с этой инструкцией часто используются инструкции прерывания `break` и иного выбора `otherwise`.

10.2.5. Инструкция организации цикла for

Инструкция `for` служит для организации циклов с заданным числом повторений. Она записывается в виде:

`for Var Nmin .. Nmax`

Эта запись означает, что выражение, помещенное в расположенное ниже место ввода, будет выполняться для значений переменной `Var`, меняющихся от `Nmin` до `Nmax` с шагом `+1`. Переменную счетчика `Var` можно использовать в исполняемом выражении.

10.2.6. Инструкция организации цикла **while**

Инструкция `while` служит для организации циклов, действующих до тех пор, пока выполняется некоторое условие. Она записывается в виде:

```
while Условие
```

Выполняемое выражение записывается в расположенное ниже место ввода.

10.2.7. Инструкция **otherwise**

Инструкция иного выбора `otherwise` обычно используется совместно с инструкцией `if`. Это поясняет следующая программная конструкция:

$$f(x) := \begin{cases} 1 & \text{if } x > 0 \\ -1 & \text{otherwise} \end{cases}$$

То есть функция $f(x)$ возвращает 1, если $x > 0$, и -1 во всех остальных случаях.

10.2.8. Инструкция прерывания **break**

Инструкция `break` вызывает прерывание выполнения программы. Чаще всего эта инструкция используется совместно с условной инструкцией `if` и инструкциями циклов `while` и `for`, обеспечивая переход в конец тела цикла.

10.2.9. Инструкция **continue**

Инструкция `continue` используется для продолжения работы после прерывания программы. Она также чаще всего используется совместно с инструкциями циклов `while` и `for`, обеспечивая возвращение в точку прерывания и продолжение вычислений.

10.2.10. Инструкция **return**

Особая инструкция `return` прерывает выполнение программы и возвращает значение операнда, стоящего следом за ней. Например, в приведенном ниже случае будет возвращаться значение 0 при $x < 0$.

```
return 0 if x<0
```

10.2.11. Инструкция **on error** и функция **error**

Инструкция `on error` позволяет создавать процедуры обработки ошибок. Эта инструкция задается в виде:

```
Выражение_1 on error Выражение_2
```

Если при выполнении `Выражения_1` возникает ошибка, то выполняется `Выражение_2`. Для обработки ошибок полезна также функция `error(S)`, которая, будучи помещенной в программный модуль, при возникновении ошибки выводит всплывающую подсказку с сообщением, хранящимся в символьной переменной `S`.

10.3. Примеры программирования в среде Mathcad

10.3.1. Простейшие примеры создания программных модулей

Несмотря на скромность набора программных средств, имеющихся в Mathcad, они дают системе именно те возможности, которые ранее попросту отсутствовали: задание специальных функций, задание различных видов циклов (в том числе вложенных), упрощение алгоритмов ряда вычислений и реализацию различных итерационных и рекурсивных процедур. Рекомендуется внимательно изучить фрагмент документа Mathcad, показанный на рис. 10.2 и иллюстрирующий часть этих возможностей.

ПРИМЕНЕНИЕ ПРОГРАММНЫХ БЛОКОВ

Применение инструкции условного выбора if в программном блоке:

$$\text{abs}(x) := \begin{cases} -x & \text{if } x < 0 \\ x & \text{otherwise} \end{cases} \quad \text{abs}(-5) = 5 \quad \text{abs}(5) = 5$$

Применение инструкции цикла for для вычисления суммы и произведения последовательности целых чисел от 1 до n:

$$\text{sum}(n) := \begin{cases} s \leftarrow 0 \\ \text{for } i \in 1..n \\ s \leftarrow s + i \end{cases} \quad \text{sum}(10) = 55 \quad \text{sum}(20) = 210$$

$$\text{prod}(n) := \begin{cases} p \leftarrow 1 \\ \text{for } i \in 1..n \\ p \leftarrow p \cdot i \\ p \end{cases} \quad \text{prod}(3) = 6 \quad \text{prod}(10) = 3.629 \times 10^6$$

Применение инструкций while и break для вычисления факториала:

$$\text{Fact}(n) := \begin{cases} f \leftarrow 1 \\ \text{while } n \leftarrow n - 1 \\ f \leftarrow f \cdot (n + 1) \\ f \end{cases} \quad \text{Fact}(3) = 6 \quad \text{Fact}(10) = 3.629 \times 10^6$$

$$F(n) := \begin{cases} f \leftarrow n \\ \text{while } 1 \\ f \leftarrow f \cdot (n - 1) \\ n \leftarrow n - 1 \\ \text{break if } n = 1 \\ f \end{cases} \quad F(3) = 6 \quad F(10) = 3.629 \times 10^6$$

Рис. 10.2. Примеры задания программных блоков

Обратите особое внимание на пример вычисления факториала. Здесь один программный модуль задается внутри другого. Вообще говоря, для нескольких подмодулей, которые должны выполняться в составе циклов, служит команда Add Line (добавить линию), добавляющая в модуль дополнительную вертикальную черту для подмодуля.

Программный модуль, в сущности, является функцией, но описанной с применением упомянутых программных средств. Она возвращает значение, определяемое последней инструкцией (если не предусмотрено иное с помощью инструкции return). Это значит, что после такого модуля, выделенного как целый блок, можно поставить знак равенства для вывода результата его работы. В блоке могут содержаться любые операторы и функции входного языка системы. Для передачи в блок значений переменных можно использовать переменные документа, которые ведут себя в блоке как глобальные.

Обычно модулю присваивается имя со списком переменных, после которого идет оператор присваивания :=. Переменные в списке являются локальными, и им можно присваивать значения при вызове функции, заданной модулем. Локальный характер таких переменных позволяет использовать для их идентификаторов

те же имена, что и у глобальных переменных документа. Однако лучше этого не делать и назначать разные имена для локальных переменных программных модулей и переменных документа.

10.3.2. Обработка ошибок в программных модулях

На рис. 10.3 показан фрагмент документа Mathcad с примерами применения инструкций `on error` и `return`, а также примером действия функции `error`, задающей вывод всплывающей подсказки при указании мышью на выражение, содержащее ошибку.

ПРИМЕНЕНИЕ ИНСТРУКЦИЙ return И on error

$F(i) := \begin{cases} \text{return "One" if } i = 1 \\ \text{return "Two" if } i = 2 \\ \text{error("No value!!!")} \text{ otherwise} \end{cases}$
 $F(1) = \text{"One"}$
 $F(2) = \text{"Two"}$
 $F(3) = \dots$
No value!!!

$S(x) := \begin{cases} \text{return } 1 \text{ if } x = 0 \\ \frac{\sin(x)}{x} \text{ otherwise} \end{cases}$
 $S(0) = 1$
 $S(1) = 0.841$
 $S(-1) = 0.841$

$y1(x) := \left(\frac{1}{x}\right) \cdot \sin(x)$
 $y(x) := 1 \text{ on error } y1(x)$
 $y(0) = 1$
 $y(1) = 0.841$
 $y(-1) = 0.841$

$f(x) := x^2 + 9$
 $x = -1$
 $R(f, x) := \text{root}(f(x), x)$
 $R(f, 1) = -3i$
ОШИБКА!

$RF(f, x) := \begin{cases} a \leftarrow x \cdot \sqrt{-1} \\ R(f, a) \text{ on error } R(f, x) \end{cases}$

$x = 1$
 $RF(f, x) = -3i$
Ошибок нет

$x = -1$
 $RF(f, x) = 3i$

Рис. 10.3. Применение инструкций `on error` и `return`

В этих примерах надо обратить внимание на два момента. Первый — возможность применения инструкции `on error` вне программного блока для задания функции $y(x) = \sin(x)/x$ с доопределением ее при $x=0$: $y(0) = 1$. Второй — применение этой инструкции для решения уравнения, имеющего комплексные корни. Когда при таком решении задается начальное значение x (действительное число), то функция `root(f, x)` вызывает ошибку. Она исправляется с помощью инструкции `on error` при задании функции $RF(f, x)$, имеющей процедуру обработки ошибок.

Таким образом, инструкция `on error` фактически играет роль процедуры обработки ошибок. Она позволяет создавать программные модули, защищенные от грубых ошибок.

10.3.3. Модуль построения точек в пространстве

На рис. 10.4 показан фрагмент документа Mathcad с примером применения программного модуля для изображения в трехмерном пространстве точек с координатами x , y и $z = f(x, y)$, где $f(x, y)$ — некоторая функция. Точки изображены на фоне параллелепипеда, внутри которого они находятся.

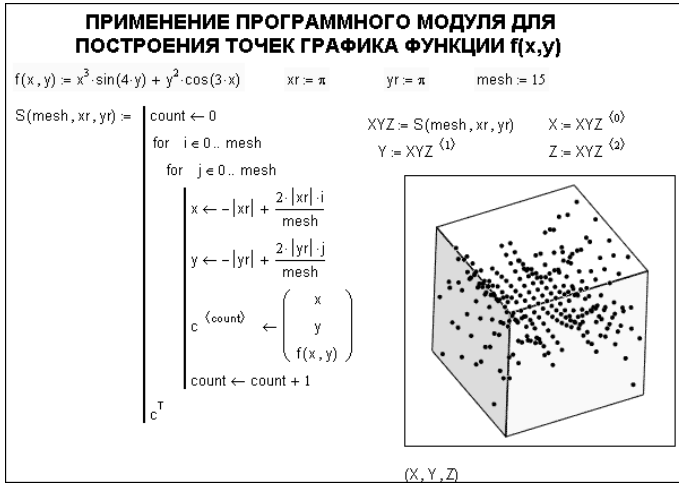


Рис. 10.4. Применение программного модуля для построения в пространстве точек с функционально связанными координатами

Этот пример иллюстрирует возможность применения внутри программного модуля графических функций — в данном случае функции mesh.

10.3.4. Модуль фурье-анализа

Теперь рассмотрим достаточно сложный, но поучительный пример применения программного блока. На рис. 10.5 показана верхняя часть документа Mathcad, в котором с помощью программного модуля вычисляются коэффициенты Фурье для заданной функции (в рассматриваемом примере — пилообразного импульса).

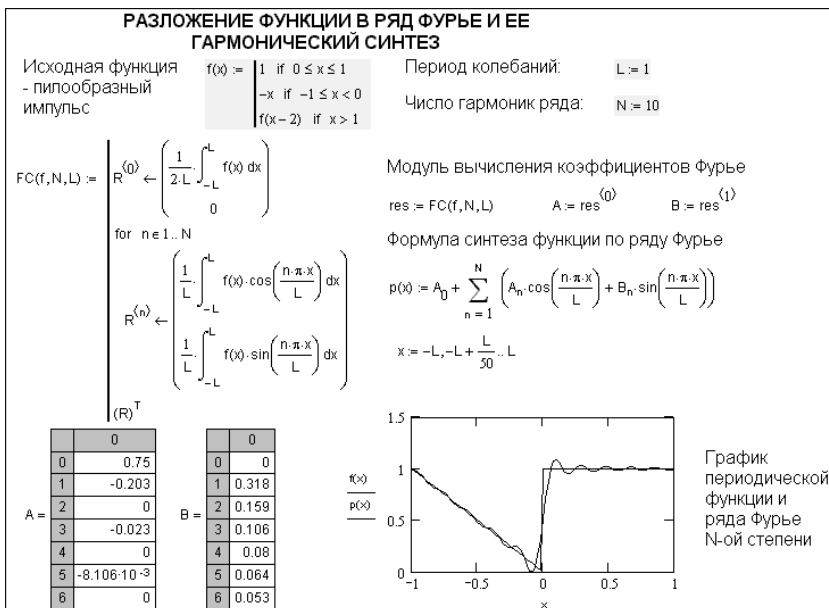


Рис. 10.5. Разложение функции в ряд Фурье

Здесь вычисление коэффициентов ряда Фурье выполнено по их классическим интегральным представлениям. Это хорошо иллюстрирует возможность применения в программных блоках весьма мощных средств входного языка системы (например, блоков с интегралами). Завершается документ реализацией гармонического синтеза сигнала. На рисунке показаны векторы коэффициентов Фурье A и B , формула гармонического синтеза исходной функции и графики исходной функции с ее представлением рядом Фурье с ограниченным числом гармоник N .

Вы получите прекрасные результаты, когда число точек спектрального анализа и синтеза порядка десятка или несколько более. Однако чем больше точек использовать, тем медленнее выполняется программа. Причина этого в том, что она использует довольно медленный алгоритм численного интегрирования. Встроенные в Mathcad функции быстрого преобразования Фурье (БПФ) при большом числе точек (сотни, тысячи) оказываются куда более быстрыми. Их и применяют при решении серьезных задач спектрального анализа и синтеза.

10.3.5. Рекурсивная генерация простых чисел

Простыми называют числа, которые делятся только сами на себя и не могут быть разложены на множители. Все они — нечетные числа, кроме единственного четного числа 2. На рис. 10.6 представлен программный модуль, который реализует рекурсивный алгоритм поиска простых чисел из множества из n натуральных чисел.

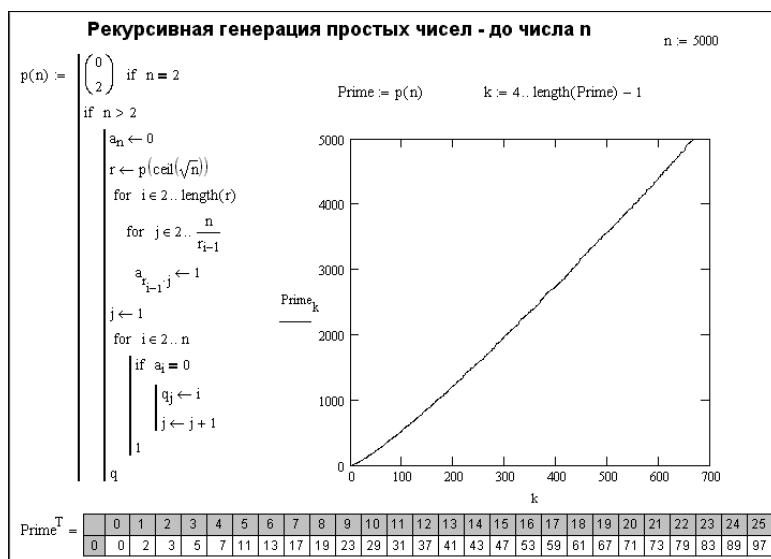


Рис. 10.6. Реализация гармонического синтеза

Интересно отметить, что зависимость значений простых чисел от значений натуральных чисел близка к линейной и представлена на рис. 10.6.

10.3.6. Программа моделирования аттрактора Лоренца

В физике хорошо известны колебательные системы — аттракторы, у которых наблюдается неустойчивый и порой весьма причудливый характер колебаний. Обсуждение таких систем явно выходит за рамки данной книги. Поэтому ограни-

чимся практическим примером. На рис. 10.7 показан начальный фрагмент документа Mathcad с программной реализацией такого аттрактора на основе решения уравнений колебаний конечно-разностным методом.

```

Аттрактор Лоренца

 $\sigma := 10$     $\gamma := 28$     $\beta := \frac{8}{3}$    iterations := 1000

lorenz(X_int, Y_int, Z_int, time_step) := for i ∈ 0..iterations
|
|   x_star ← X_int + (-σ·X_int + σ·Y_int)·time_step
|   y_star ← Y_int + (γ·X_int - Y_int - Z_int·X_int)·time_step
|   z_star ← Z_int + (-β·Z_int + X_int·Y_int)·time_step
|   X_int ← x_star
|   Y_int ← y_star
|   Z_int ← z_star
|   V0,i ← x_star
|   V1,i ← y_star
|   V2,i ← z_star
|
| V

graphit = lorenz(1, 1, 1, .01)   i := 0..iterations
X_ploti = graphit0,i   Y_ploti = graphit1,i   Z_ploti = graphit2,i
    
```

Рис. 10.7. Программа моделирования аттрактора Лоренца (начало)

Следующий фрагмент документа Mathcad с фазовым портретом колебаний в пространстве представлен на рис. 10.8. Сам фазовый портрет задан программным модулем, показанным на рис. 10.7. Нетрудно заметить, что характер колебаний и впрямь довольно причудливый. Вид колебаний можно менять в широких пределах, изменяя исходные параметры (см. рис. 10.7 сверху).

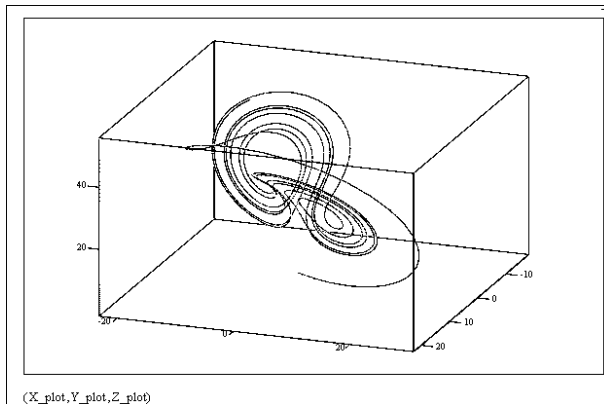


Рис. 10.8. Программа моделирования аттрактора Лоренца (окончание)

10.3.7. Построение фрактала «кукуруза»

К числу занимательных разделов математики относятся фракталы — поверхности, малые части которых как бы повторяют общую структуру поверхности. Фрактальная геометрия — особая область математической графики. Не слиш-

```

ПОСТРОЕНИЕ ФРАКТАЛА - "КУКУРУЗА"

$$x_{n+1} = x_n - h \cdot f(y_n) \quad y_{n+1} = y_n - h \cdot f(x_n) \quad \frac{dx}{dt} = -f(y) \quad \frac{dy}{dt} = -f(x) \quad f(z) := \sin(z + \tan(3 \cdot z))$$

popcorn(N,h,a) := | t ← -1
                  | d ← 2 ·  $\frac{a}{N}$ 
                  | for j ∈ 0..N
                  |   for k ∈ 0..N
                  |     N := 50
                  |     t ← t + 1
                  |     xt ← d · j - a
                  |     h := 0.05
                  |     yt ← d · k - a
                  |     a := 6
                  |     for n ∈ 0..N
                  |       xx ← xt - h · f(yt)
                  |       yy ← yt - h · f(xt)
                  |       t ← t + 1
                  |       xt ← xx
                  |       yt ← yy
                  |       zt ← xt + yt · i
                  |     z := popcorn(N,h,a)
    
```

Рис. 10.9. Программа построения фрактала «кукуруза»

ком вторгаясь в нее, рассмотрим один из документов Mathcad, в котором задано построение фрактальной поверхности, напоминающей ячейки кукурузы (рис. 10.9).

Этот пример приведен не столько для иллюстрации методов построения фракталов, сколько в качестве еще одного довода в поддержку программирования в системе Mathcad. Кроме того, это хороший пример организации рекуррентных вычислений и применения программного оператора цикла.

Фрагмент документа Mathcad с фракталом, определяемым этим программным модулем, представлен на рис. 10.10. Он и впрямь напоминает ячеистую структуру кукурузного початка.

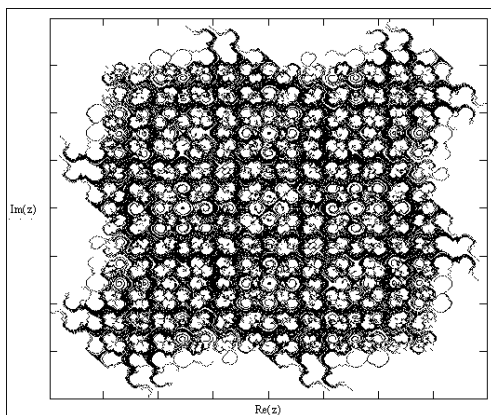


Рис. 10.10. Фрактал «кукуруза», построенный программным модулем с рис. 10.9

10.3.8. Заключительные замечания по программированию

Разумеется, задачи, описанные в этом разделе, могут решаться в системе Mathcad и без использования в явном виде программных средств. Однако эти средства нередко облегчают решение сложных задач, особенно когда имеется описание их программной реализации на каком-либо языке программирования. В этом случае несложно перевести реализацию решения задачи с этого языка на язык программирования системы Mathcad.

Вообще, надо отметить, что проблема включения в документы Mathcad программных блоков решена изящно и красиво — такие блоки подчас просто украшают документы и дают возможность пользоваться всеми средствами не только математически ориентированного входного языка Mathcad, но и классического программирования.

Много интересных и поучительных примеров задания и применения программных модулей можно найти в «быстрых шпаргалках» (QuickSheets) центра ресурсов системы (см. главу 11). Нельзя не отметить, что характер задания программных модулей в Mathcad весьма удачен: модули прекрасно вписываются в документы, выглядят просто и естественно, чего нельзя сказать о программах на обычных языках программирования.

В принципе, в Mathcad 2001i/11/12 есть возможность включения в систему функций пользователя, написанных на языке C или C++. Однако ситуация с этой возможностью очень напоминает нашу крылатую фразу «за что боролись, на то и напоролись» — ведь смысл разработки систем класса Mathcad и заключается в том, чтобы избавить пользователя от программирования на сложных языках высокого уровня. В связи с внедрением в систему Mathcad основных программных конструкций надобность в программировании на языке C++ практически отпала. Точнее, она стала прерогативой «фанатов» системного программирования.

Наконец, следует отметить еще одну важную возможность — применение откомпилированных дополнительных библиотек, расширяющих возможности ядра системы, а также специальных исполняемых модулей, которые обновляют версии системы. Их можно получить у разработчика через Интернет или электронную почту. Эти модули обычно сохраняются в основной папке Mathcad и запускаются как самостоятельные программы. После однократного исполнения они модифицируют текущую версию Mathcad, превращая ее в очередную, более мощную.

10.4. Обзор программных средств, интегрируемых с Mathcad

Программирование на собственном языке программирования является не единственным средством расширения возможностей системы Mathcad 2001. Другим важным средством является интеграция этой системы с другими программными системами.

К примеру, как бы ни были прекрасны возможности Mathcad, в подготовке текстов они намного уступают широко известным текстовым процессорам класса Word. А при хранении табличных данных многим более привычны программы электронных таблиц, например Excel из пакета офисных программ Microsoft Office. Аналогична ситуация с некоторыми другими специальными программами, в частности с графическими. Ниже дается краткая характеристика наиболее популярных программ, которые могут интегрироваться с системой Mathcad.

10.4.1. Текстовые процессоры Word

Текстовые процессоры (редакторы) класса Word [34] — одни из самых популярных офисных программ, предназначенных для подготовки текстов на самом высоком стилистическом и полиграфическом уровне. Они позволяют создавать текстовые документы с множеством стилей, с оперативным контролем граммати-

ки, стилистики и орфографии, с разнообразными табличными и графическими вставками. На фоне этих возможностей средства самой системы Mathcad 2001 в части создания текстов выглядят лишь как зачаточные.

В зависимости от обстоятельств одинаково полезной является как вставка в Mathcad документов, подготовленных в Word, так и вставка фрагментов документов Mathcad в документы текстовых процессоров Word (см. также раздел «Интеграция Mathcad с текстовым процессором Word»).

10.4.2. Электронные таблицы Excel

Табличный процессор Excel, так же как и редактор Word, входит в самый популярный пакет автоматизации офисной деятельности Microsoft Office. Excel — одна из самых мощных и гибких систем обработки электронных таблиц. Эта система может работать не только с двумерными, но и с трехмерными таблицами, представленными листами с двумерными таблицами.

Excel широко используется для подготовки прекрасно иллюстрированных финансово-экономических и иных документов. Этот процессор содержит сотни математических и экономических функций. Применение таблиц для математических расчетов всегда вызывало определенные сомнения (хотя бы в силу весьма специфических названий функций), но, будучи интегрированным с системой Mathcad, табличный процессор Excel открывает колоссальные возможности по обработке объемных и сложных данных (см. также раздел «Интеграция Mathcad с табличным процессором Excel»).

10.4.3. Графическая система Ахум

Ахум 5.0/6.0 — графическая система, созданная фирмой MathSoft — той же, что создала системы класса Mathcad. Прежде всего, эта система ориентирована на построение двумерных и трехмерных графиков, способных удовлетворить самые изысканные требования в области графической визуализации. К примеру, Ахум позволяет строить сложные трехмерные поверхности с функциональной окраской и с одновременным построением их проекции в виде линий равного уровня на плоскость, расположенную под этой поверхностью. Это заметно повышает качество визуализации построений.

Система Ахум также ориентирована на быструю и наглядную обработку данных. Она содержит множество математических и статистических функций и удобный редактор математических формул, обладает большими возможностями по импорту и экспорту данных (см. также подраздел «Интеграция Mathcad с графическим пакетом Ахум 5/6» в разделе «Интеграция Mathcad с другими программными средствами»).

10.4.4. Матричная математическая система MATLAB

MATLAB — одна из самых древних и глубоко почитаемых специалистами математических систем, ориентированных на численные расчеты. Это матричная система, которая была создана фирмой TheMathWorks, Inc.

MATLAB, будучи матричной системой, позволяет быстро и эффективно решать сложные задачи, базирующиеся на матричных математических операциях. Даже одиночное число MATLAB рассматривает как матрицу размером 1×1 . Это обеспечивает большую общность решения задач.

Профессионалы особенно почитают MATLAB (несмотря на некоторую архаичность пользовательского интерфейса) за огромный набор функций, подчас уникальных, а также за возможность поистине неограниченного и весьма простого расширения этой системы. Вам достаточно в ее редакторе задать свою новую функцию, и она «навсегда веков» становится столь же доступной, как и любая предварительно встроенная в систему функция. Благодаря этому свойству на основе системы MATLAB были созданы десятки «новых» математических систем.

Специалистов также привлекает высокая скорость выполнения вычислений — она намного выше, чем при решении подобных задач только средствами Mathcad. Таким образом, объединение возможностей систем Mathcad и MATLAB открывает перед серьезными пользователями компьютерной математики новые и почти необъятные возможности эффективного решения сложных математических задач (см. также подраздел «Интеграция Mathcad с матричной лабораторией MATLAB» в разделе «Интеграция Mathcad с другими программными средствами»).

10.4.5. Система символьной математики Maple

Как ни странно, но с маститым «патриархом» символьной математики — системой Maple — установка связей, как с компонентной системой, в Mathcad не предусмотрена. Возможно, фирма MathSoft посчитала для себя вполне достаточным альянс с фирмой Maple Software, который привел к включению в Mathcad несколько сокращенного ядра символьной математики системы Maple. Таким образом, частичная интеграция с этой системой налично.

По-видимому, ограниченный доступ к ядру Maple — это самый серьезный просчет разработчиков Mathcad. Скорее всего, это было сделано сознательно, ведь полное объединение Mathcad с Maple превратило бы новую систему в необычайно мощный инструмент, владеть которым должны суперпрофессионалы, а не рядовые пользователи. Но тогда имидж системы Mathcad как «системы для всех» был бы поставлен под вопрос, несмотря на большие перспективы роста возможностей в области компьютерной алгебры.

Тем не менее нельзя не считаться с тем, что символьные вычисления Mathcad выполняет с помощью ядра Maple (новые реализации Maple 6/7/8/9 содержит уже более 3000 (!) доступных функций, тогда как в Mathcad их порядка трех сотен). Это сравнение выглядит еще более сильно, если учесть, что функций символьной математики в Mathcad гораздо меньше — их всего-то около трех десятков.

10.5. Интеграция Mathcad с другими программными средствами

10.5.1. Интеграция Mathcad с текстовым процессором Word

Когда пользователь работает над курсовой работой, дипломным проектом, диссертацией или даже книгой, то наверняка большую часть времени он будет тратить на подготовку текста, а не на решение математических задач. Такой пользователь, скорее всего, еще не владеет той или иной математической системой, но наверняка уже всю пользуется текстовым процессором Word.

Существует ряд ограничений на подготовку русскоязычных текстов в системах Mathcad. К примеру, проверку орфографии в среде Mathcad можно использовать

только для англоязычных текстов, поскольку русскоязычный словарь в эту систему не входит.

В принципе (см. раздел «Вставка объектов» в главе 4), в этих системах можно установить объектную связь с текстовым процессором Word, используя команду Object (Объект) меню Insert (Вставка). Однако мы поступим наоборот — учитывая более важную и первичную роль текстового процессора Word для большинства пользователей, рассмотрим его интеграцию с системой Mathcad через объектную связь.

Рис. 10.11 иллюстрирует начальный этап установления объектной связи текстового процессора Word 97 с системой Mathcad. После выбора в Word команды Вставка ► Объект появляется окно объектной связи процессора Word с другими приложениями с перечнем доступных для вставки объектов. В нем необходимо выделить объект Mathcad Document (Документ Mathcad) и установить связь, щелкнув на кнопке ОК.

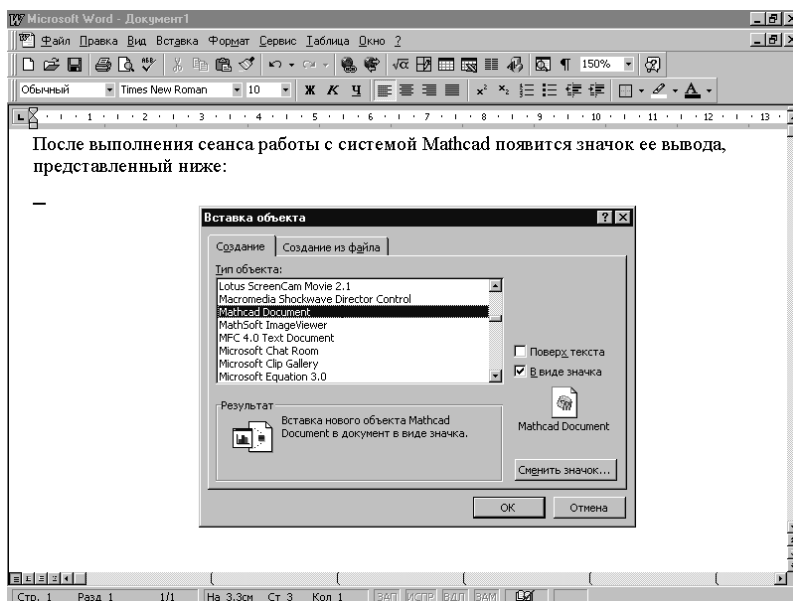


Рис. 10.11. Установка связи текстового процессора Word 97 с математической системой Mathcad

Обратите особое внимание на флажок В виде значка. Если он установлен, объект (документ Mathcad) будет представлен своим схематическим представлением — значком. В противном случае он будет представлен своим содержимым, то есть формулами, графиками и т. д. Установите этот флажок.

После этих операций в текст документа Word будет включен значок математической системы Mathcad. Одновременно появится окно системы Mathcad, в котором можно воспользоваться возможностями этой системы для подготовки нужного математического выражения или графика. Это иллюстрирует рис. 10.12, на котором приведен пример задания и вычисления в символьном виде интеграла. Обратите внимание, что при этом можно вывести палитры математических символов системы Mathcad и пользоваться всеми ее возможностями.

Завершив ввод интеграла и вычисление его символьного значения, можно вернуться в среду текстового процессора Word, закрыв окно Mathcad. Для этого, как

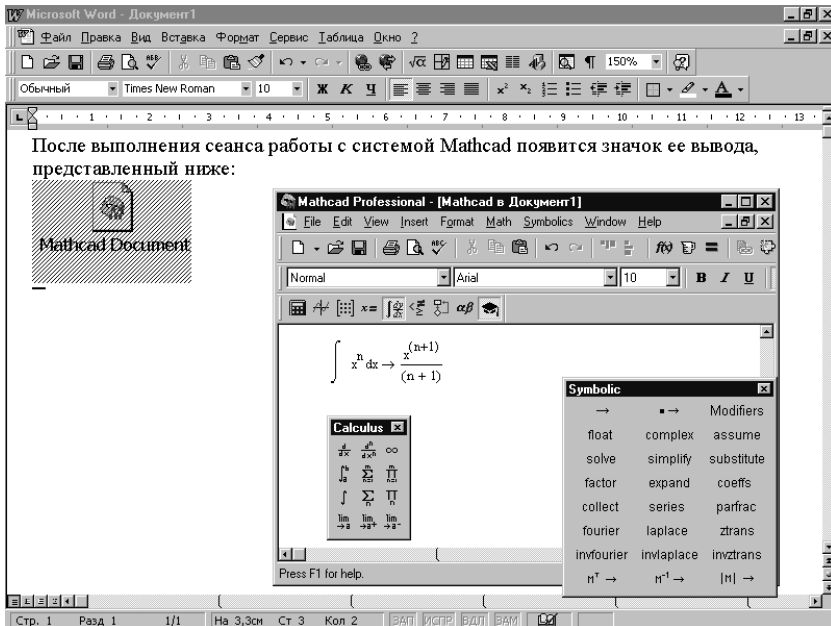


Рис. 10.12. Экран редактора Word со значком документа Mathcad и окном системы Mathcad

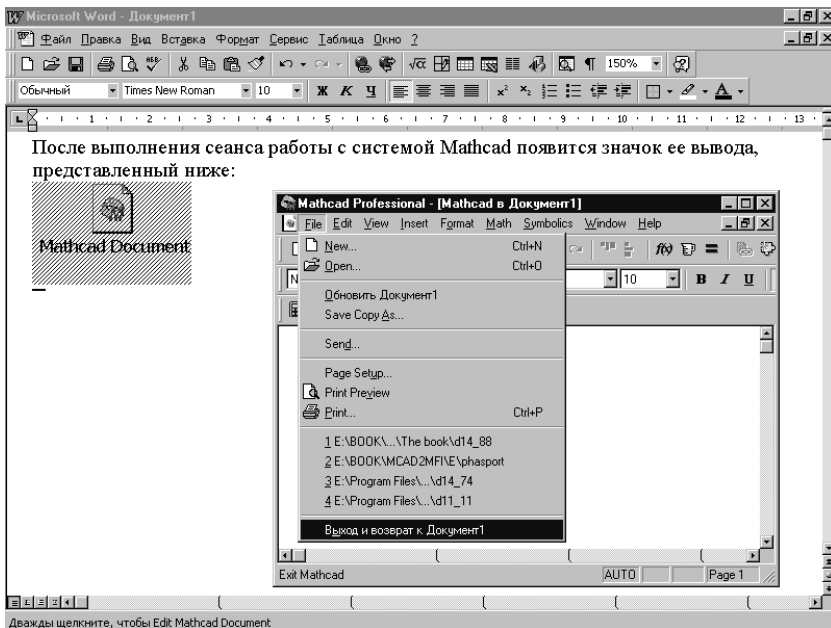


Рис. 10.13. Подготовка к закрытию окна Mathcad

показано на рис. 10.13, в окне системы Mathcad в меню File (Файл) надо выбрать команду Выход и возврат к... или щелкнуть на кнопке закрытия окна Mathcad. Все это делается при открытом окне документа Word.

После этого вид документа текстового редактора Word будет соответствовать показанному на рис. 10.14. Здесь видна пара подготовленных абзацев текста и значок, ассоциированный с документом Mathcad. Разумеется, текст можно корректировать, дополнять и т. д. в соответствии с правилами работы с текстовым редактором Word.

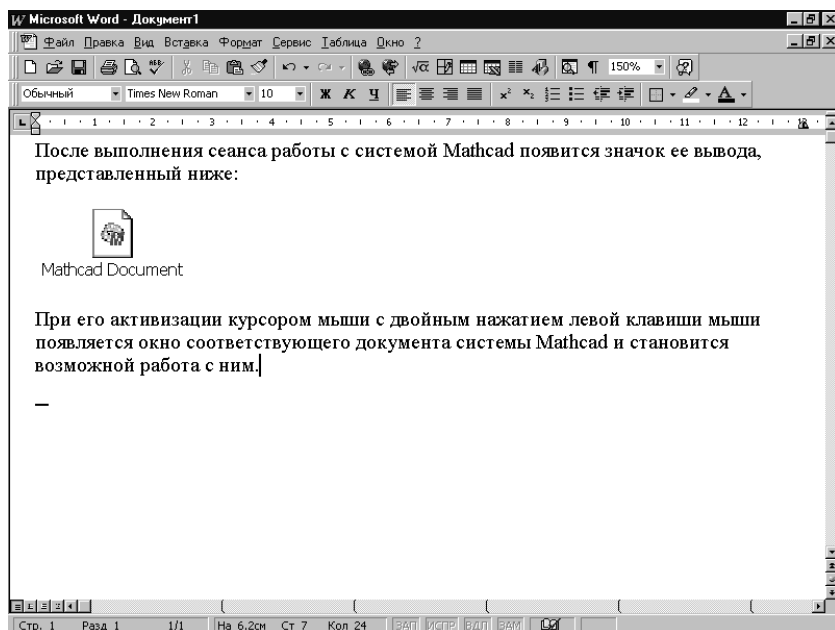


Рис. 10.14. Вид документа Word со значком, ассоциированным с документом системы Mathcad

После выделения значок, как и любой вставленный в Word объект, можно перетаскивать и изменять в размерах.

В любое время содержимое объекта можно модифицировать. Для этого достаточно дважды щелкнуть на объекте. Появится окно приложения-источника (в данном случае Mathcad) с подготовленным для редактирования документом.

Столь же просто вставить в документ Word график функции. Для отображения в документе Word не значка, а реального объекта Mathcad, при вставке нужно не устанавливать флажок В виде значка (см. рис. 10.11). Рисунок 10.15 иллюстрирует процесс внедрения в документ Word графика параболической поверхности. Как нетрудно догадаться, этот график (после фиксации ввода и выделения) можно перетаскивать и изменять в размерах.

Таким образом, вставка в документы текстового процессора объектов из математической системы Mathcad дает полноценный доступ ко всем возможностям и средствам последней. Как лучше поступать — работать в Word, вставляя объекты из Mathcad, или, работая в системе Mathcad, вставлять в нее объекты из текстового процессора Word, каждый пользователь должен решать самостоятельно.

Разумеется, таким же образом можно связать с Mathcad и другие, не упомянутые выше системы (см. раздел «Обзор программных средств, интегрируемых с системой Mathcad»). Необходимо лишь одно условие — возможность установления объектной связи (то есть приложение должно поддерживать технологию OLE).

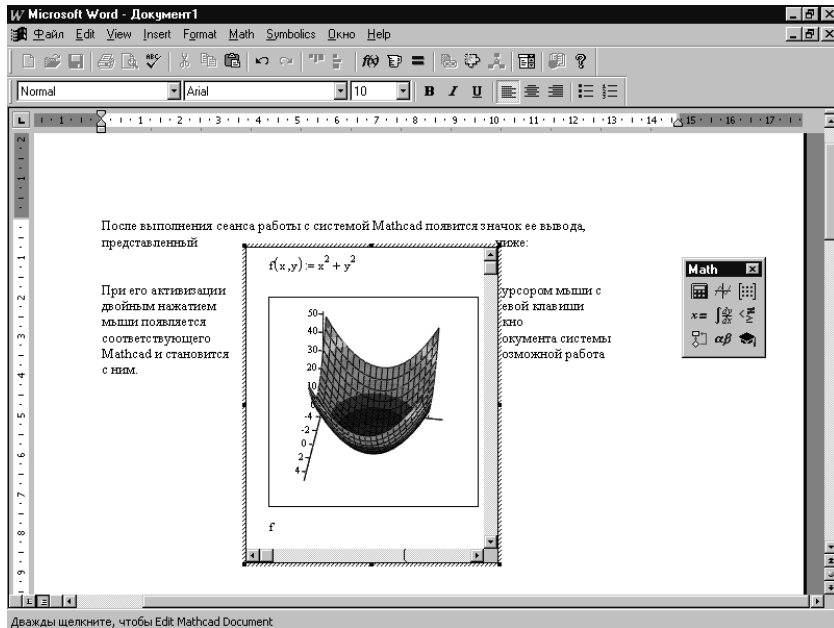


Рис. 10.15. Пример вставки в документ рисунка из системы Mathcad

Большинство приложений, разработанных для работы в среде операционной системы Windows, такой возможностью обладают. Это открывает широкие перспективы (даже для рядового пользователя) в создании сложных программных систем, решающих самые разнообразные математические задачи и создающих отчеты (документы) самого высокого качества.

10.5.2. Интеграция Mathcad с табличным процессором Excel

Еще одним широко распространенным офисным пакетом являются электронные таблицы Microsoft Excel. С системой Mathcad 2001i можно использовать расширение Mathcad Add-In for Excel, устанавливающее полноценные объектные связи между системой Mathcad и табличным процессором Excel. Для установки этого расширения достаточно вставить компакт-диск с этим расширением в устройство чтения и дождаться появления окна установки этого компонента (рис. 10.16).

Это окно дает вступительную информацию о возможности подключения табличного процессора Excel к математической системе Mathcad. Ознакомившись с этой информацией, щелкните на кнопке Next (Далее). Появится еще одно информационное окно с текстом лицензионного соглашения. Вам надо ознакомиться с текстом соглашения и щелкнуть на кнопке Yes (Принимаю). Начнется процесс установки расширения Mathcad Add-In for Excel.

Завершается установка расширения Mathcad Add-In for Excel появлением заключительного окна установки с сообщением о завершении установки и предложением заглянуть в папку, в которой находятся описание расширения и демонстрационные примеры совместной работы систем Mathcad и Excel.

Описанный процесс установки расширения Mathcad Add-In for Excel характерен и для других пакетов расширений и электронных книг, созданных для систем



Рис. 10.16. Первое окно установки расширения Mathcad Add-In for Excel

класса Mathcad. Большинство из них поставляется (точнее продается) отдельно от системы Mathcad, однако при установке Mathcad 2001 в вариантах PRO и Premium расширение Add-In for Excel устанавливается автоматически.

После установки расширения Add-In for Excel появляется возможность совместной работы Mathcad с электронными таблицами с установкой объектной связи между этими программами. Простой пример такой совместной работы иллюстрирует рис. 10.17. Здесь в Mathcad задана матрица M, которая экспортируется в Excel,

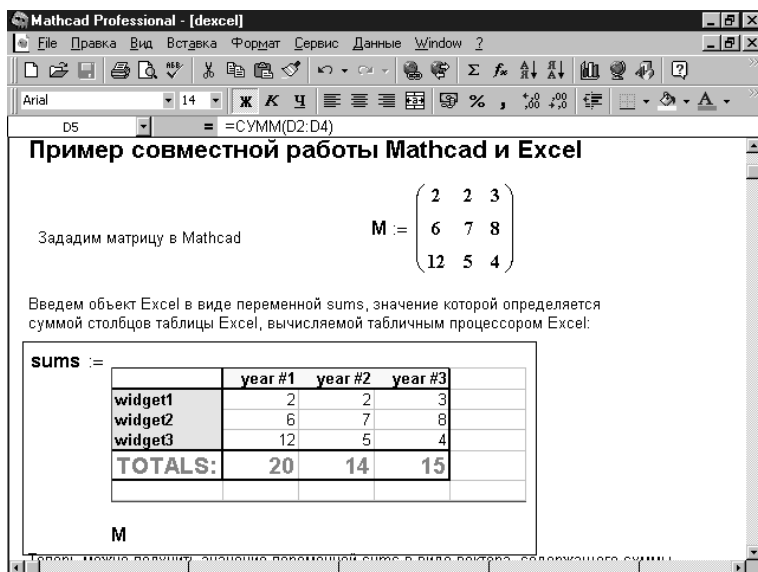


Рис. 10.17. Пример совместной работы Mathcad и Excel

а ее элементы становятся элементами электронной таблицы Excel. В этой таблице уже средствами Excel производится вычисление сумм элементов столбцов, являющихся элементами вектора-строки, который присваивается переменной *sums*.

После вставки объекта Excel (см. раздел «Вставка объектов» в главе 4) с помощью команды Object (Объект) меню Insert (Вставка) этот объект может редактироваться — например, для вычисления суммы элементов столбцов таблицы. Как видно из рисунка, в ходе редактирования электронная таблица представлена в формате Excel, а элементы интерфейса соответствуют элементам системы Excel. Запуск Excel происходит автоматически.

По завершении редактирования электронной таблицы достаточно щелкнуть вне ее области. Таблица примет вид стандартного объекта Excel, но находящегося в документе Mathcad (рис. 10.18).

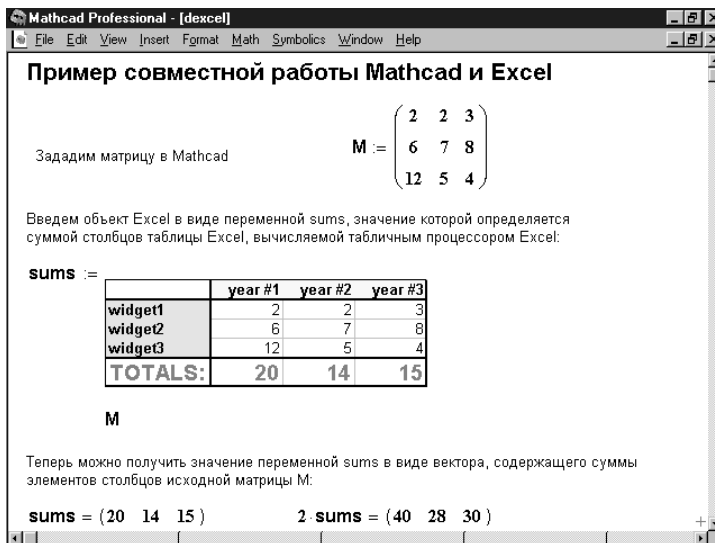


Рис. 10.18. Окно документа Mathcad с объектом Excel

Изменить значения элементов матрицы *M* можно стандартными средствами редактирования матрицы. Как только будет выполнен щелчок вне области матрицы, произойдет ее пересчет в Excel и будут получены новые результаты. Аналогично можно выполнить редактирование блока Excel. При этом, как было показано на рис. 10.17, в окне Mathcad появятся меню и панели инструментов Excel.

10.5.3. Вставка в Mathcad объектов класса ActiveX

Рисунок 10.19 иллюстрирует процесс вставки объекта ActiveX — диалогового окна для задания сигналов. Объект (в данном случае из расширения Data Acquisition Control) позволяет задавать основные параметры сигнала.

Справа на рисунке показано также окно свойств вставляемого объекта. Это окно позволяет задать рамку вокруг объекта, цвет фона и вернуть размеры объекта к исходным (объекты могут меняться в размерах и перемещаться средствами Mathcad). На рис. 10.20 представлен фрагмент документа Mathcad с графиком синусоидального сигнала, построенным по данным, заданным с помощью вставленного диалогового окна.

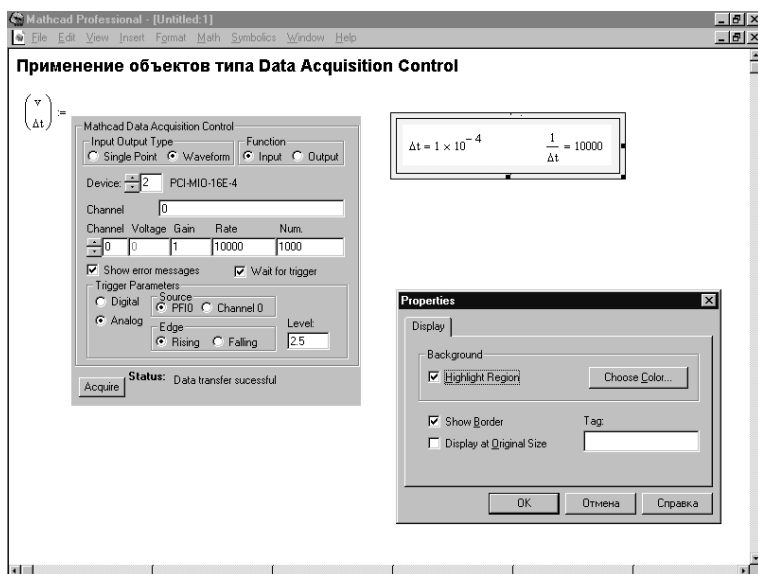


Рис. 10.19. Пример вставки объекта Data Acquisition Control

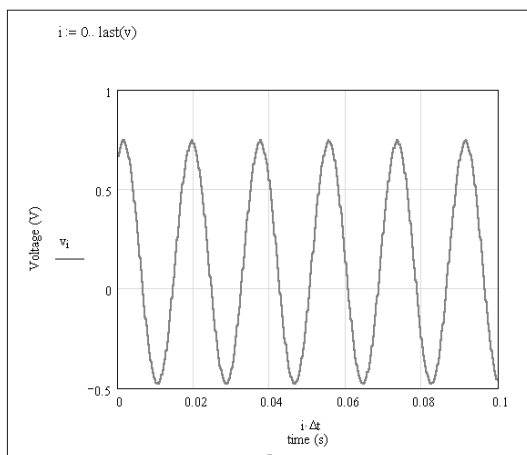


Рис. 10.20. График сигнала, созданного объектом Data Acquisition Control

Благодаря возможности вставки динамических программных объектов система Mathcad приобретает качества, характерные для визуально-ориентированных сред программирования. В частности, допускается вставка и применение различных объектов пользовательского интерфейса — диалоговых окон, текстовых полей, списков, флажков, кнопок, ползунков и т. д. На рис. 10.21 представлены примеры использования таких объектов.

10.5.4. Интеграция Mathcad с графическим пакетом Axum

Как уже отмечалось, графический пакет Axum 5/6/7, как и система Mathcad, создан фирмой MathSoft, Inc, поэтому интеграция системы Mathcad с ним не вы-

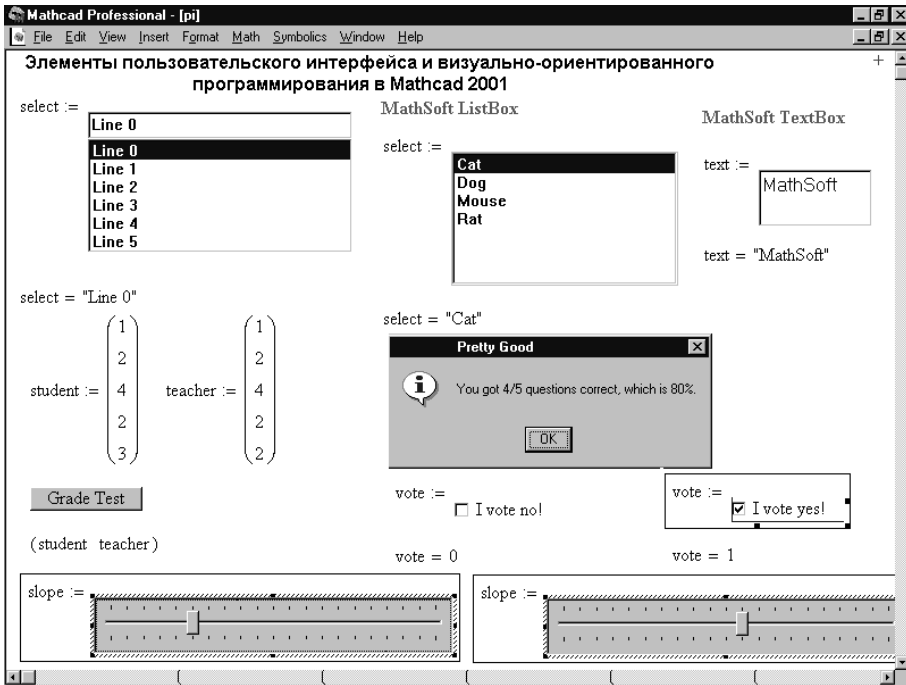


Рис. 10.21. Пример вставки объектов пользовательского интерфейса

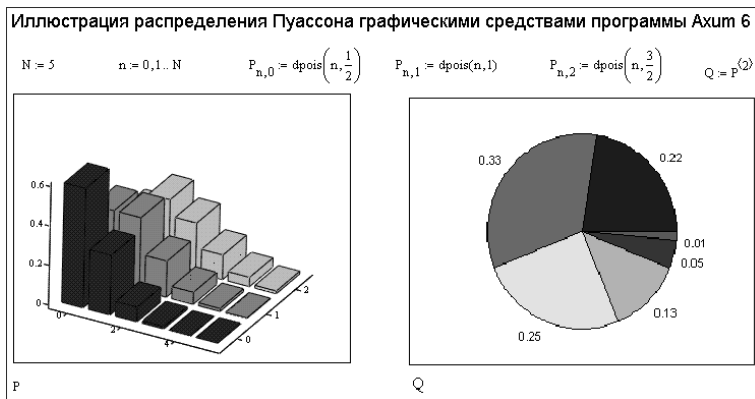


Рис. 10.22. Пример интеграции пакета Ахит 6 с Mathcad 2001

зывает никаких проблем. Теперь пользователи могут обращаться к специфическим средствам графики пакета Ахит (рис. 10.22).

В этом примере заданы наборы случайных чисел с распределением Пуассона и дана графическая иллюстрация их распределения с помощью объемной и круговой гистограмм.

10.5.5. Интеграция Mathcad с пакетом Visio

Одним из лучших пакетов научной, инженерной и иллюстративной графики является Visio. Он позволяет строить графики самого различного типа — от простых

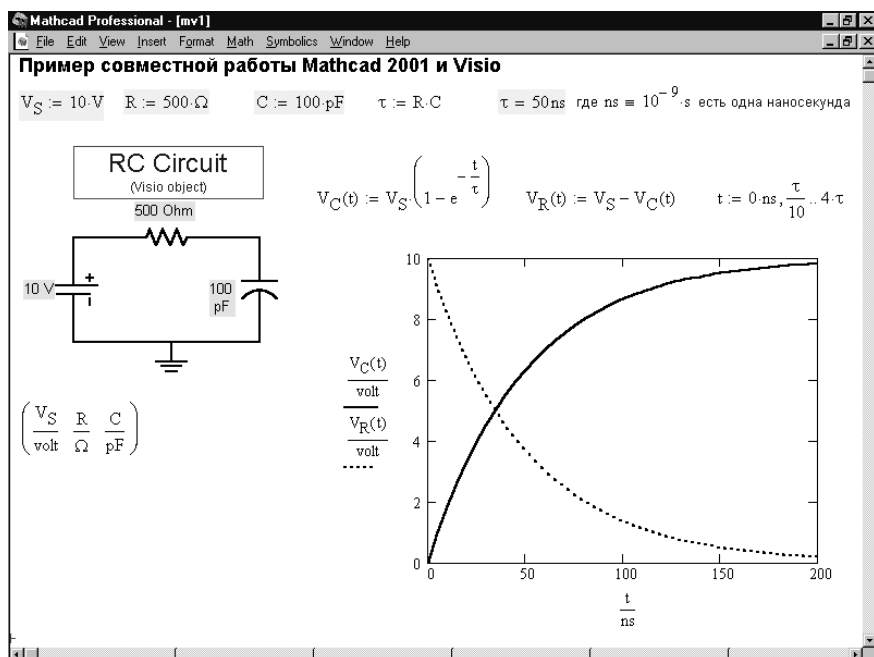


Рис. 10.23. Пример вставки рисунка Visio (электрической схемы) в документ Mathcad

диаграмм до сложных электронных схем и конструкторских чертежей. На рис. 10.23 показан документ Mathcad с простым примером применения этого пакета.

В этом примере производится расчет переходных процессов в RC -цепи средствами системы Mathcad по известным аналитическим выражениям. Рассчитываются временные зависимости напряжений на конденсаторе и резисторе и строятся их графики. Но сам чертеж схемы выполнен средствами пакета Visio. При этом заданные в Mathcad параметры схемы передаются в графический объект Visio (в схему) и отображаются в нем.

Как и в других случаях, можно перейти к редактированию встроенного объекта, дважды щелкнув на нем левой кнопкой мыши. Объект при этом приобретает тот вид, который он имеет в создавшем его приложении — в нашем случае в Visio. Это и иллюстрирует рис. 10.24.

10.5.6. Интеграция Mathcad с пакетом SmartSketch LE

Еще один графический пакет, который легко интегрируется с системой Mathcad, — это пакет SmartSketch LE. Благодаря этому пакету можно включать в состав документов Mathcad довольно сложные конструкторские чертежи. Так, в документе, показанном на рис. 10.25, можно видеть часть чертежа со стрелой и ковшем экскаватора.

На рис. 10.26 показана часть документа Mathcad, посвященного расчету фрикционной ременной передачи. Этот документ является переработкой документа, имеющегося в разделе Extending Mathcad центра ресурсов. Из рисунка убраны текстовые комментарии, так что он иллюстрирует не сами методы расчета, а лишь технику совместной работы систем Mathcad 2001i и SmartSketch LE.

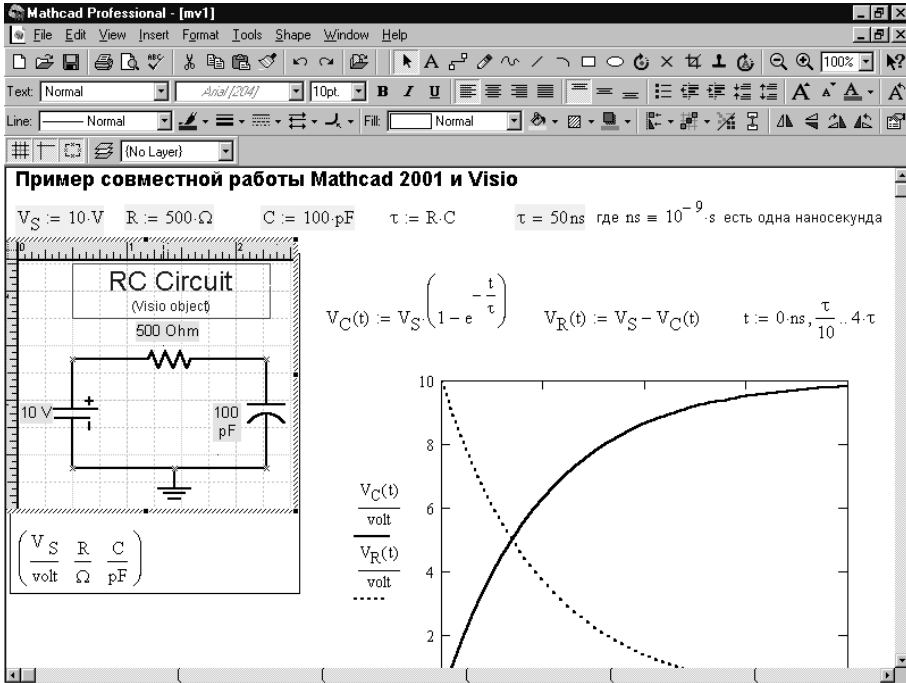


Рис. 10.24. Пример редактирования объекта Visio

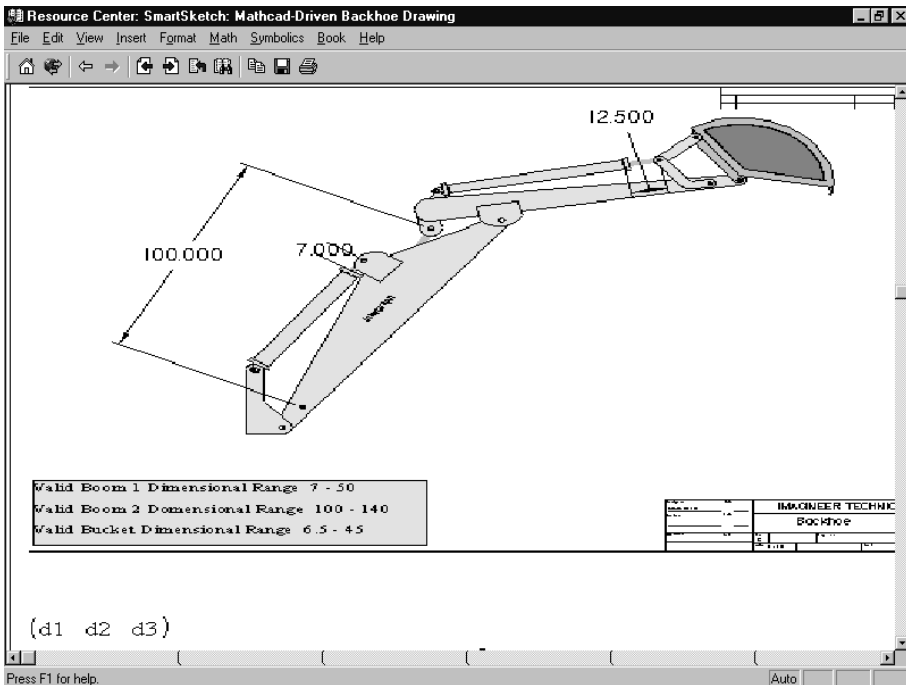


Рис. 10.25. Пример чертежа, созданного с помощью пакета SmartSketch LE в документе Mathcad

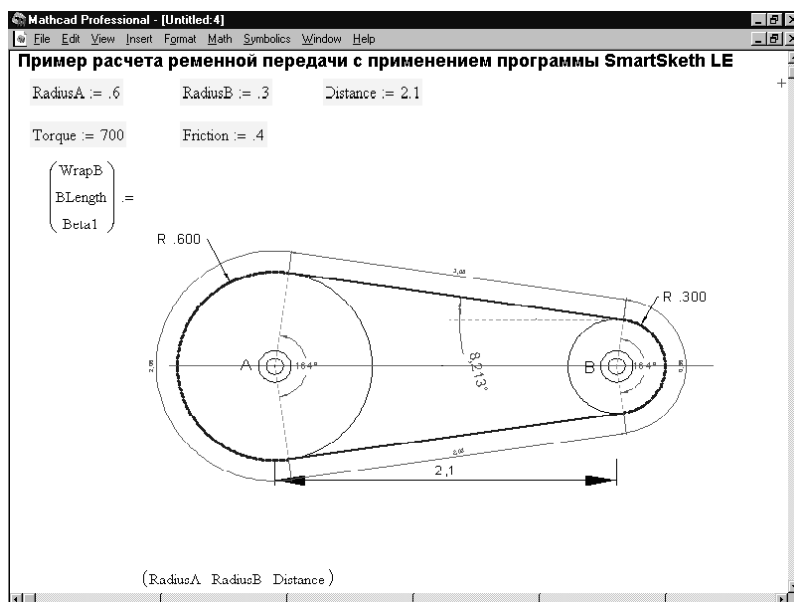


Рис. 10.26. Пример расчета фрикционной передачи (часть 1)

На рис. 10.27 показана вторая часть этого документа, содержащая уже не только дополнительные иллюстрации, но и всю достаточно сложную расчетную часть решения данной задачи.

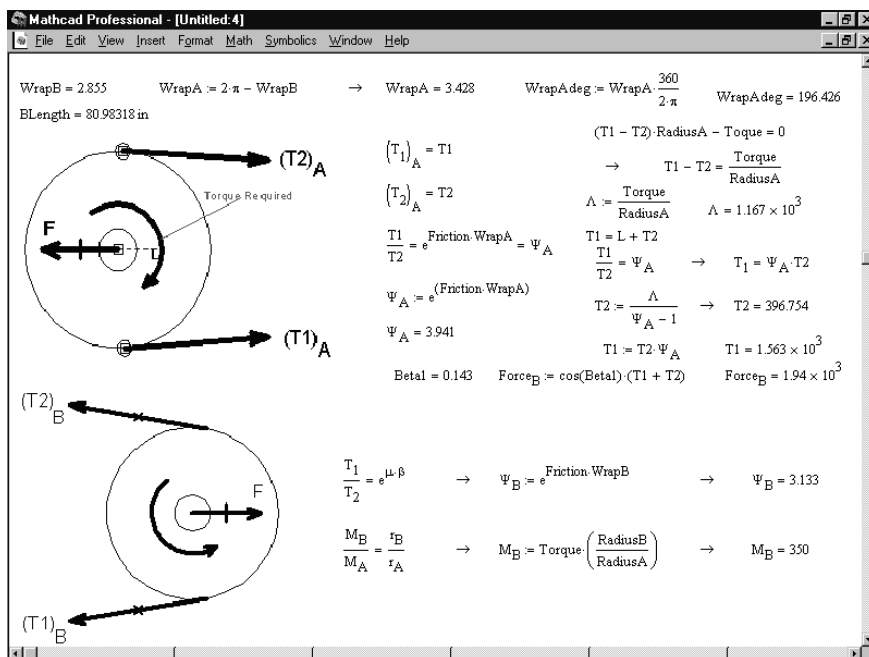


Рис. 10.27. Пример чертежа, созданного с помощью пакета SmartSketch LE в документе Mathcad

Применение различных графических пакетов для включения иллюстраций в документы Mathcad открывает поистине неограниченные возможности создания полноценных электронных уроков, статей, обзоров и даже книг.

10.5.7. Интеграция Mathcad с матричной лабораторией MATLAB

Рассмотрим интеграцию Mathcad с другой математической системой MATLAB на конкретном примере. Все, что надо сделать для использования специфических возможностей MATLAB — это включить в состав документа блоки этой системы, установив компонентную связь Mathcad. Под блоками MATLAB в данном случае понимаются строки ввода и вывода этой системы.

При загрузке такого документа из среды Mathcad 2001 можно наблюдать появление в поле объекта MATLAB имени этой системы, как показано на рис. 10.28. Обратите внимание, что уже Mathcad 2001i работает с реализацией системы MATLAB 6.5. Такая интеграция резко расширяет возможности обеих систем.

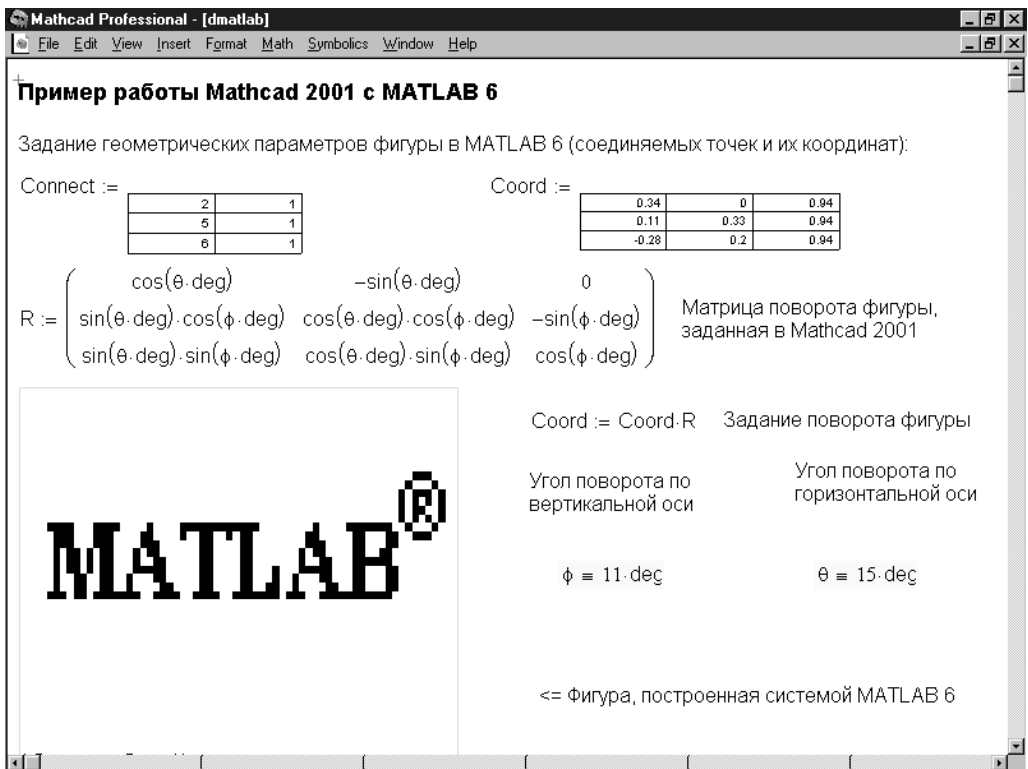


Рис. 10.28. Начало загрузки документа Mathcad с блоком системы MATLAB

На рис. 10.29 показан вид документа Mathcad с блоком системы MATLAB в процессе работы. Решается задача на построение сферы, состоящей из шести- и пятиугольников. При этом используется графика системы MATLAB, позволяющая строить сложные объемные фигуры в виде отрезков прямых, соединяющих заданные точки. Задаются номера соединяемых точек и их пространственные координаты. Для представления фигуры в заданном положении средствами Mathcad

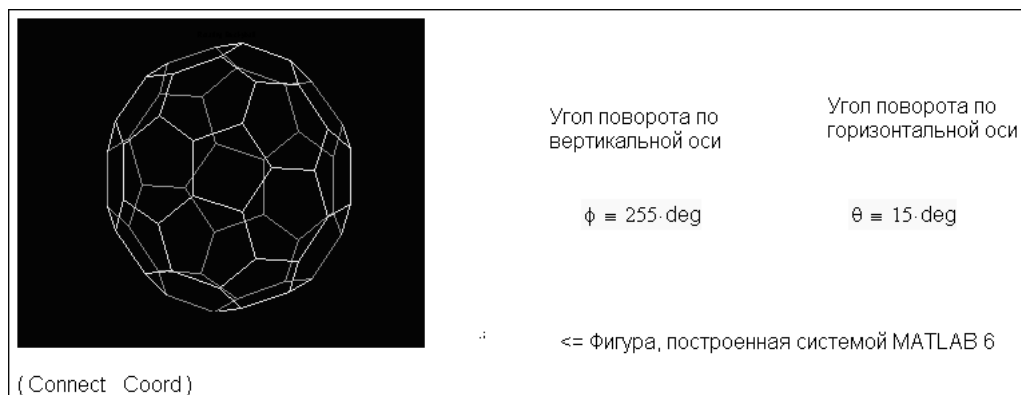


Рис. 10.29. Пример работы с документом, использующим функции матричной системы MATLAB 6

задается матрица вращения, с помощью которой и выполняется пересчет исходных координат точек вершин фигуры.

Этот пример, кстати говоря, ранее можно было реализовать в среде системного интегратора MathConnex, который из Mathcad 2001i/11 исключен (но его можно получить из Интернета, зайдя на сайт корпорации MathWorks).

10.5.8. Интеграция Mathcad с пакетом VisSim

По-видимому, одной из главных причин исключения из состава системы Mathcad системного интегратора MathConnex стали существенно расширенные возможности интеграции самой системы Mathcad с различными программными средствами — в том числе и с известной моделирующей программой VisSim.

Объем и характер данной книги не позволяет даже кратко рассмотреть эту программу, вполне заслуживающую отдельного описания. Поэтому ограничимся лишь приведением одного из примеров моделирования спектров сигналов с ее помощью (рис. 10.30).

Знакомый с MathConnex читатель найдет много общего в технологии построения моделей объектов для их моделирования, используемой в программах MathConnex и VisSim. В обоих случаях объекты задаются блоками, между которыми указываются необходимые связи. К блокам подключаются виртуальные измерительные приборы, что позволяет, в частности, наблюдать высококачественные осциллограммы, представляющие работу моделируемых объектов.

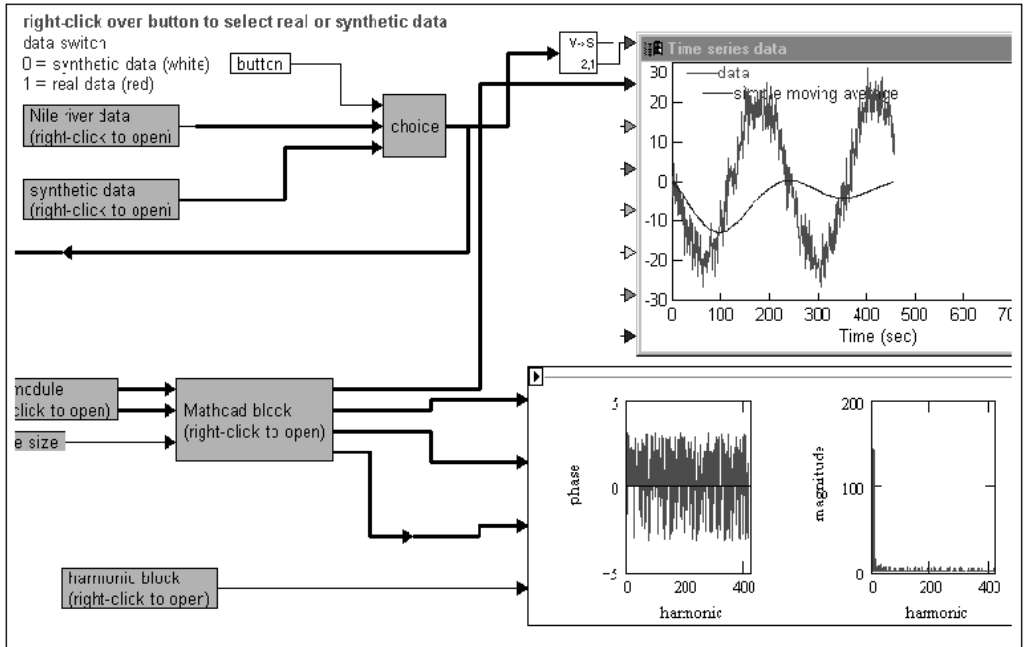


Рис. 10.30. Пример совместной работы программы VisSim с Mathcad 2001

Глава 11. Математическое моделирование

Весьма важной областью применения компьютеров является математическое моделирование в физике [26], химии, биологии и в других отраслях науки и техники. В основе математического моделирования лежат численные методы решения различных уравнений [27, 28]. В этой главе приводятся многочисленные примеры моделирования различных физических, биологических и экономических явлений и систем на базе Mathcad 2001i/11/12, а также некоторых устройств.

11.1. Основные понятия моделирования

11.1.1. Назначение и цели моделирования

Моделирование — замещение исследуемого объекта (оригинала) его условным образом, описанием или другим объектом (моделью) и изучение свойств оригинала путем исследования свойств модели. С моделями и моделированием мы сталкиваемся в нашей жизни каждый день. В детстве ребенка окружают игрушки — машинки, куклы, кубики и т. д. — модели, повторяющие отдельные свойства реально существующих предметов. Играя, ребенок получает важные знания о них. В процессе мышления человек оперирует образами объектов окружающего мира, которые являются разновидностями моделей — когнитивными (мысленными) моделями.

Реальная польза от моделирования может быть получена при выполнении двух главных условий:

- модель должна быть адекватной оригиналу, в том смысле, что должна с достаточной точностью отображать интересующие исследователя характеристики оригинала;
- модель должна устранять проблемы, связанные с физическим измерением каких-то сигналов или характеристик оригинала.

Моделирование может осуществляться с двумя главными целями:

- для изучения механизма явлений (познавательная цель);
- для управления объектом, т. е. для выработки по модели оптимальных управляемых воздействий.

В обоих случаях модель создается для определения и прогноза интересующих характеристик или сигналов объекта.

11.1.2. Основные виды моделей и их свойства

В зависимости от способа реализации все модели можно разделить на два класса.

Физические модели — предполагают, как правило, реальное воплощение тех физических свойств оригинала, которые интересуют исследователя. Физические модели упрощены и меньше и называются *макетами*. Физическое моделирование иначе называется *макетированием*.

Математические модели представляют собой формализованные описания объекта или системы с помощью некоторого абстрактного языка, например, в виде

совокупности математических соотношений или схемы алгоритма. Различают различные виды математического моделирования: вербальные (словесные), графические, табличные, аналитические и алгоритмические.

Иногда математическая модель описывается уравнениями, которые вытекают из рассмотрения физической сущности моделируемого явления или системы. Однако чаще описание моделируемых объектов и систем носит чисто формальный характер и базируется на том, что многие явления порой самой различной природы описываются уравнениями (алгебраическими, дифференциальными и иными) одного и того же вида. В этом случае говорят о *формальных* моделях.

Кроме того, явления, системы и их модели могут быть нестационарными и стационарными. *Нестационарные модели* характеризуются зависимостью их параметров от времени. У *стационарных моделей* такой зависимости нет. Естественно, что моделирование нестационарных явлений гораздо сложнее, чем стационарных.

Модели обладают рядом свойств, от которых зависит успех их применения. Отметим некоторые из них, наиболее важные.

Адекватность — это степень соответствия модели исследуемому реальному объекту. Она никогда не может быть полной. На практике модель считают адекватной, если она с удовлетворительной точностью позволяет достичь целей исследования.

Простота (сложность) — также является одной из характеристик модели. Чем большее количество свойств объекта описывает модель, тем более сложной она оказывается. Не всегда чем сложнее модель, тем выше ее адекватность. Надо стремиться найти наиболее простую модель, позволяющую достичь требуемые результаты изучения.

Потенциальность (предсказательность) — способность модели дать новые знания об исследуемом объекте, спрогнозировать его поведение или свойства.

11.1.3. Основные принципы моделирования

Моделирование базируется на нескольких основополагающих принципах. Рассмотрим их.

Принцип информационной достаточности

При полном отсутствии информации об исследуемом объекте построение его модели невозможно. С другой стороны, при наличии полной информации об объекте построение его модели не имеет смысла. Существует некоторый уровень априорной информации об объекте, при достижении которой может быть построена его адекватная модель.

Принцип осуществимости

Создаваемая модель должна обеспечивать достижение поставленной цели исследования с вероятностью, существенно отличающейся от нуля.

Принцип множественности моделей

Данный принцип является ключевым. Речь идет о том, что создаваемая модель должна отражать в первую очередь те свойства реальной системы, которые интересуют исследователя. Соответственно при использовании любой конкретной модели познаются лишь некоторые стороны реальности. Для более полного ее исследования необходим ряд моделей, позволяющий с разных сторон и с разной степенью детализации рассмотреть исследуемый объект.

Принцип агрегирования

В большинстве случаев сложную систему можно представить состоящей из агрегатов (подсистем). Для адекватного математического описания которых оказываются пригодными некоторые стандартные математические схемы.

Принцип параметризации

Этот принцип означает, что модель строится в виде известной системы, параметры которой не известны.

11.1.4. Технология моделирования

Степень реализации перечисленных принципов каждой конкретной модели может быть различной. Это зависит не только от желания исследователя, но и от соблюдения им технологий моделирования. А любая технология подразумевает определенную последовательность действий.

В настоящее время самой распространенной *технологией моделирования* является *комплексное моделирование*, под которым понимается математическое моделирование с использованием средств вычислительной техники. Соответствующие технологии комплексного моделирования представляют выполнение следующих действий:

- определение цели моделирования;
- разработку концептуальной модели;
- формализацию модели;
- программную реализацию модели;
- планирование модельных экспериментов;
- реализацию плана эксперимента;
- анализ и интерпретацию результатов моделирования.

Существуют два широких класса технологий моделирования. Первый — *имитационное моделирование* базируется на вычислении тех параметров объекта моделирования, которые описывают его поведение в рамках принятых допущений. Как правило, одним из важных допущений является неизменность структуры объекта и условий протекания моделируемых явлений. Второй класс — *ситуационное моделирование* основано на возможности изменения ситуаций в ходе моделирования. Он требует особых методов моделирования и в данной книге не рассматривается ввиду отсутствия в системе Mathcad средств для его полноценной реализации.

11.1.5. Основные методы решения задач моделирования

На этапе программной реализации модели и реализации плана экспериментов необходим выбор методов решения задач моделирования. При этом используются три основные группы методов:

- графические — оценочные приближенные методы, основанные на построении и анализе графиков;
- аналитические — решения, строго полученные в виде аналитических выражений (пригодны для узкого круга задач);
- численные — основной инструмент для решения сложных математических задач, основанный на применении различных численных методов.

Аналитическое решение удается получить редко и чаще лишь при упрощенной формулировке задачи в линейном приближении. Основным средством решения является алгоритмический подход, реализующий вычислительный эксперимент на ЭВМ. Получаемое на ЭВМ решение почти всегда содержит некоторую погрешность. Напоминаем, что есть абсолютная погрешность

$$\varepsilon = x - x_u$$

в виде разности между приближенным x и точным x_u значениями результата и относительная погрешность

$$\Delta = \varepsilon/x_u.$$

Наличие погрешности решения обусловлено рядом причин. Перечислим основные источники погрешности.

1. Математическая модель является лишь приближенным описанием реального процесса (погрешность модели).
2. Исходные данные, как правило, содержат погрешности, так как либо являются результатами экспериментов (измерений) или решениями вспомогательных задач (погрешность данных).
3. Применяемые для решения задачи методы в большинстве случаев являются приближенными (погрешность метода).
4. При вводе исходных данных в ЭВМ, выполнении операций, производятся округления (вычислительная погрешность).

Погрешности 1 и 2 — неустранимые на данном этапе решения, для их уменьшения приходится возвращаться вновь к построению математической, а и иногда и концептуальной модели, проводить дополнительное экспериментальное уточнение условий задачи.

Оценка обусловленности вычислительной задачи — еще одно обязательное требование при выборе метода решения и построении математической модели.

Пусть вычислительная задача корректна. Теоретически устойчивость задачи означает, что ее решение может быть найдено со сколь угодно малой погрешностью, если только гарантировать достаточно малую погрешность входных данных. Однако на практике их точность ограничена (и величиной гораздо большей, чем $\varepsilon_m = 2^{-P+1}$ — машинная точность, p — порядок, округление производится усечением).

Как влияют малые, но конечные погрешности входных данных на решение? Как сильно они искажают результат? Ответ на это дает понятие *обусловленности* задачи, то есть чувствительность решения вычислительной задачи к малым погрешностям входных данных.

Задачу называют *хорошо обусловленной*, если малым погрешностям входных данных отвечают малые погрешности решения, и *плохо обусловленной*, если возможны сильные изменения решения. Часто возможно ввести количественную оценку степени обусловленности — *число обусловленности* — его можно интерпретировать как коэффициент возможного возрастания погрешности в решении по отношению к вызвавшей их погрешности входных данных: если установлено неравенство между этими погрешностями

$$\Delta(y^*) \leq v_\Delta \cdot \Delta(x^*)$$

абсолютное число обусловленности или

$$\delta(y^*) \leq v_\delta \cdot \delta(x^*)$$

относительное число обусловленности (вместо погрешности могут фигурировать их границы). Для плохо обусловленных задач $\nu_\delta \gg 1$ (неустойчивость $\nu_\delta = \infty$).

При каких значениях ν_δ можно считать задачу плохо обусловленной? Это зависит от требований к точности решения и от уровня обеспечиваемой точности исходных данных.

- Если требуется найти решение с точностью 0.1 %, а входная информация задается с точностью в 0.02 %, то при $\nu_\delta = 10$ уже будет плохая обусловленность.
- Однако если исходные данные задаются с $\delta(x^*) \leq 0.0001\%$, то при $\nu_\delta = 10^3$ — задача хорошо обусловлена ($\delta(y^*) = 0.1\%$).

Вычислительные методы преобразуются к виду, удобному для программной реализации. Можно выделить следующие классы численных методов:

- *Метод эквивалентных преобразований* — исходную задачу заменяют другой, имеющей то же решение: нахождение корня нелинейного уравнения $f(x) = 0$ сводят к поиску точек глобального минимума $\Phi(x) = (f(x))^2$.
- *Методы аппроксимации* — заменяют исходную задачу другой, решение которой близко к решению исходной задачи.
- *Методы конечно-разностные*, основанные на замене производных конечными разностями, например $f'(x) \approx \frac{f(x+h) - f(x)}{h}$.
- *Прямые (точные) методы* — решение может быть получено за конечное число элементарных операций (арифметические и извлечение корня). Многие прямые методы не годятся к применению в ЭВМ из-за чувствительности к ошибкам округления.
- *Итерационные методы* — методы последовательных приближений к решению задачи. Задается начальное приближение решения, строится итерационная последовательность приближений к решению. Если эта последовательность сходится к решению, то говорят, что итерационный процесс сходится. Множество начальных приближений, для которых метод сходится, называются областью сходимости метода.
- *Метод статистических испытаний (Монте—Карло)* — основан на моделировании случайных величин и построении статистических оценок решений задач (для моделирования больших систем).

Численные методы группируются вокруг типичных математических задач: задач анализа, алгебры, оптимизации, решения дифференциальных и интегральных уравнений, обратные задач (синтез). Этот этап решения заканчивается выбором и обоснованием конкретных численных методов решения, разработкой алгоритма, которые могут быть программно реализованы средствами компьютерной техники.

11.1.6. Контроль правильности модели

Для контроля правильности полученной модели может использоваться ряд приемов:

- *Анализ размерности* — величины в левой и правой части выражения, отдельные слагаемые в каждой из частей должны иметь одинаковую размерность.

- *Проверка порядков и характеров зависимостей* — параметры и переменные, которые в данной задаче выражены величинами большого порядка малости, могут быть исключены из рассмотрения как несущественные, что часто позволяет значительно упростить модель и ее анализ. Характер изменения значений моделируемых величин должен соответствовать их реальному смыслу, не противоречить наблюдаемому данным.
- *Исследование предельных случаев* — результаты моделирования при крайних значениях параметров модели, равных, как правило, нулю или бесконечности, не должны противоречить смыслу. (Например, энергия реальной физической системы не может оказаться бесконечно большой, время протекания процесса — отрицательным и т. п.) Модель в этом случае существенно упрощается и легче для понимания.
- *Проверка замкнутости и корректности математической задачи* — система математических соотношений должна иметь единственное решение.

Задача называется *корректной*, если она удовлетворяет трем требованиям:

1. Ее решение существует при любых допустимых входных данных.
2. Это решение единственно (однозначно определено).
3. Решение непрерывно зависит от данных задачи — устойчиво по отношению к малым возмущениям входных данных.

Решение вычислительной задачи называется *устойчивым* по входным данным X , если оно зависит от входных данных непрерывным образом; т. е. для любого $\epsilon > 0$ существует $\delta = \delta(\epsilon) > 0$ такое, что всяким исходным данным x^* , удовлетворяющим условию $\Delta(x^*) < \delta$, отвечает приближенное решение u^* , для которого $\Delta(u^*) < \epsilon$.

Далеко не все практические задачи являются корректными. К ним, например, не относятся обратные задачи геофизики, астрофизики, спектрографии, распознавания образов, синтез и многие другие важные прикладные проблемы. Свойство корректности задачи имеет большое значение для выбора метода решения. К некорректным задачам неприменимы обычные численные методы вычислительной математики. Строгий анализ корректности во многих случаях математически сложен, поэтому ограничиваются проверкой соответствия количества неизвестных и связывающих их уравнений в модели.

11.2. Задача моделирования полета камня

11.2.1. Постановка задачи моделирования

Началу моделирования предшествует постановка содержательной задачи моделирования, переход от когнитивной модели к формулировке в словесной форме основных вопросов об объекте моделирования. Содержательная постановка задачи, называемая в технических дисциплинах *техническим заданием*, в дальнейшем уточняется и конкретизируется, однако принципиальные, основные положения остаются неизменными.

Дадим постановку классической физической задачи «Бросок камня», позволяющую описать полет камня, брошенного под углом к горизонту. Модель должна позволять:

- Вычислять положение камня в любой момент времени.

Исходные данные:

- масса камня;
- начальные координаты, начальная скорость и угол броска мяча.

На основе содержательной модели разрабатывается концептуальная формулировка задачи моделирования. В нашем случае движение камня может быть описано в соответствии с законами классической механики Ньютона:

- Камень будем считать материальной точкой массой m , положение которой совпадает с центром масс камня.
- Движение происходит в поле силы тяжести с постоянным ускорением свободного падения g и описывается уравнениями классической механики Ньютона.
- Движения камня происходит в одной плоскости, перпендикулярной поверхности Земли.
- Сопротивлением воздуха на первых порах пренебрегаем.

В качестве параметров движения будем использовать координаты (x, y) и скорость $v(v_x, v_y)$ центра масс камня.

Концептуальная постановка задачи на основе принятых гипотез может быть следующей: нам надо определить закон движения материальной точки массой m под действием силы тяжести, если известны начальные координаты точки x_0 и y_0 , ее начальная скорость v_0 и угол броска α_0 . Таким образом, модель является простой — объект материальная точка не имеет внутренней структуры. Учитывая типичные скорости и высоту броска камня, можно считать постоянным ускорение свободного падения. Переход от трехмерных координат к плоскости значительно упрощает решение задачи. Он вполне допустим, если камень не подкручивается при броске. Пренебрежение сопротивлением воздуха, как будет показано далее, приводит к значительной систематической ошибке результатов моделирования.

11.2.2. Построение математической модели

Теперь перейдем к составлению *математической модели* объекта — совокупности математических соотношений, описывающих его поведение и свойства. По оси x на камень не действуют никакие силы, по оси y — действует сила тяжести. Согласно законам Ньютона имеем уравнение движения по оси x и оси y :

$$m \cdot \frac{d^2x}{dt^2} = 0, m \cdot \frac{d^2y}{dt^2} = -m \cdot g, v_x = \frac{dx}{dt}, v_y = \frac{dy}{dt}, \quad (11.1)$$

при следующих начальных условиях $x(0) = x_0$, $y(0) = y_0$, $v_x(0) = v_0 \cdot \cos \alpha_0$, $v_y(0) = v_0 \cdot \sin \alpha_0$. Найти зависимости $x(t)$, $y(t)$, $v_x(t)$, $v_y(t)$.

Математическая постановка соответствует решению задачи Коши для системы обыкновенных дифференциальных уравнений с заданными начальными условиями. Известно, что решение задачи Коши существует и единственно. Количество искомых переменных равно количеству дифференциальных уравнений. Таким образом, математическая модель корректна. Решение этой задачи есть в любом учебнике физики.

11.2.3. Решение аналитическим методом

Данная задача может быть решена как аналитически, так и численно. Рассмотрим аналитическое решение. Из (11.1) запишем систему ОДУ первого порядка:

$$\frac{dv_x}{dt} = 0, \frac{dv_y}{dt} = -g, v_x = \frac{dx}{dt}, v_y = \frac{dy}{dt}. \quad (11.2)$$

После интегрирования (11.2), получим:

$$v_x(t) = C_1, v_y(t) = C_2 - g \cdot t, x(t) = C_3 + C_1 \cdot t, y(t) = C_4 + C_2 \cdot t - \frac{g \cdot t^2}{2}.$$

Определив константы интегрирования из начальных условий, окончательно запишем:

$$x(t) = x_0 + v_0 \cdot \cos \alpha_0 \cdot t, y(t) = y_0 + v_0 \cdot \sin \alpha_0 \cdot t - \frac{g \cdot t^2}{2}$$

$$v_x(t) = v_0 \cdot \cos \alpha_0, v_y(t) = v_0 \cdot \sin \alpha_0 - g \cdot t.$$

Из аналитического решения вытекает, что полет камня при отсутствии сопротивления воздуха происходит строго по параболической траектории, причем она на участках полета камня вверх и вниз симметрична.

11.2.4. Реализация математической модели «Бросок камня» в среде Mathcad

Система Mathcad имеет несколько встроенных функций для решения систем ОДУ (rkadapt, Odesolve и др). При их использовании достаточно корректно записать условие задачи и вызвать соответствующую команду — см. рис. 11.1, на котором показано решение задачи на полет брошенного камня в условиях отсутствия сопротивления воздуха с помощью функции **Odesolve**.

Mathcad в данном случае решает задачу на полет камня без учета сопротивления воздуха. Как и следовало ожидать, траектория полета камня оказывается

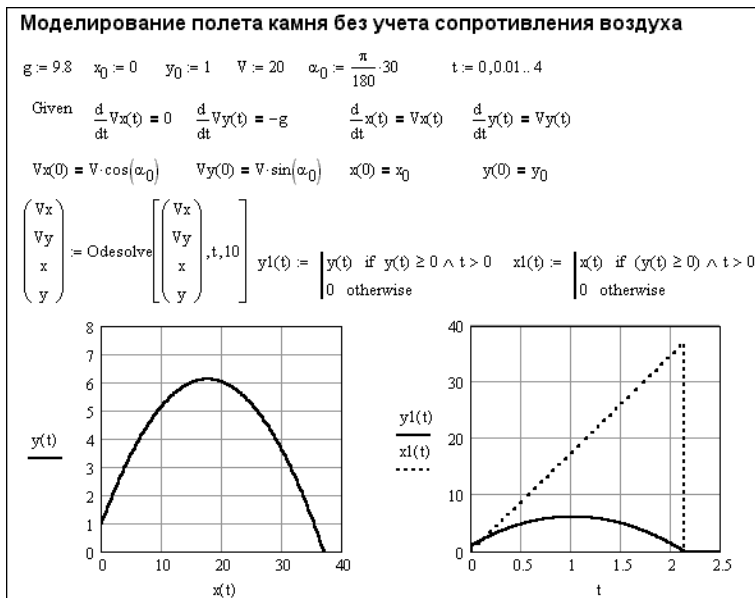


Рис. 11.1. Программная реализация численного решения в среде

квадратичной параболой. Если пользователь желает глубже ознакомиться с реализацией численных методов моделирования, он может составить свою реализацию подходящего метода, например, Эйлера или Рунге—Кутта. Однако для этого придется освоить приемы программирования в системе Mathcad — они не очень сложны, но в наши задачи пока не входят.

11.2.5. Уточнение модели «Бросок камня» и ее реализация

Теперь уточним математическую модель. Сила сопротивления воздуха F_{mp} направлена против направления движения камня:

$$m \cdot \frac{d^2x}{dt^2} = -F_{mp_x}, \quad m \cdot \frac{d^2y}{dt^2} = -m \cdot g - F_{mp_y}, \quad v_x = \frac{dx}{dt}, \quad v_y = \frac{dy}{dt}.$$

Сопротивление воздуха зависит от скорости движения тела и может быть описано следующей эмпирической формулой

$$F_{mp} = A \cdot v + B \cdot (v)^3,$$

где $A = 0.1 \text{ Н}\cdot\text{с}/\text{м}$, $B = 10^{-3} \text{ Н}\cdot\text{с}^3/\text{м}^3$.

Поскольку нелинейные задачи в аналитическом виде чаще всего не решаются, выберем численный метод, который легко реализуется функциями Mathcad.

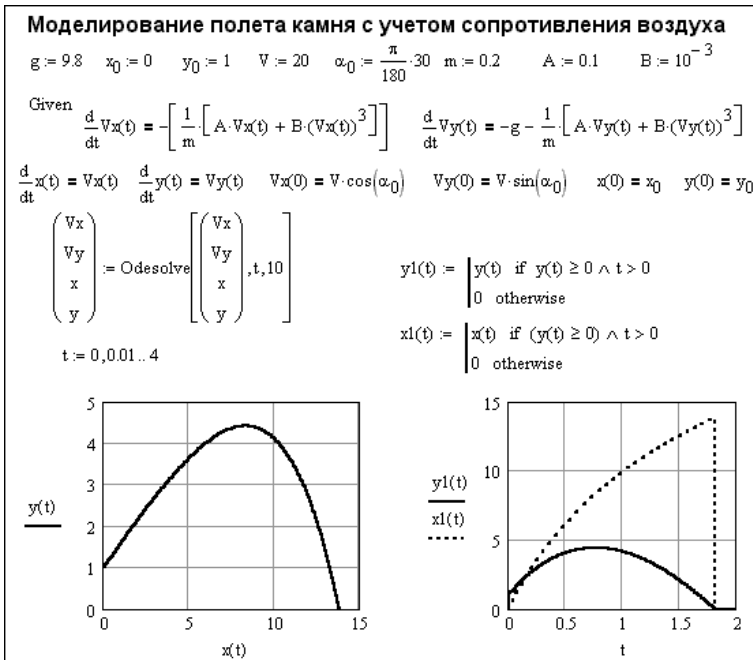


Рис. 11.2. Моделирование полета камня с учетом сопротивления воздуха

Анализ результатов решения задачи численным методом показан на рис. 11.2. Из приведенного примера явно видно, что при учете сопротивления воздуха траектория полета камня заметно отличается от параболической, она заметно круче на спаде. И камень пролетает меньшее расстояние.

11.3. Моделирование процессов на основе известных формул

Рассмотрим три характерных примера на моделирование в физике, основанное на применении известных аналитических зависимостей. В первом примере моделирование свелось просто к использованию этих зависимостей для построения графиков. В других случаях нужно искать критические моменты, что требует применения численных методов решения нелинейных уравнений.

11.3.1. Моделирование изменения параметров атмосферы

Многие процессы в природе изучены настолько хорошо, что описываются известными соотношениями или формулами. Часто, изучая то или иное явление, исследователь получает данные о нем в табличной форме и пытается получить описывающие его истинные или формальные математические зависимости (модели). Если исходные данные представлены с заметной погрешностью, что бывает чаще всего, то для этого используется аппарат регрессии (метод наименьших квадратов). Есть программы, которые выполняют регрессию сразу для множества приближающих данные функций, что позволяет отобрать лучшие из них, дающие наименьшее среднеквадратическое отклонение данных от вычисляемых по выбранной функции.

Возьмем простой пример, полезный всем, кто занимается парашютным спортом, отдыхает в горах или работает, рисуя красоты горных пейзажей. Речь идет об оценке атмосферного давления и плотности воздуха на разной высоте, от которых зависит наше состояние и даже возможность нахождения на заданной высоте. Документ рис. 11.3 показывает решение данной задачи на основе эмпирических, но достаточно точных и апробированных многими годами применения формул. Они приведены в документе.

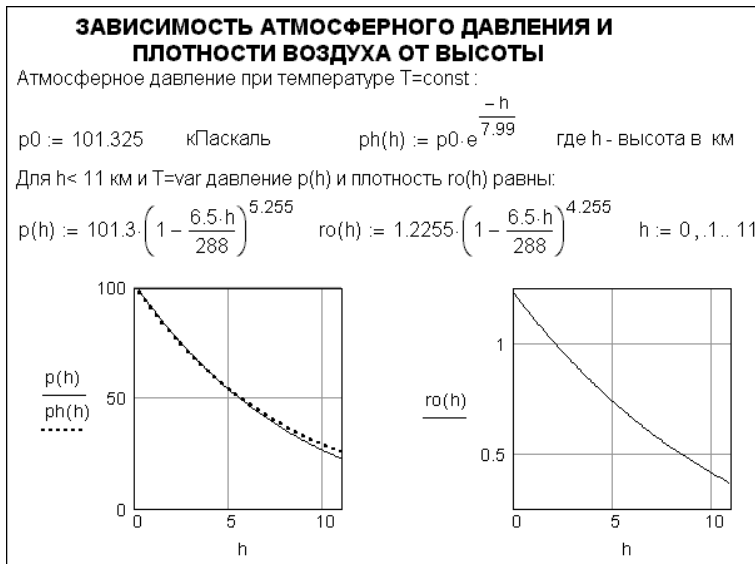


Рис. 11.3. Моделирование изменения атмосферного давления и плотности воздуха на различных высотах h

Для давления даны две формулы — одна, для $p(h)$, не учитывает изменения температуры воздуха с изменением высоты h , а другая, для $p(h)$, учитывает этот фактор. Завершают документ графики, позволяющие легко оценивать искомые параметры для разных высот h .

Многие процессы в физике и в химии описываются хорошо известными формулами — например, экспоненциальный распад радиоактивных элементов, экспоненциальный заряд и разряд конденсатора через резистор, синусоидальное колебание маятника и т. д. Опытный пользователь может в считанные минуты выполнить расчет по таким формулам в системе Mathcad и получить нужные числовые и графические данные.

11.3.2. Моделирование преодоления самолетом звукового барьера

Рассмотрим еще один пример простого моделирования с применением эмпирической зависимости. Возможно, вы наблюдали за полетом самолета и вдруг до вас доносились раскаты грома или звук взрыва при абсолютно чистом небе. Не спешите прятаться — скорее всего, эти звуки связаны с преодолением самолетом звукового барьера.

Физики давно вывели формулу, при которой это явление наблюдается. Эта формула определяет некоторое число Маха M , которое является отношением скорости тела или газовой струи к скорости звука в газе. Для самолета это число $M(u, h)$ зависит от скорости полета самолета u и высоты полета h . Как только M достигает 1, самолет преодолевает звуковой барьер, обтекание его воздухом резко меняется и это порождает звук выстрела или грома.

Рисунок 11.4 представляет документ системы Mathcad, в котором задана известная формула $M(u, h)$ и построены графики значений числа Маха от скорости полета на четырех высотах в 1, 5, 10 и 20 км (1 км = 1000 м). Пересечение этих графиков с горизонталью 1 позволяет найти момент, когда создаются условия для преодоления звукового барьера.

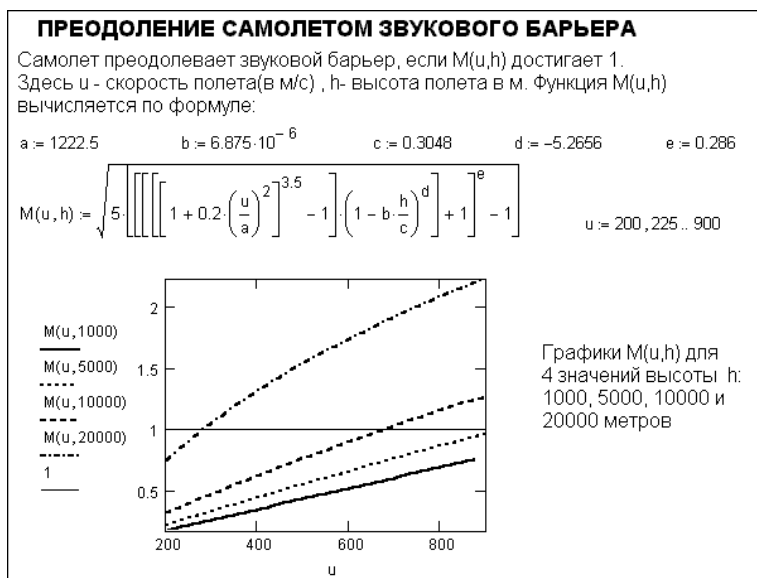


Рис. 11.4. Вычисление чисел Маха

К сожалению, найти этот момент по графикам достаточно точно трудно. Кроме того, хотелось бы получить зависимость скорости самолета от его высоты для моментов, когда он преодолевает звуковой барьер. Для этого нужно решать нелинейное уравнение $M(u, h) - 1 = 0$ относительно переменной u для заданного значения h . Мы можем воспользоваться тем или иным численным методом. Рисунок 11.5 показывает не только такое решение, но и построение по его результатам зависимости $u(h)$ при $M = 1$.

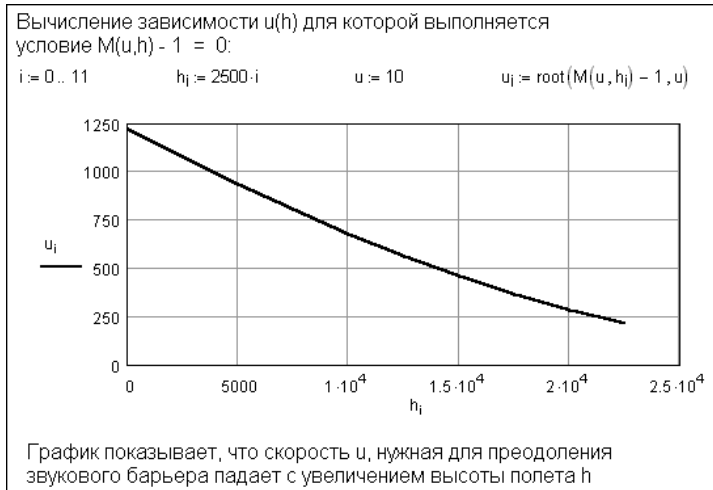


Рис. 11.5. Построение зависимости $u(h)$ для случая, когда $M=1$

Для решения используется функция системы Mathcad — `root`. Если вы пилотируете самолет, то зная высоту его полета h (в метрах), можете легко вычислить скорость u (в м/с), нужную для преодоления звукового барьера. Скорость около 1240 км/час характерна при полетах на низкой высоте. С ростом высоты полета эта скорость заметно (в несколько раз) снижается. Однако она остается настолько высокой, что преодолеть ее пока могут только реактивные самолеты.

11.3.3. Стрельба по летящей утке

Мальчик-охотник увидел летящую горизонтально на высоте h утку. Он, недолго думая, пулянул из рогатки камень в утку, не сделав поправку на время полета камня до цели. Утка благополучно миновала место встречи с камнем. Имеет ли шанс мазила-охотник все же добыть птицу для своего пропитания? Оказывается, что имеет — пролетев выше высоты полета утки, камень может попасть в бедную птицу на излете.

Эту трагикомическую ситуацию моделирует документ, представленный на рис. 11.6. Используется классическое аналитическое решение на полет камня без учета сопротивления воздуха — в данном случае нас интересует сам факт необычного попадания в цель, а вовсе не точный расчет траектории полета камня. Она, ради простоты, считается параболической.

Для нахождения момента встречи камня с уткой используется решение уравнений движения этих объектов с помощью функции `Find`. Затем строятся траектории полета камня и движения утки (последняя показана с момента, когда охотник произвел выстрел).

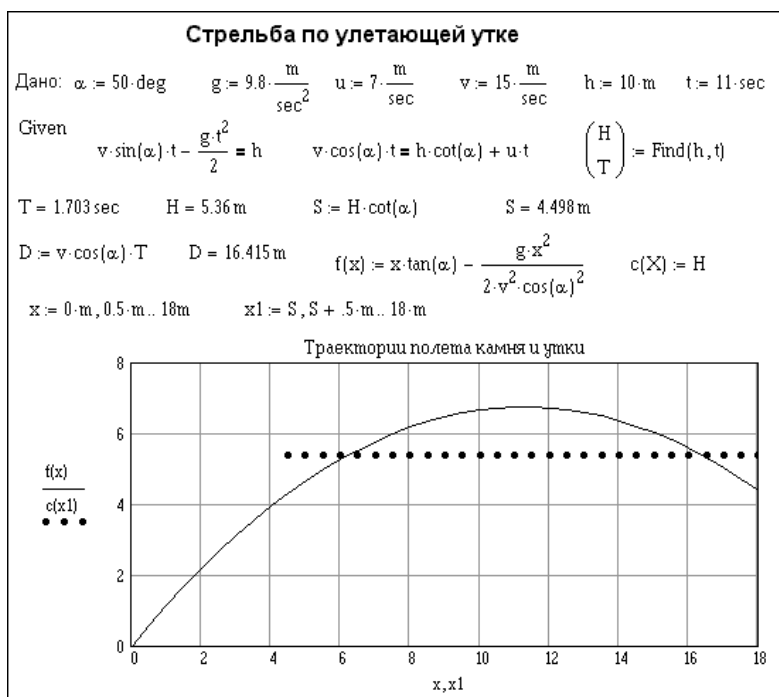


Рис. 11.6. Построение траекторий полета пули и утки

11.4. Моделирование на основе конечно-разностных методов

Выше мы уже отмечали роль конечно-разностных методов. У них решение задачи на заданном шаге основано на использовании предшествующих шагов (одного или нескольких). Пора привести примеры их применения.

11.4.1. Имитация броуновского движения частиц в плоскости

Одним из самых простых и эффектных методов является имитация броуновского движения частиц. Ограничимся 2-D случаем, когда частица может перемещаться в плоскости (x, y) . Для имитации достаточно к координатам x и y каждый раз прибавлять случайные числа (рис. 11.7). Для этого используется функция генерации случайных чисел с нормальным распределением `rndnm`.

При каждом пуске этого документа получается новая картина из 500 перемещений частицы. Легко понять, что этот вид движения частиц способствует смешению разных фрагментов растворов. Вы можете самостоятельно составить документ, в котором будет имитироваться движение частиц в пространстве, но это уже сложная задача, поскольку направления перемещения обычно не могут быть произвольными.

Хотя этот пример самый простой в этой главе, он демонстрирует моделирование при случайных приращениях параметров. Все другие примеры основаны на фиксированном изменении значения независимой переменной, например, времени.

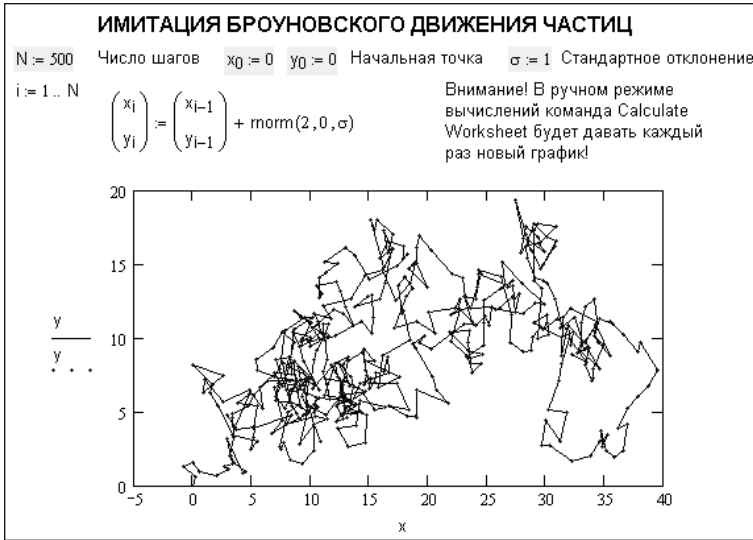


Рис. 11.7. Имитация броуновского движения частиц в плоскости

11.4.2. Имитация броуновского движения частиц в пространстве

Рисунок 11.8 иллюстрирует более сложный случай моделирования броуновского движения частиц в пространстве. Здесь приходится учитывать, что наличие молекул приводит к существованию различных допустимых направлений движения частиц. В представленном документе это учтено заданием матрицы Step возможных шагов.



Рис. 11.8. Имитация броуновского движения частиц в пространстве

Обратите внимание на то, что каждый новый пуск данного документа порождает новую картину диффузии.

Отсутствие времени в решении задач броуновского движения не случайно. Дело в том, что это движение обладает удивительным свойством — его характер практически не зависит от времени. Разумеется, в определенных пределах, если расстояния, проходимые частицами, заметно превышают размеры атомов и молекул и меньше размеров нашей планеты или вселенной. Можно во много раз уменьшить время наблюдения, и при этом мы будем видеть по-прежнему хаотические движения частиц, хотя и на меньшем расстоянии, чем при значительном времени наблюдения. Для таких процессов говорят, что они *самоподобны* [17].

11.4.3. Моделирование диффузии

Еще один красивый пример моделирования физико-химических явлений, которые мы наблюдаем повседневно — анализ диффузии вещества тонкой пластины в вещество раствора, в который опущена пластина (рис. 11.9).

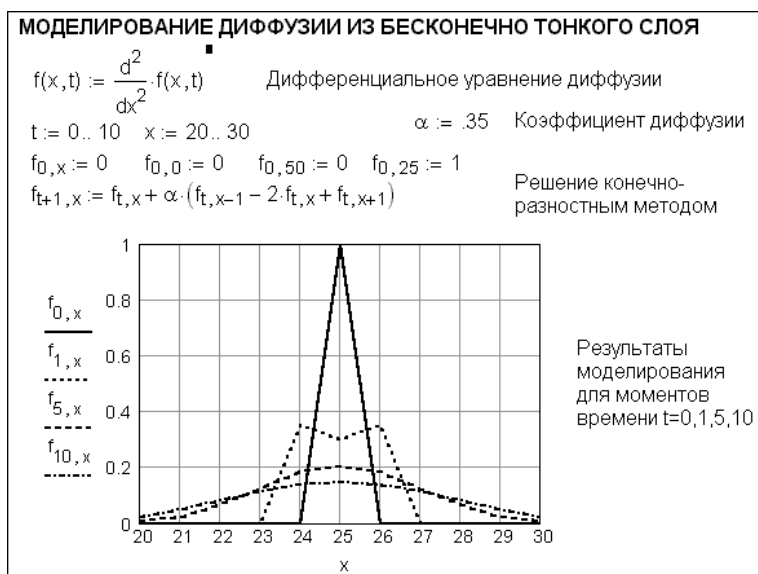


Рис. 11.9. Моделирование диффузии из тонкого слоя

Диффузия описывается дифференциальным уравнением второго порядка в частных производных. Это уравнение представлено в начале документа и помечено знаком ■. Он означает, что эта формула явно не используется и записана как комментарий. На самом деле уравнение решается конечно-разностным методом, и такое решение позволяет строить графически распределение концентрации диффундирующего материала в материал раствора в разные моменты времени. Хорошо видно, как диффундирующий материал постепенно растворяется и перемещается по обе стороны от первоначального расположения.

Немного сложнее решение задачи на моделирование диффузии вещества из толстой пластины. Ее решение представлено на рис. 11.10.

Приведенные примеры дают наглядное представление о том, что даже в отсутствии средств решения дифференциальных уравнений в частных производных ре-

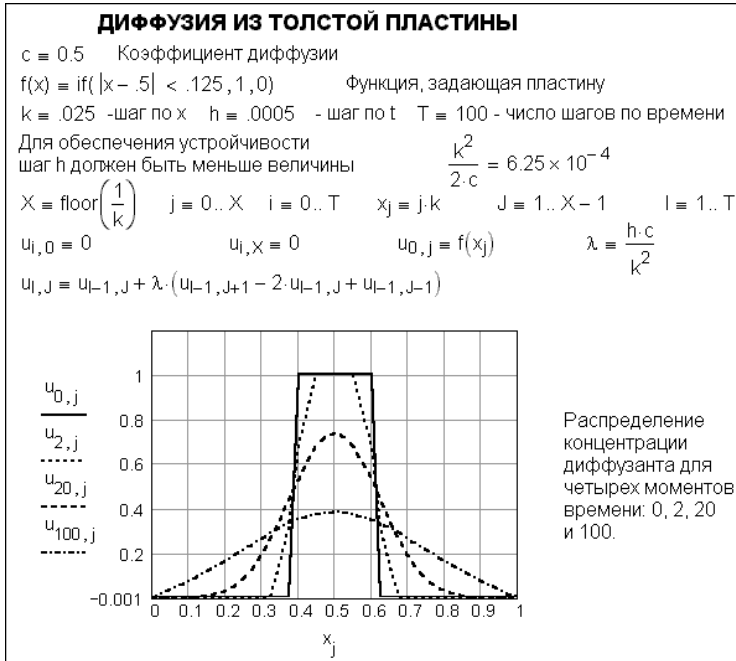


Рис. 11.10. Моделирование диффузии из толстого слоя

шение некоторых задач, таких как диффузия, возможно конечно-разностными методами. Напоминаем, что в новейшей версии Mathcad 11 уже есть встроенная функция `pdsolve` для решения подобных задач.

11.4.4. Моделирование торможения автомобиля

Многие из нас имеют автомобиль. Когда, заметив пешехода или собаку, перебегающих дорогу, мы жмем на тормоза, то нас волнует один вопрос — как быстро наш автомобиль потеряет свою энергию и остановится. Решение этой задачи для автомобиля с дисковым тормозным устройством представлено документом рис. 11.11.

Процесс потери энергии в общем случае связан с решением нелинейного дифференциального уравнения. Физики любят использовать для такого решения простые конечно-разностные методы, при которых решение выражается рекуррентной формулой. Фактически они соответствуют решению простым методом Эйлера. Результаты решения представлены в конце рис. 11.11.

Важно отметить, что в данном расчете используются размерные величины (энергия определяется в джоулях). Для расчетов с размерными величинами в состав Mathcad входят файлы, содержащие глобальные определения размерных величин. В конце документа (рис. 11.11) показан пример использования такого файла. В нем содержится определение многих размерных величин, относящихся к физике (их больше, чем это нужно для нашего конкретного примера).

11.4.5. Моделирование падения парашютиста

Многие любят парашютный спорт. Бросок из самолета и перегрузки, возникающие при открытии парашюта, доставляют им неопишуемое наслаждение и повы-

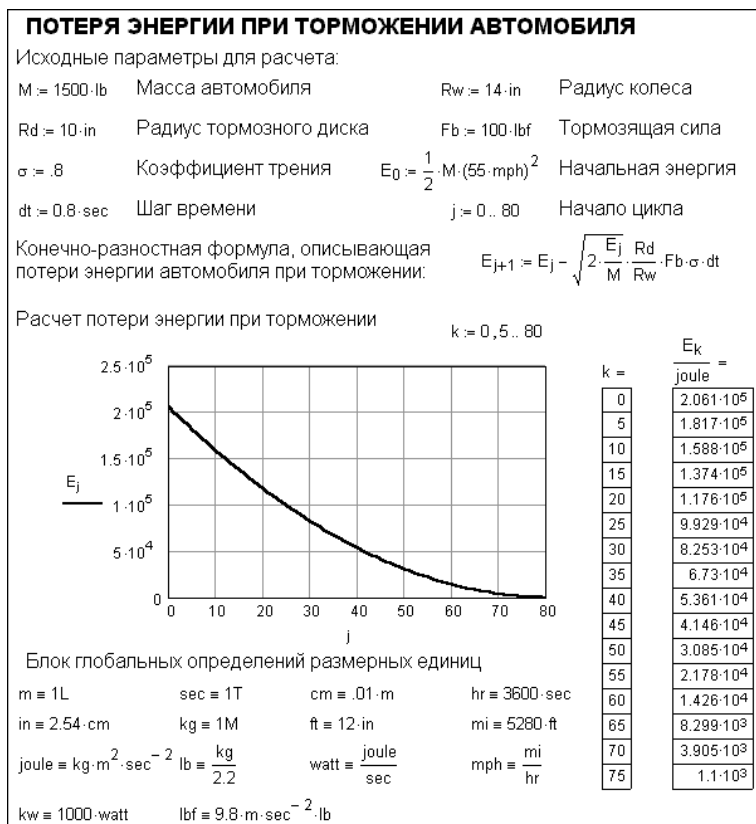


Рис. 11.11. Расчет потери энергии тормозящего автомобиля

шают содержание адреналина в крови. Поэтому таких читателей заинтересует моделирование процесса падения парашютиста, описанное ниже.

Пусть на высоте y_0 из самолета выпрыгивает парашютист и в свободном полете падает вниз. Спустя 5 секунд он дергает кольцо парашюта и испытывает заметные перегрузки из-за раскрытия парашюта и резкого снижения скорости падения. Моделирование падения парашютиста основывается на решении нелинейной системы дифференциальных уравнений и требует учета сопротивления воздушной среды до и после открытия парашюта. Оно выполнено в документе рис. 11.12 с применением конечно-разностного метода решения.

В результате моделирования получены следующие зависимости (рис. 11.13): зависимость высоты парашютиста от времени, зависимость скорости и ускорения падения от времени и зависимость резкости (второй производной от скорости) падения от времени. Резкость падения характеризуют перегрузки, которые испытывает парашютист в полете. Они резко возрастают сразу после раскрытия парашюта, и это прекрасно известно всем, кто прыгал с парашютом.

Этот пример дает наглядное представление о применении размерных величин.

11.4.6. Моделирование эффекта Доплера

Когда мимо нас мчится поезд, мы замечаем, что при его приближении тональность гудка локомотива повышается, а при удалении поезда — понижается. Это

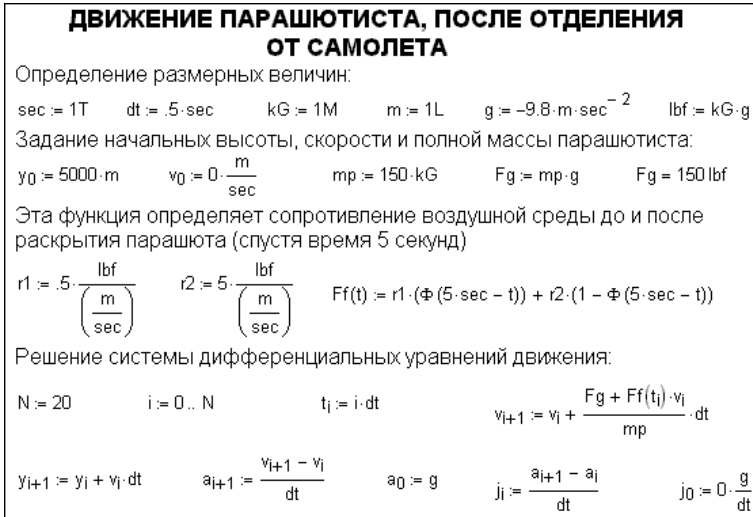


Рис. 11.12. Моделирование падения парашютиста (начало документа)

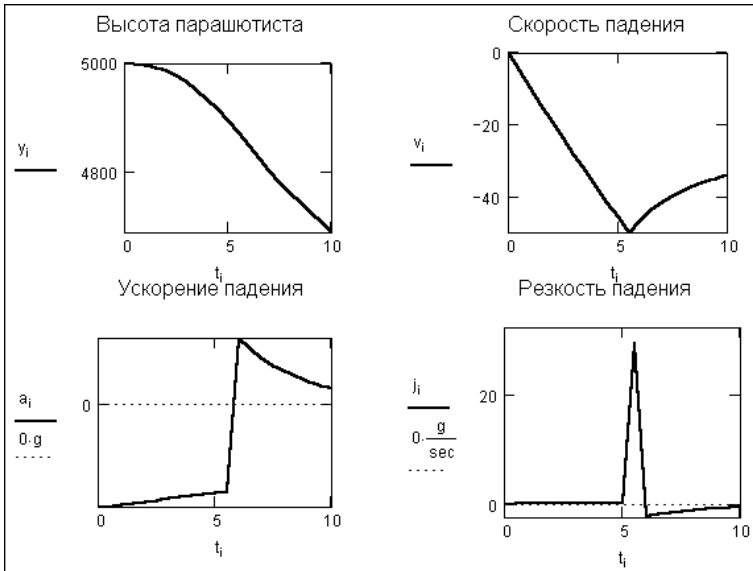


Рис. 11.13. Данные о падении парашютиста (конец документа рис. 11.13)

можно трактовать как «сгущение» волн звука при приближении поезда и их «разрежение» при его удалении. Данный эффект в физике получил название эффекта Доплера.

На рис. 11.14 представлен начальный фрагмент документа Mathcad с иллюстрацией возникновения эффекта Доплера. Здесь показан случай проявления эффекта Доплера в двумерном пространстве — классический пример движения поезда.

На рисунке даны основные теоретические вычисления, относящиеся к эффекту Доплера в двумерном и трехмерном пространстве. Для двумерного пространства дается оценка частот f_H и f_L для случаев приближения поезда и его удаления

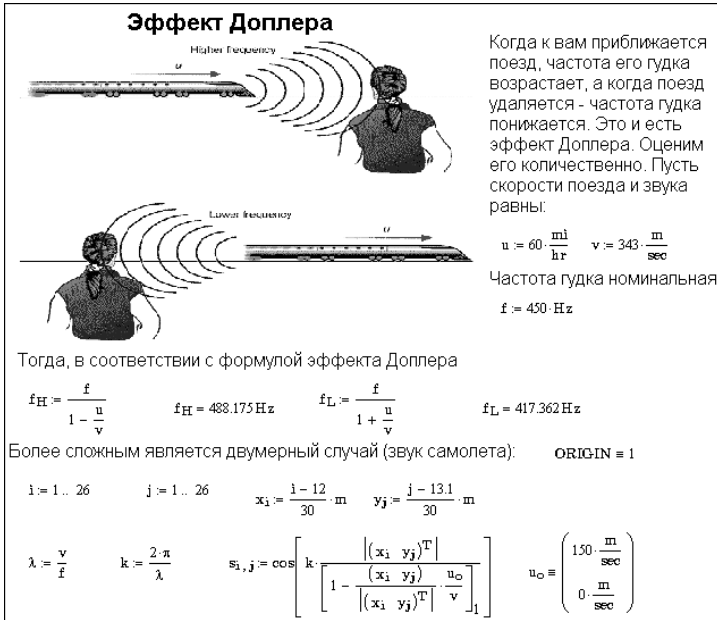


Рис. 11.14. Иллюстрация эффекта Доплера в двумерном и трехмерном случаях

при номинальной частоте гудка $f = 450 \text{ Гц}$. Нетрудно заметить, что изменение частоты достаточно заметно для ее оценки на слух.

Более сложен случай проявления эффекта Доплера в трехмерном пространстве. Соответствующие расчеты даны в нижней части рис. 11.15. Завершающий фрагмент документа Mathcad с пространственной диаграммой изменения скорости звука при проявлении эффекта Доплера в пространстве показан на рис. 11.15.

Эффект Доплера относится к фундаментальным понятиям физики. Он проявляется в изменении частоты не только звуковых, но и иных волновых колебаний.

К сожалению, рис. 11.15 демонстрирует наличие «ложки дегтя» в той «бочке» возможностей, которую дает Mathcad 2001, — рисунок поверхности получается

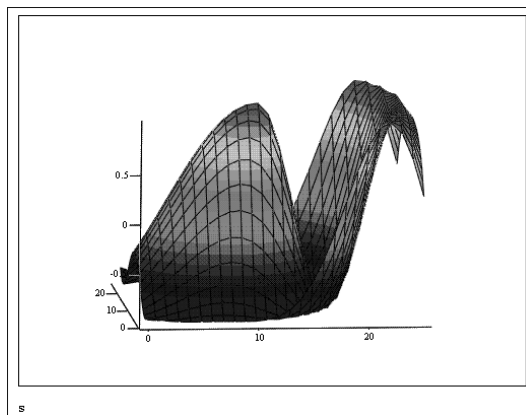


Рис. 11.15. Пространственная диаграмма проявления эффекта Доплера в трехмерном случае

слепым. Mathcad не позволяет вводить аннотации по осям трехмерного графика, а потому интерпретация назначения осей остается за пользователем. Ситуация усложняется еще и тем, что вращение графика мышью может привести к смене исходного положения осей, что запутывает интерпретацию результатов вычислений.

11.4.7. Обтекание крыла самолета воздухом

Многие с трудом понимают, как крылья самолета могут поднимать в воздух такую машину. Между тем существование подъемной силы крыла можно легко проверить на практике — высуньте согнутую на манер крыла ладонь из мчавшегося автомобиля или вагона поезда. Вы заметите, как напор воздуха стремится поднять ладонь руки. Но можно описать это и конечно-разностной моделью, представленной на рис. 11.16 и прекрасно иллюстрирующей возможности еще первых версий системы Mathcad.



Рис. 11.16. Моделирование профиля крыла и его обтекания потоком воздуха

Эта модель слишком специфична, чтобы подробно описывать ее в данной книге. Она понятна аэродинамикам, а для прочих «простых смертных» является наглядной иллюстрацией причин возникновения подъемной силы. Из-за разного расстояния, которое проходит воздушный поток над и под крылом, возникает разница воздушного давления. Она и создает подъемную силу крыла.

11.5. Моделирование колебательных систем

11.5.1. Анализ линейной колебательной системы

Известно множество *линейных систем* и *устройств*, создающих почти синусоидальные колебания — самые простые из известных. Это струна музыкальных инструментов, маятник часов, LCR-колебательный контур, колеблющаяся молекула

вещества и т. д. Все эти устройства и системы при малых амплитудах колебаний можно описать линейным дифференциальным уравнением второго порядка, вид которого представлен в заголовке рис. 11.17. Там же даны типичные решения этого уравнения с помощью блока Given и функции Odesolve системы Mathcad.

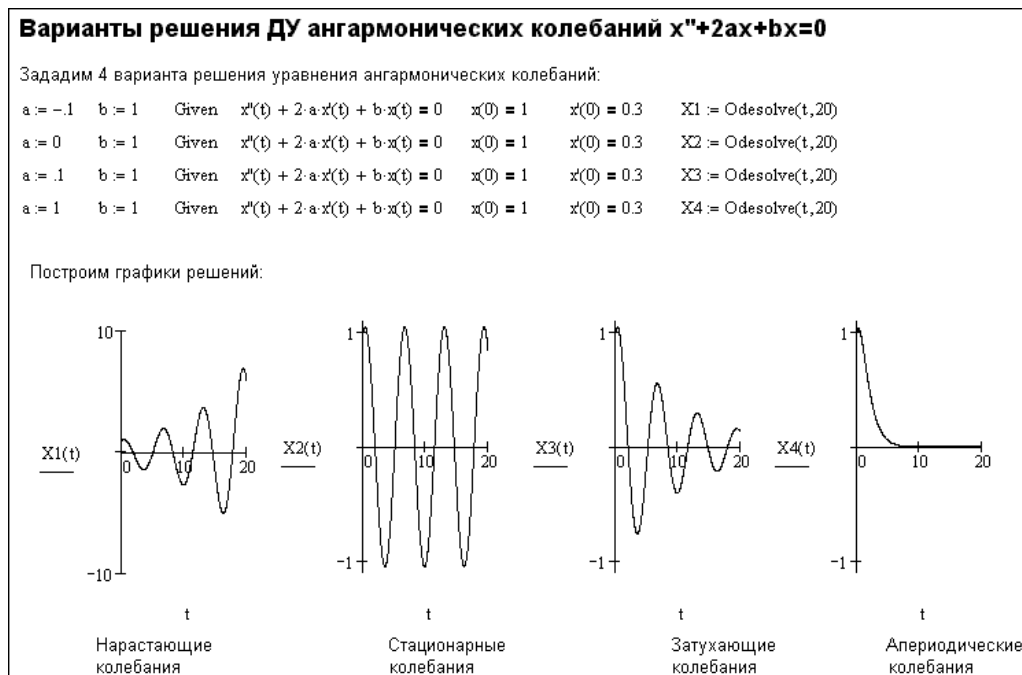


Рис. 11.17. Решения дифференциального уравнения второго порядка, описывающего поведение линейных колебательных систем

Поведение линейной системы сильно зависит от параметра a — затухания. При его отрицательных значениях амплитуда колебаний нарастает по экспоненциальному закону. При $a = 0$ создаются незатухающие синусоидальные колебания. Однако этот процесс нестабилен — малейшее изменение в ту или иную сторону приводит либо к нарастанию колебаний, либо к их затуханию. При больших положительных a (теоретически $a > 0.25$) переходный процесс в системе становится аperiодическим. Все эти случаи можно анализировать аналитически, но численный метод решения с помощью функции Odesolve намного проще и нагляднее.

11.5.2. Анализ нелинейной колебательной системы Ван—дер—Поля

А теперь рассмотрим поведение нелинейной колебательной системы второго порядка. Характер нелинейности системы может быть самым различным. Классическим стал анализ нелинейных систем, описываемых нелинейным дифференциальным уравнением второго порядка — уравнением Ван—дер—Поля. Рис. 11.18 показывает документ системы Mathcad, в котором такое уравнение решается при параметре $\mu = 0,5$. Этот параметр задает характер решения, как и начальные условия для $x(t)$ и $dx(t)/dt$. При положительных μ колебания в системе нарастают, но вследствие нелинейности системы их амплитуда ограничивается, а форма становится заметно отличной от синусоидальной. Решение выполняется с помощью функции Odesolve.

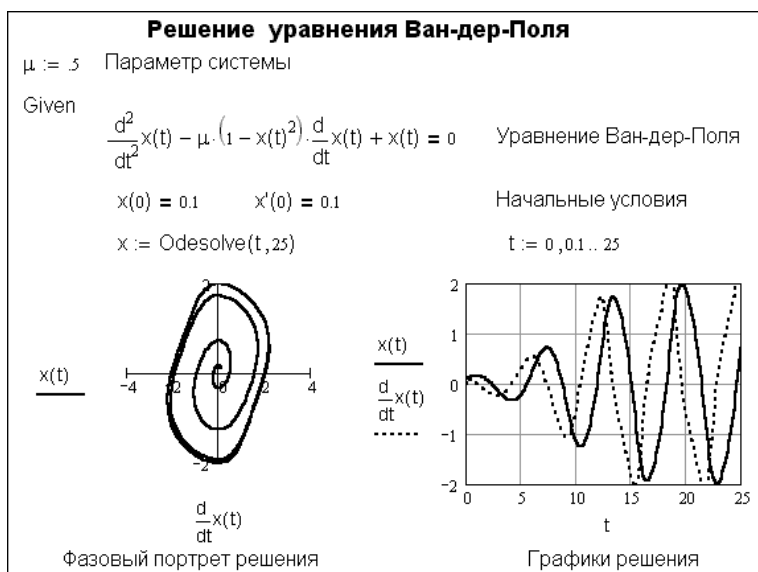


Рис. 11.18. Решение уравнения Ван—дер—Поля

Дифференциальное уравнение второго порядка можно разбить на два уравнения первого порядка. Этот случай решения уравнения Ван—дер—Поля был представлен на рис. 7.26 и 7.27. Оба варианта решения совершенно равноценны. В случае, показанном на рис. 7.26, поведение системы моделируется при отрицательном параметре $\mu = -0,1$. В этом случае возникшие вначале колебания затухают во времени. А случай рис. 7.27 отличается от рис. 11.18 еще и разными начальными условиями.

Системы, колебания в которых возникают без внешних воздействий, принято называть *автономными системами*. Помимо систем класса Ван—дер—Поля к ним относятся и описанный выше генератор колебаний на туннельном диоде и большинство автогенераторов синусоидальных и релаксационных колебаний. В частности выполненных на электронных лампах и полевых транзисторах.

11.5.3. Моделирование системы Даффинга с внешним воздействием

Поведение неавтономных нелинейных систем второго порядка, находящихся под внешним воздействием, может быть очень сложным. В этом можно убедиться на примере системы Даффинга, описывающей процессы в нелинейных резонаторах с внешним воздействием — например в лазерных резонаторах. Пример численного моделирования процессов в такой системе дан на рис. 11.19.

Дифференциальное уравнение Даффинга второго порядка имеет дополнительный кубический член в левой части, а правая часть представляет внешнее косинусоидальное воздействие. Форма колебаний такой системы довольно сложна из-за наложения внутренних колебаний на внешние, причем частоты колебаний сильно различаются. В итоге время от времени может наступать автосинхронизация колебаний, но из-за нелинейности системы и изменения амплитуды собственных колебаний может наблюдаться срыв синхронизации, сопровождаемый скачкообразными и довольно хаотическими изменениями параметров системы. Тем не менее

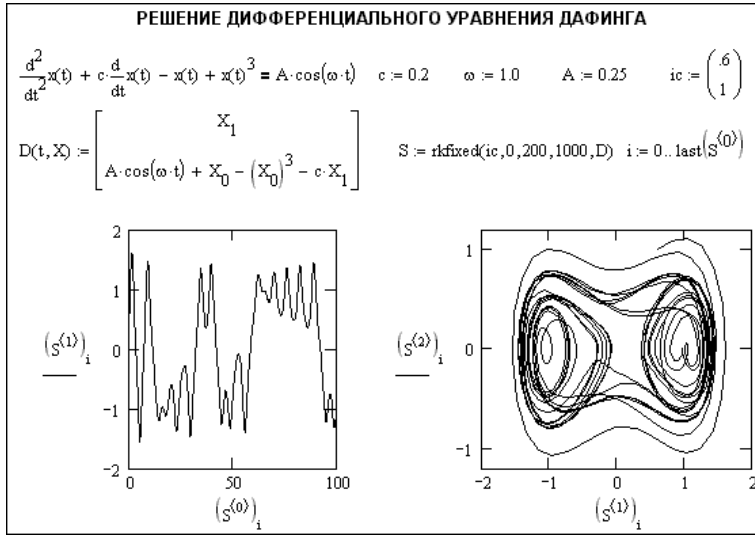


Рис. 11.19. Решение уравнения Даффинга

фазовый портрет системы имеет два фокуса, соответствующих более низкочастотной компоненте колебаний. Эти фокусы соответствуют статистической оценке наиболее вероятных видов (*мод*) колебаний.

11.5.4. Хаос и моделирование аттрактора Лоренца

Броуновское движение частиц, моделирование которого мы уже провели, и колебания в системе Даффинга является проявлениями *хаоса* в природе. Наблюдая за изменениями курса акций, сходами ледников и снежных лавин или за колебаниями температуры, мы нередко убеждаемся в том, что наряду с вполне предсказуемыми изменениями того или иного параметра (например, повышения температуры летом и понижения зимой) нередко наблюдаются хаотические изменения, которые трудно или невозможно заранее предвидеть.

Иногда «развал» казалось бы устойчивой системы приводит к резким изменениям ее поведения — наш «черный вторник» или обвал рубля в 1988 году тому наглядные примеры. Как и крупный террористический акт в центре одной из самых стабильных стран мира — США. Существует достаточно обоснованное мнение, что хаотическое поведение систем куда больше характерно для природы, чем стационарное, происходящее с неизменяемыми во времени параметрами. Так что хаос стал одним из важных объектов изучения современной наукой. Его моделирование осуществляется на основе численных методов.

Чем сложнее система и чем большим количеством дифференциальных уравнений она описывается, тем больше вероятность возникновения в системе хаотических режимов — даже если она автономна. Изучение этого вопроса показало, что уже в системах из трех дифференциальных уравнений возможно возникновение хаотических режимов даже в автономных системах. Наглядным примером этого является аттрактор Лоренца, который мы уже рассмотрели как пример программирования — см. рис. 10.7 и 10.8.

Другой пример моделирования аттрактора Лоренца представлен на рис. 11.20. Тут аттрактор Лоренца описан уже как система из трех дифференциальных урав-

нений. При определенных значениях параметров σ и b и начальных параметров переменных поведение аттрактора (он в этом случае называется *странным аттрактором*) очень напоминает хаотические колебания в системе Дафинга.

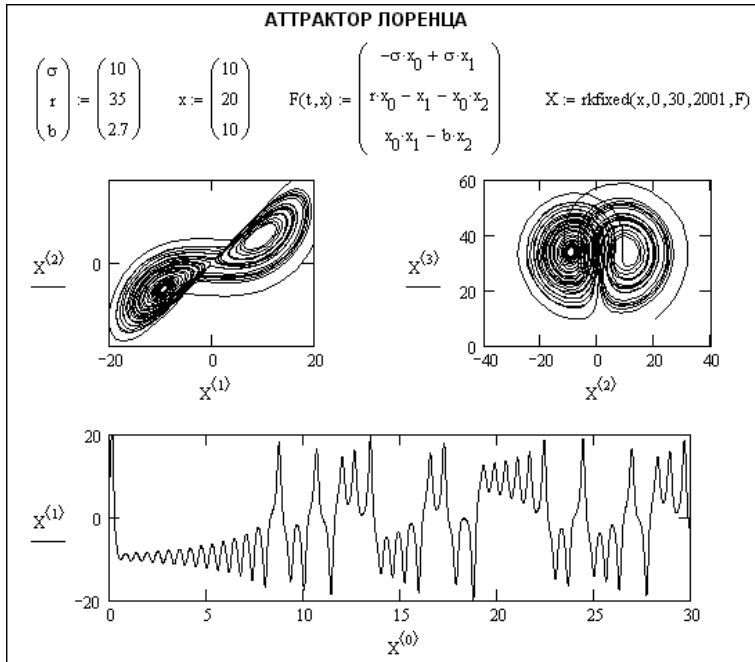


Рис. 11.20. Моделирование аттрактора Лоренца, заданного системой из трех дифференциальных уравнений

Аттрактором в теории колебаний называется притягивающая область в фазовом пространстве. Причины неустойчивости аттракторов связаны с экспоненциальной неустойчивостью системы в малых областях фазового пространства. При этом наблюдаются хаотические переходы из одной области фазового пространства в другие области, но при этом колебания могут не выходить из некоторой более обширной области фазового пространства. «Обвал» системы означает переход в некоторое состояние, резко отличающееся от других состояний, т. е. выход за пределы ограниченного фазового состояния системы. Такое состояние может оказаться устойчивым и приводит к переходу системы в статическое состояние, при котором изменения ее параметров отсутствуют.

11.6. Моделирование фундаментальных физических явлений

11.6.1. Моделирование математического маятника с анимацией

Но вернемся к системам, поведение которых все же предсказуемо и описывается решением систем дифференциальных уравнений, описывающих состояния системы — они так и именуется *уравнениями состояния*. И рассмотрим классиче-

скую физическую задачу на колебания математического маятника. Физически это камень или иной тяжелый предмет, подвешенный на веревке, закрепленной сверху. Маятник настольных часов тоже хорошая иллюстрация на применение математического маятника в реальном устройстве.

Рисунок 11.21 содержит (в первой его половине) постановку задачи на колебания маятника. Они описываются нелинейным дифференциальным уравнением, что существенно осложняет анализ колебаний. Однако при малой амплитуде колебаний система линеаризируется и мы сравнительно легко получаем уравнение колебаний в операторной форме.

Математический маятник

Из фундаментальных уравнений для моментов движения:

$$\frac{d}{dt} l_z = \sum_{k=1}^n m_k (F_k) \quad \sum_{k=1}^n m_k (F_k) = -M \cdot g \cdot l \cdot \sin(\phi) \quad l_z = M \cdot \omega_{\tau} \cdot l$$

$$\omega_{\tau} = l \cdot \dot{\phi} \quad l_z = M \cdot l^2 \cdot \dot{\phi} \quad \frac{d}{dt} l_z = M \cdot l^2 \cdot \frac{d^2}{dt^2} \phi \quad M \cdot l^2 \cdot \frac{d^2}{dt^2} \phi(t) = -M \cdot g \cdot l \cdot \sin(\phi)$$

получим нелинейное дифференциальное уравнение колебаний маятника:

$$\frac{d^2}{dt^2} \left(\phi(t) + \frac{g}{l} \right) \cdot \sin(\phi) = 0$$

Разложив $\sin(\phi)$ в ряд:

$$\sin(\phi) \text{ series, } \phi = 0 \rightarrow 1 - \frac{1}{6} \phi^3 + \frac{1}{120} \phi^5$$

Поскольку угол отклонения маятника намного меньше величины 2π , то при $\sin \approx f$ имеем:

$$\frac{d^2}{dt^2} \phi(t) + \frac{g}{l} \phi(t) = 0$$

Введя обозначения $L = \text{laplace}(f(t), t, s)$, $C_1 = f(0)$, $C_2 = \text{diff}(f(0), 0)$, имеем:

$$C_1 = \phi(0) \quad C_2 = \frac{d}{dt} \phi(0) \quad L \cdot s^2 - C_1 \cdot s - C_2 + \frac{g}{l} \cdot L = 0$$

Решая это уравнение относительно переменной L имеем:

$$L \cdot s^2 - C_1 \cdot s - C_2 + \frac{g}{l} \cdot L \text{ solve, } L \rightarrow (C_1 \cdot s + C_2) \cdot \frac{1}{(s^2 \cdot l + g)}$$

Рис. 11.21. Постановка задачи на моделирование математического маятника

Решение этого уравнения при воздействии на маятник толчка представлено на рис. 11.22. Оно иллюстрируется графиком колебаний — отклонением угла нити от вертикального положения. Нетрудно заметить, что форма колебаний синусоидальна.

Колебания маятника легко проиллюстрировать графически, используя технику анимации графиков. Такие иллюстрации производят большое впечатление на учащихся школ и даже вузов и университетов. Рисунок 11.23 поясняет подготовку к анимации. Здесь показано уравнение движения груза (жирной точки) и окно подготовки фреймов (кадров) анимации. В этом окне, после выделения рисунка маятника и нажатия кнопки Animation, можно наблюдать подготовку фреймов (кадров) анимации.

По окончании подготовки фреймов появляется окно анимационного проигрывателя, показанное на рис. 11.24. В нем и можно наблюдать колебания маятника в пределах одного периода. При этом маятник отклоняется вправо на угол до 60° , а затем влево на угол до -60° и возвращается в исходное состояние (угол 270°). Ри-

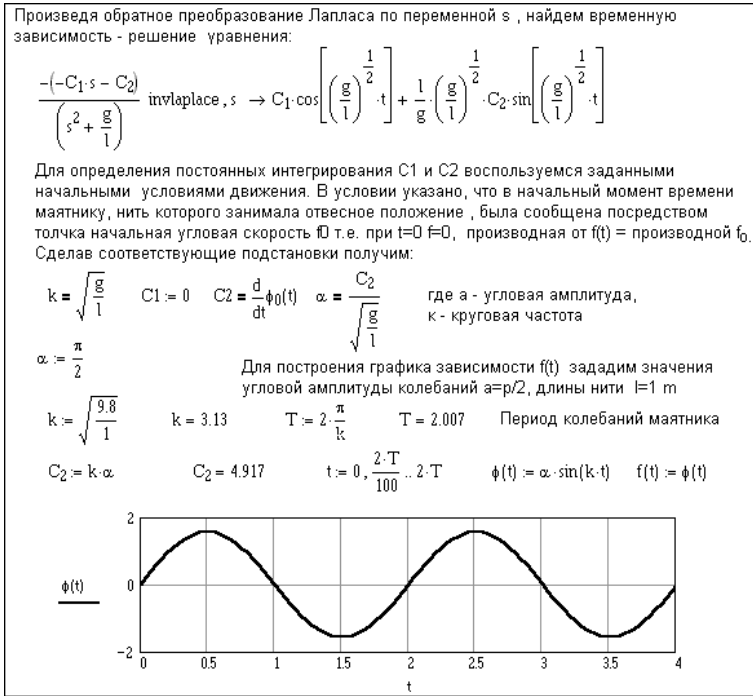


Рис. 11.22. Решение линеаризованного уравнения математического маятника и его графическое представление

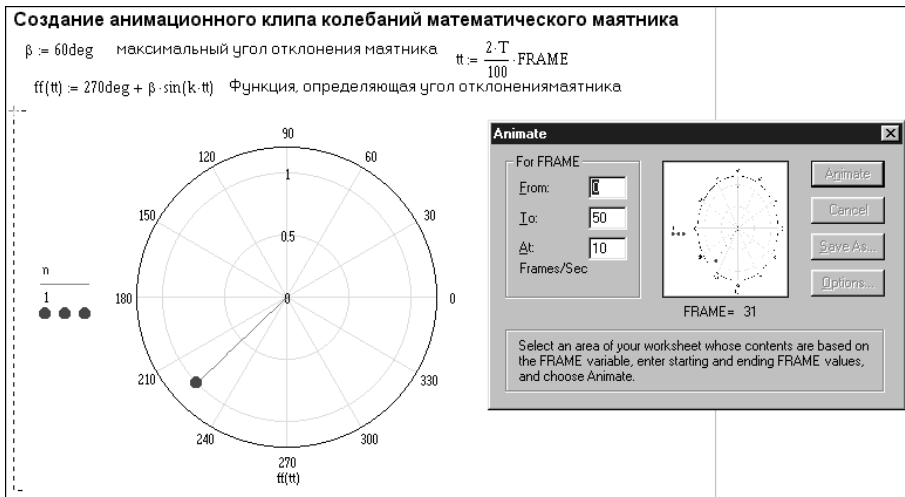


Рис. 11.23. Подготовка к анимации колебаний математического маятника

сунок 11.24 при этом иллюстрирует некоторый промежуточный кадр анимации (всего задано 50 фреймов на 1 цикл колебаний).

Но насколько линейная модель маятника отличается от более точной модели, основанной на решении нелинейных дифференциальных уравнений маятника?

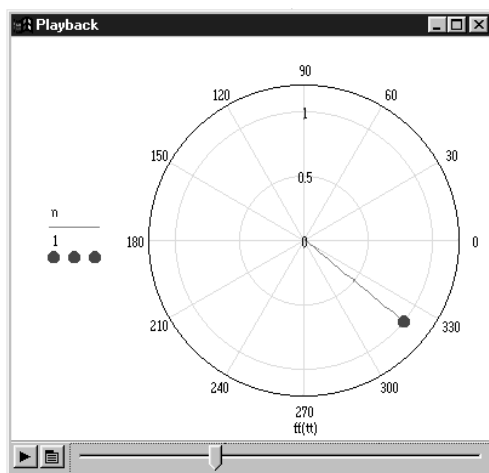


Рис. 11.24. Наблюдение колебаний маятника с помощью анимационного проигрывателя

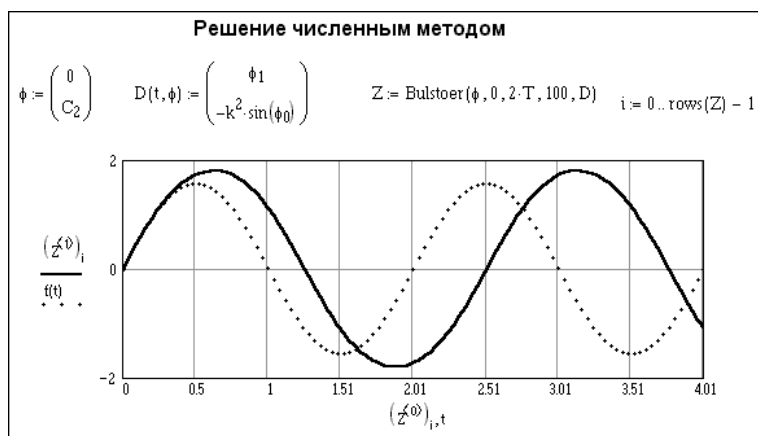


Рис. 11.25. Численное решение системы нелинейных дифференциальных уравнений математического маятника

Ответ на этот вопрос дает рис. 11.25, на котором представлено решение этих уравнений численным методом.

Нетрудно заметить, что решение по нелинейной модели качественно почти не отличается от решения по линейной модели — форма колебаний очень близка к синусоидальной. Однако амплитуда колебаний и их период заметно отличаются от значений, характерных для линейной модели.

11.6.2. Моделирование рассеивания альфа-частиц

Теперь рассмотрим пример из микромира, фундаментальное значение которого трудно переоценить. Речь идет о рассеивании альфа-частиц тонкой золотой фольгой.

Сейчас мы уже знаем, что материя состоит из атомов, которые имеют положительно заряженное «массивное» ядро, вокруг которого по орбитам движутся отрицательно заряженные электроны. Однако когда гипотеза об атомном строении ве-

щества была только-только высказана, не хватало одного важного момента — экспериментального доказательства существования атомов с их ядрами. В результате сложнейших (по тем временам) экспериментов это удалось доказать. Правда, не прямо, а косвенно (прямое доказательство было получено намного позже с помощью электронных микроскопов, позволивших наблюдать и фотографировать молекулы и атомы).

Ученые разумно предположили, что если бомбардировать тонкую пластину вещества положительно заряженными альфа-частицами, то они должны отклоняться от прямолинейного пути, отталкиваясь от ядра атомов при прохождении мимо ядер на близком расстоянии. Более того, расчеты показывали, что некоторые частицы могли даже отскакивать от ядра и менять направление полета, двигаясь в сторону источника альфа-частиц.

С помощью современных СКМ мы можем промоделировать эту ситуацию. Это поясняет начало документа системы Mathcad, представленное на рис. 11.26. Здесь сформулирована задача моделирования альфа-частицами тонкой золотой фольги и аналитически получены формулы для составляющих силы Лоренца F (силы отталкивания альфа-частицы от ядра атома).



Рис. 11.26. Первая часть документа, моделирующего рассеивание альфа-частиц

На второй части документа (рис. 11.27) представлена подготовка дифференциальных уравнений, описывающих траекторию движения альфа-частиц, и их решение с помощью функции Bulstoer. Решение для нескольких вариантов «запуска» альфа-частиц с разными расстояниями от центра ядра хранится в матрице Z.

На третьей части документа (рис. 11.28) построены графики траекторий альфа-частиц. Моделирование блестяще подтверждает предсказанное поведение частиц — часть их испытывает отклонение от прямой их полета, которая была бы в случае отсутствия ядра. А некоторые частицы даже испытывают изменение направления движения на прямо противоположное. Датчики частиц, установленные в разных местах физической установки, и впрямь фиксировали такие изменения траектории, что и явилось подтверждением атомной теории строения веществ и материалов.



Рис. 11.27. Вторая часть документа, моделирующего рассеивание альфа-частиц

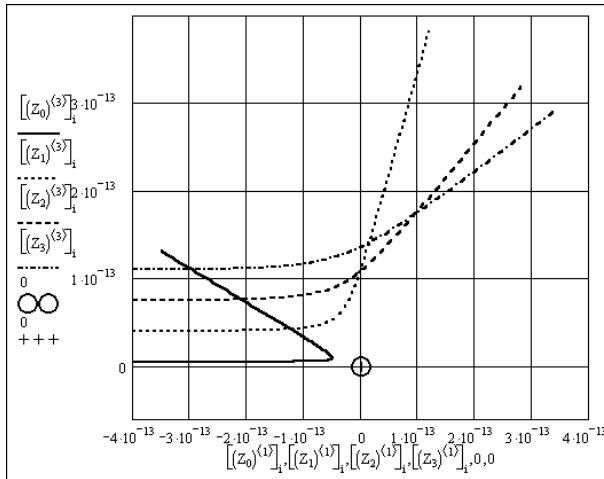


Рис. 11.28. Третья часть документа, моделирующего рассеивание альфа-частиц

Разумеется, и описанная модель является не абсолютно точной. Например, значительные погрешности могут возникнуть, если направить частицу точно в ядро. Если скорость частицы велика, частица может преодолеть силу отталкивания ядром и будет поглощена им. Новые обстоятельства такого поглощения (вплоть до радиоактивного распада вещества) описанная модель просто уже не учитывает.

11.6.3. Разделение изотопов

В ядерной физике *изотопами* называют атомы, имеющие одинаковый электрический заряд, но разную массу. Одной из важных задач ядерной физики является разделение изотопов. Рисунок 11.29 содержит начало документа, посвященного

описанию масс-спектроскопического метода разделения изотопов. Этот метод основан на особенностях траектории ионов, вылетающих из точки А и отклоняемых магнитным полем. Система дифференциальных уравнений, описывающих движение ионов, представлена на рис. 11.29.

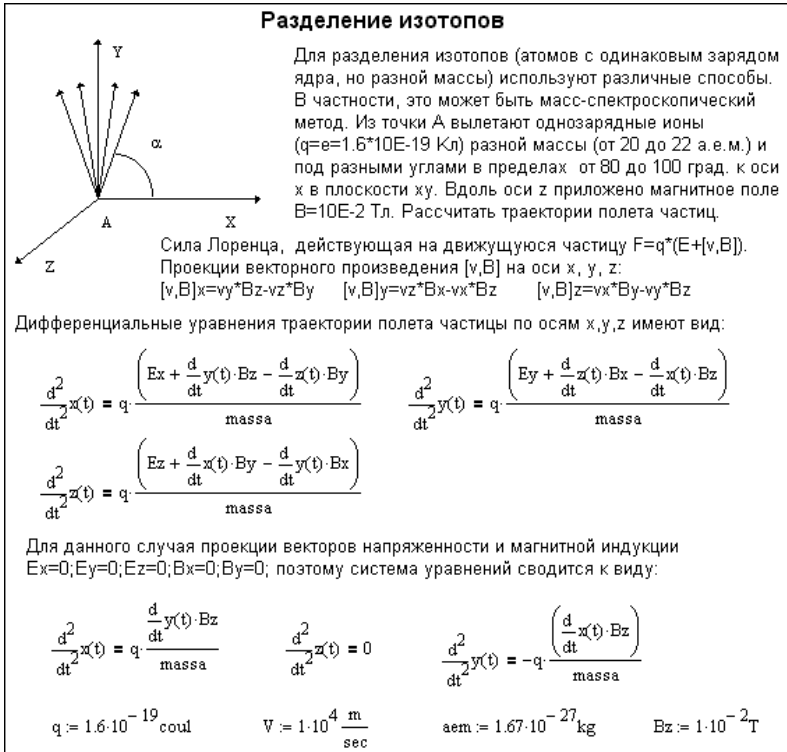


Рис. 11.29. Первая часть документа, моделирующего масс-спектроскопический метод разделения изотопов

Во второй (рис. 11.30) и третьей (рис. 11.31) частях документа содержатся вектора и решения дифференциальных уравнений движения ионов (изотопов) для случаев двух углов, под которыми вылетают частицы, и трех значений масс их.

Заключительная четвертая часть документа (рис. 11.32) показывает построение траекторий движения ионов с разной массой, вылетающих под разными углами.

Как видно из рис. 11.32, ионы одинаковой массы, вылетающие под разными углами, «собираются» в одну точку. Остается, как говорится, «подставить стаканчик» и собрать нужные изотопы.

11.6.4. Движение частиц в магнитном поле

Многие, даже достаточно сложные задачи физики имеют аналитические решения. Такова задача на движение в равномерном магнитном поле частицы, имеющей электрический заряд. Итак, пусть частица с зарядом q и массой M влетает под углом α в область, где действует магнитное поле с напряженностью H . Рисунок 11.33 показывает построение траекторий для двух таких частиц с разной массой. Движение частиц происходит под действием силы Лоренца.

Рассмотрим случай для двух углов, под которыми вылетают частицы при трех значениях масс частиц:

$$j := 0..1 \quad \alpha_j := (80 + j \cdot 20) \text{ deg} \quad i := 1..3 \quad \text{massa}_i := (19 + i) \cdot \text{aem}$$

Проекция вектора скорости на оси координат:

$$V_{x_j} := V \cdot \cos(\alpha_j) \quad V_{y_j} := V \cdot \sin(\alpha_j)$$

Вектора начальных условий для двух углов вылета частиц:

$$f1 := \begin{pmatrix} 0 \\ V_{x_0} \cdot 1 \frac{\text{sec}}{\text{m}} \\ 0 \\ V_{y_0} \cdot 1 \frac{\text{sec}}{\text{m}} \end{pmatrix} \begin{matrix} x(0) \\ \frac{d}{dt} x(0) \\ y(0) \\ \frac{d}{dt} y(0) \end{matrix}$$

$$f2 := \begin{pmatrix} 0 \\ V_{x_1} \cdot 1 \frac{\text{sec}}{\text{m}} \\ 0 \\ V_{y_1} \cdot 1 \frac{\text{sec}}{\text{m}} \end{pmatrix} \begin{matrix} x(0) \\ \frac{d}{dt} x(0) \\ y(0) \\ \frac{d}{dt} y(0) \end{matrix}$$

$$D1(t, f1) := \begin{pmatrix} f1_1 \\ f1_3 \cdot Bz \\ q \frac{\text{massa}_1}{\text{massa}_1} \\ f1_3 \\ f1_1 \cdot Bz \\ -q \frac{\text{massa}_1}{\text{massa}_1} \end{pmatrix} \begin{matrix} x' \\ x'' \\ y' \\ y'' \end{matrix}$$

$$D2(t, f2) := \begin{pmatrix} f2_1 \\ f2_3 \cdot Bz \\ q \frac{\text{massa}_1}{\text{massa}_1} \\ f2_3 \\ f2_1 \cdot Bz \\ -q \frac{\text{massa}_1}{\text{massa}_1} \end{pmatrix} \begin{matrix} x' \\ x'' \\ y' \\ y'' \end{matrix}$$

$$Z1 := \text{Bulstoer}(f1, 0, 10 \cdot 10^{-5}, 100, D1)$$

$$Z2 := \text{Bulstoer}(f2, 0, 10 \cdot 10^{-5}, 100, D2)$$

Рис. 11.30. Вторая часть документа, моделирующего масс-спектрометрический метод разделения изотопов

$$D3(t, f1) := \begin{pmatrix} f1_1 \\ f1_3 \cdot Bz \\ q \frac{\text{massa}_2}{\text{massa}_2} \\ f1_3 \\ f1_1 \cdot Bz \\ -q \frac{\text{massa}_2}{\text{massa}_2} \end{pmatrix} \begin{matrix} x' \\ x'' \\ y' \\ y'' \end{matrix}$$

$$D4(t, f2) := \begin{pmatrix} f2_1 \\ f2_3 \cdot Bz \\ q \frac{\text{massa}_2}{\text{massa}_2} \\ f2_3 \\ f2_1 \cdot Bz \\ -q \frac{\text{massa}_2}{\text{massa}_2} \end{pmatrix} \begin{matrix} x' \\ x'' \\ y' \\ y'' \end{matrix}$$

$$Z3 := \text{Bulstoer}(f1, 0, 10 \cdot 10^{-5}, 100, D3)$$

$$Z4 := \text{Bulstoer}(f2, 0, 10 \cdot 10^{-5}, 100, D4)$$

$$D5(t, f1) := \begin{pmatrix} f1_1 \\ f1_3 \cdot Bz \\ q \frac{\text{massa}_3}{\text{massa}_3} \\ f1_3 \\ f1_1 \cdot Bz \\ -q \frac{\text{massa}_3}{\text{massa}_3} \end{pmatrix} \begin{matrix} x' \\ x'' \\ y' \\ y'' \end{matrix}$$

$$D6(t, f2) := \begin{pmatrix} f2_1 \\ f2_3 \cdot Bz \\ q \frac{\text{massa}_3}{\text{massa}_3} \\ f2_3 \\ f2_1 \cdot Bz \\ -q \frac{\text{massa}_3}{\text{massa}_3} \end{pmatrix} \begin{matrix} x' \\ x'' \\ y' \\ y'' \end{matrix}$$

$$Z5 := \text{Bulstoer}(f1, 0, 10 \cdot 10^{-5}, 100, D5)$$

$$Z6 := \text{Bulstoer}(f2, 0, 10 \cdot 10^{-5}, 100, D6)$$

Рис. 11.31. Третья часть документа, моделирующего масс-спектрометрический метод разделения изотопов

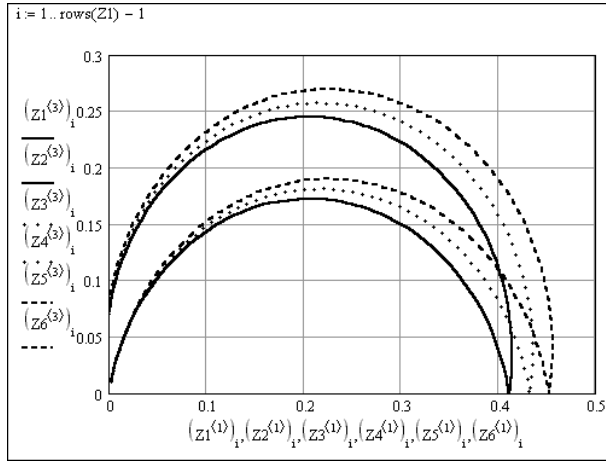


Рис. 11.32. Четвертая часть документа, моделирующего масс-спектроскопический метод разделения изотопов с построением траекторий полета ионов

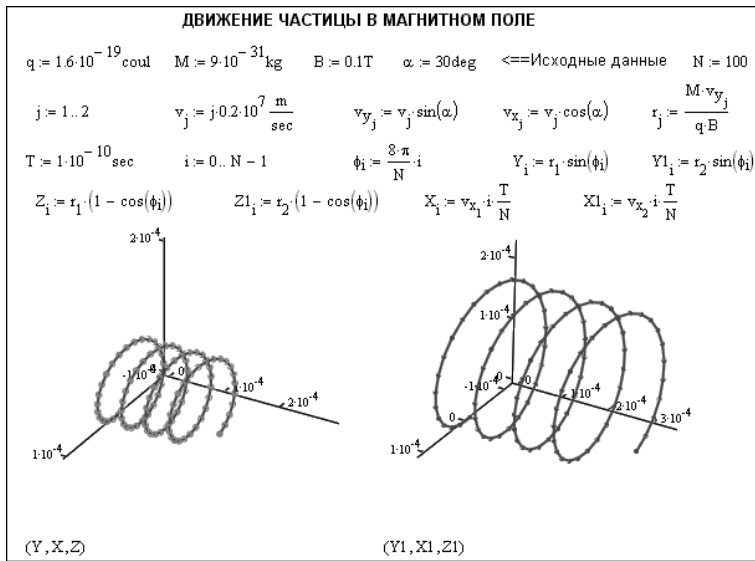


Рис. 11.33. Движение частиц с электрическим зарядом в магнитном поле

Как видно из рис. 11.33, частицы вдоль оси X движутся по траектории, соответствующей пространственной спирали. Чем выше начальная скорость частицы, тем выше диаметр спирали. Этот пример дает хорошее представление о визуализации сложного движения в пространстве.

11.6.5. Моделирование дифракции на щели

Mathcad прекрасно подходит для моделирования и визуализации различных оптических явлений. К таким явлениям относится дифракция на щели. Пусть в темной комнате с затемненным окном имеется щель в окне, через которую в ком-

нату проникает луч света и падает на экран, установленный на расстоянии L от щели. Если щель тонка и свет обладает волновыми свойствами, то будет наблюдаться дифракция — явление изменения интенсивности света на поверхности экрана, вызванное интерференцией световых волн.

Явление дифракции хорошо изучено и формулы, описывающие его, есть в учебниках по физике и оптике. Воспользовавшись этими формулами, построим наглядную модель дифракции, вначале в двумерном приближении. Представленный на рис. 11.34 график наглядно отражает изменение интенсивности освещения в разрезе щели.

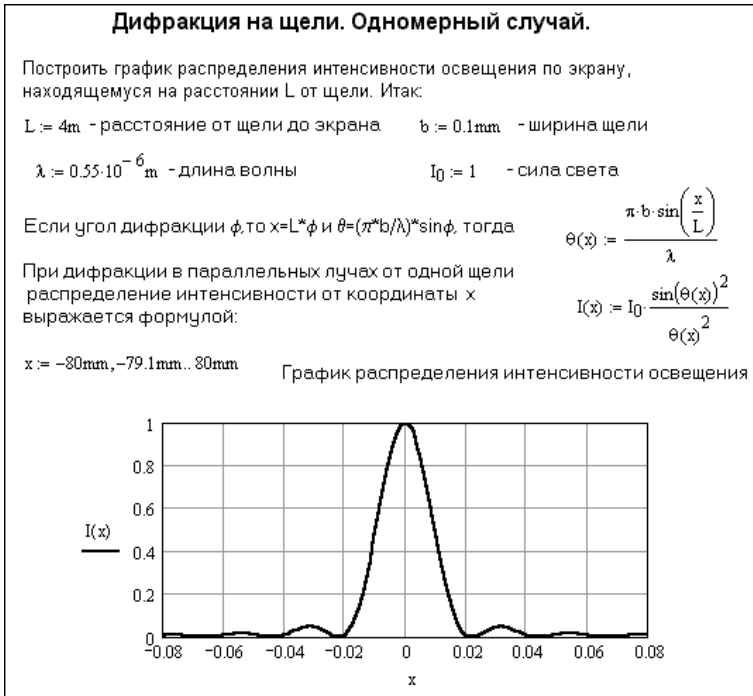


Рис. 11.34. Моделирование дифракции на щели (двумерный случай)

Можно повысить степень визуализации явления дифракции, перейдя к трехмерному случаю. Построения для него представлены на рис. 11.35.

Нетрудно заметить, что в рамках принятых допущений кривая распределения интенсивности света, приведенная на рис. 11.34, справедлива в любом сечении поверхности рис. 11.35, в которой интенсивность освещения характеризуется высотой поверхности. Иными словами, по длине щели интенсивность освещения неизменна и краевые эффекты на концах щели в рамках принятой модели просто не учитываются.

11.6.6. Интерференция света, излучаемого малым отверстием

Часто, наблюдая свет, излучаемый отверстием малого размера, мы видим кольца интерференции, именуемые кольцами Ньютона. Рисунок 11.36 иллюстрирует известные из оптики соотношения для распределения интенсивности света и фазы в функции от радиального расстояния от центра отверстия в любую сторону.

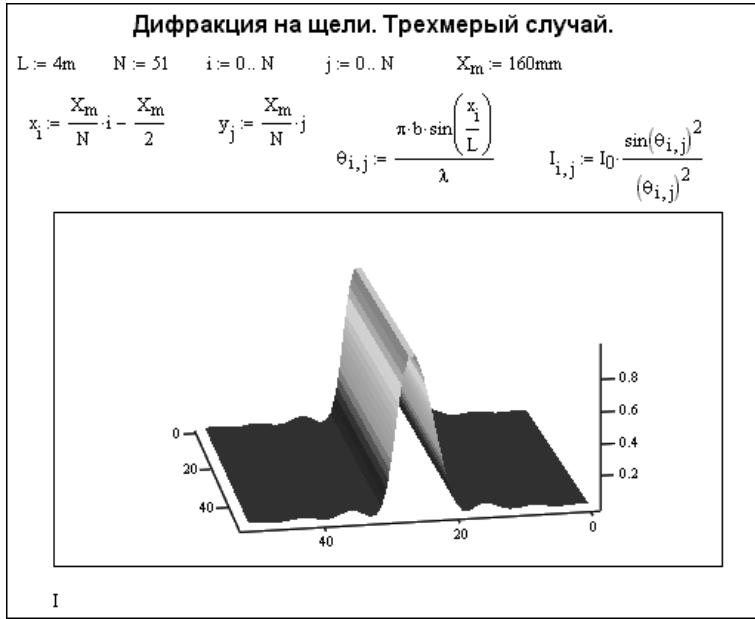


Рис. 11.35. Моделирование дифракции на щели (трехмерный случай)

Нетрудно заметить, что интенсивность света максимальна в центре кольца и испытывает колебания по мере удаления от него. Эти колебания напоминают смещенную вверх, на величину амплитуды, косинусоиду с нарастающей частотой.



Рис. 11.36. К интерференции света, излучаемого отверстием малого размера

Такой характер интерференции и порождает в плоскости экрана характерные кольца Ньютона. Их построение показано на рис. 11.37 для двух случаев — в виде контурного графика на плоскости и объемного графика. Оба графика дают наглядную картину интерференции, очень близкую к наблюдаемой на практике (но без функциональной окраски).

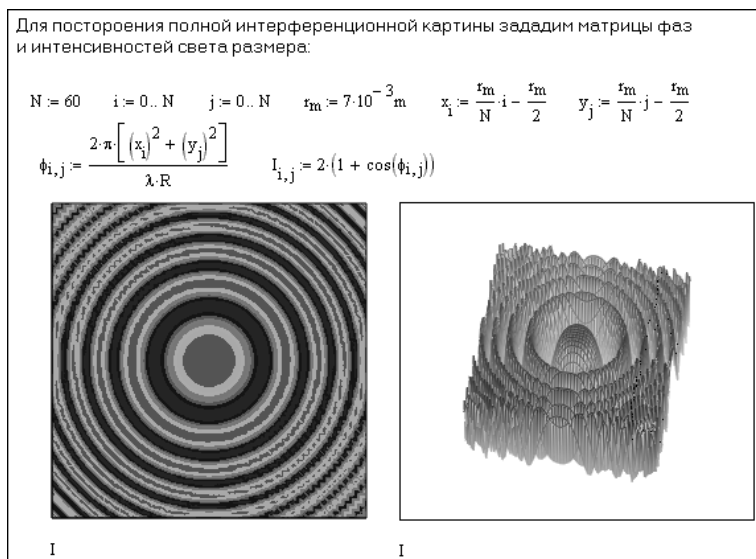


Рис. 11.37. Интерференционные кольца Ньютона на плоскости и в пространстве

Читатель должен четко понимать, что на самом деле представленные данные (как и все результаты моделирования в данной главе) являются лишь имитацией моделируемых процессов. Сколь бы наглядными и красочными они ни были, эти данные не являются прямыми наблюдениями физических процессов.

11.7. Моделирование биологических и экономических систем

11.7.1. Модель системы «хищник—жертва» Лотки—Вольтерра

Теперь рассмотрим типичную земную задачу о совместном проживании хищников и их жертв. Поскольку жертвы поедаются хищниками, число жертв начинает сокращаться, а число хищников — расти. Однако так не может продолжаться долго. Через некоторое время хищникам начинает не хватать пищи и их популяция перестает расти и даже уменьшается. В итоге жертвы начинают размножаться более интенсивно и их число растет. Далее эти процессы повторяются и в них обнаруживается периодичность.

Одной из первых моделей (1925—1927 гг.) такой системы «хищник—жертва» стала модель Лотки и Вольтера (рис. 11.38). Пусть y_0 и y_1 — число жертв и хищников. Предположим, что относительный прирост жертв y_0'/y_0 равен $a - by_1$, где $a > 0$ — скорость размножения жертв в отсутствие хищников, by_1 ($b > 0$) — потери от хищников. Развитие популяции хищников зависит от количества пищи (жертв), при отсутствии пищи ($y_0 = 0$) относительная скорость изменения популяции хищников равна $y_1'/y_1 = c$, где $c > 0$. Наличие пищи компенсирует убывание хищников и при $y_0 > 0$ имеем $y_1'/y_1 = (-c + dy_0)$, где $d > 0$.

Рассмотренная модель достаточно универсальна. Она может описывать не только изменение популяций хищников и жертв, но и поведение конкурирующих фирм, рост народонаселения, численность воюющих армий, изменение экологи-

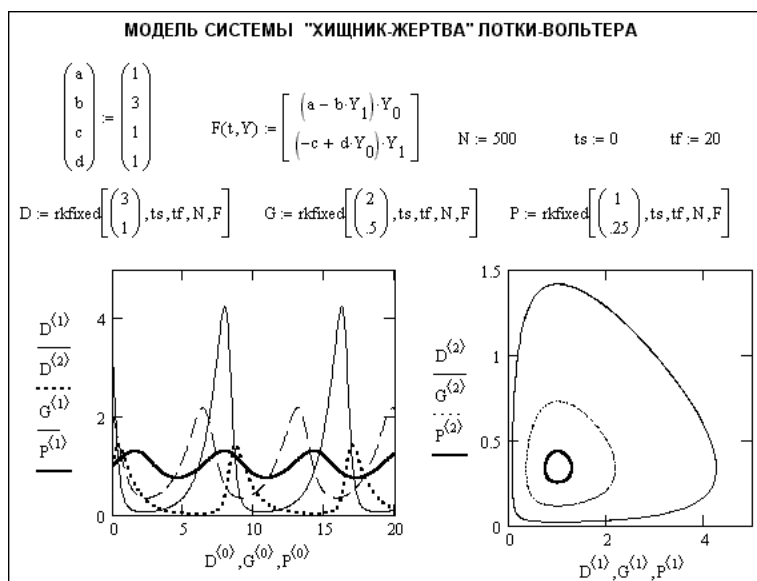


Рис. 11.38. Моделирование системы «хищник—жертва» по модели Лотки—Вольтерра

ческой обстановки, развитие науки и пр. Рекомендуется поэкспериментировать с этой моделью и убедиться, что моделируемые процессы могут иметь не только колебательный, но и аperiodический характер.

11.7.2. Модель системы «хищник—жертва» с логистической поправкой

Колебания популяций хищников и жертв на самом деле наблюдаются не всегда. Нередко мы наблюдаем стабильное количество тех и других, хотя процесс съедения жертв хищниками идет постоянно. Такой случай требует введения некоторой логистической поправки, которая учитывается в несколько иной модели системы «хищник—жертва», представленной на рис. 11.39.

Дополнительный параметр α в этой модели позволяет управлять затуханием осцилляций (колебаний) модели. Нетрудно заметить, что при указанных параметрах модели колебательный процесс в модели явно затухает и устанавливается длительное равновесие между числом хищников и жертв. Фазовый портрет приобретает устойчивый *фокус*. Форма фазового портрета свидетельствует о довольно малой нелинейности этой системы. Поэтому колебания напоминают затухающую синусоиду. Однако при $\alpha < 0$ образуется неустойчивый фокус и колебания начинают нарастать.

11.7.3. Модель системы «хищник—жертва» Холлинга—Тэннера

Еще одна нелинейная модель системы «хищник—жертва» была предложена Холлингом и Тэннером — рис. 11.40. Эта модель имеет две важные особенности. Ее нелинейность довольно сильна, что видно из вида фазового портрета, витки которого заметно отличны от эллипсов.

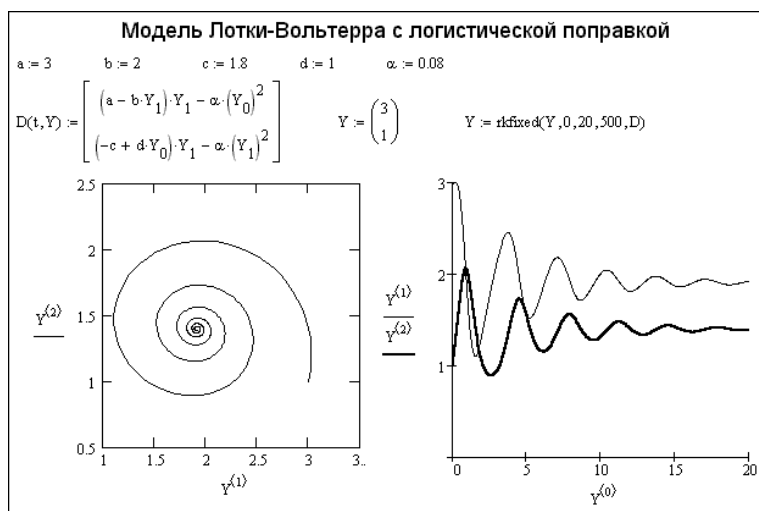


Рис. 11.39. Моделирование системы «хищник—жертва» по модели Лотки—Вольterra с логистической поправкой

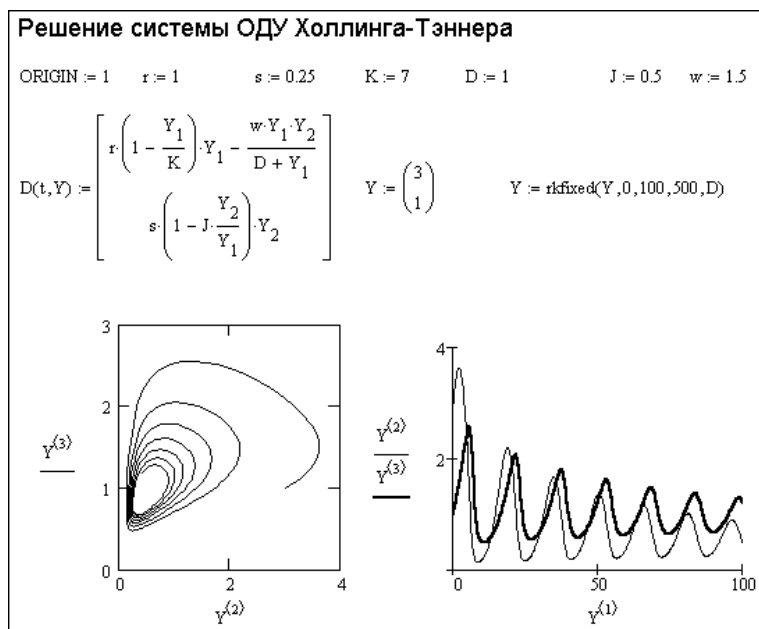


Рис. 11.40. Моделирование системы «хищник-жертва» по модели Холлинга—Тэннера

Главное свойство этой модели заключается в том, что в конечном счете колебания задаются предельным циклом фазового портрета, который может быть устойчивым. Он и определяет амплитуду колебаний, которые устанавливаются в стационарном режиме работы системы. При этом колебания могут как затухать во времени (пример чего и приведен), так и возрастать во времени, приближаясь при этом к стационарным колебаниям.

11.7.4. Моделирование замкнутой экономической системы

В поведении биологических и экономических систем есть много общего. Поэтому при обсуждении моделей экономических систем ограничимся описанием модели замкнутой практически линейной экономической системы, которая описывается дифференциальным уравнением второго порядка (рис. 11.41).

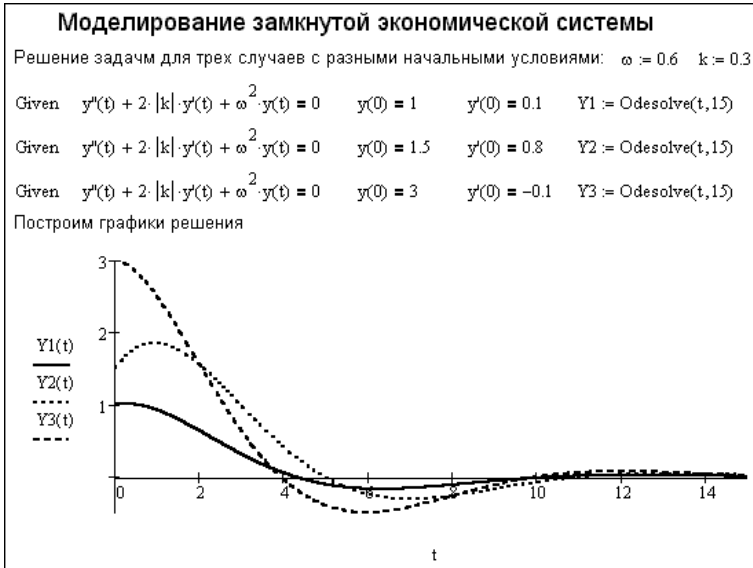


Рис. 11.41. Моделирование замкнутой экономической системы

Эта модель характеризуется двумя параметрами — круговой частотой осцилляций ω и параметром затухания k . Поведение системы существенно зависит как от этих параметров, так и от начальных условий. Так, оно может иметь как аperiодический, так и колебательный характер. В сущности, в этой модели для нас уже нет ничего нового — это еще одна модель системы второго порядка.

11.8. Моделирование экономических процессов

11.8.1. Моделирование закона Мура на основе нелинейной регрессии

На заре разработки микропроцессоров один из основателей корпорации Intel Гордон Мур высказал предположение, что количество транзисторов на кристалле микропроцессора будет удваиваться за некоторый промежуток времени. Это типичный эмпирический степенной (экспоненциальный) закон представляется простым выражением:

$$F(y, N, yu) = N \cdot 2^{\frac{y}{yu}}$$

Здесь N — количество транзисторов на кристалле в некоторый год y (условно считаем его нулевым), $F(y)$ — число транзисторов на кристалле спустя y лет и yu — срок (в годах и долях года), за который число транзисторов возрастает вдвое.

Уточнение закона Мура на первый взгляд может показаться технической задачей. Но это совсем не так. Крупнейшая микроэлектронная корпорация мира Intel, потратившая на научные исследования в 2002 году свыше 4,5 миллиарда долларов, сделала этот сугубо эмпирический закон своим знаменем и средством прогноза своего развития. Известность этого закона настолько широка, что он стал нарицательным и скорее характеризует социальные аспекты развития микроэлектроники, чем чисто технические закономерности. А потому оценка закона Мура имеет большой интерес для социологов, экономистов и тех, кто интересуется историей и тенденцией развития современной информационной техники.

Попытаемся выяснить, насколько справедлив этот закон. Ниже представлена таблица, дающая представление о динамике роста числа транзисторов (в тысячах штук) на кристалле микропроцессоров корпорации Intel с момента появления в 1971 году первого микропроцессора 4004. Таблица охватывает тридцатилетний период разработки процессоров корпорацией Intel и представляет лишь небольшую выборку из числа созданных Intel изделий.

Тип микропроцессора	Тысяч транзисторов	Год разработки	Параметр y
4004	$N_0 = 2,3$	1971	0
8008	3,5	1972	1
8080	6	1974	3
8088	29	1979	8
286	134	1982	11
386	275	1986	15
486	1200	1989	18
Pentium	3500	1993	22
Pentium PRO	5500	1995	24
Pentium II	7500	1997	26
Pentium III	9500	1999	28
Pentium 4	42000	2000	29
Pentium 4 M	75000	2001	30

Данные приведенной таблицы впечатляют сами по себе. Но насколько они соответствуют представленной формуле? И возможен ли по ним прогноз? Попробуем ответить на эти вопросы.

К сожалению, данные таблицы неравномерно распределены во времени (y в годах) и в значительной степени случайны, что препятствует применению простых методик нелинейной регрессии и простых функций линейного (относительно искомых параметров) предсказания, таких как функция predict в системе Mathcad. Нелинейная регрессия такого вида не реализована даже в таких маститых системах компьютерной математики, как Maple. Но, к счастью, в системе Mathcad 2001i/11/12 никаких проблем в ее проведении нет, если не считать больших вычислительных погрешностей и возможности переполнения разрядной сетки даже такой системы компьютерной математики, фигурирующей с числами, имеющими предельные значения до 10^{308} .

На рис. 11.42 представлен документ системы Mathcad 2002i с математической иллюстрацией «закона Мура». В левом верхнем углу документа задана формула «закона Мура» и в аналитическом виде вычислены ее частные производные по искомым параметрам N и u . Затем заданы векторы FI (функции и ее производных, нужных для реализации алгоритма нелинейной регрессии), числа лет, прошедших с 1971 года, Vy и числа тысяч транзисторов на кристалле процессора VN . С помощью функции `genfit`, использующей эти данные, вычислены параметры $N0$ и u . Начальные условия, сильно влияющие на точность регрессии, задаются вектором VS — они содержат стартовые значения параметров N (в тысячах штук) и значения u .

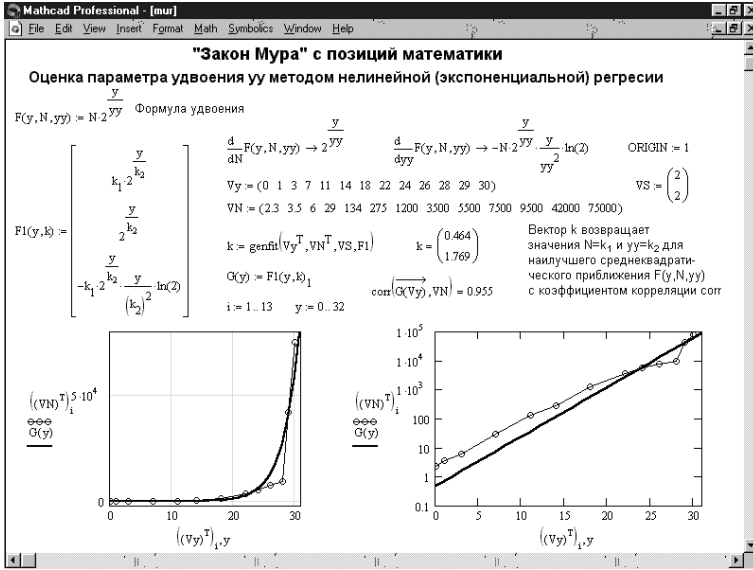


Рис. 11.42. Нелинейная регрессия для зависимости, описывающей увеличение числа транзисторов на кристалле процессора корпорации Intel от времени

Левый график задает число транзисторов как функцию от параметра u (время удвоения) в линейном масштабе. При этом расчетный график имеет типично экспоненциальный вид. Он показывает особенно резкое нарастание числа транзисторов в микропроцессорах начиная с 90-х годов прошлого века. До этого времени график малопредставителен.

Интереснее выглядит расчетный график в логарифмическом масштабе (справа внизу). Он превращается в кривую, наклон которой определяется параметром u . В течение первых примерно двадцати лет расчетный график приближающей функции и график, построенный по точкам реальных данных, идут практически параллельно, что свидетельствует о справедливости оценок Мура на протяжении этого времени. При этом реальное число транзисторов в серийных микропроцессорах оказывалось несколько большим, чем при расчетной оценке. Найденное время u при этом составило 1.769 года, т. е. удвоение числа транзисторов происходило примерно в два года. Ранние оценки времени в полтора года и даже в один год нелинейной регрессией не подтверждаются — не случайно корпорация Intel отказалась от них!

Однако в период между 22 и 28 условными годами развития (или между 1993 и 1999 гг.) стал наблюдаться заметный устойчивый спад в темпах роста числа тран-

зисторов на кристалле. Он проявляется в постепенном отклонении точек реальных данных вниз. Первые двадцать с небольшим лет развития процессоров соответствовали эволюционному пути развития технологии микропроцессоров — от процессора 4004 до Pentium. С появлением процессоров Pentium архитектура и сложность процессоров резко возросли, в частности из-за существенного увеличения команд мультимедиа-расширения MMX. Число транзисторов на кристалле превысило один из первых технологических и даже психологических барьеров в один миллион штук.

В результате возникло заметное отставание реальной технологии от закона Мура. В этот период особенно усилились сомнения в его принципиальной справедливости. Но в последние примерно три года корпорация Intel добилась резкого скачка в совершенствовании технологии, перейдя на производство микропроцессоров новой архитектуры — Pentium 4, производимых по технологии с разрешением 0,13 мкм на больших кремниевых дисках диаметром 300 мм. Стали внедряться новые технологические достижения. И теряющая крутизну кривая роста числа транзисторов резко скакнула вверх именно в эти последние три года.

Итак, математический анализ закона Мура подтверждает справедливость подмеченной Муром зависимости. Правда, надо отметить, что экспоненциальный рост числа транзисторов (и иных параметров микросхем) от времени очень чувствителен к параметру u . Его приближенные значения от 1 до 2 лет ведут к чудовищным просчетам (см. далее)! Да и сама процедура нелинейной регрессии для такой зависимости оказывается очень чувствительной к ошибкам машинных расчетов. В этом нетрудно убедиться, слегка меняя исходные данные или начальные приближения для u и N . Тем не менее близкое к единице значение коэффициента корреляции $\text{corr} = 0.955$ говорит о том, что зависимость (1) при полученных значениях N_0 и u не так уж и плохо соответствует исходным парам данных.

Характерной особенностью нелинейной регрессии оказывается большая погрешность в начальной области расчетов и резкое отличие расчетного параметра N в первые годы от реальных значений числа транзисторов на кристаллах первых микропроцессора. Так, выброс вниз даже небольшого числа исходных точек в правой области графика (где число транзисторов очень велико) ведет к тому, что большинство исходных точек в левой графика (где число транзисторов мало) располагается сверху расчетной зависимости — что прекрасно видно из правого графика рис. 2.???

Устранить этот недостаток можно добрым «дедовским» методом — взяв за левую точку прямой графика в логарифмическом масштабе точку первого отсчета и подобрав крутизну прямой на глаз по наилучшему положению в облаке исходных точек (отсчетов). Считая за нулевой 1971 год и за $N_0 = 2.3$ тысячи число транзисторов первого микропроцессора 4004, попробуем методом проб подобрать с помощью Mathcad логарифмическую прямую, на которую хорошо укладываются данные за первые годы развития процессоров и которая исходит из точки $(0, N_0)$. Результат представлен на рис. 11.43 сплошной тонкой линией. Названия процессоров на графике проставлены с помощью графического редактора.

Результат оказывается просто поразительным. Оказывается, целых 22 года число транзисторов и впрямь увеличивалось вдвое за каждые $u = 2$ года. При этом исходные точки укладываются почти точно на представляющую их приближенную зависимость. Таким образом, проведенная специалистами Intel коррекция закона Мура была вполне обоснованной и довольно точной. Однако до года или даже до полутора лет время удвоения u за этот период никогда не падало. Следовательно,

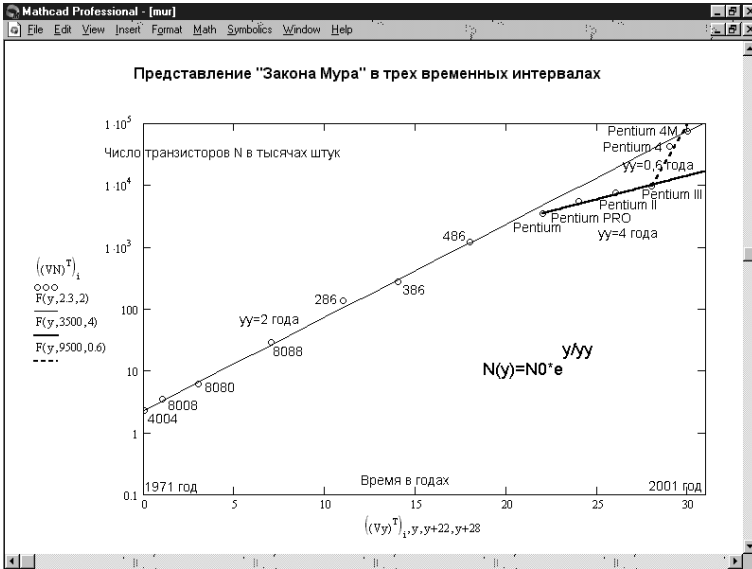


Рис. 11.43. Результат представления закона Мура отрезками прямых в логарифмическом масштабе

строго математически начальные прогнозы Мура были очень неточны, что низко-лько не умаляет их рекламное и эмоциональное значение.

Однако с появлением процессоров класса Pentium действие даже скорректиро-ванного закона Мура стало грубо нарушаться. Попробуем подобрать прямую для хорошего представления реалий роста числа транзисторов после этого периода, задав в качестве отчета момент появления процессоров Pentium — точка (22, 1200). Оказывается (жирная прямая на рис. 11.42), время удвоения возросло почти вдвое и составило $y = 4$ года. Это стало свидетельством предкризисной ситуации в разработке микропроцессоров и развитии их технологии. В какой-то степени это оправдывает отказ Intel от новых названий процессоров очередного поколе-ния — как известно, Pentium II и Pentium III сохранили отношение к звучному имени Pentium процессоров пятого поколения, которые и стали переломной точ-кой в законе Мура.

Однако, как известно, период спада в промышленности всегда сменяется пери-одом резкого подъема, увы, в дальнейшем вновь переходящего в спад. С перехо-дом от производства процессора Pentium III к Pentium 4 в 1999 году кривая роста числа транзисторов во времени резко скакнула вверх, начиная с точки (28, 9500). Это, очевидно, связано с очередным усложнением процессоров, появлением но-вых команд поддержки Интернета и потокового расширения и в процессорах Pen- tium III и Pentium 4. Время удвоения уменьшилось до рекордно малой величины в 0,6 года, но его оценка пока математически весьма недостоверна из-за малого объема выборки данных после 1999 года — всего 3 точки.

В целом, усредняя параметр y на весь тридцатилетний период развития мик-ропроцессоров, можно признать, что данные нелинейной регрессии достаточно корректны. Последуем за специалистами Intel и попытаемся дать прогноз роста числа транзисторов на кристалле микросхем на основании нашего приближения закона Мура выражением (1). Это показано на рис. 11.44 для первого десятилетия (рисунок слева) и для следующего десятилетия (рисунок справа).

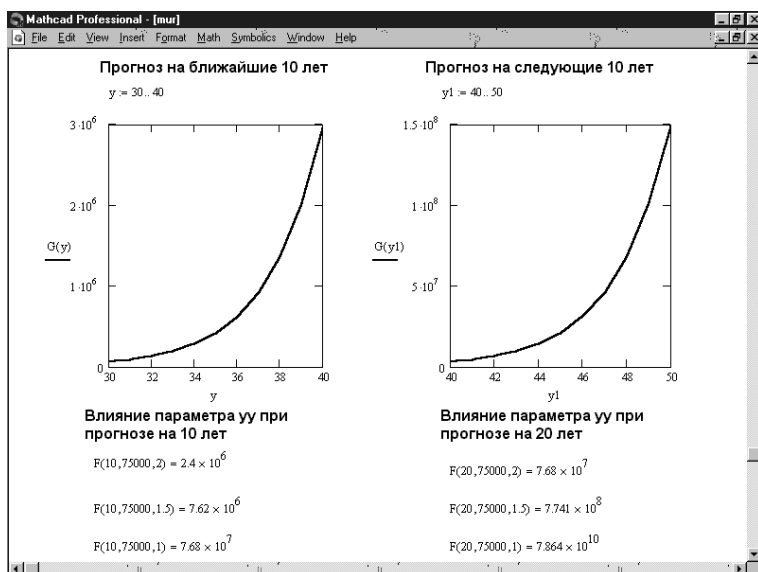


Рис. 11.44. Прогноз роста числа транзисторов на кристалле микросхемы на первые и на последующие десять лет по данным нелинейной регрессии

Прогноз на первые десять лет выглядит более или менее реалистичным. Можно ожидать появления микропроцессоров, на кристалле которых будет до 3 миллиардов транзисторов. Но вот прогноз на следующее десятилетие выглядит скорее фантастическим, чем реальным, — судя по нему, к 2020 году число транзисторов на кристалле достигнет примерно 140 миллиардов!

Увы, но поводов для сомнения в таком прогнозе более чем достаточно. Математически они вызваны критичностью закона Мура к параметру — времени удвоения u . Под рисунками рис. 11.44 показаны примеры расчета по формуле закона Мура на конец 10- и 20-летнего периода предсказания для разных $u = 1, 1.5$ и 2 года. Расхождения в оценках достигают соответственно примерно 32 и более чем 1000 раз! В этих условиях точность прогноза гарантировать нельзя и выводами закона Мура надо пользоваться с большой осторожностью.

Физическим ограничением на рост числа транзисторов может оказаться очередной предел геометрического разрешения в производстве интегральных микросхем. Даже с учетом внедряемой новейшей технологии литографии со сверхжестким ультрафиолетовым излучением этот барьер перемещается от 0.1 мкм до 0.03—0.05 мкм. При этом толщина подзатворного диэлектрика микроскопических полевых транзисторов, уже ныне составляющая менее 5 атомных слоев, достигнет предельного значения в один атомный слой.

Наконец, главный козырь к приостановке действия закона Мура — компьютерные системы из дискретных скорее всего превратятся в распределенные системы. В результате надобность в сверхмощных одиночных процессорах попросту упадет. И закону Мура придется подчиниться другому, тоже отнюдь не точному в математическом смысле, закону о переходе количества в качество.

Весной 2003 года Гордон Мур сам признал, что необходима и дальнейшая коррекция его закона, причем ее факторы полностью отвечают тем, которые были описаны в материалах по закону Мура в первой книге данной серии. В частности, такая коррекция необходима в связи с развитием новых технологий в производст-

ве микросхем — в частности электронно-ионной технологии с применением рентгеновских излучений с малой длиной волны. Там же были описаны прогнозы по этой зависимости.

11.8.2. Закон Мура и метод предсказания Бурга

Попытаемся улучшить предсказание данных по закону Мура и предпримем попытку все же применить для этого неплохо себя зарекомендовавший авторегрессионный метод Бурга. Для этого можно попытаться выполнить предварительную интерполяцию имеющихся данных сплайнами. Это поясняет документ, начало которого представлено на рис. 11.45.

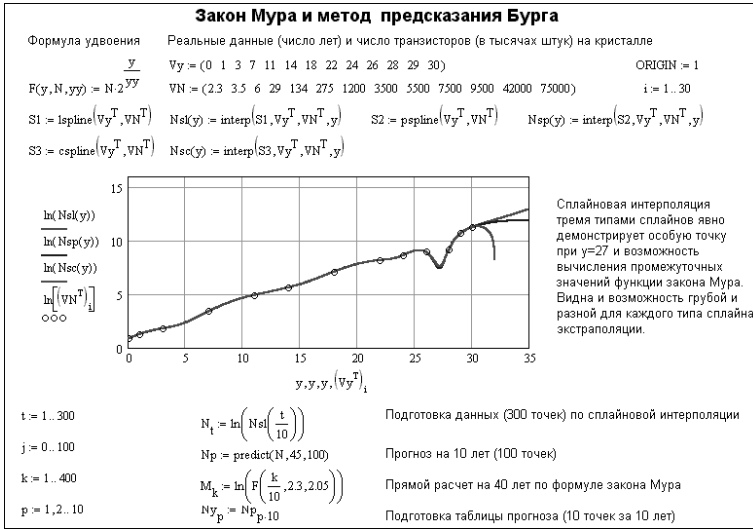


Рис. 11.45. Начало документа по прогнозу закона Мура методом предсказания Бурга

Здесь данные приведенной выше таблицы интерполируются с применением трех видов сплайнов — с линейным, квадратическим и кубическим продолжениями. Замечательно, что в пределах отрезка интерполяции кривые ее сливаются, т. е. интерполяцию можно проводить любым типом сплайнов. Это вполне закономерно, поскольку при интерполяции всегда используются кубические сплайны и лишь продолжение у них возможно разное.

Можно предположить, что возможна экстраполяция и самими сплайнами. Но рис. 11.45 наглядно показывает, что такая экстраполяция явно неудовлетворительна. Каждый сплайн дает свою кривую экстраполяции, и мы не располагаем никакими данными о том, что какая-либо экстраполяция предпочтительна.

В связи с этим будем использовать сплайновую интерполяцию только для увеличения точек, нужных для применения функции predict, реализующей авторегрессионный метод Бурга для данных в виде равноотстоящих узловых точек. Под рисунком 11.45 представлены выражения, задающие подготовку 300 точек данных, прогноз по последним 100 точкам и вычисления прямо по формуле закона Мура. На рис. 11.46 показаны итоговые графические зависимости для прогноза этим комбинированным методом.

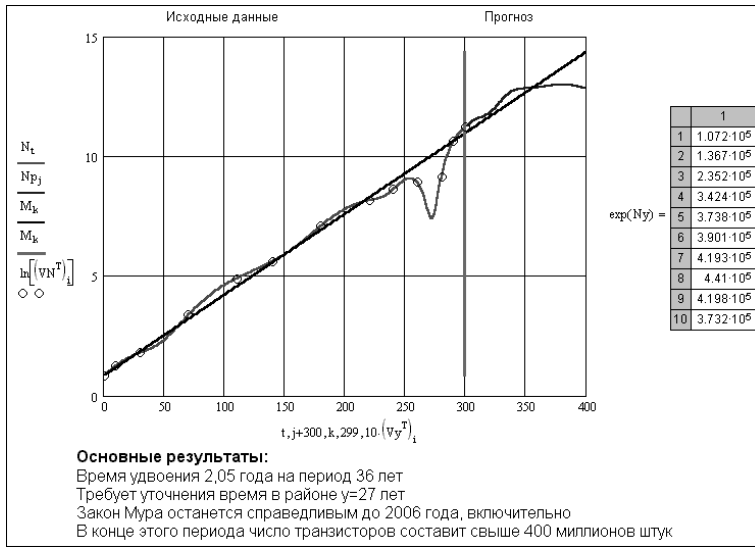


Рис. 11.46. Конец документа по прогнозу закона Мура методом предсказания Бурга

Данные прогноза в этом случае очень интересны. Видно, что закон Мура сохраняет свою силу по крайней мере еще 6—7 лет, но после этого прирост числа транзисторов прекращается. При этом максимальное число транзисторов на кристалле достигнет 420 миллионов. Такие данные неплохо согласуются с оценкой ряда специалистов Intel — они говорят, что прогноз Мура должен оставаться верным на ближайшие несколько лет, даже если не будут введены принципиально новые технологические достижения. Но если они будут введены, то этот закон может порадовать нас успехами корпорации Intel и на более длительный период времени.

11.8.3. Оптимальные экономико-математические модели

Современная экономика широко использует математические методы и самые разнообразные математические модели. В отличие от естественных наук возможности экспериментальных исследований в общественных дисциплинах ограничены. Экономический эксперимент в масштабах страны может привести к кризису и социальным потрясениям, а в рамках отдельной фирмы — к убыткам или краху. Поэтому моделирование экономических процессов, предварительный анализ возможных последствий тех или иных управленческих решений особенно важны. Рассмотрим примеры некоторых типичных задач экономического моделирования, наиболее часто встречаемые на практике.

Большую группу в моделировании экономических процессов составляют задачи, относящиеся к методам принятия оптимальных решений, исследованию операций. В повседневной практике хозяйствования требуется выбрать производственную программу, поставщиков, распределение ресурсов, маршрут транспортировки. Требование оптимальности в планировании и управлении приводит к задачам *оптимального (математического) программирования* — разделу прикладной математики, занимающемуся условной оптимизацией.

Необходимо найти такое управленческое решение $X = \{x_1, x_2, \dots, x_n\}$, которое в некоторой области допустимых решений D обеспечивало бы наилучшее значение

некоторого критерия оптимальности — экономического показателя. Такими экономическими показателями чаще всего являются «максимум прибыли», «минимум затрат», «максимум рентабельности» и т. д. Задача условной оптимизации в общем виде может быть записана так: найти максимум или минимум функции

$$f(X) = f(x_1, x_2, \dots, x_n) \quad (11.3)$$

при ограничениях

$$\varphi_i(x_1, x_2, \dots, x_n) (\leq, =, \geq) b_i, \text{ где } i = 1..m, \quad (11.4)$$

$$x_j \geq 0, j = 1..n. \quad (11.5)$$

Условия (11.5) может и не быть, но чаще всего переменные в экономическом моделировании должны быть не отрицательными. Выбор оптимального управленческого решения в конкретной производственной ситуации требует решения задачи оптимального программирования. Если функция и ограничения (11.3)—(11.5) линейные, то проблема сводится к задаче *линейного программирования*. К математическим задачам линейного программирования приводят различные производственные и хозяйственные ситуации, которые требуют оптимального использования ограниченных ресурсов (задачи о планировании выпуска продукции, о смесях, транспортная задача и так далее).

В общем случае задача линейного программирования может быть сформулирована следующим образом: найти максимум или минимум *целевой функции*

$$f(X) = \sum_{j=1}^n c_j \cdot x_j \quad (11.6)$$

при ограничениях

$$\sum_{j=1}^n a_{ij} \cdot x_j = b_i, x_j \geq 0, b_i \geq 0, i = 1..m, j = 1..n. \quad (11.7)$$

Для решения задач линейного программирования разработано большое количество различных методов. При анализе моделей с двумя или тремя переменными часто используются графические построения на плоскости или в пространстве. Среди универсальных методов решения наиболее распространен *симплексный метод*. Симплекс это многоугольник, перемещаемый в пространстве решения таким образом, чтобы, сужаясь, мог охватить точку искомого решения, отвечающую решению задачи с заданной погрешностью.

При этом методе задается некоторое начальное приближение, удовлетворяющее всем ограничениям задачи, но не обязательно оптимальное. Оптимальность результата достигается последовательным улучшением исходного варианта за определенное число шагов (итераций). Направление перехода от одной итерации к другой выбирается на основе критерия оптимальности целевой функции задачи.

Реализовывать симплекс-метод вручную — громоздко и сложно. Системы компьютерной математики имеют средства решения задач оптимизации, в том числе и симплекс-методом. Рассмотрим примеры решения нескольких типичных задач линейного программирования с помощью таких средств.

11.8.4. Решение задач максимизации объема продукции

Рассмотрим вполне типовую для малого бизнеса задачу на максимизацию объема выпуска изделий некоторым малым предприятием в денежном эквиваленте. Постановка задачи и ее решение представлены в документе рис. 11.47.

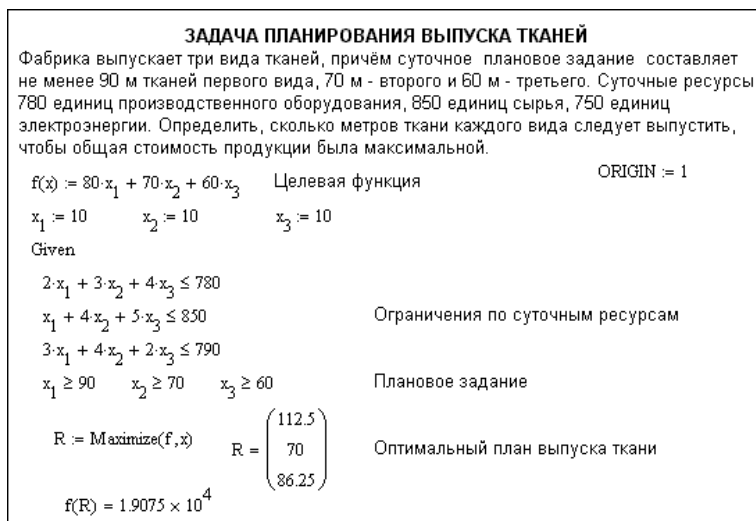


Рис. 11.47. Решение задачи планирования выпуска тканей

В задаче ищется максимальный объем выпуска тканей — также в денежном эквиваленте.

11.8.5. Решение задач минимизации ресурсов

Подобным описанному образом решаются и задачи на минимизацию ресурсов производства. Пример решения такой задачи на минимизацию числа автомобильных перевозок дан на рис. 11.48.

Другой такой задачей является задача минимизации стоимости смеси, например бензина. Стандартом требуется, что октановое число бензина А-76 должно быть не ниже 76, а содержание серы — не более 0.3 %. Для изготовления бензина используется смесь из четырех компонентов. Данные о компонентах приведены в таблице:

Характеристика	Компонент бензина			
	1	2	3	4
Октановое число	68	72	80	90
Содержание серы, %	0,35	0,35	0,3	0,2
Ресурсы, т	700	600	500	300
Себестоимость, ден. ед./т	40	45	60	90

Требуется определить, сколько тонн каждого компонента следует использовать для получения 1000 т автомобильного бензина А-76, чтобы его себестоимость была минимальной [16]. Решение задачи представлено на рис. 11.49.

11.8.6. Решение транспортной задачи

На практике часто встречается так называемая транспортная задача. В n пунктах — складах поставщиков находится определенное количество S_i ($i = 1..n$) еди-

ЗАДАЧА МИНИМИЗАЦИИ ЧИСЛА АВТОМОБИЛЬНЫХ СМЕН

На участок строящейся дороги необходимо вывезти 20 000 м³ каменных материалов. В районе строительства есть три карьера с запасами 8000 м³, 9000 м³, 10000 м³. Для погрузки материалов используются экскаваторы, имеющие производительность 250 м³ в смену на карьерах 1 и 2 и 500 м³ в смену в карьере 3. На погрузку материалов для рассматриваемого участка для экскаваторов выделен общий лимит 60 смен. Для перевозки 10000 м³ материалов с карьера 1 требуется 1000 смен автомобилей, из карьера 2 - 1350, из карьера 3 - 1700 смен автомобилей. Найти оптимальный план перевозок, обеспечивающий минимальное количество автомобильных смен. Объем материалов нормируем относительно 1=10 000 м³ для уменьшения вычислительных погрешностей.

ORIGIN := 1 $f(x) := 1000 \cdot x_1 + 1350 \cdot x_2 + 1700 \cdot x_3$ Целевая функция

$x_1 := 10$ $x_2 := 10$ $x_3 := 10$

Given

$x_1 + x_2 + x_3 = 2.0$ Потребность в материалах

$40 \cdot x_1 + 40 \cdot x_2 + 20 \cdot x_3 \leq 60$ Ограничение по сменам экскаваторов

$0 \leq x_1 \leq 0.8$ $0 \leq x_2 \leq 0.9$ $0 \leq x_3 \leq 1.0$ Запасы материалов в карьерах

$R := \text{Minimize}(f, x)$ $R = \begin{pmatrix} 0.8 \\ 0.2 \\ 1 \end{pmatrix}$ Количество вывозимых материалов с карьеров в 10000 м³

$f(R) = 2.77 \times 10^3$ Число автомобильных смен

Рис. 11.48. Решение задачи на минимизацию автомобильных перевозок

ЗАДАЧА О СМЕСЯХ

Стандартом требуется, что октановое число бензина А-76 должно быть не ниже 76, а содержание серы - не более 0.3%. Для изготовления бензина используется смесь из четырех компонентов. Требуется определить, сколько тонн каждого компонента следует использовать для получения 1000 т автомобильного бензина А-76, чтобы его себестоимость была минимальной.

$f(x) := 40 \cdot x_1 + 45 \cdot x_2 + 60 \cdot x_3 + 90 \cdot x_4$ Целевая функция

$x_1 := 10$ $x_2 := 10$ $x_3 := 10$ $x_4 := 10$

Given

$x_1 + x_2 + x_3 + x_4 = 1000$ Условие получения заданного количества бензина

$68 \cdot x_1 + 72 \cdot x_2 + 80 \cdot x_3 + 90 \cdot x_4 \geq 76 \cdot 1000$ Ограничение по октановому числу бензина

$0.35 \cdot x_1 + 0.35 \cdot x_2 + 0.3 \cdot x_3 + 0.2 \cdot x_4 \leq 0.3 \cdot 1000$ Ограничение по содержанию серы

$0 \leq x_1 \leq 700$ $0 \leq x_2 \leq 600$ $0 \leq x_3 \leq 500$ $0 \leq x_4 \leq 300$ Ограничения по числу компонентов

$R := \text{Minimize}(f, x)$ $R = \begin{pmatrix} 0 \\ 571.429 \\ -2.842 \times 10^{-14} \\ 142.857 \\ 285.714 \end{pmatrix}$ $f(R) = 5.7143 \times 10^4$ Минимальная себестоимость

Рис. 11.49. Решение задачи о смесях

ниц некоторого однородного продукта. Этот продукт потребляется m потребителями в определенном объеме B_j ($j = 1..m$). Известны расходы на перевозку единицы продукта из i -го склада j -му потребителю, которые равны C_{ij} и приведены в

таблице транспортных расходов. Требуется составить такой план перевозок, при котором полностью удовлетворяются заказы потребителей с минимальными транспортными затратами.

Пусть на трех складах хранится 310, 260 и 280 единиц груза соответственно. Требуется его доставить пяти потребителям, заказы которых равны 180, 80, 200, 160, 220 единиц. Стоимости перевозки единицы груза со склада потребителю указаны в транспортной таблице.

Разработать модель, описывающую затраты при перевозке грузов со складов потребителям и позволяющую оптимизировать затраты на транспортировку. Исходные данные к задаче следующие:

- груз на всех складах одинаковый;
- количество груза на каждом i -м складе;
- заказ каждого j -го потребителя;
- стоимость перевозки груза с i -го склада j -му потребителю.

Примем следующие гипотезы:

- считаем, что пропускная способность дороги от каждого склада не ограничена;
- длительность перевозки от склада к потребителю не учитывается при выборе предпочтительного плана перевозок;
- общее количество грузов на складах всегда больше или равно заказу потребителей.

Введем обозначения:

- n — количество поставщиков;
- m — количество потребителей;
- A_i — поставки от i -го поставщика всем потребителям, ограниченные S_i — количеством груза на складе;
- B_j — заказ j -го потребителя — поставки ему от всех поставщиков,
- X_{ij} — перевозки от i -го поставщика j -му потребителю;
- C_{ij} — цена поставки единицы груза от i -го поставщика j -му потребителю.

Требуется обеспечить полное выполнение всех заказов B_j при минимальных затратах на перевозку грузов. Общие затраты на перевозку равны

$$F(X) = \sum_{i=1}^n \sum_{j=1}^m C_{i,j} \cdot x_{i,j},$$

где $F(X)$ — минимизируемая функция, зависящая от $n \times m$ переменных X_{ij} . Очевидно, что при этом должны выполняться ограничения:

$$A_i = \sum_{j=1}^m x_{i,j} \leq S_i \text{ — нельзя поставить груза больше, чем есть на складе;}$$

$$B_j = \sum_{i=1}^n x_{i,j} \text{ — потребитель должен получить точное количество заказанного груза;}$$

$$x_{i,j} \geq 0 \text{ — товаропоток не может быть отрицательным.}$$

Математическая постановка соответствует решению задачи линейного программирования — условия оптимизации описываются системами линейных уравнений и неравенств. Решения задачи будем проводить в среде пакета Mathcad, ис-

пользуя универсальную встроенную функцию `Minimize`. Документ, решающий данную задачу, представлен на рис. 11.50.



Рис. 11.50. Решение транспортной задачи

11.8.7. Задачи целочисленного программирования с булевыми переменными

Если искомые параметры (переменные) должны иметь только целые значения, то для их нахождения надо применять методы решения задач целочисленного программирования. Их достаточно легко реализовать, например, с помощью табличного процессора Excel. Однако Mathcad специальных реализаций таких методов не имеет.

Однако в некоторых практических задачах целочисленного программирования искомые переменные могут принимать не любые целые значения, а лишь значения 0 — ответ «нет» и 1 — ответ «да». Такие переменные называют логическими или булевыми. Одной из задач с такими переменными является *задача о назначениях*. В подобных случаях требуется решить, как распределить рабочих по различным рабочим местам, чтобы общая выработка была наибольшей или затраты на зарплату наименьшими, и как выбрать из нескольких возможных вариантов инвестиционных проектов (например, закупки станков для модернизации цеха) наиболее эффективный проект.

В общем случае можно подобные задачи сформулировать следующим образом:

Предлагается n управленческих решений x_i , каждое из которых позволяет получить эффект от его реализации $C_1, C_2 \dots C_n$. В наличии имеется m видов ресурсов в количестве S_j . Управленческое решение требует для своей реализации объем ресурсов V_{ij} , где $i = 1..n, j = 1..m$. Необходимо так распорядиться имеющимися ресурсами, чтобы максимизировать эффект от принимаемых решений.

Оптимизируемая переменная может принимать только два значения:

$$x_i = \begin{cases} 1, & \text{если решение следует принять.} \\ 0, & \text{если решение нецелесообразно.} \end{cases}$$

Тогда задача оптимизации будет следующей:

Найти максимум целевой функции

$$F(X) = \sum_{i=1}^n C_i \cdot x_i \text{ — экономическая эффективность, при ограничениях,}$$

$$\sum_{i=1}^n B_{i,j} \cdot x_i \leq S_j, j = 1..m \text{ — по ресурсам,}$$

$$x_i \in \{0,1\}, i = 1..n \text{ — булевый характер переменных.}$$

Такая постановка соответствует задаче дискретного линейного программирования с булевыми переменными. Она может быть решена как обычная задача целочисленного программирования, например, методом ветвей и границ с соответствующими ограничениями, или специальными методами, в частности, методом полного или сплошного перебора, методом Балаша и т. д.

Рассмотрим пример на выбор инвестиционного проекта. Имеется четыре инвестиционных проекта, каждый из которых требует затрат материальных и трудовых ресурсов. Количество ресурсов ограничено и не позволяет реализовать все проекты сразу. Выбрать для реализации оптимальные по суммарному экономическому эффекту проекты. Конкретные числовые данные приведены в таблице, представленной ниже.

Показатели	Варианты инвестиционных проектов				Запасы
	1	2	3	4	
Материальные ресурсы	200	180	240	250	800
Трудовые ресурсы	10	15	22	28	50
Прибыль	65	80	90	210	

Математически проблема соответствует задаче дискретного линейного программирования с булевыми переменными для приведенных выше целевых функций.

Учитывая небольшое количество возможных вариантов, а также использование программных средств для автоматизации вычислений, будем решать задачу *методом полного или сплошного перебора*.

Метод заключается в переборе всех возможных вариантов сочетаний допустимых значений переменных, проверке выполнения для каждого ограничений и вычислении в удовлетворительных случаях соответствующих значений целевой функции. Из полученного множества значений выбирается максимальное (или минимальное), а набор значений переменных для него и будет решением задачи. В общем случае число вычислительных процедур при полном переборе быстро растет и равняется $N = 2^n \cdot (m + 1)$, где n — число переменных, m — число ограничений. Метод имеет простой алгоритм и может быть легко реализован с использованием средств программирования пакета Mathcad (рис. 11.51).

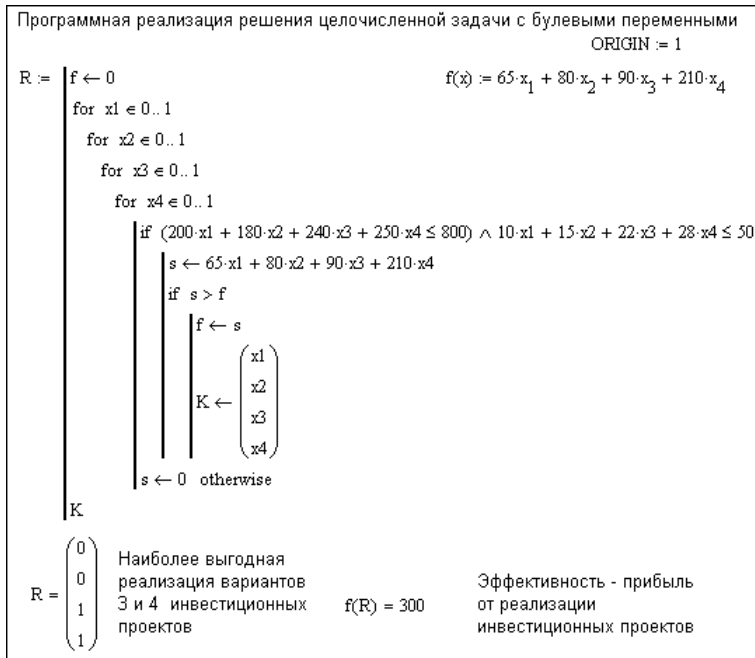


Рис. 11.51. Пример решения задачи целочисленного программирования с булевыми переменными методом прямого перебора

11.9. Сетевые модели в оптимизации управленческих решений

11.9.1. Задача поиска кратчайшего пути

Для представления различных технических объектов, описания процессов и функционирования систем часто используются графовые модели. К модели в виде графа можно свести и многие практические экономические задачи. К таким проблемам относятся задача поиска кратчайшего пути в заданной транспортной системе, задачи о распределении потока в сети, сетевые модели планирования последовательности работ, задача коммивояжера и другие.

В общем виде задача формулируется следующим образом. Имеется некоторое количество пунктов, соединенных определенным образом одно- или двунаправленными связями. Каждая связь имеет определенный вес — длину. Требуется найти кратчайший путь из пункта i в пункт j .

При составлении математической модели задачи необходимо учитывать, что маршрут должен быть непрерывным, а каждый промежуточный пункт на пути следования может быть посещен только один раз. Транспортная система в задаче является ориентированным графом — 2-полюсной сетью, где N_1 — вход, N_n — выход, весовые коэффициенты c_{ij} ребер δ_{ij} являются длинами пути между пунктами i и j , требуется определить кратчайший путь из N_1 в N_n . Сопоставим каждому ребру графа булеву переменную, т. е. $\delta_{ij} \in \{0,1\}$. Если ребро входит в маршрут, то $\delta_{ij} = 1$, иначе $\delta_{ij} = 0$. Тогда целевая функция, которая минимизируется при поиске кратчайшего пути, имеет вид:

$$F = \sum_i \sum_j c_{ij} \cdot \delta_{ij}.$$

Все пункты маршрута можно разделить на начальный, промежуточный и конечный. Очевидно, что в каждом промежуточном пункте должно быть по одному входящему и исходящему ребру, а для начального и конечного пунктов может быть только одно исходящее или входящее ребро соответственно. Математически эти ограничения могут быть записаны следующим образом:

- для перечисления всех k входящих в i -й пункт маршрута ребер $\sum_k \delta_{ki} = 1$,
 $i = 2 \dots n$;
- для перечисления всех j исходящих из i -го пункта ребер $\sum_j \delta_{ij} = 1$,
 $i = 1 \dots n - 1$.

Если же i пункт не входит в кратчайший маршрут, то соответствующая сумма как для входящих, так и исходящих из вершины графа ребер должна быть равна нулю. Тогда для любого пункта сети, кроме начального и конечного, должно выполняться условие:

$$\sum_k \delta_{ki} - \sum_j \delta_{ij} = 0.$$

В начальном пункте

$$\sum_j \delta_{1j} = 1,$$

в конечном

$$\sum_k \delta_{kn} = 1 \text{ и } \delta_{ij} \geq 0 \text{ для всех } i \text{ и } j.$$

От переменных δ_{ij} достаточно потребовать только неотрицательности. Из-за ограничений в решении могут быть получены только значения нуля либо единицы. Таким образом, мы получили обычную задачу линейного программирования, которую можно решить без наложения требований целочисленности.

Очевидно, что к подобной формулировке, а точнее, соответствующей математической модели можно свести самые разнообразные задачи, в том числе планирование последовательности выполнения технологических процессов и работ. Вес ребер графа при этом может иметь самый различный смысл: продолжительность, трудоемкость, стоимость и т. д.

Пусть требуется найти кратчайший маршрут из пункта А в пункт В, если схема движения и расстояния между объектами заданы рис. 11.52 [24]. Под рисунком представлено решение этой задачи.

11.9.2. Задача о распределении потоков в сетях

В задачах подобного типа требуется найти оптимальный вариант транспортировки продукта по сети определенной конфигурации. В этом случае элементы сети имеют следующие характеристики: c_{ij} — стоимость транспортировки единицы продукции для ребра сети между вершинами i и j , D_{ij} — пропускная способность этого ребра, в общем случае ограниченная в пределах $0 \leq D_{ij} \leq \infty$ (если ребро между данными вершинами i и j графа отсутствует, то пропускная способность равна нулю, если поток ничем не ограничен — то бесконечности). Очевидно, что в этом случае должно выполняться требование *сохранения потока* — суммарный поток, входящий и выходящий из узла, должны быть равны. Пусть x_{ij} — поток в ребре графа, тогда для промежуточной вершины сети

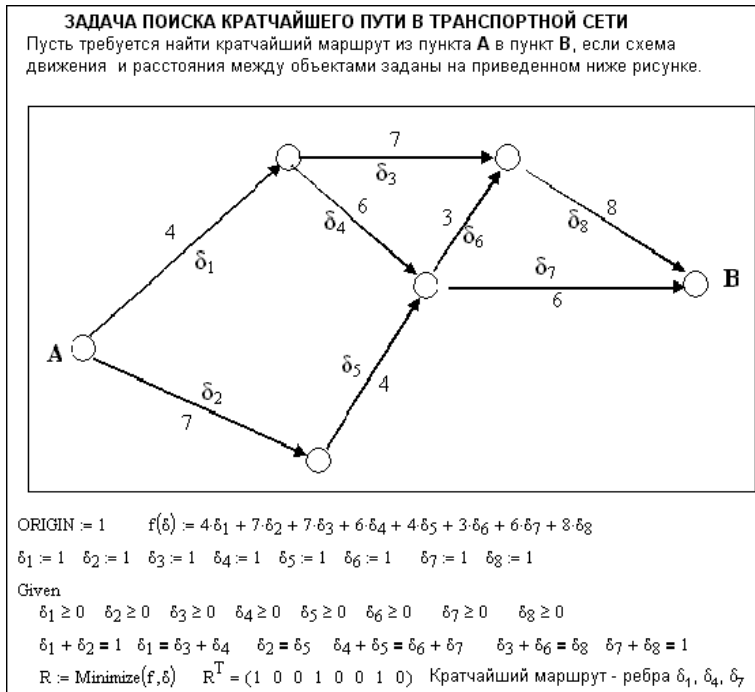


Рис. 11.52. Решение задачи на поиск кратчайшего маршрута

$$\sum_k x_{ki} - \sum_j x_{ij} = 0,$$

где k — перечисление всех входящих, j — всех исходящих ребер для вершины i .

Для потока в любом ребре требуется, чтобы

$$0 \leq x_{ij} \leq D_{ij}.$$

Для начальной и конечной вершины очевидно необходимо выполнение условия

$$\sum_j x_{1j} = A_1,$$

где A_1 — максимальный выходной поток, создаваемый исходной вершиной сети, необходимо, чтобы он был меньше, чем суммарная пропускная способность всех исходящих из вершины ребер,

$$\sum_k x_{kn} = B_n,$$

где B_n — максимальный поток, потребляемый конечной вершиной сети, он также не должен превышать пропускной способности входящих ребер.

Возможны различные постановки задачи оптимизации — минимизации стоимости транспортировки и максимизации потока. Получаем соответственно две формулировки математической модели задачи.

1. Минимизация стоимости:

$F = \sum_i \sum_j c_{ij} \cdot x_{ij}$ — минимизируемая целевая функция — общая стоимость транспортировки.

Ограничения:

$A_i = B_n$ — поток не может накапливаться в промежуточных вершинах, т. е.

$$\sum_j x_{1j} = \sum_k x_{kn},$$

$0 \leq x_{ij} \leq D_{ij}$ — по пропускной способности;

$$\sum_k x_{ki} - \sum_j x_{ij} = 0 \text{ — сохранение непрерывности потока.}$$

2. *Максимизация потока:*

$F = \sum_k x_{kn}$ — максимизируемая целевая функция — суммарный поток, входящий в конечный узел.

$$\sum_i \sum_j c_{ij} \cdot x_{ij} \leq C_s \text{ — суммарные затраты не должны превысить величины имеющихся средств } C_s.$$

Ограничения:

$A_i = B_n$ — поток не может накапливаться в промежуточных вершинах, т. е.

$$\sum_j x_{1j} = \sum_k x_{kn},$$

$0 \leq x_{ij} \leq D_{ij}$ — по пропускной способности;

$$\sum_k x_{ki} - \sum_j x_{ij} = 0 \text{ — сохранение непрерывности потока.}$$

Рассмотрим задачу на поиск максимального потока для системы автодорог, представленных на рисунке документа рис. 11.53, где цифрами обозначена максимальная пропускная способность участков транспортной сети, тысяч машин в день.

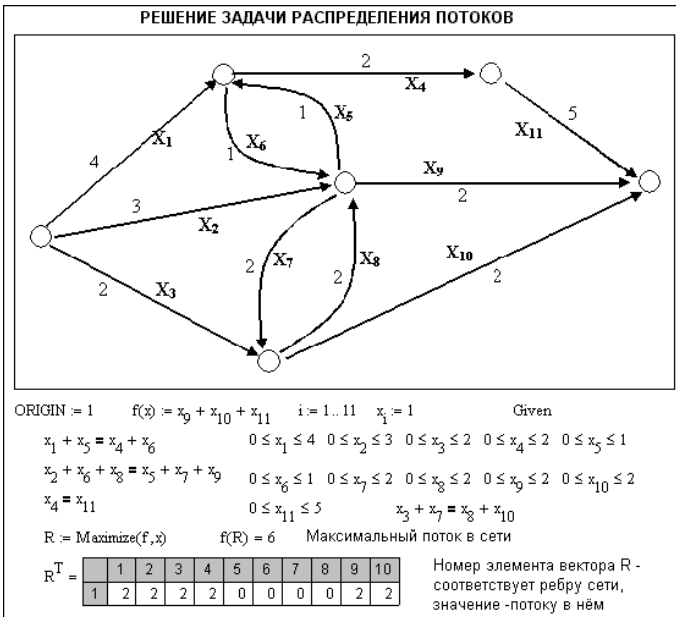


Рис. 11.53. Решение задачи на поиск максимального потока для системы автодорог

Заданный граф частично ориентирован. Для того чтобы прийти к математической модели, необходимо преобразовать граф в ориентированную сеть. Это можно сделать, заменив каждое неориентированное ребро — дорогу с двусторонним движением двумя ориентированными — односторонними полосами движения, каждая с исходной пропускной способностью. Дороги x_4 и x_5 стали односторонними, так как возможность противоположного направления движения в данной задаче для них несущественна.

Задача может быть решена геометрически: согласно теореме, максимальная пропускная способность сети равна минимальной пропускной способности сечений сети. Аналитическое решение сводится к методам линейного программирования. Кроме того, тогда возможно определить соответствующие ему потоки в каждом ребре сети.

Сравнение максимально возможного потока, исходящего из начального узла сети, с результатом решения ($9 > 6$) показывает, что данная транспортная сеть требует дополнительного расширения для его пропуска.

Глава 12. Обработка сигналов и расчет электронных устройств

Mathcad может эффективно использоваться для проверки математических методов обработки сигналов и расчета радиоэлектронных устройств. В этой главе даны основы и наглядные примеры такого применения. Это рассмотрение не претендует на полноту и является введением в эту интересную область применения систем компьютерной математики. Заинтересовавшийся этим читатель может обратиться ко многим применениям Mathcad в электро- и радиотехнике, размещенным в Интернете и книгах [1, 10, 11, 22, 41, 43].

12.1. Создание, моделирование и представление сигналов

12.1.1. Моделирование ступени и прямоугольного импульса

Как известно, передача сообщений осуществляется с помощью сигналов той или иной природы, например электрических или световых. Сигналы могут быть *аналоговыми* (непрерывными) и *дискретными*, т. е. представляемыми дискретными уровнями. Например, сигнал о положении монеты (орел или решка) будет дискретным и двоичным. Если этих уровней много (например грани игрального кубика), можно говорить о цифровом представлении информации. Сигналы, мгновенные значения которых представлены числами, принято называть *цифровыми сигналами*. Аналоговые сигналы характеризуется плавным и непрерывным изменением их параметров, например это температура воздуха или величины электрического тока или напряжения для электрических сигналов.

Простейшим дискретным сигналом является ступенька. Ступенька с единичной амплитудой в момент $t = 0$ в Mathcad создается функцией Хевисайда (рис. 12.1). Для перемещения времени появления ступеньки на время t_0 достаточно записать аргумент функции Хевисайда как $(t - t_0)$ — на рис. 12.1 пример дан для $t_0 = 5$.

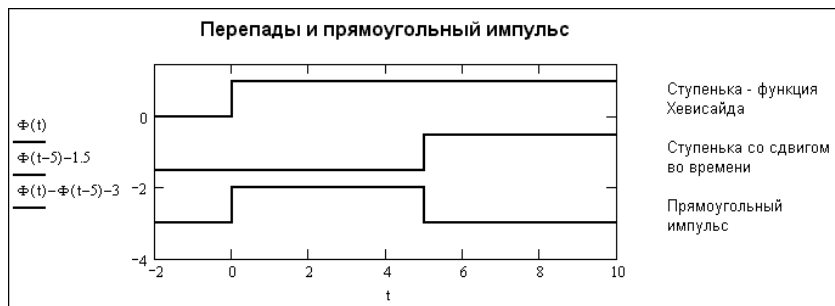


Рис.12.1. Сигналы в виде ступени и прямоугольного импульса

Сигнал в виде прямоугольного импульса единичной амплитуды легко создать как разность двух функций Хевисайда. Этот случай также показан на рис. 12.1. Логические операторы и функция `if` также удобны для создания таких сигналов.

12.1.2. Синусоидальный (гармонический) сигнал

Наиболее распространенным непрерывным сигналом является синусоидальный или гармонический электрический сигнал $u(t) = U_M \cdot \sin(\omega \cdot t + \phi)$, где U_M — амплитуда синусоидального сигнала, ω — круговая частота и ϕ — фаза. Временная зависимость синусоидального сигнала представлена на рис. 12.2.

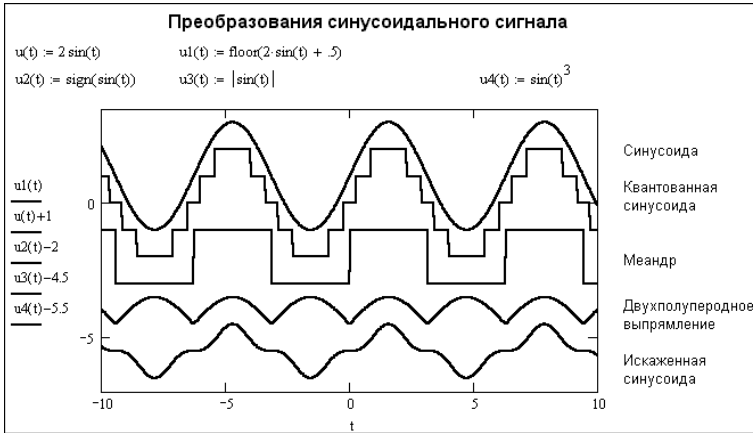


Рис. 12.2. Синусоидальный сигнал и его некоторые преобразования

Круговая частота связана с обычной частотой выражением $\omega = 2 \cdot \pi \cdot f = 2 \cdot \pi / T$. Частота $f = 1/T$ — это число периодов T синусоидального сигнала в единицу времени (секунду или с). Она измеряется в герцах (Гц). Один герц — это один период колебаний в секунду (единица названа в честь Герца, теоретически обосновавшего существование электромагнитных волн, распространяющихся в пространстве).

Фундаментальное значение синусоидального сигнала состоит в том, что этот сигнал является *стационарным*. Это значит, что все его параметры U_M , ω и ϕ являются постоянными величинами и от времени не зависят. Этот сигнал определен во времени в пределах от $-\infty$ до $+\infty$. Он периодический $u(t) = u(t + T)$ и симметричный $u(t) = -u(-t)$. Можно сказать, что синусоида описывает простейшее (а потому названное *гармоническим*) движение, параметры которого неизменны.

На практике используется великое множество и несинусоидальных сигналов, например импульсные сигналы пилообразной, прямоугольной и иной формы. Математик Фурье строго доказал, что такие периодические сигналы могут быть представлены суммой синусоидальных сигналов с кратной их частоте повторения частотой $k \cdot f_1$, где $k = 1, 2, \dots$ — целое число и f_1 — частота повторения сигнала. Эти сигналы называют *гармониками*, а значение k — номерами гармоник. Синусоидальный сигнал с частотой f_1 есть *первая гармоника*, а сигналы с более высокими частотами называют *вышшими гармониками*. Линейные преобразования сигналов не меняют состав гармоник, называемый *спектром*, а нелинейные приводят к его изменению, т. е. появлению новых гармоник.

12.1.3. Квантование и преобразование синусоидального сигнала

Аналоговый сигнал можно *квантовать*, т. е. представлять его рядом ступенек, высота которых задается уровнем сигнала в начале каждой ступеньки (в момент *выборки*) и остается неизменной на протяжении каждой ступеньки. Для квантова-

ния синусоидального сигнала по амплитуде удобно использовать функцию округления floor — см. пример на рис. 12.2. Здесь для наглядности амплитуда синусоиды взята равной 2, что дает 5 ступенек квантованного сигнала. Вообще говоря, число ступенек определяется двойной амплитудой синусоиды плюс одну ступеньку.

Возможны также различные нелинейные преобразования синусоидального сигнала, например, его абсолютное значение порождает сигнал, характерный для двухполупериодного выпрямления, а функция знака sign (от полного имени signum — знак) позволяет формировать прямоугольные симметричные импульсы — меандр.

12.1.4. Модуляция синусоидальных сигналов

Основные параметры синусоидального сигнала могут непрерывно и плавно меняться во времени. Такое изменение называется *модуляцией* сигнала. Например, амплитудная модуляция описывается выражением:

$$u(t) = U_m(t) \cdot \sin(\omega \cdot t + \varphi), \quad (12.1)$$

где $U_m(t)$ — зависимость амплитуды от времени. Сигнал такого вида называется *амплитудно-модулированным*. Строго говоря, он является уже не стационарным и даже не синусоидальным.

Если $U_m(t)$ несет некоторую информацию, то говорят о наложении этой информации на синусоидальный сигнал. И если такой сигнал имеет высокую частоту $f = 1/T$, то сигнал может распространяться в свободном пространстве как электромагнитная волна со скоростью света (300 000 км/с). На этом и основана радиосвязь. Расстояние, которое проходит волна за один период, называется *длиной волны*. Отсюда пошло деление волн на короткие и ультракороткие, средние и длинные волны.

Часто степень модуляции оценивается коэффициентом модуляции — отношением амплитуды модулирующего синусоидального колебания к амплитуде модулируемого колебания с более высокой частотой. На рис. 12.3 показан модулированный по амплитуде сигнал с его верхней огибающей.

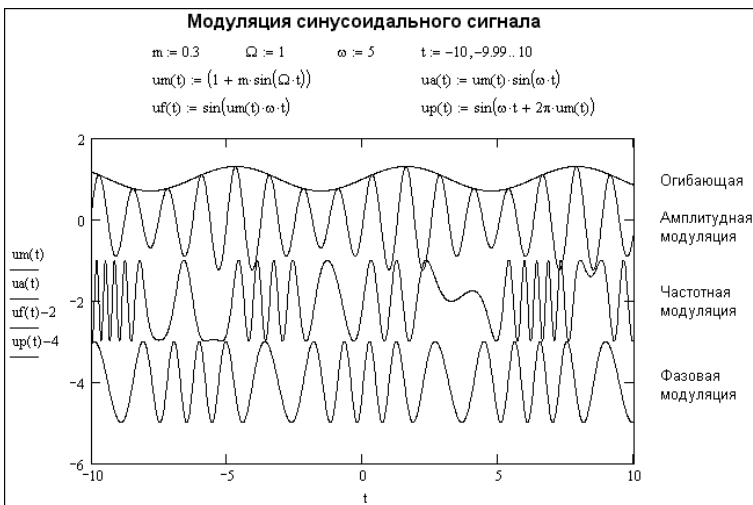


Рис. 12.3. Виды модуляции синусоидального сигнала

Заметим, что синусоидальный сигнал может моделироваться еще и по частоте, и по фазе. Этому соответствует *частотная модуляция* (рис. 12.3, средняя кривая) и *фазовая модуляция* (рис. 12.3, нижняя кривая), которые широко используются на практике в радиотехнических системах. Обратите внимание на то, что в примере частотной модуляции виден эффект перемодуляции — заметное отклонение формы сигнала от синусоидальной. Фазовая модуляция по форме сигнала напоминает частотную модуляцию, и в самом деле оба вида модуляции сопутствуют друг другу.

12.1.5. Создание сигналов комбинациями элементарных функций

Различные комбинации элементарных функций позволяют моделировать многие виды сигналов. На рис. 12.4 показаны примеры применения комбинаций тригонометрических и обратных тригонометрических функций для моделирования сигналов ряда типовых форм.

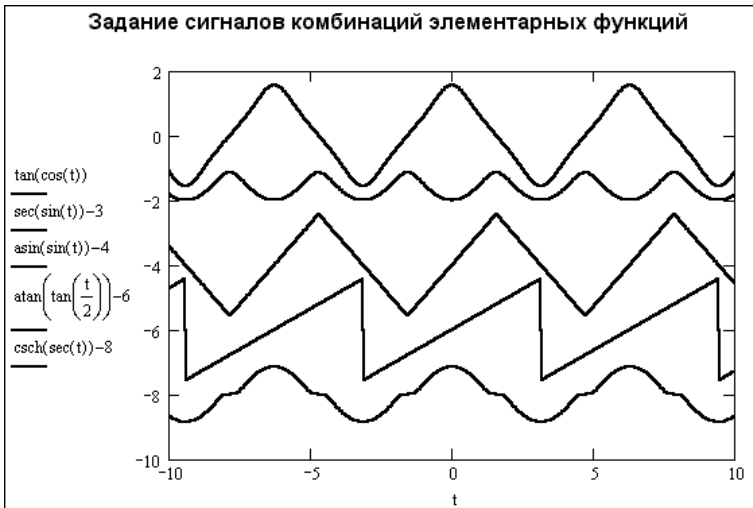


Рис. 12.4. Сигналы, создаваемые комбинациями элементарных функций

Некоторые из представленных сигналов, например треугольные и пилообразные сигналы, можно формировать и иначе. Однако формирование их из периодических тригонометрических функций иногда дает существенные преимущества — такие сигналы обрабатываются символьными (аналитическими) методами и легко делаются периодическими.

12.1.6. Выборки сигнала и теорема Котельникова

При цифровой обработке сигналов часто производят их *выборку* (вырезку) в определенные моменты времени (рис. 12.5, сверху). Они могут равномерно или неравномерно отстоять друг от друга. Сигнал может быть периодическим (с периодом T), но может быть и однократным или непериодическим.

Выборку электрических сигналов и их представление в виде чисел или кодов конечной разрядности выполняют так называемые *аналого-цифровые преобразователи* — АЦП. В результате на выходе АЦП мы имеем дискретный сигнал, представленный потоком чисел (кодов). Главные показатели АЦП — это их разряд-

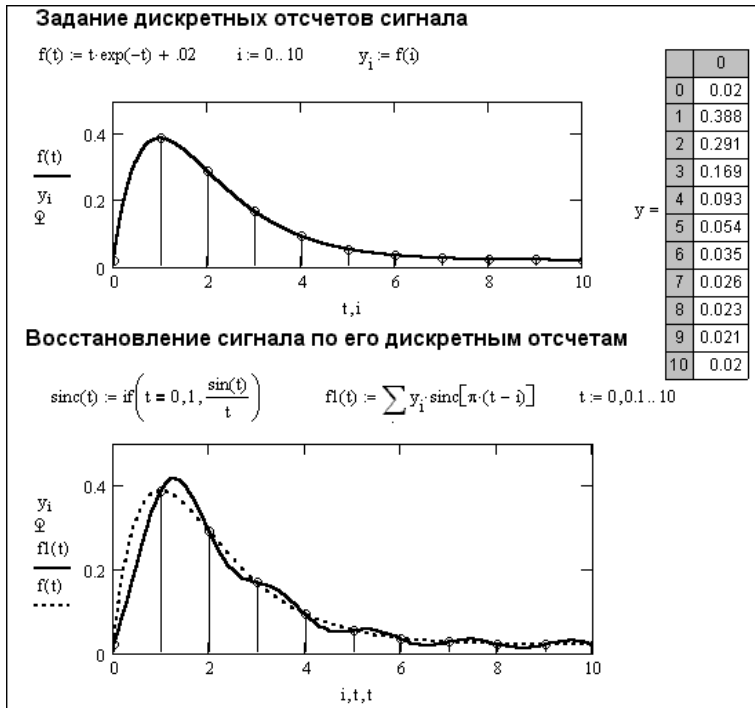


Рис. 12.5. Задание непрерывного сигнала его выборками и его восстановление на основе теоремы Котельникова

ность (число уровней квантования, обычно выражаемое в двоичном виде) и скорость выполнения преобразований (число операций в секунду).

Обратное преобразование цифровой информации в аналоговую выполняют *цифро-аналоговые преобразователи* — ЦАП. Для наиболее распространенных электрических сигналов АЦП и ЦАП выпускаются в виде больших интегральных микросхем. Их применение позволяет легко реализовать на практике описанные в данной главе математические методы обработки сигналов.

Как часто надо делать равномерные выборки произвольного сигнала, чтобы после преобразования в цифровую форму, а затем снова в аналоговую была сохранена форма сигнала? Ответ на этот важный вопрос дает теорема об отсчетах или теорема Котельникова (за рубежом именуемая также теоремой Найквиста): «Если спектр сигнала $e(t)$ ограничен высшей частотой f_b , то он без потери информации может быть представлен дискретными отсчетами с числом, равным $2 \cdot f_b$ ». При этом сигнал восстанавливается по его отсчетам $e(k \cdot dt)$, следующим с интервалом времени dt , с помощью фильтра низких частот, реализующего восстановление по формуле:

$$e(t) = \sum_{k=-\infty}^{\infty} e(k \cdot dt) \frac{\sin(\pi(t - k \cdot dt) / dt)}{\pi(t - k \cdot dt) / dt}.$$

Для восстановления непрерывного сигнала по его выборкам достаточно располагать функцией $\text{sinc}(x) = \sin(x)/x$ с учетом ее особого значения $\text{sinc}(x) = 1$ при $x = 0$. Такая функция есть в системе Mathcad 11/12, но ее нет в предшествующих версиях системы Mathcad. Однако, как показано на рис. 12.5 (часть документа снизу), такую

функцию несложно задать с помощью функции *if*. Заодно на рис. 12.5 снизу показано восстановление сигнала по его отсчетам на основе теоремы Котельникова.

Рисунок 12.5 показывает, что даже при небольшом числе отсчетов (в нашем случае их 11) восстановление сигнала (или, можно сказать, его интерполяция) происходит вполне прилично, хотя и не идеально. Интерполирующая кривая получается довольно плавной и непрерывной и точно проходит через точки исходных отсчетов. Увы, злополучный эффект Гиббса (колебания интерполирующей кривой, не присущие исходному сигналу) и здесь имеет место. Их причиной является ограниченное число выборок. При увеличении числа выборок точность восстановления можно существенно улучшить, что часто и делается на практике.

К примеру, популярные сейчас оптические компакт-диски содержат высококачественные записи речи и музыки в цифровой форме. Высшая частота звуковых сигналов 20 кГц. Частота дискретизации их 44,1 кГц (более 44000 выборок в секунду). Каждая выборка представляется (квантуется) с числом уровней $2^{16} = 65536$ (разрядность квантования 16 бит). Полученный поток цифровых данных и записывается в двоичном виде на оптический диск. В итоге компакт-диск при лазерном считывании информации позволяет воспроизводить сколько угодно раз звуки любимых певцов и оркестров с невиданным ранее качеством. Более того, возможна их многократная перезапись без малейшей потери качества. Таким образом, положения теоремы Котельникова это отнюдь не теоретическая абстракция, а реальный метод представления и восстановления сигналов.

12.1.7. Понятие о случайных сигналах

Рассмотренные выше сигналы являются *детерминированными* и их значения в любой момент времени можно точно вычислить. Временные зависимости или сигналы, один из параметров которых случайным образом зависит от времени, называют *случайными сигналами*. Среди таких сигналов особенно часто встречается сигнал в виде «белого шума». В простейшем случае это сигнал, который генерируется генератором случайных чисел — функция $\text{rnd}(a)$, где a — максимальное значение сигнала (минимальное в данном случае — 0). Амплитудные значения такого сигнала случайно распределены во времени по случайному закону. Бывает и белый шум, у которого частотный спектр сигнала равномерен.

Особо надо выделить *случайные стационарные сигналы*. Под этим, прямо скажем не вполне понятным, определением кроются сигналы, параметры которых, несмотря на изменение одного из них или ряда параметров по случайному закону, не содержат трендов. Подробное описание таких сигналов можно найти в [20].

Наконец, возможны сигналы, у которых случайные компоненты распределены по каким-то конкретным законам распределения. Они часто встречаются в специальных приложениях радиосвязи, но рассмотрение систем с такими сигналами далеко выходит за рамки данного раздела.

12.2. Основы спектрального метода моделирования сигналов и линейных систем

12.2.1. Схема спектрального метода

Методы обработки и моделирования сигналов ныне широко применяются в различных сферах науки и техники. Достаточно отметить анализ сигналов в меди-

цине, обработку сигналов сейсмографических исследований, анализ и распознавание звуковых и видеосигналов и изображений.

В общем случае *спектральный метод* моделирования линейных систем реализуется следующей диаграммой:

$$s_{\text{input}}(t) \rightarrow \text{DFT} \rightarrow \dot{F}(\omega) \rightarrow \dot{K}(\omega) \cdot \dot{F}(\omega) \rightarrow \text{IFT} \rightarrow s_{\text{output}}(t).$$

Входной сигнал подвергается прямому преобразованию Фурье DFT (Direct Fourier Transform) и раскладывается на множество гармоник, комплексные амплитуды которых задаются функцией $\dot{F}(\omega)$. Данный процесс означает *декомпозицию* сигналов, т. е. его разложение на отдельные гармонические составляющие. Он соответствует переходу от временного представления сигнала к его частотному представлению.

Частотные составляющие сигналы подаются на вход анализируемой линейной системы, которая характеризуется комплексным коэффициентом передачи $\dot{K}(\omega)$. Реакция системы характеризуется выражением $\dot{K}(\omega) \cdot \dot{F}(\omega)$. Эти функции часто раскладываются на действительную и мнимую составляющие, что позволяет задавать их амплитудно-частотными и фазочастотными характеристиками — АЧХ и ФЧХ.

Поскольку моделируется линейная система, то вычисление $\dot{K}(\omega) \cdot \dot{F}(\omega)$ означает вычисление реакции линейной системы на каждую частотную составляющую входного сигнала. При этом АЧХ сигнала (его спектр) и системы перемножаются, а фазовые сдвиги складываются. Полученный отклик системы (*композиция* его составляющих) в частотной форме подвергается обратному преобразованию Фурье IFT (Inverse Fourier Transform). В результате вычисляется временная зависимость сигнала на выходе системы $s_{\text{output}}(t)$.

Возможны самые разнообразные методы реализации описанной выше методики спектрального моделирования. Так, если $\dot{K}(\omega) = 1$, то можно говорить только о спектральном моделировании, анализе и синтезе сигналов, что нередко представляет самостоятельный интерес — например, для получения спектров произвольных сигналов и изучения их возможностей. По-разному могут осуществляться преобразования Фурье — по точным формулам, по приближенным формулам метода прямоугольников, по уточненным формулам, с помощью быстрого преобразования Фурье FFT (Fast Fourier Transform) и т. д.

Если сигнал $s_{\text{input}}(t)$ содержит шумовую компоненту, то данная схема анализа и синтеза сигналов способна решать такую задачу, как очистка сигнала от шума и его компрессия за счет ограничения спектра.

12.2.2. Программа точного вычисления коэффициентов Фурье

Для получения коэффициентов Фурье удобно пользоваться встроенными операторами интегрирования системы Mathcad. На рис. 12.6 приведено начало документа, открывающего обширные возможности в изучении спектрального анализа и синтеза различных сигналов. Здесь представлена программа, которая вычисляет коэффициенты разложения сигнала $f(t)$ в ряд Фурье при заданном полупериоде сигнала T (в дальнейшем $T = 1$) и заданном числе гармоник N . Эта программа заносит вектор коэффициентов Фурье A в нулевой столбец возвращаемой матрицы, а коэффициенты Фурье B в первый столбец.

Теперь мы можем провести исследование возможностей спектрального метода на основе преобразований Фурье на примерах типовых форм сигналов. Это позволит оценить нужное число гармоник и возможности данного метода.

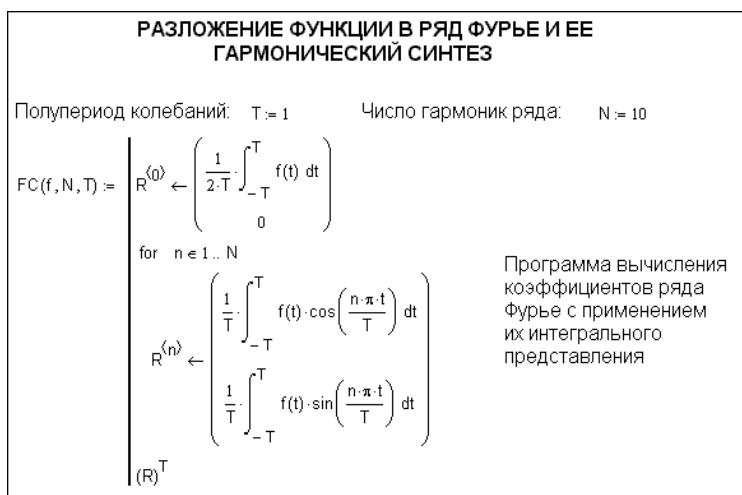


Рис. 12.6. Программный блок вычисления коэффициентов Фурье

12.2.3. Спектральный анализ и синтез пилообразного импульса

Начнем с рассмотрения сигнала пилообразного импульса, уровень которого падает со значения 1 до 0 за первый полупериод и равен 1 во время второго полупериода. Задание такого сигнала, вычисление его спектра и восстановление по формуле синтеза представлены на рис. 12.7. Отчетливо видно возникновение эффекта Гиббса — появление нарастающих к моменту $t = 0$ колебаний. Максимального значения они достигают в точке разрыва (перепада) — это и есть точка $t = 0$ для данного сигнала. Амплитуда колебаний достигает 9 % от перепада в точке разрыва.

Если сигнал имеет форму меандра, то амплитуда колебаний будет достигать 9 % от двойного перепада или 18 % от амплитуды меандра. Это очень искажает сигнал — см. анализ ниже.



Рис. 12.7. Спектральный анализ и синтез пилообразного импульса

12.2.4. Спектральный анализ и синтез меандра

Рассмотрим спектральный анализ и синтез прямоугольного симметричного импульса — меандра. На рис. 12.8 показано задание этого импульса с помощью функции *if*. На первом полупериоде значение импульса равно 1, а на втором —1. Таким образом, полный (двойной) перепад импульса равен 2. Это удваивает пульсации, вызванные эффектом Гиббса, так что максимальная амплитуда пульсаций достигает уже 18 %. Ясно, что здесь 10 гармоник явно недостаточно для хорошего представления импульса. В частности, отчетливо видна конечная крутизна синтезируемого колебания в точках разрыва исходного импульса.



Рис. 12.8. Спектральный анализ и синтез прямоугольного импульса — меандра при числе гармоник $N = 10$

Может показаться, что увеличение числа гармоник позволит улучшить представление такого импульса. На рис. 12.9 показан случай, когда число гармоник увеличено в 2.5 раза и достигает $N = 25$. В самом деле, крутизна синтезируемого колебания в точках разрывов (перепадов) возросла в 2.5 раза и импульс стал лучше напоминать меандр. Но эффект Гиббса сохранил свое «зловредное» лицо — частота пульсаций возросла в 2.5 раза, но их амплитуда осталась прежней.

По выведенным значениям элементов векторов коэффициентов **A** и **B** можно судить о спектральном составе импульсов. Так, рассмотренный выше пилообразный импульс имеет нулевые четные коэффициенты в векторе **A**. А меандр имеет вообще все нулевые коэффициенты в векторе **A**. В принципе, это означает, что их можно вообще не вычислять.

12.2.5. Спектральный анализ и синтез короткого прямоугольного импульса

Спектральный анализ и синтез короткого (длиною в четверть периода) прямоугольного импульса представлен на рис. 12.10. Для этого импульса характерны нулевые значения всех четных коэффициентов в векторах **A** и части четных коэффициентов в векторе **B**. Проявление эффекта Гиббса весьма заметно.



Рис. 12.9. Спектральный анализ и синтез прямоугольного импульса — меандра при числе гармоник N = 25

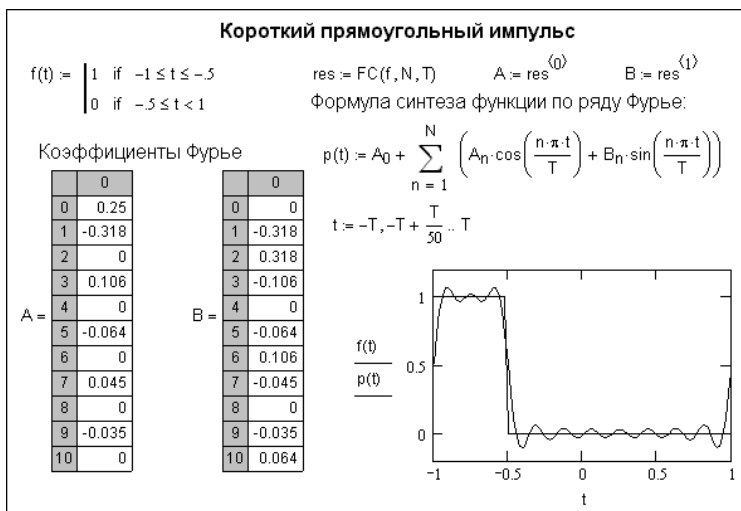


Рис. 12.10. Спектральный анализ и синтез прямоугольного импульса с длительностью в четверть периода

Вообще говоря, чем короче прямоугольный импульс, тем больше гармоник нужно для его представления.

12.2.6. Спектральный анализ и синтез «выпрямленной» синусоиды

Очень широкое распространение находит «сигнал» в виде выпрямленной синусоиды. Такой формы напряжение характерно, например, для двухполупериодного выпрямителя, работающего на активную нагрузку. Спектральный анализ такого сигнала позволяет оценить уровень фона и необходимость в сглаживании и фильтрации такого напряжения.

Рисунок 12.11 показывает спектральный анализ и синтез такого «сигнала» с помощью описываемой методики. Для этого «сигнала» характерны большое значение постоянной составляющей ($A_0 = 0.637$), нулевые нечетные коэффициенты в векторе A и все нулевые коэффициенты B . Это указывает на сравнительно легкую фильтрацию такого сигнала с целью выделения его постоянной составляющей. При этом для представления «сигнала» достаточно использовать небольшое число гармоник — в нашем случае $N = 5$.

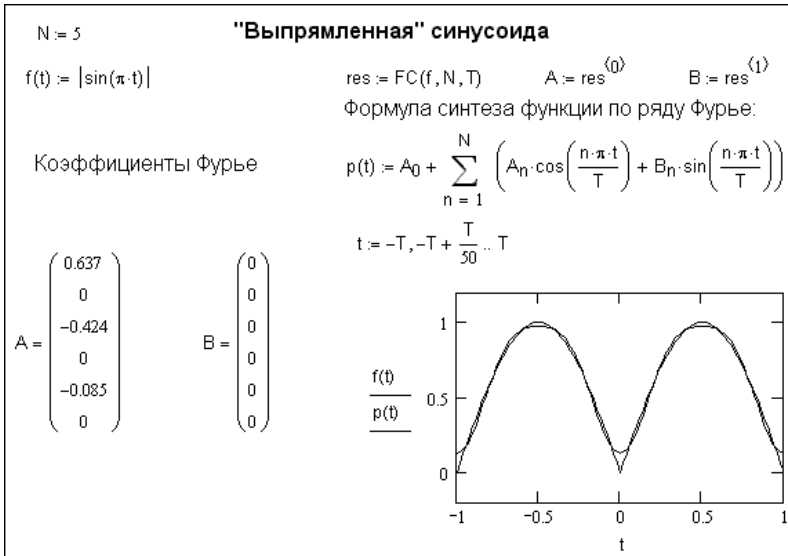


Рис. 12.11. Спектральный анализ и синтез «сигнала» выпрямленной синусоиды

Почему мы тут взяли слово «сигнал» в кавычки? Дело в том, что обычно сигналами называют временные зависимости, несущие информацию. В данном случае данная зависимость представляет не столько сигнал, сколько продукт работы электронного устройств — двухполупериодного выпрямителя. При этом стараются освободить этот продукт от всех гармоник, выделив только постоянную составляющую сигнала — компоненту, не меняющуюся во времени и к информационным процессам не относящуюся.

12.2.7. Спектральный анализ и синтез радиоимпульса

В заключение наших экспериментов рассмотрим спектральный анализ и синтез радиоимпульса, представленного пачкой из трех периодов синусоидальных колебаний (рис. 12.12).

Фурье-синтез такого сигнала выглядит вполне удовлетворительным, пожалуй, даже хорошим, уже при малом числе гармоник (в нашем случае $N = 10$). Хотя в начале и в конце радиоимпульса эффект Гиббса себя проявляет достаточно явно. Причина неплохих результатов в данном случае вполне очевидна — исходный сигнал является синусоидой и легко представим малым числом гармоник — тоже синусоид, из коих явно доминирует первая гармоника с частотой синусоидальных сигналов в пачке.

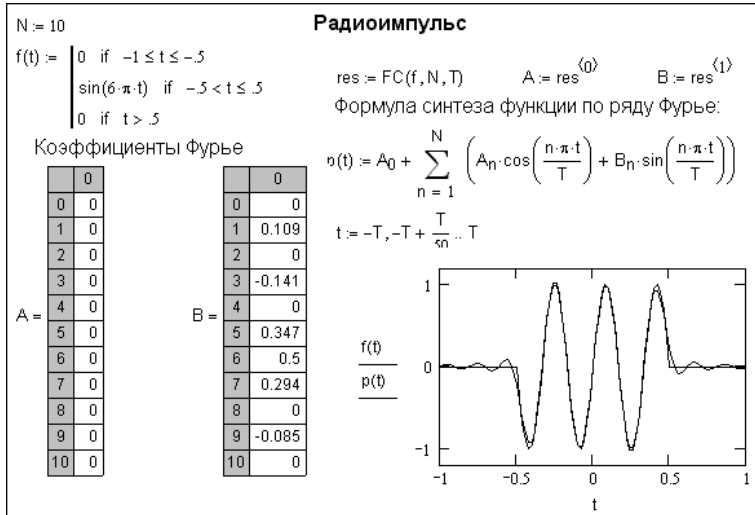


Рис. 12.12. Спектральный анализ и синтез радиоимпульса

12.3. Эффект Гиббса и борьба с ним

12.3.1. Математическая природа эффекта Гиббса

Мы уже столько раз упоминали эффект Гиббса, что пора разобраться с причинами его возникновения и обсудить способы борьбы с ним. В том, что с этим эффектом надо всерьез бороться, говорит тот факт, что амплитуда пульсаций синтезированных сигналов (самых по себе не имеющих пульсаций) достигает 9 %, а двойная амплитуда даже 18 % и практически не меняется при увеличении числа гармоник — просто выбросы при пульсациях становятся короче. Между тем известно множество задач, где точность приближения должна составлять значения, намного меньшие 1 %, а порою не превышать и тысячных долей процента.

Поскольку пульсации эффекта Гиббса наиболее явно проявляются в моменты скачков анализируемой функции, ограничимся рассмотрением случая представления скачка

$$f(t) = \begin{cases} 0, & t < 0 \\ 1, & t \geq 0 \end{cases}$$

ограниченным спектром. Ограничение спектра можно учесть, введя в преобразование Фурье общего вида

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{i\omega t} d\omega$$

умножение $F(\omega)$ на прямоугольное частотное окно

$$W(\omega) = \begin{cases} 1, & |\omega| \leq \gamma \\ 0, & |\omega| > \gamma \end{cases}$$

Это окно задает резкое ограничение спектра. Опуская детали вывода, которые можно найти в [34—37], найдем, что в этом случае

$$f(t) \approx \frac{1}{2} + \frac{1}{\pi} Si(\lambda \cdot t),$$

где функция $Si(t)$ известна как интегральный синус

$$Si(z) = \int_0^z \frac{\sin(x)}{x} dx.$$

Задание функции $Si(t)$ и построение зависимости $f(t)$ для данного случая представлено на рис. 12.13.

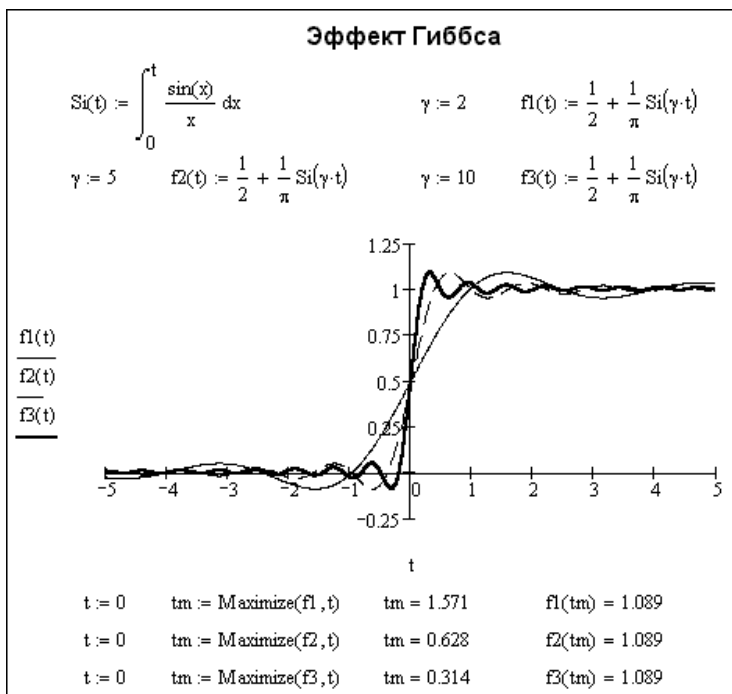


Рис. 12.13. К пояснению эффекта Гиббса

Итак, как следует из рассмотренного, эффект Гиббса существует, увы, как теоретически обоснованная реальность. И связан он прежде всего с неудачно подобранной (или просто поневоле существующей) формой частотного окна, резко ограничивающего число используемых при спектральном синтезе гармоник (частот), — прямоугольного.

Внизу рис. 12.13 представлен расчет экстремальной точки графика функции $f(t)$ при разных значениях параметра γ . Нетрудно заметить, что ордината максимума остается равной 1.089, т. е. относительная амплитуда-выброс для единичного перепада достигает примерно 9%, что уже отмечалось.

Наша русская поговорка «Клин клином вышибают» наводит на мысль, что, применяя более совершенные частотные окна, чем простейшее прямоугольное, можно ослабить эффект Гиббса или даже избавиться от него. И это совершенно

верный метод. К этой возможности мы будем возвращаться неоднократно. К сожалению, введение частотных окон неизбежно ухудшает реставрацию сигналов на участках их быстрых изменений.

А закончим мы этот раздел на слегка грустной нотке — оказывается, эффект Гиббса «всеяден» — он встречается в разложениях сигнала не только по синусоидальному базису, но и по другим базисам — в том числе и характерным для вейвлет-преобразований сигналов. Так что с его проявлением мы столкнемся еще не раз.

12.3.2. Подавление эффекта Гиббса с помощью сигма-множителей

К сожалению, встроенные в Mathcad функции прямого и обратного быстрого преобразований Фурье не устраняют эффект Гиббса. Более того, они критичны к выбору числа гармоник и числа отсчетов сигналов. Желательно, чтобы число отсчетов было равно 2^m , где m — натуральное число. Это не всегда удобно, особенно если функция задана малым числом отсчетов. В связи с этим нередко полезна обычная реализация спектрального анализа. Но перед ее описанием рассмотрим методы подавления эффекта Гиббса.

Одним из давно известных методов подавления эффекта Гиббса является применение сигма-множителей, уменьшающих амплитуды гармоник по мере роста их номера n при приближенном вычислении коэффициентов Фурье методом прямоугольников (рис. 12.14). На представленном документе даны ссылки на работы, в которых предложены методы борьбы с эффектом Гиббса.



Рис. 12.14. Синтез меандра по 10 гармоникам

Реализация метода сигма-множителей представлена рис. 12.15. Нетрудно заметить, что эффект Гиббса резко ослабляется, но небольшие его проявления все же остаются.

12.3.3. Подавление эффекта Гиббса с помощью множителей Фейера

Другой метод борьбы с эффектом Гиббса при использовании синтеза рядом Фурье заключается в применении множителей Фейера. Их определение и применение представлено на рис. 12.16.



Рис. 12.15. Синтез меандра по 10 гармоникам с сигма-множителями

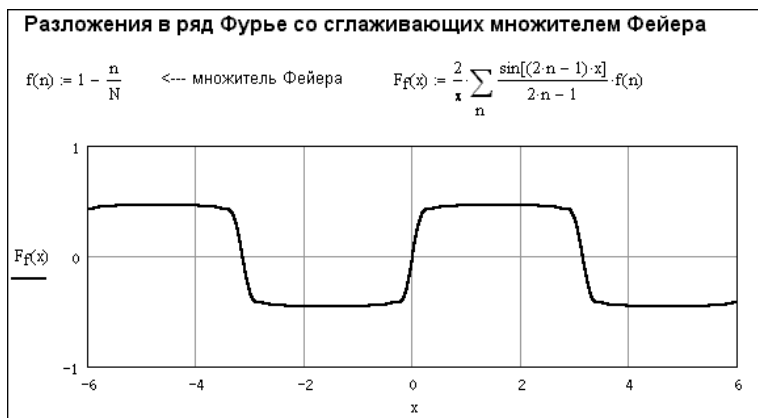


Рис. 12.16. Синтез меандра по 10 гармоникам с множителями Фейера

Этот метод обеспечивает практически полное подавление эффекта Гиббса. Но его недостатком является несколько скошенная вершина импульсов.

12.3.4. Улучшенное моделирование сигналов в применении их интерполяции

Кардинальное улучшение моделирования сигналов при спектральном методе достигается заменой дискретных отсчетов функции на плавную функцию $y(t)$, получаемую при той или иной методике интерполяции. При этом появляется возможность задавать произвольное число отсчетов по интерполируемой функции $y(t)$ и получить сколь угодно большое число гармоник для ее синтеза.

Одним из первых таких методов был метод Филона, при котором интерполяция проводилась с помощью полиномов второго порядка. Однако этот метод оказался сложным и редко применялся на практике. Более простым является метод кусочно-линейной интерполяции, который и реализован ниже [2]. В принципе, Mathcad позволяет легко реализовать и сплайновую интерполяцию, но проверка

показала, что кардинального улучшения по сравнению с линейной интерполяцией не наблюдается, тогда как время синтеза заметно возрастает.

В этом методе отсчеты берутся для несколько иной функции, но на практике весьма близкой к исходной — той, которой принадлежат исходные точки сигнала. Несмотря на эти оговорки, такой прием дает блестящие результаты — при достаточно большом числе выбранных для синтеза гармоник синтезируемая функция почти повторяет исходную, а эффект Гиббса исчезает практически полностью.

В документе рис. 12.17 показана реализация этого улучшенного метода для сигнала, отсчеты которого размещены в векторе Y_i . Коэффициенты Фурье рассчитываются методом прямоугольников. В документе приняты меры по уменьшению времени вычислений за счет отказа от умножений на ноль. Кроме того, расчетные формулы даны в виде, принятом в радиотехнике, так что частоту повторения сигнала f_1 можно задавать прямо в Герцах.

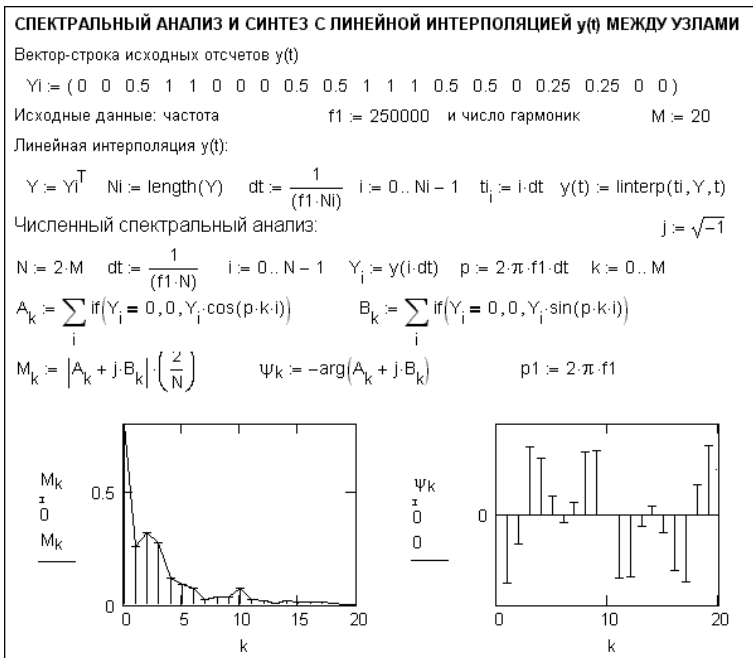


Рис. 12.17. Улучшенный спектральный анализ с линейной интерполяцией функции между узлами (начало документа)

Как видно из конца этого документа, показанного на рис. 12.18, степень совпадения исходной и синтезируемой функций очень высока. Это свидетельствует о высокой степени достоверности моделирования сложного сигнала на основе описанного подхода.

В приведенном примере использован достаточно сложный сигнал со скачками, полками и участками линейного изменения. В реальной ситуации функция $y(t)$ чаще всего является довольно плавной и эффект Гиббса проявляется в гораздо меньшей мере, чем в примененном тестовом примере. Это говорит о том, что все описанные методы имеют право на жизнь.

В представленную на рис. 12.18 формулу спектрального синтеза сигнала введен множитель, учитывающий изменение спектра сигнала при прохождении его через

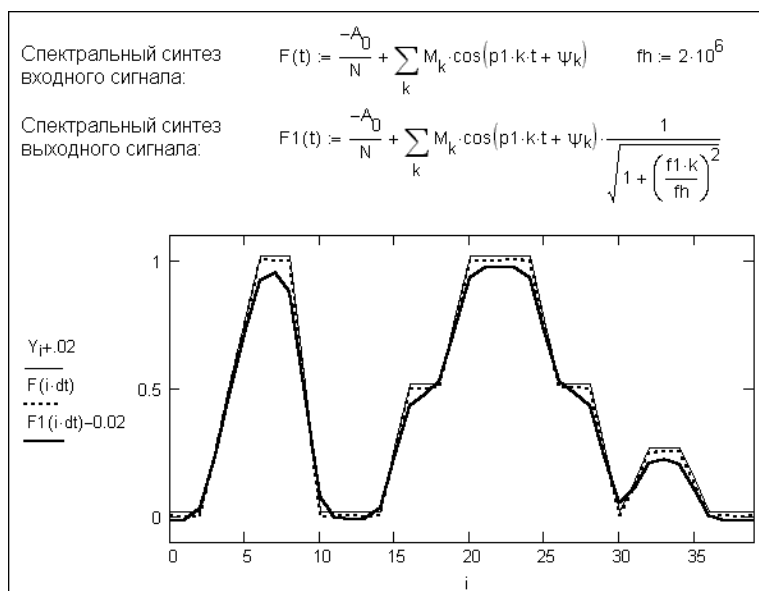


Рис. 12.18. Спектральный синтез на основании расчета спектра (конец документа)

низкочастотный фильтр или усилитель, ослабляющий высокие частоты (граничная частота fh). В результате получена временная зависимость сигнала с учетом искажений его этим устройством. Читатель может попробовать свои силы в решении подобных задач для других случаев, например при прохождении сигнала через многокаскадный усилитель. Таким образом, документ рис. 12.17 и 12.18 по существу является программой спектрального моделирования не только сигналов, но и искажающих или обрабатывающих их устройств.

12.4. Специальные приложения спектрального анализа и синтеза сигналов

12.4.1. Ограничения и недостатки преобразования Фурье

Вспомним, что в основе представления периодического сигнала рядом Фурье с ограниченным числом членов лежит соотношение:

$$y(t) \sim \sum_{n=0}^N c_n e^{in}, \quad (12.2)$$

где i — мнимая единица. Функция $y(t)$ представляется суммой произведений синусоид, представленных членом e^{in} , на коэффициенты Фурье c_n . Они вычисляются как:

$$c_n = \frac{1}{2\pi} \int_0^{2\pi} y(t) e^{-in} dt. \quad (12.3)$$

Коэффициенты Фурье — комплексные числа. Они описывают набор (спектр) гармонических сигналов, называемых гармониками. Число n — номер гармоники

(0, 1, 2, ...). Теоретически ряд Фурье имеет бесконечное число членов (гармоник), но на практике оно всегда конечно и равно N . Поэтому ряд (12.2) записывается как приближенный. Гармоники, представляющие собой синусоиды с разной частотой, кратной частоте первой гармоники, и разной фазой, образуются умножением c_n на e^{in} .

С позиций точного представления преобразованием Фурье произвольных сигналов и функций можно отметить целый ряд его недостатков:

- неприменимость к анализу нестационарных сигналов;
- преобразование Фурье даже для одной заданной частоты требует знание сигнала не только в прошлом, но и в будущем, что является теоретической абстракцией;
- в условиях практически неизбежного ограничения числа гармоник или спектра колебаний точное восстановление сигнала после прямого и обратного преобразований Фурье теоретически (и тем более практически) невозможно, в частности, из-за появления эффекта Гиббса;
- базисной функцией при разложении в ряд Фурье является гармоническое (синусоидальное) колебание, которое математически определено в интервале времени от $-\infty$ до $+\infty$ и имеет неизменные во времени параметры;
- численное интегрирование во временной области от $-\infty$ до $+\infty$ при прямом преобразовании Фурье (ППФ) и от $-\infty$ до $+\infty$ в частотной области при обратном преобразовании Фурье (ОПФ) встречает большие вычислительные трудности;
- отдельные особенности сигнала (например, разрывы или пики) вызывает незначительные изменения частотного образа сигнала во всем интервале частот от $-\infty$ до $+\infty$, которые «размазываются» по всей частотной оси, что делает их обнаружение по спектру практически невозможным;
- ясно, что такая плавная базисная функция, как синусоида, в принципе вообще не может представлять перепады сигналов с бесконечной крутизной, хотя такие сигналы (например, прямоугольные импульсы) применяются весьма широко;
- единственным приспособлением к представлению быстрых изменений сигналов, таких как пики или перепады, является резкое увеличение числа гармоник, которые оказывают влияние на форму сигнала и за пределами локальных особенностей сигнала;
- по составу высших составляющих спектра практически невозможно оценить местоположение особенностей на временной зависимости сигнала и их характер;
- для нестационарных сигналов (а таковых сейчас большинство) трудности ППФ и ОПФ (и, соответственно, быстрого преобразования Фурье (БПФ)) многократно возрастают.

Небольшие разрывы (ступеньки) на синусоидальном или любом плавно изменяющемся сигнале трудно обнаружить в его фурье-спектре, ибо они создают множество высших гармоник очень малой амплитуды — рис. 12.19. Сигнал здесь получен суммированием синусоиды с меандром, который моделируется выражением вида $0.1 \cdot \text{sign}(\sin(x))$. Таким образом ступенька имеет величину 0.1 от амплитуды синусоиды и видна на осциллограмме сигнала и его реставрации (для устранения слияния кривые раздвинуты по вертикали).

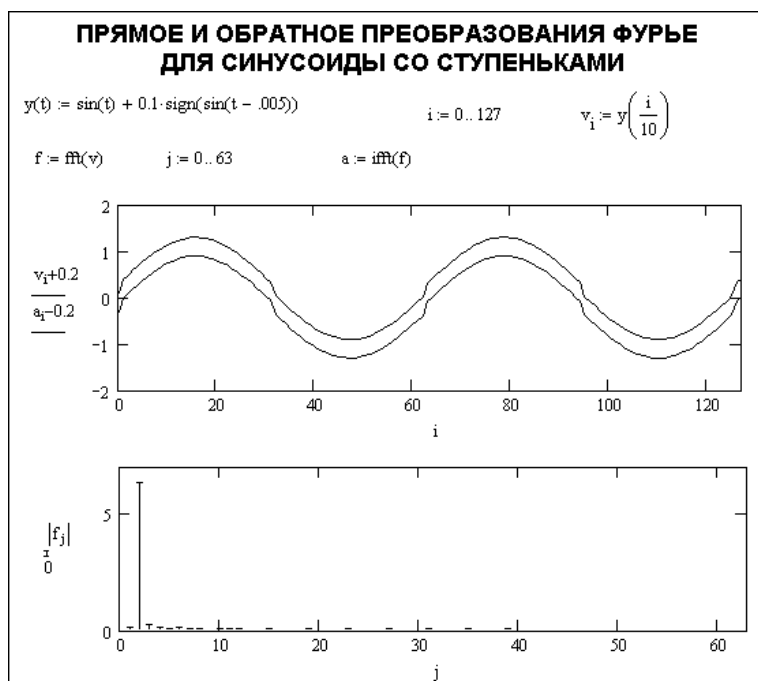


Рис. 12.19. Спектральный анализ и синтез синусоидального сигнала с небольшими ступеньками при переходе через нуль

Спектр таких сигналов (см. спектрограмму внизу рис. 12.19) содержит едва заметные высокочастотные составляющие спектра, по которым распознать локальную особенность сигнала и тем более ее место и характер практически невозможно. Составляющие спектр особенности как бы размазаны по оси частот.

12.4.2. Кратковременное (оконное) преобразование Фурье

Проблемы спектрального анализа и синтеза сигналов, ограниченных во времени, частично решаются переходом к так называемому кратковременному или *оконному преобразованию Фурье*. Идея этого преобразования очень проста — временной интервал существования сигнала разбивается на ряд промежутков — временных окон. В каждом промежутке вычисляется свое преобразование Фурье. Если в каком-то окне существовали частотные составляющие некоторого сигнала, то они будут присутствовать в спектре. А если нет — будут отсутствовать. Таким образом можно перейти к частотно-временному представлению сигналов, которое является особым разделом техники обработки сигналов.

Кратковременное (оконное) преобразование выполняется с использованием выражения:

$$A(\omega) = \int_{-\infty}^{\infty} y(t) \cdot w(t - b) \cdot e^{-i\omega t} dt. \quad (12.4)$$

Здесь, в отличие интеграла Фурье, функция $y(t)$ под знаком интеграла дополнительно умножается на оконную функцию $w(t-b)$. Параметр b окна задает его сдвиг на временной оси. Обычно задается ряд фиксированных значений b в пределах полного окна. Например, для простейшего *прямоугольного окна* функция $w(t-b)$ в пределах окна дает 1, а за пределами окна просмотра — 0. При этом для каждого окна мы получаем свой набор комплексных амплитуд сигнала в частотной области.

Казалось бы, раз оконное преобразование Фурье дает нам частотно-временное представление сигналов, то достаточно им и ограничиться. И не нужно было открывать вейвлет-преобразования, о которых уже писалось и будет еще писаться?

Однако ситуация не так проста! Она упирается в известный принцип неопределенности Гейзенберга. Согласно ему, невозможно получить одновременно высокое частотное и высокое временное разрешение. Выбирая окно с малой шириной по времени, мы получаем высокое временное разрешение, но низкое частотное разрешение. Взяв окно с большой шириной во времени, получаем хорошее разрешение по частоте, но плохое во времени. Оконное преобразование оперирует с окнами, имеющими одинаковую ширину, а потому данное противоречие для него неразрешимо. Мы вернемся к этой проблеме после рассмотрения современных методов представления спектра. А пока отметим еще один принципиальный недостаток оконного преобразования Фурье- в нем по-прежнему используется единственная базисная функций — синусоида со всеми ее отмеченными ранее недостатками.

Mathcad, к сожалению, не имеет средств для проведения оконного преобразования Фурье в своем ядре. Но эти средства есть в пакете расширения Signal Processing, описанном в книге [11].

12.4.3. Фильтрация зашумленных аналоговых сигналов

Под фильтрацией сигнала в общем случае подразумевается выделение полезного сигнала из его смеси с мешающим сигналом (часто шумом). Наиболее распространенный тип фильтрации — частотная фильтрация. Если известна область частот, занимаемых полезным сигналом, достаточно выделить эту область и подавить те области, которые заняты шумом.

Начальный фрагмент документа Mathcad, показанный на рис. 12.20, иллюстрирует технику фильтрации с применением БПФ. В начале документа синтезируется двухчастотный сигнал, представленный 128 отсчетами вектора q . Затем к этому сигналу добавляется шум, получаемый с помощью генератора случайных чисел, и формируется вектор s из 128 отсчетов зашумленного сигнала. Путем использования прямого БПФ сигнал с шумом преобразуется из временной области в частотную, что создает вектор f из 64 частотных составляющих. Затем выполняется фильтрующее преобразование, эффективность которого оценивается параметром фильтрации a . Отфильтрованный сигнал (вектор g) подвергается обратному БПФ, в результате чего создается вектор выходного сигнала h .

В конце документа (рис. 12.21) сравниваются временные зависимости исходного и выходного сигналов. Хотя абсолютной идентичности между ними нет, видно, что выходной сигнал почти повторяет входной и в значительной мере избавлен от высокочастотных шумовых помех, маскирующих полезный сигнал.

Алгоритмы, реализованные в данном примере, находят широкое применение на практике. Они лежат в основе создания фильтров для аналоговых сигналов.



Рис. 12.20. Фильтрация аналогового сигнала с применением БПФ (начало документа)

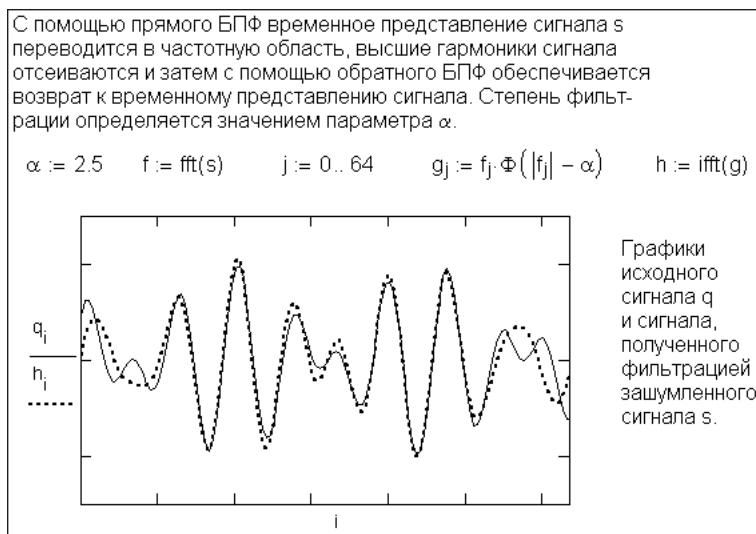


Рис. 12.21. Фильтрация аналогового сигнала с применением БПФ (окончание документа)

12.4.4. Цифровая фильтрация с помощью БПФ

Аналоговые фильтры обычно имеют довольно сложную аппаратную реализацию. В них применяются громоздкие и нетехнологичные колебательные контуры

Заметим, что в данном примере речь идет не о фильтрации сигнала на фоне шумов, а о прохождении сложного составного сигнала через цифровые фильтры. Графики в конце документа иллюстрируют характер искажений сигнала при этом.

12.4.5. Фильтрация пространственного образа с применением БПФ

До сих пор мы приводили примеры применения одномерного БПФ. Однако возможно и двумерное преобразование Фурье, позволяющее решать более сложные задачи. Рассмотрим типичную задачу на сложную обработку изображений.

Система Mathcad 2001i/11 реализует двумерное БПФ, если в операциях БПФ использовать не векторы, а матрицы. Однако у ранних версий Mathcad таких возможностей нет, поэтому воспользуемся более наглядной реализацией двумерного БПФ с использованием функций одномерного БПФ, применяемого по строкам и столбцам матрицы поверхности, искаженной шумами. На рис. 12.24 показан фрагмент документа Mathcad, реализующий фильтрацию поверхности с помощью такого БПФ.

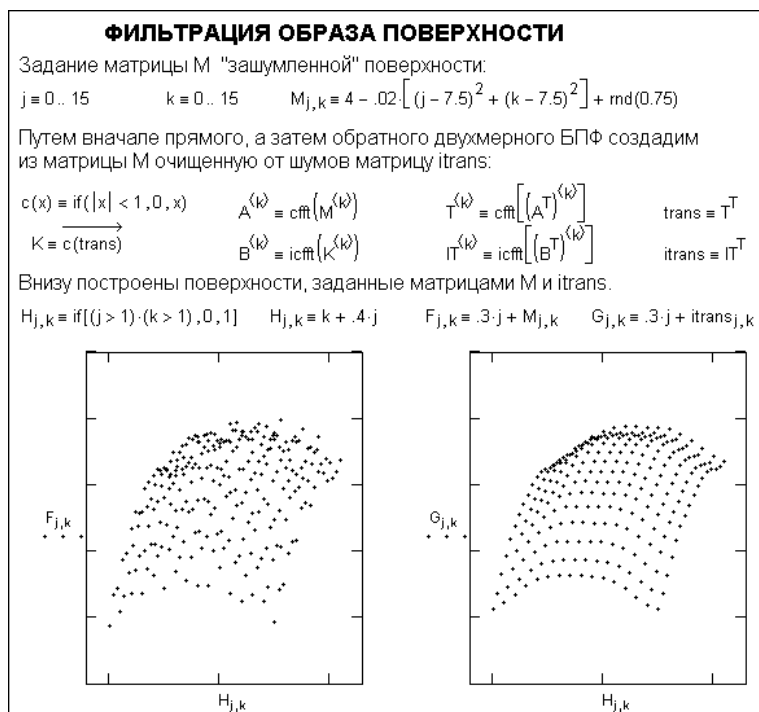


Рис. 12.24. Фильтрация пространственного образа

В начале документа создается матрица поверхности М со случайными отклонениями каждой точки. Внизу слева построена эта поверхность. Она имеет вид пятна, в котором отдаленно угадываются общие контуры поверхности. Регулярность расположения точек-крестиков на ней вообще не заметна. В средней части документа показана реализация математического аппарата фильтрации. График справа внизу иллюстрирует результат фильтрации. На нем отчетливо видны не только общая форма поверхности (типа «выпуклый лист»), но и явная регулярность распо-

ложения точек. Это и есть результат фильтрации. Возможно, что двумерное преобразование Фурье позволит улучшить этот результат. Заинтересованный читатель может проверить это.

Полезно отметить, что графики поверхности здесь также получены без применения стандартных средств построения поверхностей системы Mathcad. Поэтому без какой-либо доработки (кроме выбора шрифта комментариев) этот пример можно использовать и в ранних версиях системы.

12.5. Инженерные методы спектрального анализа

12.5.1. Спектральный анализ методом Берга

Для некоторых простых зависимостей $y(t)$ амплитуды гармоник могут выражаться аналитически [42, 43]. Примером служат отрезки синусоиды, получаемые выделением только верхней ее части. На практике такие колебания широко используются в радиотехнике, где обрезание нижней части синусоиды обусловлено работой электронных приборов (например, ламп или транзисторов) в нелинейном режиме.

Доля периода косинусоиды, используемой для анализа гармоник, оценивается углом отсечки θ (измеряется в радианах). Он, к примеру, равен π , если обрезается вся нижняя половина косинусоиды. Удобно вычислять относительную амплитуду k -й гармоники (по отношению к усеченной амплитуде косинусоидального импульса). Этот параметр для разных значений k впервые был вычислен А. И. Бергом.

Во фрагменте документа Mathcad, показанном на рис. 12.25, представлены формулы для коэффициентов Берга α_0 (относительная постоянная составляющая сигнала), α_1 (относительная амплитуда первой гармоники) и α_n (относительная амплитуда n -й гармоники). Коэффициенты Берга являются функциями нормированного угла отсечки.

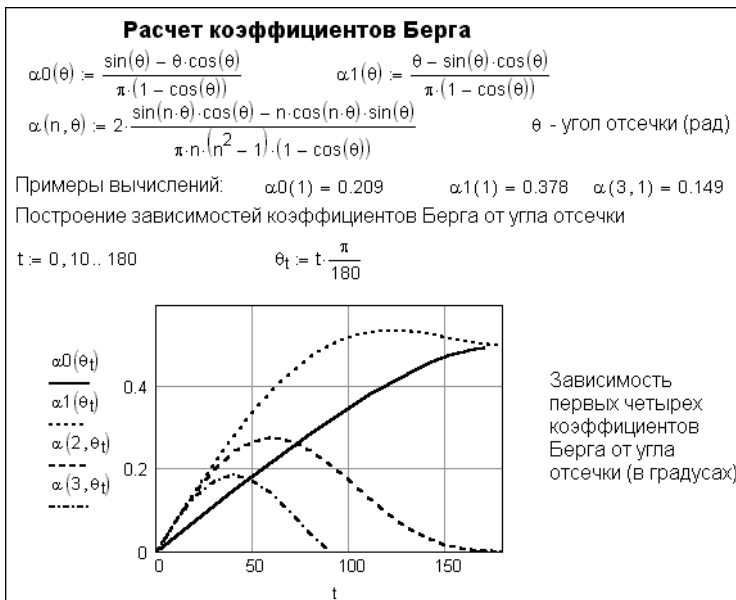


Рис. 12.25. Спектральный анализ методом Берга

График в этой части этого фрагмента дает наглядное представление об изменении первых четырех коэффициентов Берга с изменением угла отсечки. Нетрудно заметить существование максимумов у этих зависимостей. Например, максимальная амплитуда первой гармоники будет при угле отсечки 120 градусов, второй гармоники — 60 градусов и т. д. Знать эти углы полезно при проектировании умножителей частоты (например, удвоителей или утроителей), работа которых основана на фильтрации одной из высших гармоник.

Насколько точно коэффициенты Берга представляют собой отрезки косинусоиды? Ответ на этот вопрос дает вторая часть документа, представленная на рис. 12.26. Здесь представлена функция синтеза отрезков косинусоиды по коэффициентам Берга для трех значений углов при числе используемых гармоник $N = 8$.

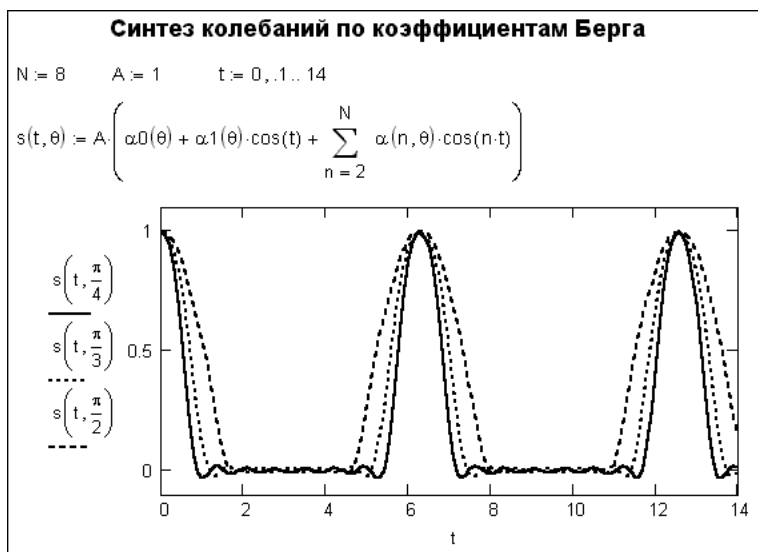


Рис. 12.26. Спектральный синтез по коэффициентам Берга

Нетрудно заметить, что синтез по коэффициентам Берга и впрямь неплохо описывает отрезки косинусоиды. Если увеличить число гармоник N с 8 до более высоких значений, то можно существенно повысить точность синтеза.

12.5.2. Спектральный анализ методом пяти ординат

Метод пяти ординат является распространенным в радиотехнике инженерным методом оценки малой нелинейности усилительных устройств. Он заменяет трудоемкие ручные расчеты амплитуд гармоник вычислением их по простым формулам, в которых задается лишь 5 ординат зависимости $y(x)$, описывающей нелинейное устройство, на входе которого действует синусоидальный сигнал с заданной амплитудой x_{\max} и смещением x_0 .

Недостатком этого метода в классическом варианте является необходимость задания нелинейной функции точно пятью ординатами. Кроме того, трудоемким является пересчет амплитуд гармоник при изменении амплитуды входного сигнала, что нужно для расчета зависимости коэффициента нелинейных искажений от амплитуды сигнала на входе усилителя.

На рис. 12.27 показана первая часть документа Mathcad, реализующего этот метод. В отличие от классического варианта здесь нелинейность задается любым числом ординат — от пяти и более. Данные о ней задаются как элементы векторов X и Y, хранящих значения функции $y(x)$. Эта функция задается сплайнами, что вполне оправдано, поскольку она имеет слабую нелинейность.

Конец документа представлен на рис. 12.28. Здесь заданы формулы метода пяти ординат и показана возможность множественного расчета этим методом с целью вычисления зависимости коэффициента гармоник от амплитуды синусоидального напряжения, воздействующего на нелинейное устройство, например усилитель.

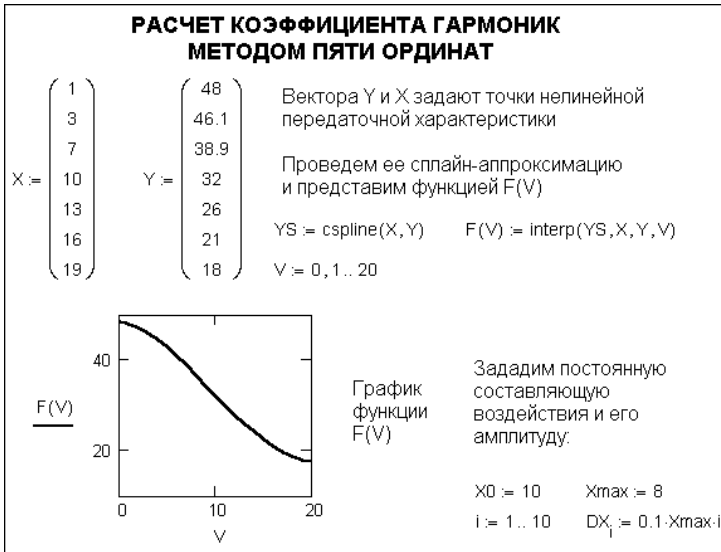


Рис. 12.27. Начало документа с реализацией метода пяти ординат

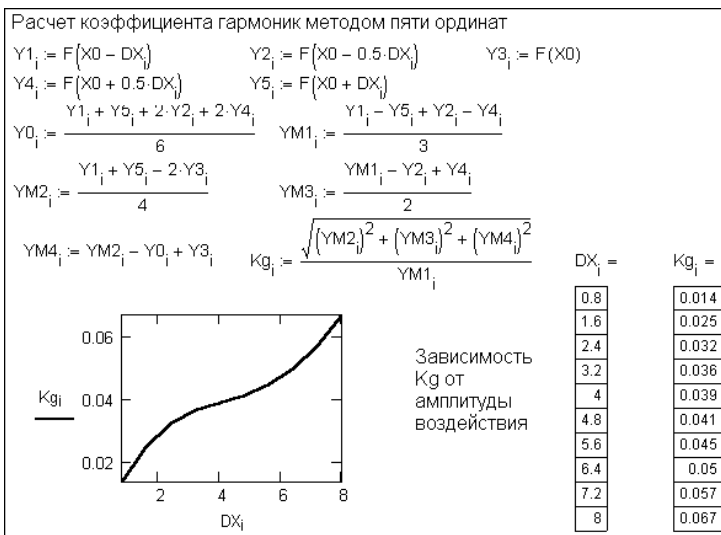


Рис. 12.28. Окончание документа с реализацией метода пяти ординат

Метод пяти ординат достаточно прост при однократных вычислениях коэффициента гармоник. Его реализация в системе Mathcad позволяет выполнять такие вычисления многократно.

12.5.3. Спектральный анализ методом двенадцати ординат

Если нелинейность зависимости $y(x)$ значительна, то можно использовать еще один инженерный метод спектрального анализа — *метод двенадцати ординат*. Этот метод основан на простых вычислениях коэффициентов Фурье по двенадцати ординатам функции $y(x)$, абсциссы которых расположены равномерно на отрезке определения $y(x)$.

На рис. 12.29 представлено начало документа Mathcad с реализацией метода двенадцати ординат. В нем задан вектор ординат и дана часть вычислений — в основном подготовительных.

СПЕКТРАЛЬНЫЙ АНАЛИЗ МЕТОДОМ 12 ОРДИНАТ

$y := \begin{pmatrix} 0 \\ 1 \\ 4 \\ 5 \\ 4 \\ 3 \\ 3.5 \\ 3.5 \\ 3 \\ 1 \\ 0.5 \\ 0.2 \\ 0 \end{pmatrix}$	<p>Вектор y задает 12 ординат некоторой функции, которая периодически повторяется с периодом $T=1$ (условная единица)</p> <p>Метод 12 ординат позволяет найти синусные и косинусные составляющие спектрального разложения данной функции, используя при этом довольно простые расчетные формулы, позволяющие быстро провести вычисления</p>																								
	<table style="width: 100%; border: none;"> <tr> <td>$s0 := y_{12}$</td> <td>$s1 := y_1 + y_{11}$</td> <td>$s2 := y_2 + y_{10}$</td> <td>$s3 := y_3 + y_9$</td> </tr> <tr> <td>$s4 := y_4 + y_8$</td> <td>$s5 := y_5 + y_7$</td> <td>$s6 := y_6$</td> <td>$d1 := y_1 - y_{11}$</td> </tr> <tr> <td>$d2 := y_2 - y_{10}$</td> <td>$d3 := y_3 - y_9$</td> <td>$d4 := y_4 - y_8$</td> <td>$d5 := y_5 - y_7$</td> </tr> <tr> <td>$o0 := s0 + s6$</td> <td>$o1 := s1 + s5$</td> <td>$o2 := s2 + s4$</td> <td>$o3 := s3$</td> </tr> <tr> <td>$r0 := s0 - s6$</td> <td>$r1 := s1 - s5$</td> <td>$r2 := s2 - s4$</td> <td>$q1 := d1 + d5$</td> </tr> <tr> <td>$q2 := d2 + d4$</td> <td>$q3 := d3$</td> <td>$n1 := d1 - d5$</td> <td>$n2 := d2 - d4$</td> </tr> </table>	$s0 := y_{12}$	$s1 := y_1 + y_{11}$	$s2 := y_2 + y_{10}$	$s3 := y_3 + y_9$	$s4 := y_4 + y_8$	$s5 := y_5 + y_7$	$s6 := y_6$	$d1 := y_1 - y_{11}$	$d2 := y_2 - y_{10}$	$d3 := y_3 - y_9$	$d4 := y_4 - y_8$	$d5 := y_5 - y_7$	$o0 := s0 + s6$	$o1 := s1 + s5$	$o2 := s2 + s4$	$o3 := s3$	$r0 := s0 - s6$	$r1 := s1 - s5$	$r2 := s2 - s4$	$q1 := d1 + d5$	$q2 := d2 + d4$	$q3 := d3$	$n1 := d1 - d5$	$n2 := d2 - d4$
$s0 := y_{12}$	$s1 := y_1 + y_{11}$	$s2 := y_2 + y_{10}$	$s3 := y_3 + y_9$																						
$s4 := y_4 + y_8$	$s5 := y_5 + y_7$	$s6 := y_6$	$d1 := y_1 - y_{11}$																						
$d2 := y_2 - y_{10}$	$d3 := y_3 - y_9$	$d4 := y_4 - y_8$	$d5 := y_5 - y_7$																						
$o0 := s0 + s6$	$o1 := s1 + s5$	$o2 := s2 + s4$	$o3 := s3$																						
$r0 := s0 - s6$	$r1 := s1 - s5$	$r2 := s2 - s4$	$q1 := d1 + d5$																						
$q2 := d2 + d4$	$q3 := d3$	$n1 := d1 - d5$	$n2 := d2 - d4$																						

Рис. 12.29. Начало документа с реализацией метода двенадцати ординат

Окончание документа с реализацией метода двенадцати ординат представлено на рис. 12.30. Нетрудно заметить, что этот метод позволяет найти коэффициенты Фурье для приближенного описания функции.

Из рисунка видно, что синтезируемая по методу двенадцати ординат функция неплохо описывает исходную функцию, узловые точки которой были взяты за основу спектрального анализа этим методом. Разумеется, это справедливо, если функция, подлежащая анализу, достаточно гладкая. Метод двенадцати ординат широко применяется до сих пор в электротехнике, хотя для системы Mathcad он выглядит как восстановленный довоенный автомобиль.

12.6. Вейвлет-обработка сигналов

12.6.1. Пример вейвлет-разложения и реконструкции прямоугольного импульса

Mathcad стали первыми массовыми системами, в ядро которых были включены функции прямого и обратного дискретного вейвлет-преобразования на основе

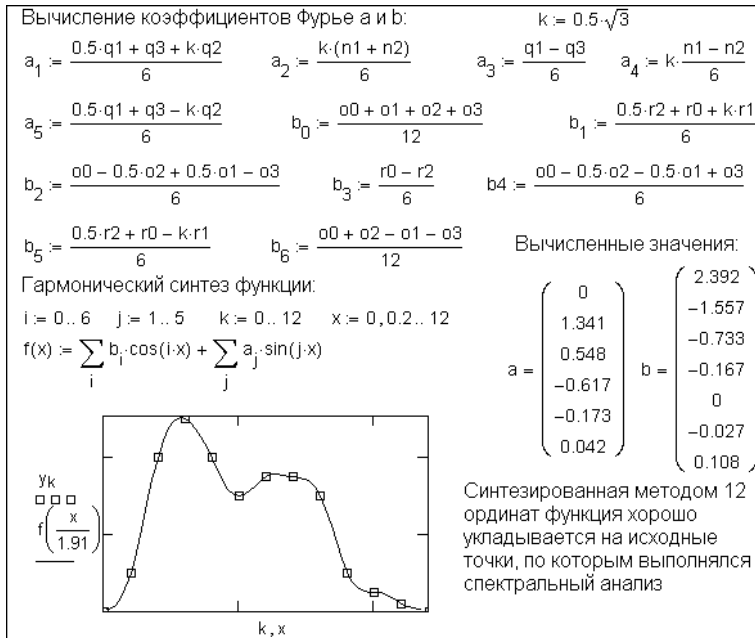


Рис. 12.30. Окончание документа с реализацией метода двенадцати ординат

вейвлетов Добеши порядка 4 (DB4 или db4). Мы уже описывали функции вейвлет-преобразований в главе 8, так что продолжим эту тему ради постепенного освоения возможностей вейвлетов. Пока же отметим, что вейвлеты это уже третий способ синтеза сигналов на основе ряда с произведениями отсчетов на базисные функции [38—41]. Два таких способа — синтез на основе базиса Котельникова и на основе базиса Фурье мы уже рассмотрели.

Рисунок 12.31 показывает пример осуществления прямого дискретного вейвлет-преобразования для прямоугольного импульса. Для наглядности результаты преобразования представлены в виде графиков пяти первых вейвлет-коэффициентов. Нетрудно убедиться, что для этого типа вейвлета коэффициенты представлены прямоугольными (или ступенчатыми) волнами разной амплитуды, длительности и формы.

Некоторые детали разложения сразу бросаются в глаза:

- очевидно, что форма вейвлет-коэффициентов благоприятна для синтеза скачков и импульсов, содержащих скачки;
- вейвлет-коэффициенты представляют собой прямоугольные колебания с явно кратной частотой осцилляций;
- число максимально возможных уровней разложения невелико;
- вейвлет-коэффициенты хранятся в матрице, и их всегда можно выделить (например для построения графиков отдельных коэффициентов или организации обработки и реконструкции сигналов).

Вторая часть документа, представленная на рис. 12.32, демонстрирует простую технику удаления части коэффициентов путем обнуления соответствующих ячеек матрицы вейвлет-коэффициентов. Это позволяет ограничить число используемых коэффициентов значением L и тем самым менять уровень композиции (реконст-

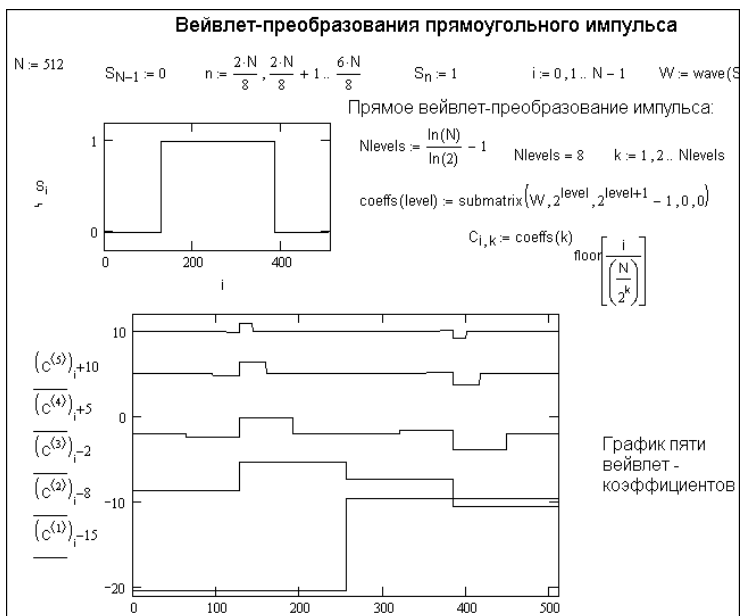


Рис. 12.31. Пример вейвлет-декомпозиции для прямоугольного импульса (начало документа)

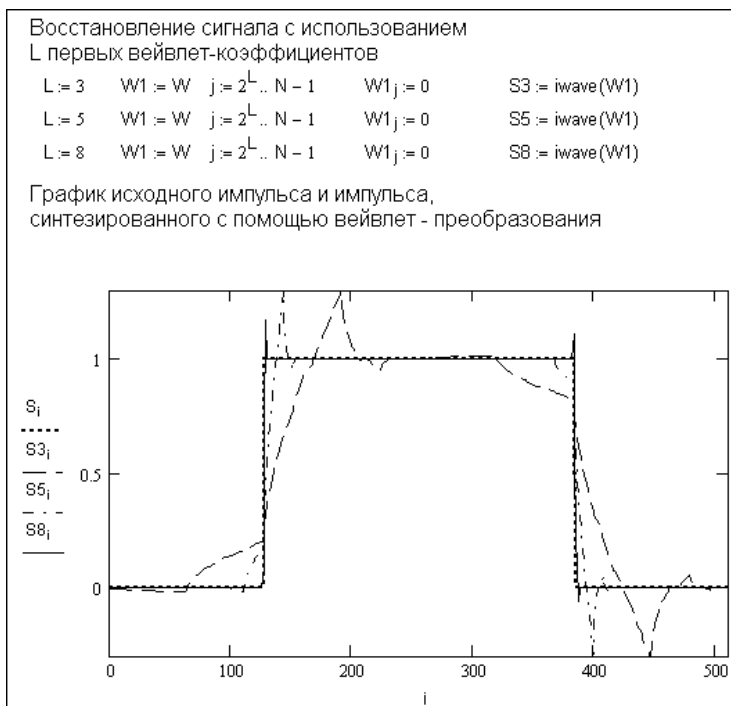


Рис. 12.32. Пример реконструкции сигнала — прямоугольного импульса (конец документа)

рукции) сигнала. Рис. 12.32 дает временные зависимости реставрированного сигнала для трех уровней $L = 3, 5$ и 8 .

Из представленного рисунка видно, что ограничение уровня реставрации ведет к появлению характерных для эффекта Гиббса выбросов в местах, где сигнал имеет скачки уровня. Чем ближе уровень реставрации к максимальному значению $Nlevel$, тем точнее реставрация. Что касается формы выбросов, то она иная, чем при эффекте Гиббса фурье-синтеза. Если у последнего эта форма характерна для затухающей синусоиды и не имеет острых верхушек, то при вейвлет-синтезе форма колебаний определяется формой самого вейвлета. Грубые вейвлеты DB4 имеют выбросы с острыми верхушками.

12.6.2. Пример вейвлет-разложения и реконструкции прямоугольного импульса с линейно нарастающей вершиной

Теперь рассмотрим несколько более сложный сигнал — импульс с крутыми нарастанием и спадом и линейно растущей вершиной (линейным трендом в формулировке, принятой в анализе временных рядов). Такой сигнал можно рассматривать как суперпозицию прямоугольного и пилообразного сигналов. Здесь (см. рис. 12.33) вновь построены пять вейвлет-коэффициентов, представляющих декомпозицию сигнала.

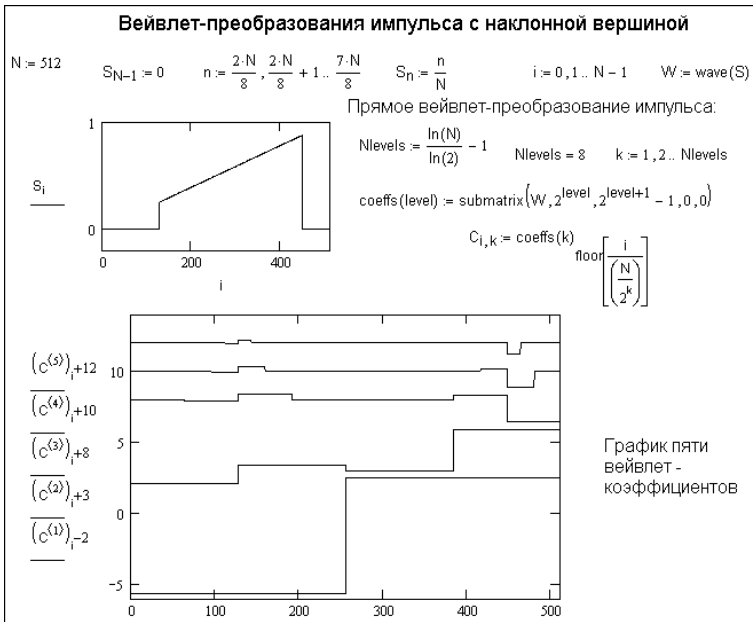


Рис. 12.33. Пример декомпозиции импульса с линейно нарастающей вершиной (начало документа)

Тут любопытно отметить, что, несмотря на отсутствие линейных участков у самих вейвлет-коэффициентов, линейно нарастающая часть импульса приближается превосходно, без малейших намеков на ступенчатость — рис. 12.34. Здесь показано восстановление сигнала для трех уровней $L = 3, 6$ и 8 .

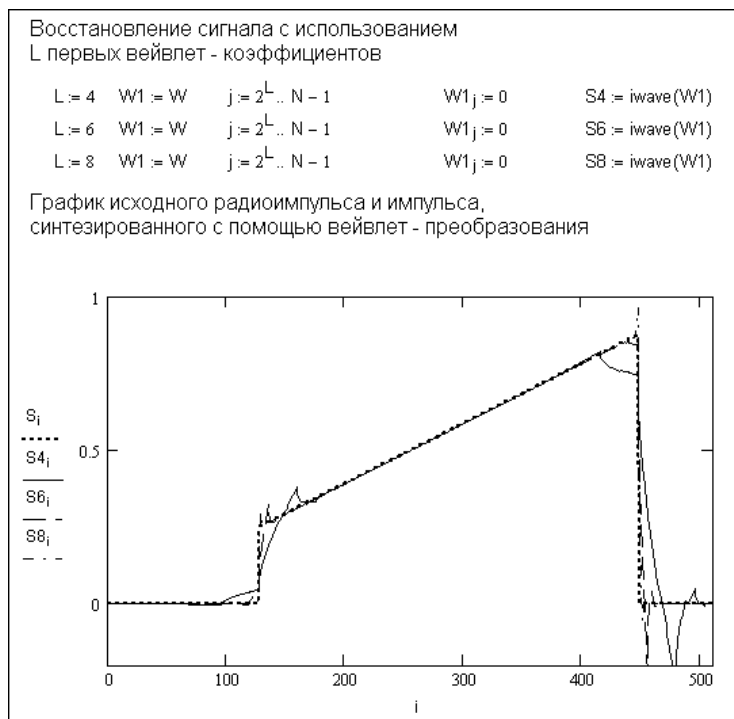


Рис. 12.34. Пример реставрации импульса с линейно нарастающей вершиной (конец документа)

12.6.3. Вейвлет-разложение и реконструкция радиоимпульса

А теперь рассмотрим вейвлет-преобразования для еще одного характерного сигнала — радиоимпульса, представляющего собой вырезку синусоидального сигнала. Соответствующий документ с прямым вейвлет-преобразованием представлен на рис. 12.35. По-прежнему для данного типа вейвлета коэффициенты представляют собой прямоугольные формы.

Несмотря на такую форму вейвлет-коэффициентов они весьма недурно воспроизводят довольно сложный сигнал в виде радиоимпульса. Это видно из окончания документа рис. 12.36, представленного на рис. 12.35. Здесь реконструированный сигнал дан для уровней реконструкции $L = 3, 5$ и 8 .

В этом примере, как и в других, видно проявление эффекта Гиббса и в результатах вейвлет-синтеза. Однако форма колебаний тут несколько иная — она определяется временной зависимостью вейвлетов. Можно заметно уменьшить эффект Гиббса в данном случае, выбрав более гладкие вейвлеты.

12.6.4. Сравнение вейвлет- и фурье-анализа и синтеза

Итак, мы проиллюстрировали процесс дискретных вейвлет-преобразований для трех видов сигналов. Все они представляют трудности для стандартного преобразования Фурье из-за наличия скачков и резкого проявления эффекта Гиббса. Тем не менее во всех случаях простейшие функции преобразований (для всего одного типа вейвлета из многих десятков известных типов) дали вполне приличные результаты.

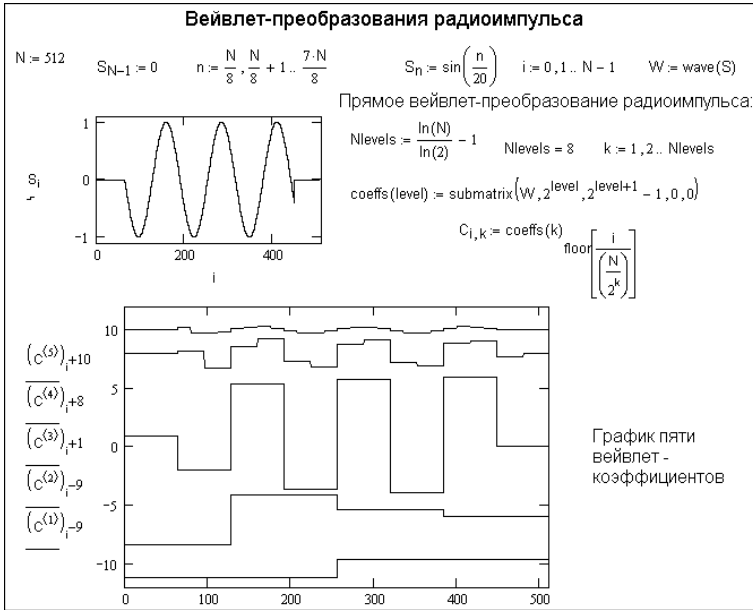


Рис. 12.35. Пример декомпозиции радиопульса (начало документа)

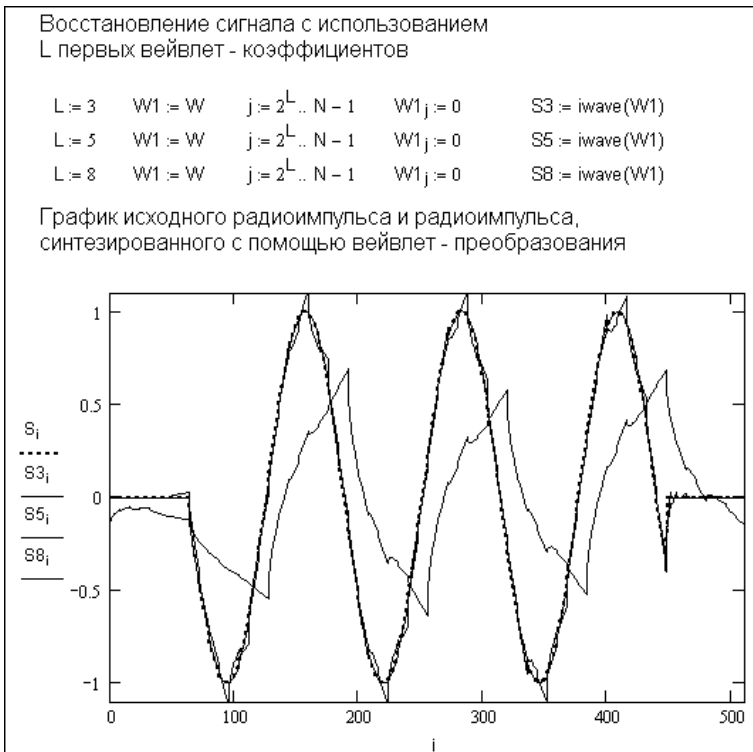


Рис. 12.36. Пример реконструкции радиопульса (конец документа)

В некоторой мере мы даже преуспели в выделении вейвлет-коэффициентов и в удалении части из них, что, по существу, означает осуществление процесса вейвлет-фильтрации. Последняя, естественно, сопровождается некоторыми искажениями — отличиями реконструированного сигнала от оригинального. Порою весьма заметными. Это хорошо видно во всех временных диаграммах реставрированного сигнала для различных уровней реставрации.

В целом можно отметить, что фурье-анализ и синтез дает хорошие результаты для плавно изменяющихся сигналов, а вейвлет-анализ и синтез для сигналов с перепадами и разрывами. Последние более характерны для цифровых сигналов, находящихся широкое применение. Более глубокий анализ, выполненный в главе 15, показывает, что для любого вида сигнала вейвлет-преобразование позволяет получить лучшие результаты, по сравнению с преобразованием Фурье. Это, в частности, относится к эффективности сжатия сигналов и подавления шумов.

Однако пока за пределами нашего рассмотрения остались такие важные вопросы, как:

- применение вейвлетов других классов (кроме DB4 сейчас известны десятки других типов вейвлетов);
- сравнение выгоды от применения вейвлет-преобразований или фурье-преобразований при решении важных конкретных задач обработки сигналов и изображений, например компрессии сигналов и очистки их от шума;
- использование спектров Фурье и вейвлет-спектров для анализа особенностей сигналов и прогноза временных рядов.

Изучение этих вопросов требует более основательного знакомства с практикой как преобразований Фурье, так и вейвлет-преобразований. Этому посвящены главы 14 и 15 книги [11]. В них читатель и найдет ответы на эти и многие другие вопросы.

12.7. Обработка рисунков

12.7.1. Общие замечания по обработке рисунков

В системах компьютерной математики работа с рисунками подобна работе с сигналами. Рисунки получают с помощью сканеров, цифровых фотоаппаратов, графических редакторов и т. д. Они могут быть представлены матрицами, задающими координаты их точек — *пикселов*. Для представления черно-белых рисунков достаточно одной матрицы с плотностями черного цвета, тогда как цветной рисунок должен представляться тремя матрицами — для плотностей красного (Red), зеленого (Green) и синего (Blue) цветов. Таким образом, для создания рисунков реализуется *цветовая схема RGB*.

Для обработки рисунков широко используются специальные графические редакторы, например PhotoShop, Ulead Photo Impact и др. Разумеется, по большому счету, Mathcad не составит им конкуренцию в части оперативной обработки рисунков. Однако такая обработка в среде Mathcad имеет одно важное преимущество — она математически прозрачна, что незаменимо при создании новых алгоритмов обработки изображений и познании уже применяемых.

12.7.2. Обработка монохромных изображений

Mathcad 2001i/11/12 с помощью функции READBMP позволяет считывать монохромные изображения, представленные файлами с расширением .BMP. Эта функ-

ция возвращает матрицу, размер которой определяется размером рисунка. Каждый элемент возвращаемой матрицы соответствует пикселу исходного изображения и имеет значение кода плотности черного цвета от 0 до 255.

Если указать имя этой матрицы в месте ввода шаблона рисунка, то будет построено монохромное изображение рисунка, показанное в левой части рис. 12.37. В этом фрагменте документа показана также часть матрицы M с данными изображения Моны Лизы с ее загадочной улыбкой.



Рис. 12.37. Обработка монохромного изображения

Над матрицей изображения могут выполняться все доступные для матриц преобразования. Например, можно получить матрицу M1, дающую негативное изображение (на рисунке в центре):

$$M1 := \frac{-M}{2} + 175.$$

С помощью функции submatrix можно из матрицы выделить подматрицу S меньшего размера:

$$S := \text{submatrix}(M, 60, 160, 50, 150)$$

Этот случай обработки изображения (вырезание части изображения) представлен на рисунке справа. Разумеется, для правильной работы этого документа нужно наличие файла с рисунком и правильное указание доступа к нему в функции READBMP.

12.7.3. Обработка цветных изображений

Для построения цветных изображений в системе Mathcad используется известный метод разложения на составляющие цветовой модели RGB. При нем изображение представляется суммой трех составляющих с красным (Red), зеленым (Green) и синим (Blue) цветами. Соответственно функция READRGB обеспечивает считывание цветных изображений из файлов формата BMP и возвращает встроенный массив D, содержащий три монохромных изображения, представляющих яркости трех указанных цветов (рис. 12.38). Их простое сложение на одном рисунке просто повторяет исходное изображение.

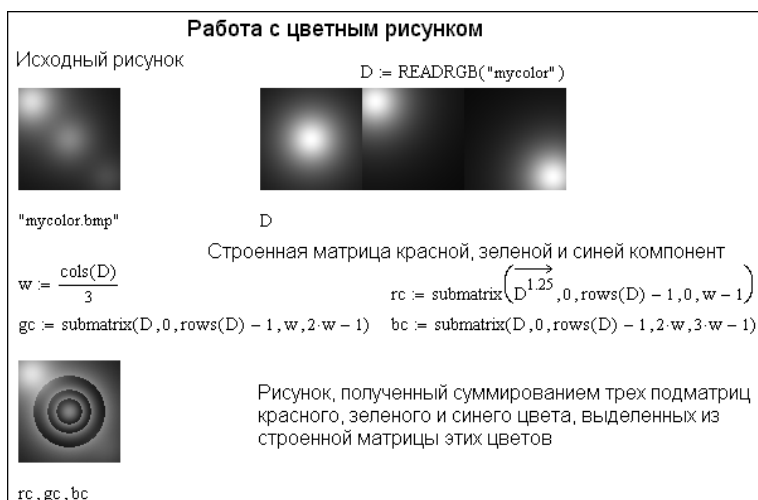


Рис. 12.38. Пример обработки цветного изображения

С помощью функции `submatrix` несложно выделить из массива `D` три массива `rc`, `rg` и `rb`, несущие информацию отдельно по каждому из трех цветов. После этого можно обрабатывать их по заданному алгоритму. Такая обработка возможна и в процессе задания подмассивов. Так, на рисунке подмассив красного цвета получен возведением каждого элемента массива `D` в степень 1.25. Если код интенсивности цвета какого-то элемента массива становится больше 255, из него вычитается число 256. Таким образом, происходит «внезапное» резкое изменение цвета, создающее при суммировании массивов характерное, напоминающее интерференцию, изменение цвета красного пятна в центре синтезированного рисунка, как показано в левом нижнем углу рисунка (к сожалению, черно-белого).

12.7.4. Функции для работы с файлами и матрицами рисунков

Mathcad 2001i/11/12 помимо вышеупомянутых функций `READBMP` и `READRGB` поддерживает еще ряд функций для работы с файлами и матрицами рисунков. Это функции считывания файлов разного формата со словом `READ` в имени, а также функции записи матриц рисунков в виде файлов со словом `WRITE` в названии. Применение этих функций достаточно очевидно и подобно описанному для функций `READBMP` и `READRGB`. При этом поддерживается расширенный набор форматов файлов: `BMP`, `PCX`, `JPG` и `TIF`. Это позволяет вести обработку файлов как монохромных, так и цветных изображений на вполне профессиональном уровне. Полный набор функций для работы с файлами и матрицами рисунков представлен в приложении Б.

Обработка представленных матрицами изображений может выполняться самыми разнообразными математическими методами. Например, для фильтрации изображений могут применяться быстрое преобразование Фурье и волновые преобразования. Более полное описание методов математической и художественной обработки изображений, к сожалению, выходит за рамки данной книги. Тем не менее мы рассмотрим перспективные методы обработки рисунков на основе вейвлет-преобразований.

12.7.5. Пример вейвлет-обработки и компрессии изображения

Для многих вейвлет-обработка изображений кажется настолько сложной, что ее нереально выполнять в системе Mathcad. Тем более что в ядре системы имеется всего две функции для прямого `wave` и обратного `iwave` преобразований одномерного массива. Ниже представлен документ, который показывает, что это не так и что вейвлет-обработка изображений вполне возможна в системе Mathcad даже без применения специальных пакетов расширения.

На рис. 12.39 представлено начало документа, в котором выполняется прямое двумерное вейвлет-преобразование сигнала изображения, которое хранится в графическом файле формата BMP, и задается число (в процентах) удаляемых вейвлет-коэффициентов `thres`. Идея преобразования достаточно проста — двумерный массив изображения `M` развертывается в одномерный и задается функция двумерного вейвлет-преобразования `wave2D`. После этого мы получаем матрицу вейвлет-коэффициентов `N`.

Вейвлет-преобразования монохромного изображения

`M := READBMP("lena")` Считываем файл исходного изображения

`thres := 95` Задаем число (в %) отсекаемых вейвлет-коэффициентов

Вычислим размеры изображения (в пикселях):
`width := cols(M)` `width = 256` `height := rows(M)` `height = 256`

Определим вспомогательные функции:

$$\text{Makepad}(n,m) := \begin{cases} \text{for } x \in 1..m \\ \text{for } y \in 1..n \\ A_{x,y} \leftarrow 0 \\ A \end{cases}$$

$$\text{ComputePad}(n) := \begin{cases} x \leftarrow \frac{\log(n)}{\log(2)} \\ 0 \text{ if } (x - \text{floor}(x)) = 0 \\ 2^{\text{floor}(x+1)} - n \text{ otherwise} \end{cases}$$

`Padx(A) :=` $\begin{cases} n \leftarrow \text{ComputePad}(\text{cols}(A)) \\ A \text{ if } n = 0 \\ \text{augment}(A, \text{Makepad}(n, \text{rows}(A))) \text{ otherwise} \end{cases}$ `M1 := Padx(M)` `cols(M1) = 256`

`Pady(A) :=` $\begin{cases} n \leftarrow \text{ComputePad}(\text{rows}(A)) \\ A \text{ if } n = 0 \\ \text{stack}(A, \text{Makepad}(\text{cols}(A), n)) \text{ otherwise} \end{cases}$ `M1 := Pady(M)` `rows(M1) = 256`

`wave2D(A) :=` $\begin{cases} \text{for } i \in 1.. \text{cols}(A) \\ A^{(i)} \leftarrow \text{wave}(A^{(i)}) \\ A \leftarrow A^T \\ \text{for } i \in 1.. \text{cols}(A) \\ A^{(i)} \leftarrow \text{wave}(A^{(i)}) \\ A^T \end{cases}$ `N := wave2D(M1)` Зададим и выполним двумерное вейвлет-преобразование

M =

	71	72	73	74	75
1	100	97	97	96	103
2	100	97	97	96	103
3	101	99	97	97	102
4	97	96	98	102	99
5	97	99	99	96	98
6	95	101	100	99	101
7	99	101	100	99	97
8	99	99	97	99	97
9	99	98	97	98	98
10	100	99	98	98	102
11	99	96	95	101	95

N =

	71	72	73
1	77.37	105.54	-273.91
2	28.33	171.02	-65.87
3	15.77	-38.45	-29.07
4	3.78	-36.07	-76.67
5	11.71	-18.71	-46.51
6	0.75	6.71	-2.63
7	6.74	-10.76	-0.05
8	-10.5	-87.34	167.55
9	10.04	-43.27	-9.27
10	-2.33	-5.17	-6.76
11	4.22	-5.84	1.41

Рис. 12.39. Прямое вейвлет-преобразование изображения из графического файла

Для пояснения смысла вейвлет-преобразований в левой части рис. 12.39 представлена исходная матрица изображения `M` и матрица вейвлет-коэффициентов `N`. Элементы матрицы `M` — целые числа от 0 до 255, которые характеризуют яркость каждой точки изображения — от черного цвета 0 до белого 255. Элементы матрицы `N` — вейвлет-коэффициенты, которые представляют собой по существу временные зависимости.

Следующая часть документа (рис. 12.40) демонстрирует технику компрессии изображения и его восстановления. Здесь после вычисления вспомогательных функций задается вычисление порога th . Если какой-то вейвлет-коэффициент имеет абсолютное значение меньше th , ему просто задается нулевое значение. В результате получается разреженная матрица N , файл которой можно подвергнуть компрессии любым архиватором.

The screenshot shows the Mathcad Professional interface with several user-defined functions and a matrix N .

Vectorize(A) :=
$$\begin{cases} \text{for } i \in 1.. \text{cols}(A) \\ \quad v \leftarrow A^{(i)} \text{ if } i = 1 \\ \quad v \leftarrow \text{stack}(v, A^{(i)}) \text{ otherwise} \\ \quad v \end{cases}$$
 Зададим функцию векторизации матрицы и проведем векторизацию:
 $v := \text{Vectorize}(N)$

Vabs(v) :=
$$\begin{cases} \text{for } i \in 1.. \text{rows}(v) \\ \quad v_i \leftarrow -v_i \text{ if } v_i < 0 \\ \quad v \end{cases}$$
 Зададим функцию вычисления абсолютного значения элементов вектора:
 $v := \text{Vabs}(v)$ $v := \text{sort}(v)$

$\text{skip} := (\text{cols}(M1) - \text{width}) \cdot \text{rows}(M1) + (\text{rows}(M1) - \text{height}) \cdot \text{cols}(M1)$

Вычисление порогового значения вейвлет-коэффициентов:

$th := v \cdot \text{skip} + \text{floor}\left(\frac{\text{thresh}}{100} \cdot \text{width} \cdot \text{height}\right)$ $th = 42.513$

Threshold(A, t) :=
$$\begin{cases} \text{for } i \in 1.. \text{rows}(A) \\ \quad \text{for } j \in 1.. \text{cols}(A) \\ \quad \quad A_{i,j} \leftarrow 0 \text{ if } |A_{i,j}| < t \\ \quad \quad A \end{cases}$$
 Определение функции для порогового ограничения вейвлет-коэффициентов:
 $N := \text{Threshold}(N, th)$

iwave2(A) :=
$$\begin{cases} \text{for } i \in 1.. \text{cols}(A) \\ \quad A^{(i)} \leftarrow \text{iwave}(A^{(i)}) \\ \quad A \leftarrow A^T \\ \quad \text{for } i \in 1.. \text{cols}(A) \\ \quad \quad A^{(i)} \leftarrow \text{iwave}(A^{(i)}) \\ \quad A^T \end{cases}$$
 Определим функцию для вычисления двумерного обратного вейвлет-преобразования:
 $M1 := \text{iwave2}(N)$
 $M1 := \text{submatrix}(M1, 1, \text{height}, 1, \text{width})$

	71	72	73
0	105.54	-273.91	0
1	171.02	-65.67	-49.48
2	0	0	0
3	0	-76.67	87.97
4	0	-46.51	0
5	0	0	0
6	0	0	0
7	-87.34	167.55	-66.36
8	-43.27	0	0
9	0	0	0
10	0	0	0
11	0	0	0
12	0	0	0
13	0	0	-57.66
14	0	0	0
15	0	59.42	0

Рис. 12.40. Обратное вейвлет-преобразование изображения с компрессией

На рис. 12.41 построено исходное изображение (по матрице M) и изображение, полученное в результате вейвлет-преобразований и компрессии (по матрице $M1$). Нетрудно заметить, что несмотря на высокую степень компрессии (использовалось только 5 % ненулевых вейвлет-коэффициентов, что соответствует степени компрессии 20) изображение достаточно отчетливо, хотя и имеет заметные погрешности в сравнении с оригиналом. Помимо округления передачи полутонов заметны светлые точки на наиболее темных участках изображения. Это типичное проявление эффекта Гиббса, жесткости примененных вейвлетов DB4 и резкого порога удаления вейвлет-коэффициентов.

Указанные погрешности изображения заметно уменьшаются при уменьшении степени компрессии. На рис. 12.42 показано сравнение оригинального изображения с изображением, в котором сохранено только 20 % коэффициентов (степень сжатия 5). В этом случае разницу в изображениях приметить довольно трудно.



Рис. 12.41. Исходное (слева) и преобразованное (справа) изображения при степени компрессии 20



Рис. 12.42. Исходное (слева) и преобразованное (справа) изображения при степени компрессии 5

12.8. Основы применения операторного метода

12.8.1. Вводные замечания к операторному методу

Многие линейные системы (управления, электро- и радиотехнические) успешно анализируются и моделируются операторным методом. Заметим, что он основан на замене дифференциальных уравнений алгебраическими уравнениями. При этом часто используется такой прием: система, устройство или цепь, находящаяся под воздействием, описывается в виде линейных уравнений в комплексном виде. При этом $\sigma + j\omega$ (или $\sigma + j\omega$) формально заменяется на символ оператора Лапласа s , что позволяет получить систему алгебраических уравнений в операторной форме. Затем с помощью обратного преобразования Лапласа можно получить импульсную характеристику (реакцию на единичный импульс) анализируемого объекта, а взяв от нее интеграл — переходную характеристику (реакцию на единичный скачок).

Некоторые преобразования Лапласа Mathcad легко выполняет в режиме символьных вычислений. Ниже представлены преобразования, выполняемые в командном режиме:

1	$\frac{1}{s}$	1
t	$\frac{1}{s^2}$	t
$\exp\left(\frac{-t}{\tau}\right)$	$\frac{1}{\left(s + \frac{1}{\tau}\right)}$	$\exp\left(\frac{-t}{\tau}\right)$
$1 - \exp\left(-\frac{t}{\tau}\right)$	$\frac{1}{[s \cdot (s \cdot \tau + 1)]}$	$1 - \exp\left(\frac{-t}{\tau}\right)$
$\sin(\omega \cdot t)$	$\frac{\omega}{(s^2 + \omega^2)}$	$\sin(\omega \cdot t)$
$\exp\left(\frac{-t}{\tau}\right) \cdot \sin(\omega \cdot t)$	$\frac{\omega}{\left[\left(s + \frac{1}{\tau}\right)^2 + \omega^2\right]}$	

В большинстве примеров как прямое, так и обратное преобразования благополучно выполняются. Но уже в последнем примере (тоже отнюдь не слишком сложном) Mathcad отказывается от вычислений и выводит окно с надписью «No symbolic result was found» (Результат в символьной форме не находится). В таких случаях приходится обращаться к реализации преобразований Лапласа в численной форме.

12.8.2. Основные характеристики линейных систем

Основной характеристикой линейных систем является передаточная операторная характеристика $y(s)$. Заменяя в ней s на $i\omega$, можно вычислить амплитудно-частотную характеристику АЧХ как модуль полученного выражения и фазочастотную характеристику как аргумент этого выражения. Пример этого представлен на рис. 12.43.

На рис. 12.43 показано также теоретическое выражение для переходной характеристики ПХ системы $h(t)$, которая определяет реакцию системы на входное воздействие в виде единичного скачка (функции Хевисайда). Mathcad легко вычисляет ПХ систем с простыми передаточными характеристиками, например $hi(t)$ для нашего примера. Как уже отмечалось, обратное преобразование Лапласа над $y(s)$ создает импульсную характеристику системы, а над $y(s)/s$ - переходную (умножение $y(s)$ на $1/s$ означает интегрирование). К сожалению, такие аналитические преобразования далеко не всегда возможны, особенно если функция $y(s)$ содержит дополнительные переменные (см. примеры выше в разделе 12.8.1).

Вычисление $h(t)$ по приведенной теоретической формуле осуществляется довольно медленно. Причина кроется в трудоемкости вычисления несобственного интеграла для ПХ, который имеет бесконечный верхний предел. Поэтому не рекомендуется задавать слишком малую погрешность вычислений TOL (в нашем слу-

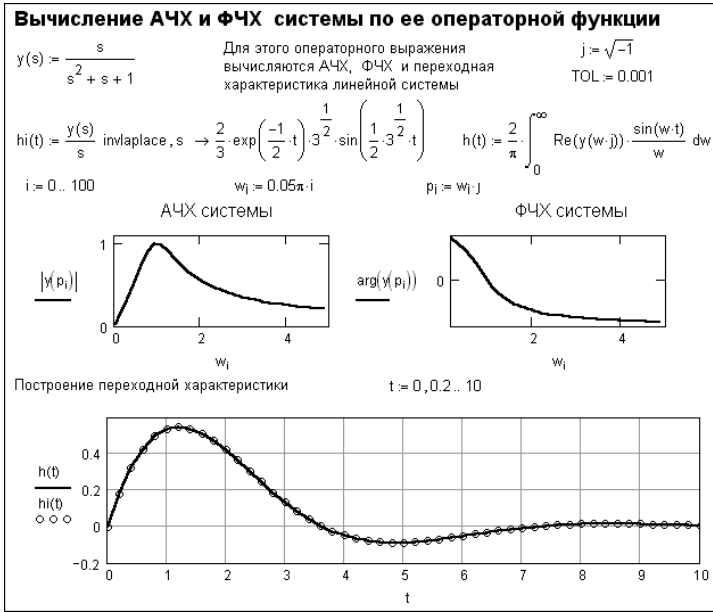


Рис. 12.43. Построение АЧХ, ФЧХ и ПХ линейной системы по ее операторной передаточной характеристике

чае $\text{TOL} = 0.001$). В связи с указанным обстоятельством разумно применять и другие методы расчета ПХ, например, описанный ниже. Можно также заменить бесконечный предел на конечный, если исходная функция ограничена во времени (в нашем случае замена бесконечности на 10 уже вполне пригодна).

12.8.3. Программа построения диаграмм Бode

Прежде чем привести пример на еще один операторный метод вычисления переходной характеристики линейной системы, рассмотрим построение диаграмм Бode — так называют логарифмические амплитудно-частотные и фазочастотные характеристики линейных систем. Линейная система обычно характеризуется передаточной операторной характеристикой $F(s)$. На рис. 12.44 представлен программный модуль системы Mathcad, который строит диаграммы Бode для заданной функции F в диапазоне частот от минимальной MinFreq до максимальной MaxFreq . Заметим, что нулевую частоту MinFreq брать при логарифмическом масштабе нельзя.

Рисунок 12.45 дает пример построения диаграммы Бode для некоторой системы с заданной передаточной характеристикой $GH(s)$. Рисунок слева показывает логарифмическую амплитудно-частотную, а справа фазо-частотную характеристики данной системы.

Вид диаграмм Бode для специалиста говорит о многом. Но подробное описание тонкостей операторного метода в задачи данной книги не входит. Отметим лишь, что диаграмма Бode для усилителя с цепью разорванной обратной связи позволяет судить об устойчивости усилителя после его охвата этой обратной связью.

12.8.4. Построение переходной характеристики на основе БПФ

Рисунок 12.46 показывает программный модуль, в котором преобразование производится разбивкой передаточной характеристики на $N = 128$ частей и при-

```

Построение диаграммы Бode для функции F(s)
BodePlot(F, MinFreq, MaxFreq) :=
  ndk ← 20
  n ← ndk · ceil( log( MaxFreq / MinFreq ) )
  r ← ln( MaxFreq / MinFreq )
  for i ∈ 0.. n
    ω ← MinFreq · ei·r/n
    Zi,0 ← ω
    V ← F(j·ω)
    Zi,1 ← 20 · log(|V|)
    Zi,2 ← arg(V) · 180 / π
  for i ∈ 1.. n
    Zi,2 ← Zi,2 - 360 if (|Zi,2 - Zi-1,2|) > 300
  Z
    
```

Рис. 12.44. Программный модуль для построения диаграмм Бode

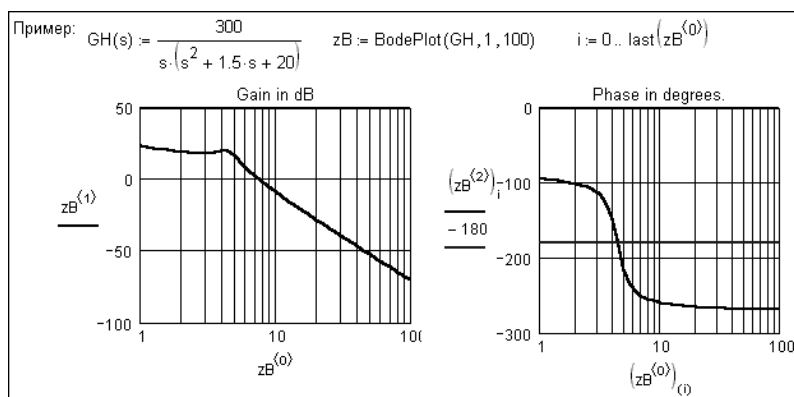


Рис. 12.45. Пример построения диаграмм Бode

менением быстрых преобразований Фурье. Число N можно менять, но оно должно быть двойкой в целой степени. Входными параметрами функции TimeResponse является имя передаточной операторной функции F и время T_{\max} , на котором определяется переходная характеристика.

В деталях алгоритма преобразования несложно разобраться, изучив команды программного модуля. Функция TimeResponse возвращает матрицу из двух столбцов. Нулевой столбец содержит отсчеты времени t , а первый — отсчеты переходной характеристики $h(t)$.

На рис. 12.47 представлен пример построения переходной характеристики системы второго порядка, заданной достаточно сложной передаточной характеристикой. Она содержит ряд параметров, что препятствует получению аналитического выражения системой Mathcad. Тем не менее такое выражение существует и приведено на рис. 12.47 в виде функции пользователя $f(t)$.


```

Функция вычисления переходной характеристики по F(s)
TimeResponse(F, Tmax) :=
  N ← 128
  FFTtime ← 1.5 · Tmax
  T ←  $\frac{\text{FFTtime}}{N}$ 
  epst ← 0.05
   $\sigma \leftarrow \frac{-\ln(0.1 \cdot \text{epst})}{\text{FFTtime}}$ 
  for i ∈ 0..  $\frac{N}{2}$ 
     $\omega \leftarrow \frac{i \cdot 2 \cdot \pi}{\text{FFTtime}}$ 
    SFi ← F(σ - j · ω)
  SF $\frac{N}{2} \leftarrow \text{Re}\left(\text{SF}\frac{N}{2}\right)$ 
  ft ←  $\frac{\text{ifft}(\text{SF})}{T \cdot \sqrt{N}}$ 
  for k ∈ 0..  $\frac{N \cdot T_{\text{max}}}{\text{FFTtime}}$ 
    Zk,0 ← k · T
    Zk,1 ← eσ · k · T · ftk
  Z
  
```

Рис. 12.46. Программный модуль для преобразования операторной передаточной характеристики в переходную характеристику



Рис. 12.47. Пример построения переходной характеристики по передаточной характеристике

Если система находится под внешним воздействием и ее переходная характеристика известна, то ее реакцию на такое воздействие вычислить довольно просто — например, используя интеграл Дюамеля.

12.9. Примеры электро- и радиотехнических расчетов

Сфера применения Mathcad в радиотехнических расчетах исключительно велика. Уже первые версии Mathcad позволяли выполнять массу таких расчетов, например, расчет индуктивностей и емкостей, построение графиков вольт-амперных характеристик нелинейных приборов, расчет статических и динамических характеристик простых цепей и так далее [2]. Повторять их в данной книге нет смысла — ныне пользователь Mathcad вполне может сделать это самостоятельно, причем в считанные минуты. Поэтому рассмотрим примеры, требующие определенных и порой немалых усилий по их реализации в среде Mathcad. Среди этих примеров есть оригинальные примеры, а есть взятые и доработанные из электронных книг и полученные из Интернета.

12.9.1. Моделирование поля двух электрических зарядов

Начнем с классического примера — построение линий электрического поля двух электрических зарядов. Это скорее пример на электрофизику, но от нее лишь шаг к радиотехнике. Данная задача достаточно проста, поскольку теоретическая электрофизика дает явное выражение для напряженности поля в плоскости (x, y) в виде функции $\phi(x, y)$. Эта формула приведена на рис. 12.48 в упрощенном и полном виде.

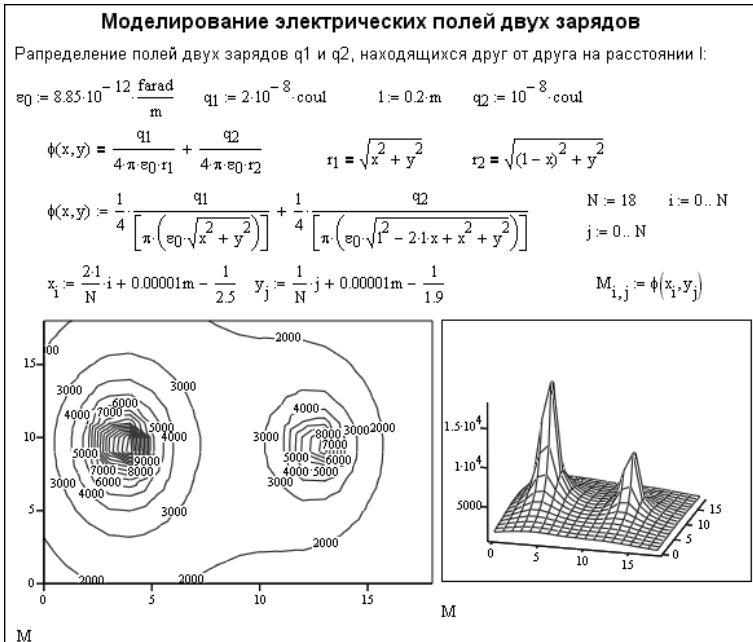


Рис. 12.48. Электрическое поле двух точечных зарядов

Остается построить по ней графики распределения напряженности электрического поля. Они и даны в двух наглядных видах — в виде контурного и в виде трехмерного графика.

12.9.2. Расчет АЧХ и ФЧХ импеданса параллельного LCR-контура

Трудно переоценить роль резонансных LCR-контуров. В современном радио-передатчике и радиоприемнике их множество и они составляют основу многих радиоэлектронных устройств — резонансных фильтров, избирательных усилителей и др. LCR-контуры бывают двух основных типов — параллельные и последовательные. Они различаются способом соединения реактивных элементов — индуктивности L и емкости C .

Рисунок 12.49 представляет документ, в котором выполняется расчет АЧХ и АЧХ параллельного LCR-контура по общеизвестным из радиотехнической литературы формулам. Ось круговых частот $\omega = 2\pi f$ при построении АЧХ и ФЧХ задана в логарифмическом масштабе.

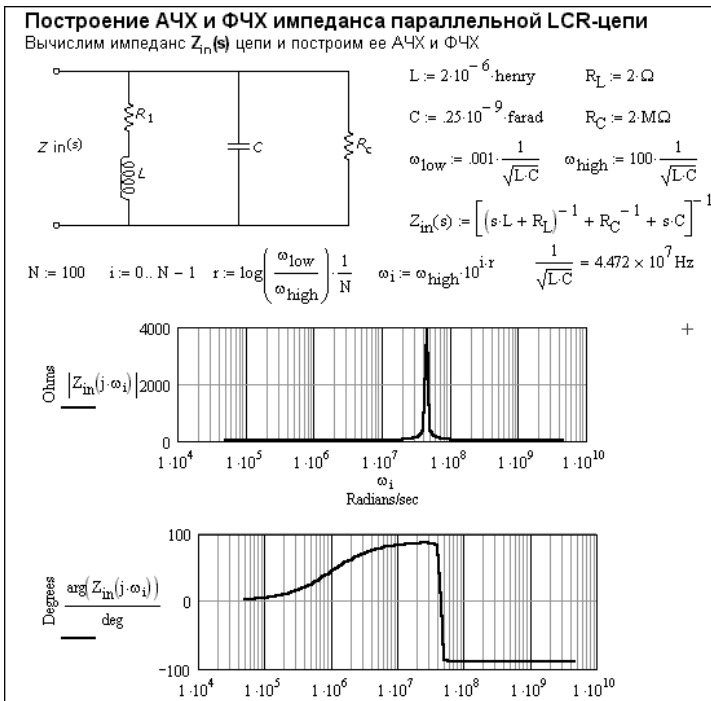


Рис. 12.49. Построение АЧХ и ФЧХ параллельного LCR-контура

При малых частотах ω импеданс цепи мал и определяется значением резистора $R_L = R$. При этом фазовый сдвиг ϕ близок к 0. При повышении частоты модуль полного сопротивления цепи возрастает из-за влияния индуктивности L . Однако емкость C , влияние которой поначалу незаметное, при высоких частотах понижает модуль импеданса. На некоторой частоте

$$\omega_0 = \frac{1}{\sqrt{L \cdot C}}$$

наступает резонанс и импеданс контура становится активным и большим. На АЧХ появляется пик, а фаза испытывает резкое изменение в районе частоты ω_0 , где она равна 0. После этого модуль импеданса падает, причем решающую роль здесь играет емкость C . Резонанс часто используется для создания избирательных цепей, например входных цепей супергетеродинных радиоприемников.

12.9.3. Построение АЧХ и ФЧХ импеданса последовательного LCR-контура

Рисунок 12.50 представляет документ, в котором строится АЧХ и ФЧХ последовательного LCR-контура. Здесь также используются известные выражения для импеданса такого контура, а частота задается в логарифмическом масштабе.

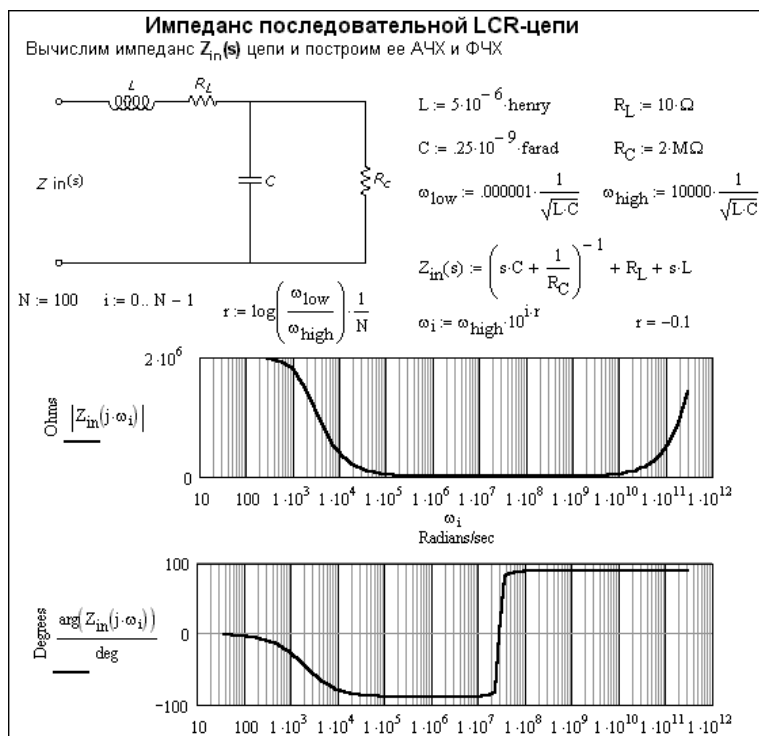


Рис. 12.50. Построение АЧХ и ФЧХ последовательного LCR-контура

Поведение этой цепи в широком диапазоне частот довольно необычно. При низких частотах, близких к 0, импеданс цепи имеет действительный характер и определяется значением R_C . Фазовый сдвиг при этом близок к 0. По мере повышения частоты импеданс падает преимущественно из-за влияния емкости C . По мере повышения частоты начинается повышение импеданса из-за влияния индуктивности L . В итоге в довольно широкой полосе частот модуль импеданса цепи мал.

Однако в районе резонансной частоты ω_0 фазовый сдвиг в узкой области частот быстро изменяется и меняет свой знак.

Рисунок 12.51 показывает детальное построение АЧХ и ФЧХ в суженном диапазоне частот и значений модуля импеданса. Теперь хорошо видна область резо-

нанса, в которой модуль на частоте ω_0 падает до значения сопротивления RL . Если к цепи подключен источник переменного напряжения, то на этой частоте будет пик тока.

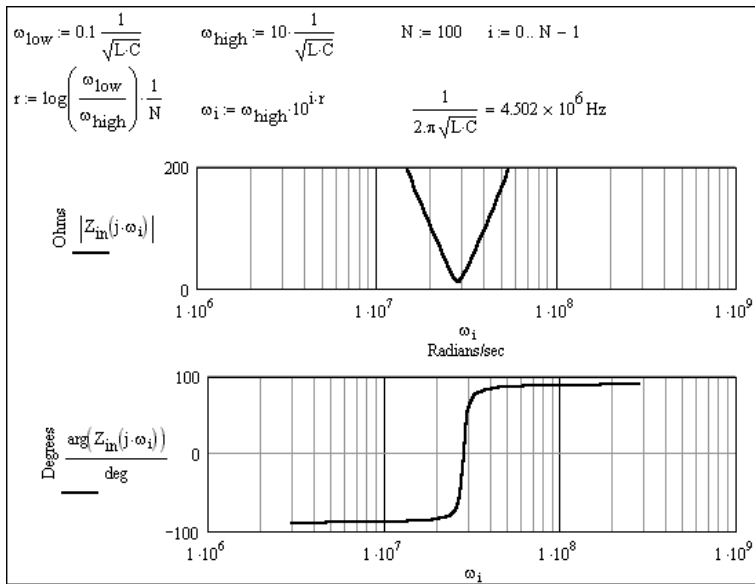


Рис. 12.51. Построение АЧХ и ФЧХ последовательного LCR-контура вблизи резонансной частоты

На применении резонансных контуров базируется построение полосовых фильтров в радиоприемниках, сложных многоконтурных фильтров сосредоточенной селекции, фильтров для подавления гармоник в радиопередающих устройствах и многих других радиотехнических устройств.

12.9.4. Реакция резонансного контура на синусоидальное воздействие

LCR-контур часто работает под внешним воздействием в виде синусоидальных сигналов. Причина этого в том, что даже один высокодобротный контур превращает импульсный сигнал в синусоидальный. В связи с этим разберем актуальную задачу — вычислим реакцию колебательного контура на синусоидальный сигнал, как отличающийся по частоте от резонансной частоты контура, так и с частотой, равной резонансной частоте контура.

Рисунок 12.52 показывает начало документа, в котором решается эта поучительная задача для случая, когда частота воздействия f далека от резонансной (она более чем вдвое ниже). Воздействием является перепад синусоидального сигнала — он начинается с момента $t = 0$, имеет амплитуду U_m и частоту f .

В данном случае избрано решение задачи путем моделирования поведения контура. Для этого подготовлено и решено линейное дифференциальное уравнение второго порядка для LCR-контура, находящегося под колебательным воздействием. Оно сведено к двух дифференциальным уравнениям первого порядка и решается с помощью функции *Vulstoer*. В решении $u(t)$ — напряжение на конденсаторе C (выходе) контура. Контур в данном случае ведет себя как несогласован-

«Звон» контура нередко используется в электронных устройствах. Например, в электронных осциллографах «звенящие» контуры применялись для создания временных меток на экране осциллографической трубки, что было нужно для выполнения измерений временных интервалов.

А теперь рассмотрим случай, когда частота меандра равна резонансной частоте контура. Этот случай моделирования представлен на рис. 12.55. Картина переходных процессов на этот раз резко изменилась — она очень похожа на случай реакции контура на синусоидальный сигнал, показанный на рис. 12.53.

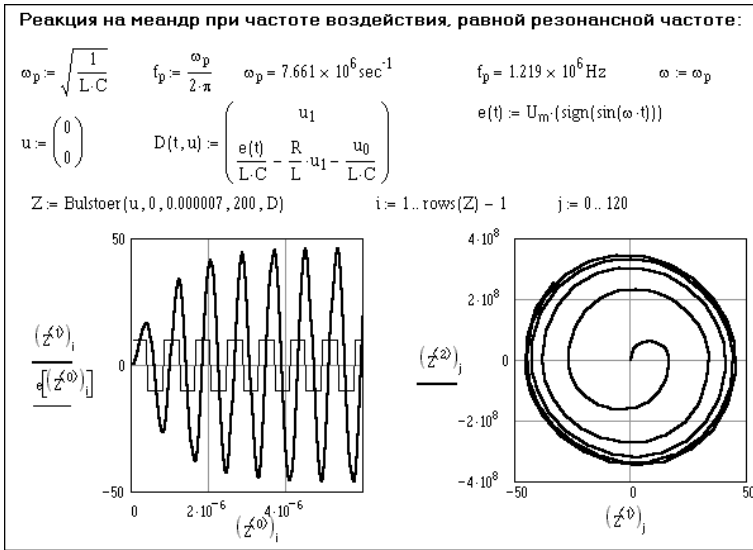


Рис. 12.55. Вычисление реакции LRC-контура на меандр с частотой повторения, равной резонансной частоте контура

Физически такая метафора вполне понятна — на резонансной частоте контур ведет себя как высокоизбирательная цепь, выделяющая из меандра его первую гармонику и реагирующую в основном на нее. Другие гармоники меандра контуром просто подавляются и слабо влияют на вид переходного процесса. При желании с помощью описанной программы читатель может выполнить моделирование поведения контура на любые другие виды воздействий.

12.9.6. Расчет микрополосковых линий

С разработкой печатных плат и гибридно-пленочных микросхем основным типом линий передачи и задержки и даже обычных проводников стали микрополосковые линии. Они создаются в виде плоских проводников, расположенных на поверхности диэлектрической пластины или между такими пластинами. Скорость перемещения сигналов в таких линиях меньше скорости света c и рассчитывается по простым формулам:

$$v/c = 1/\sqrt{\epsilon} \text{ для симметричной линии и}$$

$$v/c = 1/\sqrt{0.475 \cdot \epsilon + 0.67} \text{ для несимметричной линии.}$$

Однако расчет волнового сопротивления линий или ширины полоски по заданному волновому сопротивлению выполняется по довольно сложным формулам. Ниже мы рассмотрим несколько документов Mathcad для выполнения этих расчетов.

На рис. 12.56 представлен расчет ширины проводящей полоски несимметричной микрополосковой линии. Заданы толщина диэлектрика *Height* и диэлектрическая постоянная ϵ . Предполагается, что снизу линия крепится к металлическому основанию или диэлектрическая пластина имеет снизу слой металла.

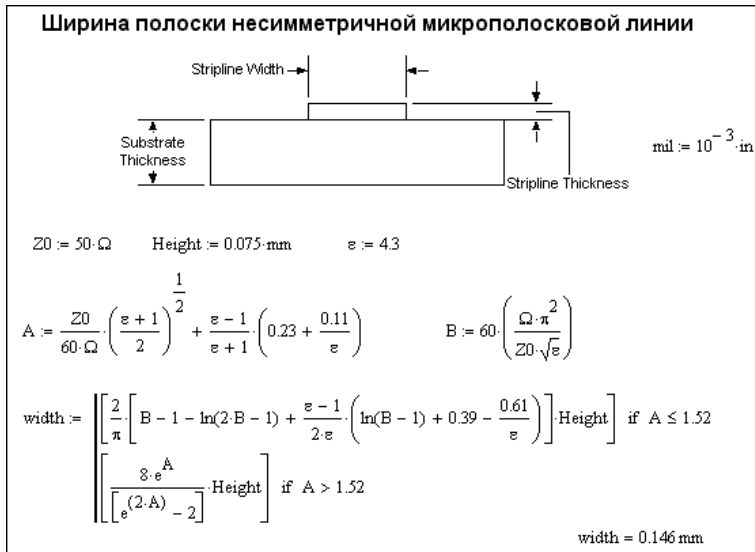


Рис. 12.56. Расчет ширины проводящей полоски микрополосковой линии

Расчет волнового сопротивления симметричной микрополосковой линии по заданной ширине полоски иллюстрирует рис. 12.57.

Для расчета ширины проводящей полоски симметричной микрополосковой линии можно использовать документ, представленный на рис. 12.58 (подразумевается наличие проводящих слоев снаружи диэлектрических пластин).

Иногда применяются несимметричные микрополосковые линии, у которых проводящая полоска заключена между диэлектрическими пластинами разной толщины. Рисунок 12.59 показывает расчет волнового сопротивления такой линии.

Наконец, рис. 12.60 показывает расчет ширины проводящей полоски несимметричной микрополосковой линии со смещенной центральной полоской.

Стоит отметить, что существуют и несколько иные расчетные формулы для микрополосковых линий передачи. Их и методики расчета можно найти в [2, 20].

12.9.7. Расчет основных параметров каскодного усилителя

Продолжая тему расчета линейных электронных устройств, рассмотрим типичный расчет параметров малосигнального каскодного усилителя упрощенным аналитическим методом. Малосигнальность усилителя означает, что мы отказываемся от учета нелинейности устройства, что возможно, поскольку такие усилители работают обычно при малых сигналах как на входе, так и на выходе. В результате



Рис. 12.57. Расчет волнового сопротивления симметричной микрополосковой линии



Рис. 12.58. Расчет ширины проводящей полоски микрополосковой линии



Рис. 12.59. Расчет волнового сопротивления несимметричной микрополосковой линии со смещенной центральной полоской

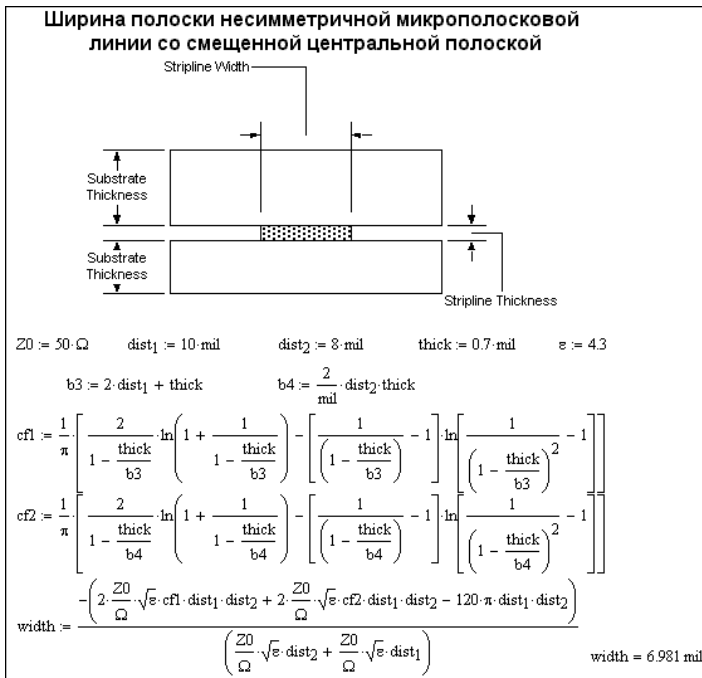


Рис. 12.60. Расчет ширины проводящей полоски несимметричной микрополосковой линии со смещенной центральной полоской

расчет усилителя можно выполнить по классической методике расчета линейных устройств на переменном токе.

Рассматриваемый здесь документ состоит из нескольких частей. Вначале представим схему усилителя — рис. 12.61 (она строится в обычном графическом редакторе и импортируется в документ Mathcad просто как рисунок). Усилитель можно рассматривать как каскад с общим эмиттером, работающий на каскад с общей базой. Эти каскады интегрированы как по постоянному току, так и переменному току.

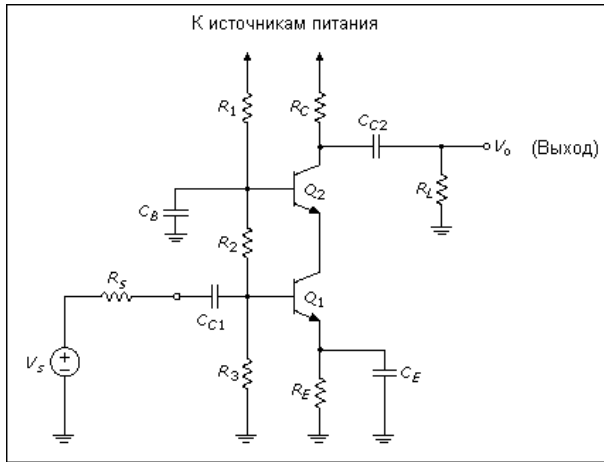


Рис. 12.61. Схема каскодного усилителя

Типовой расчет граничной частоты усилителей предполагает разбивку их эквивалентной схемы на отдельные ячейки и расчет граничной частоты каждой ячейки. Эквивалентная схема каскодного усилителя и ее преобразование в три ячейки с разными постоянными времени представлена на рис. 12.62.

Расчетная часть документа представлена на рис. 12.63. Здесь после вычисления ряда вспомогательных величин вычислен один из важнейших параметров — коэффициент усиления в области средних частот A_M . Расчет этого усилителя на низких частотах достаточно тривиален и особого интереса не представляет, поскольку такой усилитель используется обычно как высокочастотный. Поэтому в конце расчетной части вычислены граничные высшие частоты как звеньев усилителя (см. рис. 12.62, б), так и всего усилителя в целом — частота f_H .

Расчет показывает, что при используемых транзисторах (явно не очень высокочастотных) частота среза усилителя составляет около 7,5 МГц, что характерно для типового применения усилителя в качестве усилителя телевизионных сигналов. Модуль коэффициента усиления чуть выше 20.

12.9.8. Расчет на постоянном токе двухтранзисторной цепи

Еще одна двухтранзисторная цепь показана на рис. 12.64. В зависимости от внешнего обрамления эта цепь может выполнять функции каскодного усилителя, стабилизатора тока в нагрузке R_{C2} и иного устройства.

Один из методов расчета таких цепей базируется на ее линеаризации. При этом нелинейные эмиттерные переходы представляются в виде источника постоянного напряжения, включенного последовательно с резистором, величина которого зада-

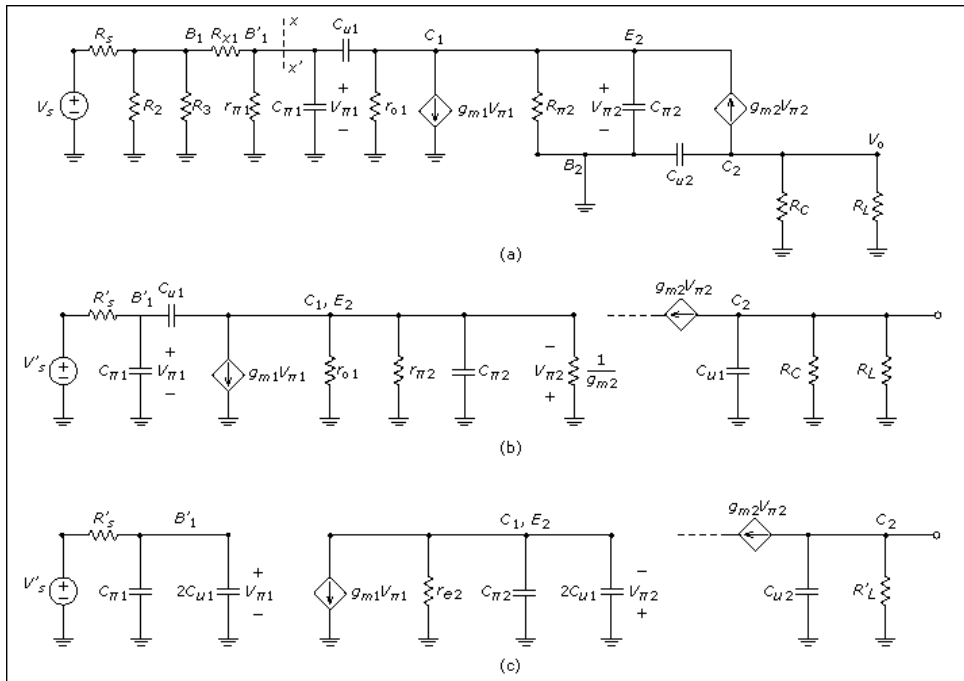


Рис. 12.62. Эквивалентная схема каскодного усилителя и ее преобразование в три ячейки

Зададим параметры схемы и транзисторов: $\beta = 150$

$R_S = 3.9 \text{ k}\Omega$	$R_E = 3.3 \text{ k}\Omega$	$C_{C1} = 1.1 \text{ }\mu\text{F}$	$V_{CC} = 15 \text{ volt}$	$R_1 = 18 \text{ k}\Omega$	$R_C = 5.8 \text{ k}\Omega$
$C_{C2} = 1.1 \text{ }\mu\text{F}$	$V_{BE1} = 7 \text{ volt}$	$R_2 = 3.9 \text{ k}\Omega$	$R_L = 3.9 \text{ k}\Omega$	$C_B = 9 \text{ }\mu\text{F}$	$V_{BE2} = 7 \text{ volt}$
$R_3 = 8.2 \text{ k}\Omega$	$C_E = 10 \text{ }\mu\text{F}$	$V_T = 25 \text{ mV}$	$C_\mu = 2.2 \text{ pF}$	$r_x = 47 \text{ }\Omega$	$C_\pi = 14 \text{ pF}$

Расчет вспомогательных величин

$V_{B1} = V_{CC} \frac{R_3}{R_1 + R_2 + R_3}$	$V_{B1} = 4.086 \text{ volt}$	$I_{E1} = \frac{V_{B1} - V_{BE1}}{R_E}$	$I_{E1} = 1.026 \text{ mA}$	$I_{E2} = I_{E1}$	$I_{E2} = 1.026 \text{ mA}$
$V_{B2} = V_{CC} \frac{R_2 + R_3}{R_1 + R_2 + R_3}$	$V_{B2} = 6.03 \text{ volt}$	$V_{C1} = V_{B2} - V_{BE2}$	$V_{C2} = V_{CC} - I_{E2} R_C$	$V_{C1} = 5.33 \text{ volt}$	$V_{C2} = 9.048 \text{ volt}$
$\xi_m = \frac{I_{E1}}{V_T}$	$\xi_m = 0.041 \text{ mho}$	$r_\pi = \frac{\beta}{\xi_m}$	$r_\pi = 3.654 \text{ k}\Omega$	$r_e = \left(\frac{1}{r_\pi} + \xi_m \right)^{-1}$	
$R'_S = \left[r_\pi^{-1} + \left[r_x + \left(R_3^{-1} + R_2^{-1} + R_S^{-1} \right)^{-1} \right]^{-1} \right]^{-1}$	$R'_S = 1.124 \times 10^3 \text{ }\Omega$	$R'_L = \frac{R_C R_L}{R_C + R_L}$	$R'_L = 2.332 \text{ k}\Omega$		
$A_M = -\xi_m R'_L \frac{r_\pi}{R_2 R_3 + R_S r_\pi + r_x + \left(\frac{1}{R_2} + \frac{1}{R_3} + \frac{1}{R_S} \right)^{-1}}$	$A_M = -26.777$				

Расчет граничных высших частот звеньев усилителя и граничной частоты всего усилителя:

$f_1 = \frac{1}{2\pi R'_S (C_\pi + 2C_u)}$	$f_1 = 7.699 \text{ MHz}$	$f_2 = \frac{1}{2\pi r_e (2C_\mu + C_\pi)}$	$f_2 = 357.412 \text{ MHz}$	$f_3 = \frac{1}{2\pi C_\mu R'_L}$	$f_3 = 31.022 \text{ MHz}$
$f_H = \frac{1}{\sqrt{\frac{1}{f_1^2} + \frac{1}{f_2^2} + \frac{1}{f_3^2}}}$	$f_H = 7.47 \text{ MHz}$				

Рис. 12.63. Расчет параметров каскодного усилителя

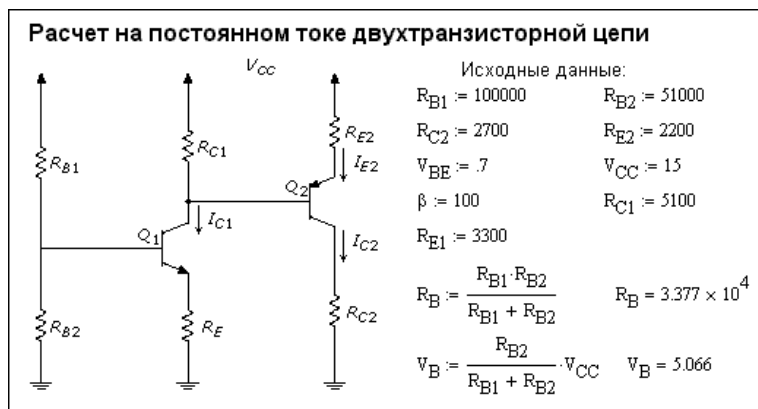


Рис. 12.64. Двухтранзисторная цепь

ет наклон (и дифференциальное сопротивление) перехода. Это позволяет методом Кирхгофа составить систему линейных уравнений для неизвестных токов и напряжений и затем решить ее с помощью функции Find, задав перед этим начальные значения искомых параметров. Реализация этого метода показана на рис. 12.65.

Параметры этой цепи на переменном токе можно найти методом, описанным выше для каскадного усилителя. Вообще говоря, схема рис. 12.64 по существу также является основой каскадного усилителя, с той разницей, что она выполнена на разнополярных транзисторах.

Расчет параметров из системы уравнений:

$V_{CE1} := 5$	$V_{BC2} := 4$	$I_{C2} := .001$	$I_{B2} := .00001$	$I_{E2} := .001$
$V_{EC2} := 5$	$I_{C1} := .001$	$I_{B1} := .00001$	$I_{E1} := .001$	

Given

$-V_B + I_{B1} \cdot R_B + V_{BE} + I_{E1} \cdot R_{E1} = 0$	$V_{BC2} + V_{BE} = V_{EC2}$
$-V_{CC} + I_{E2} \cdot R_{E2} + V_{EC2} + I_{C2} \cdot R_{C2} = 0$	$I_{E1} = I_{C1} + I_{B1}$
$V_{CE1} + I_{E1} \cdot R_{E1} = V_{BC2} + I_{C2} \cdot R_{C2}$	$I_{C1} = \beta \cdot I_{B1}$
$(I_{C1} - I_{B2}) \cdot R_{C1} = I_{E2} \cdot R_{E2} + V_{BE}$	$I_{E2} = I_{C2} + I_{B2}$
$I_{C2} = \beta \cdot I_{B2}$	

Find($V_{CE1}, V_{EC2}, V_{BC2}, I_{C1}, I_{E1}, I_{B1}, I_{C2}, I_{E2}, I_{B2}$) =

$\begin{pmatrix} 5.08965 \\ 3.379772 \\ 2.679772 \\ 1.189465 \times 10^{-3} \\ 1.201359 \times 10^{-3} \\ 1.189465 \times 10^{-5} \\ 2.360875 \times 10^{-3} \\ 2.384484 \times 10^{-3} \\ 2.360875 \times 10^{-5} \end{pmatrix}$

Рис. 12.65. Расчет двухтранзисторной цепи на постоянном токе

12.9.9. Расчет универсального активного фильтра

Большинство радиотехнических устройств, относящихся к линейным, давно хорошо изучены и для их расчета и проектирования имеются готовые формулы. Их так много, что даже многотомные справочники не вместят описания всех возможных вариантов таких устройств и относящихся к ним формул. Это еще один повод отказа от подготовки книг с примерами конкретных расчетов в радиотехнике. Читателю гораздо важнее освоить методику таких расчетов, что позволит применить их в своей практике и на нужных ему своих собственных примерах. Благо Mathcad, что уже отмечалось, позволяет легко выполнять вычисления даже по довольно сложным формулам.

Учитывая сказанное, рассмотрим расчет универсального активного фильтра на операционных усилителях (рис. 12.66). Таких фильтров описано невообразимое множество, так что рассмотренный фильтр лишь один из многих. Его выбор для примера просто связан с тем, что он достаточно универсален и позволяет реализовать сразу четыре основных вида фильтров — фильтр низких частот, высокие частот, полосовой и задерживающий (режекторный). Большое число операционных усилителей в таком фильтре ныне не помеха — отечественная и зарубежная промышленность уже давно выпускает счетверенные операционные усилители. А вот универсальность фильтра — важное его достоинство.

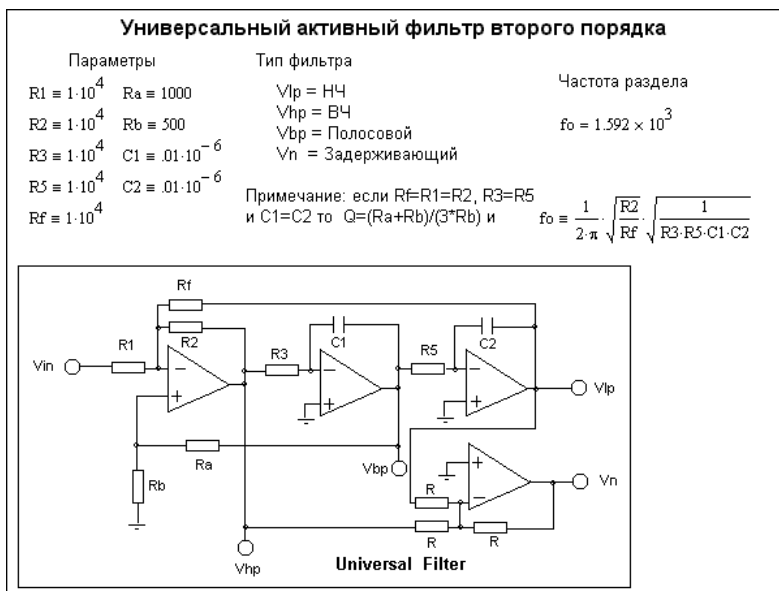


Рис. 12.66. Схема универсального активного фильтра и его исходные данные

Расчет амплитудно-частотных характеристик фильтра для возможных вариантов его применения представлен на рис. 12.67. Здесь даны соответствующие расчетные формулы и графики АЧХ.

Наконец, на рис. 12.68 представлены формулы для расчета фазочастотных характеристик фильтра и дано построение самих характеристик. Обратите внимание на то, что углы этих характеристик заданы в градусах, а не в радианах. Так принято при расчете технических электронных устройств.

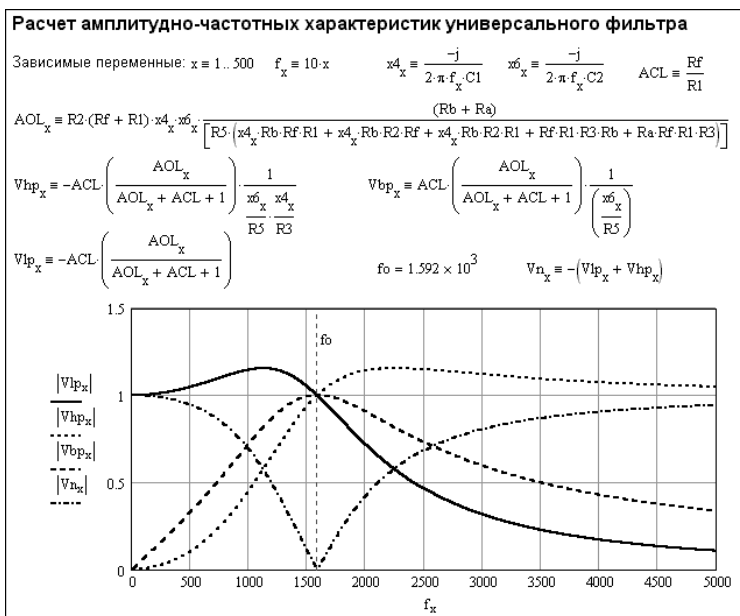


Рис. 12.67. Расчет амплитудно-частотных характеристик универсального фильтра

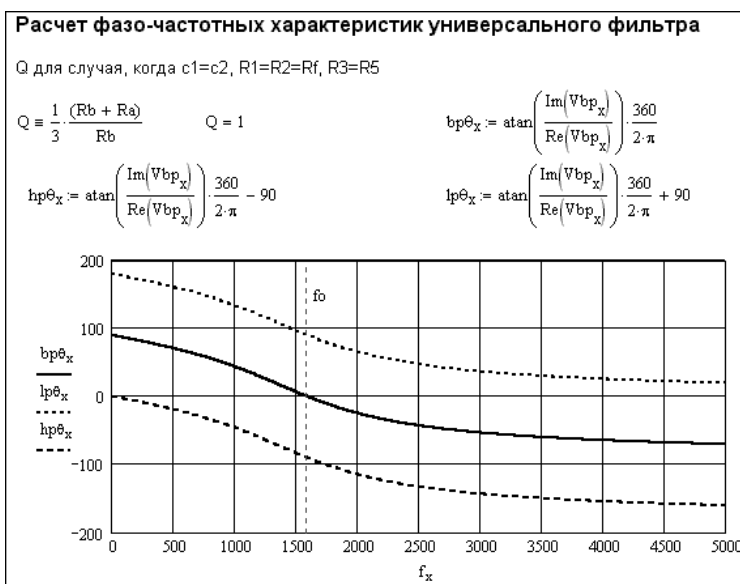


Рис. 12.68. Расчет фазочастотных характеристик универсального фильтра

12.9.10. Расчет нелинейной диодной цепи

К сожалению, даже простые диодные цепи относятся к нелинейным цепям, для расчета которых приходится применять численные методы, поскольку анали-

тических решений чаще всего не имеется. В качестве иллюстрации этого тезиса рассмотрим простейшую цепь, где диод подключен к источнику постоянного напряжения через резистор (рис. 12.69).

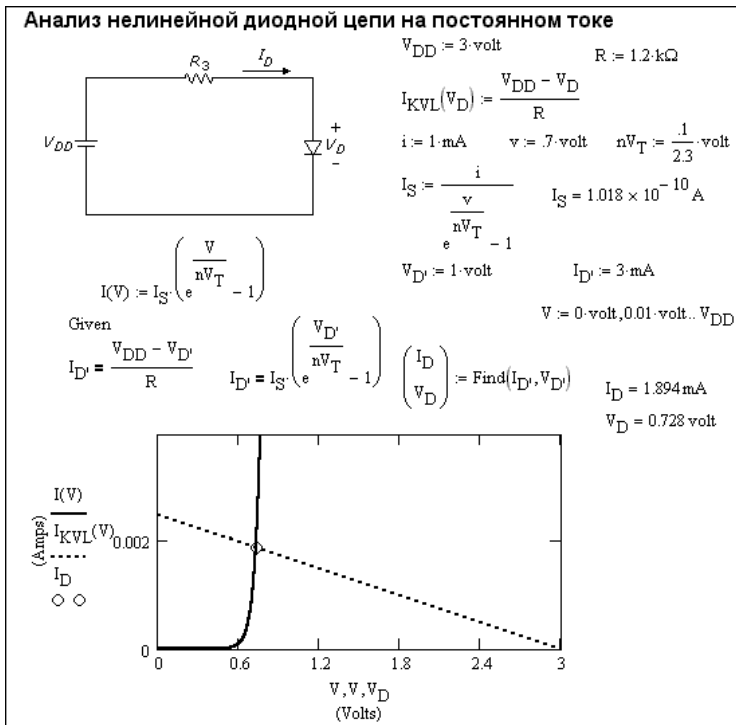


Рис. 12.69. Простейшая нелинейная диодная цепь

Диод обладает экспоненциальной зависимостью падения напряжения на нем от тока — $V(I)$. Обычно в справочниках в качестве основного параметра диода дано напряжение на нем при заданном токе. Это позволяет легко найти обратный ток (ток насыщения I_S), что показано на рис. 12.69. Далее на нем показано решение нелинейного уравнения и его графическая интерпретация. Они позволяют найти ток через диод и напряжение на нем. Графически эти параметры определяются из точки пересечения ВАХ диода с линией нагрузки резистора R .

12.10. Моделирование нелинейной цепи на туннельном диоде

12.10.1. Подготовка к моделированию цепи

Наш следующий пример описывает очень простое и интересное устройство — генератор на туннельном диоде, имеющем нелинейную N-образную вольт-амперную характеристику (ВАХ). Такая характеристика свойственна приборам, именуемым «негатронами».

Пусть туннельный диод подключен к источнику постоянного напряжения E через последовательно соединенные резистор R и индуктор L . Пусть далее R и E

выбраны так, что рабочая точка туннельного диода находится на падающем участке его ВАХ. Это обстоятельство принципиально важно, поскольку дифференциальная проводимость туннельного диода тогда является отрицательной. Физически это означает, что туннельный диод отдает энергию во внешнюю цепь и в ней возможно возникновение автоколебаний.

Если учесть емкость C туннельного диода (совместно с емкостью монтажа и нагрузочной емкостью), такая схема будет описываться следующей системой нелинейных дифференциальных уравнений:

$$\frac{di}{dt} = \frac{E - iR - u}{L}, \quad \frac{du}{dt} = \frac{i - I(u)}{C}.$$

Нелинейность этой системы обусловлена тем, что во втором уравнении ток туннельного диода $I(u)$ нелинейно зависит от напряжения на нем и емкости $C - u$. Зависимость $I(u)$ есть N -образная вольт-амперная характеристика туннельного диода.

Документ рис. 12.70 позволяет рассчитать переходные процессы при включении рассмотренной схемы, таблично заданной N -образной ВАХ туннельного диода. Она задается своими табличными значениями — элементами векторов U и I . N -образная ВАХ получается сплайн-интерполяцией с кубической экстраполяцией. Решение системы дифференциальных уравнений обеспечивается методом Рунге—Кутта с фиксированным шагом с помощью функции `rkfixed`.

МОДЕЛИРОВАНИЕ ЦЕПИ НА ТУННЕЛЬНОМ ДИОДЕ

$U := \begin{pmatrix} 0 \\ .2 \\ .4 \\ .6 \\ .8 \\ 1.0 \\ 1.2 \end{pmatrix}$	$I := \begin{pmatrix} 0 \\ .01 \\ .004 \\ .001 \\ .0009 \\ .003 \\ .01 \end{pmatrix}$	<p>Векторы U и I таблично задают N-образную вольтамперную характеристику туннельного диода. Она используется для сплайн-аппроксимации:</p> <p>$k := 0..6$ $IS := cspline(U, I)$ $V := -1, -0.5..1.2$ $J(V) := interp(IS, U, I, V)$</p>
<p>Моделирование работы цепи:</p>		
$D(t, Y) := \begin{bmatrix} \frac{E(t) - Y_0 \cdot R - Y_1}{L} \\ \frac{(Y_0 - interp(IS, U, I, Y_1))}{C} \end{bmatrix}$		
$Y := \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad Y := rkfixed(Y, 0, 30 \cdot 10^{-9}, 300, D)$		

Рис. 12.70. Моделирование генератора на туннельном диоде (начало документа)

12.10.2. Моделирование режима почти синусоидальных колебаний

Начнем со случая, когда рабочая точка туннельного диода установлена на падающем участке ВАХ. В этом случае дифференциальная проводимость туннельного диода отрицательна и в контуре RLC -цепи могут возникать колебания. При достаточно малой их амплитуде и при достаточно высокой добротности контура они могут быть незатухающими и почти гармоническими.

Результаты моделирования для этого случая (рис. 12.71) в этом документе представлены в двух формах. Первая имеет вид *фазового портрета* — положение каждой точки решения отмечается на графике в плоскости ВАХ. При этом для

каждой точки решения по одной оси двумерного графика откладывается изменяемый параметр, а по другой производные параметры (в нашем случае это напряжение на емкости C и ток, протекающий через нее. Для чисто гармонических колебаний предельный цикл колебаний на фазовой плоскости должен иметь форму эллипса (или при равных масштабах осей — окружности).

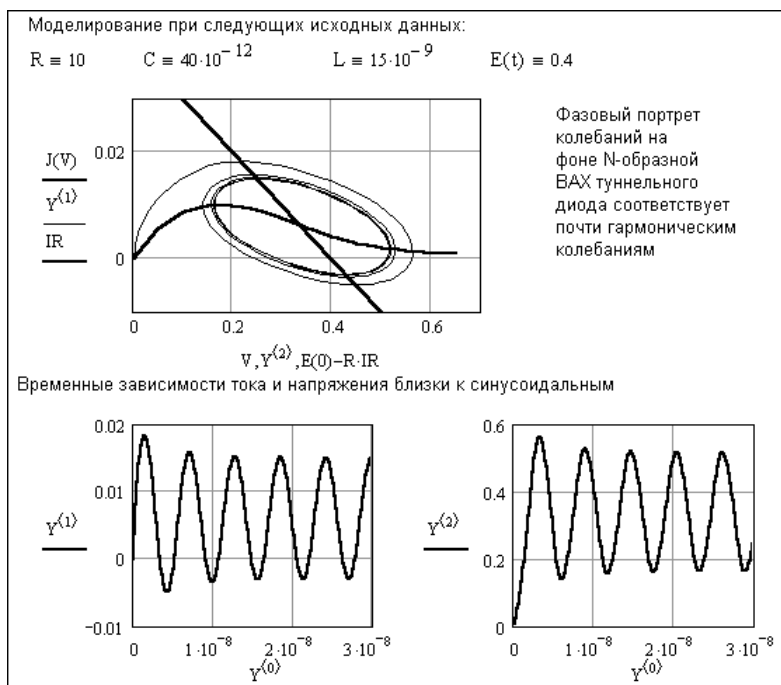


Рис.12.71. Моделирование генератора на туннельном диоде (случай генерации почти синусоидальных колебаний)

В фазовой плоскости строится и линия нагрузки, которая описывается уравнением $U(I) = E - I \cdot R$. ВАХ и линия нагрузки на рис. 12.71 построены жирными линиями, а фазовый портрет — тонкой линией. Линия нагрузки пересекает ВАХ в единственной точке на падающем участке ВАХ.

Отличие установившегося фазового портрета от эллипса небольшое, что, с одной стороны, говорит о малом влиянии нелинейности, а с другой — о почти гармоническом характере колебаний. Вторая форма — временные зависимости напряжения на туннельном диоде и тока в цепи конденсатора C . Эти зависимости близки к гармоническим ($C = 40$ пФ, $L = 15$ нГ) в конце заметной стадии установления колебаний, которая имеет характер затухающих колебаний (фазовый портрет при этом имеет вид закручивающейся спирали). Установившийся цикл колебаний называют *предельным циклом*.

Следует отметить, что область применения описанного документа неизмеримо шире, чем моделирование конкретной схемы генератора на туннельном диоде. Практически с его помощью можно моделировать основную цепь на любом приборе с N - или Λ -образной ВАХ (например, индуктивный релаксатор на лавинном транзисторе, генераторы на N - и Λ -диодах и транзисторах, их аналогах и так далее). Достаточно лишь задать данные их ВАХ и соответствующие исходные данные.

Стоит, к примеру, уменьшить емкость C в два-три раза, а индуктивность увеличить, как генератор перейдет в режим генерации релаксационных колебаний. Колебания теперь возникают более резко, фазовый портрет начинается сразу с предельного цикла, форма которого заметно отличается от эллиптической. Колебания при этом относятся к типу релаксационных и возникают сами по себе, без какого-либо внешнего воздействия. Практически после первого цикла колебаний устанавливается их стационарный режим. Вы можете самостоятельно проверить этот случай.

12.10.3. Моделирование нелинейного усилителя

Теперь рассмотрим более интересный случай, когда линия нагрузки расположена на восходящей части ВАХ с положительной проводимостью. При постоянном $E(t)$ такое положение рабочей точки устойчиво и не способно вызвать колебания в цепи. Но, предположим, что наряду с постоянным напряжением $E = 0.2$ В на цепь подается переменное напряжение с амплитудой 0.2 В и частотой 100 МГц. Это будет приводить к перемещению рабочей точки по ВАХ, и в какие-то моменты времени она начнет попадать на падающий участок ВАХ. В итоге могут возникнуть колебания, что и показано на рис. 12.72.

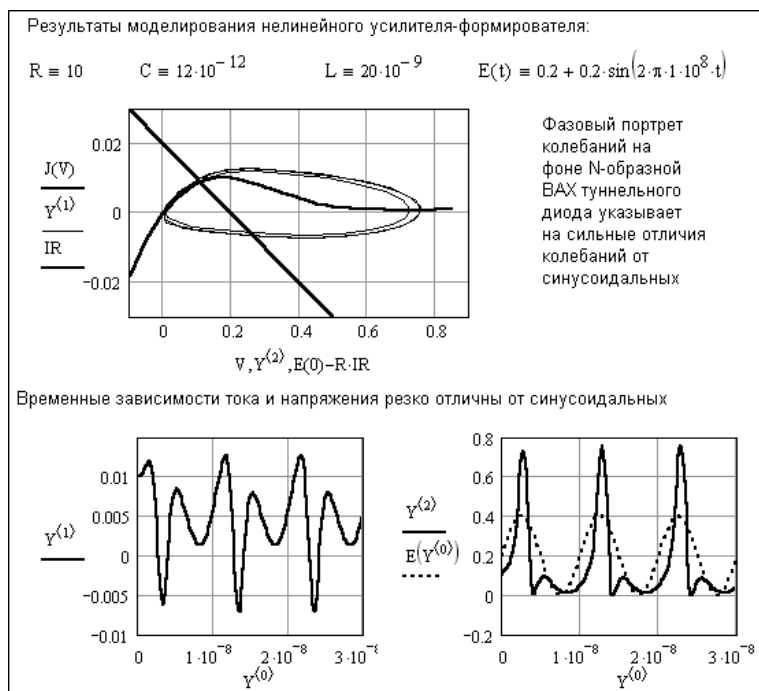


Рис. 12.72. Моделирование нелинейного усилителя на туннельном диоде

Этот случай можно рассматривать с двух позиций. Во-первых, он характеризует работу цепи как нелинейного усилителя. В установившемся режиме фазовый портрет колебаний заметно отличается от эллипса, так что форма колебаний резко отлична от синусоидальной. Во-вторых, можно рассматривать цепь как ждущий релаксационный генератор, который запускается верхушками синусоидального входного сигнала.

12.10.4. Моделирование ждущего релаксационного генератора

Рассмотрим генерацию импульсов цепью с туннельным диодом в ждущем режиме (рис. 12.73). Здесь поначалу линия нагрузки располагается на восходящем участке ВАХ с положительной дифференциальной проводимостью, так что поначалу цепь стабильна. Затем напряжение $E(t)$ увеличивается наложением на постоянную составляющую импульса. Линия нагрузки при этом перемещается с исходного положения (сплошная прямая) в положение (пунктирная прямая), при котором рабочая точка оказывается на падающем участке и теряет устойчивость.

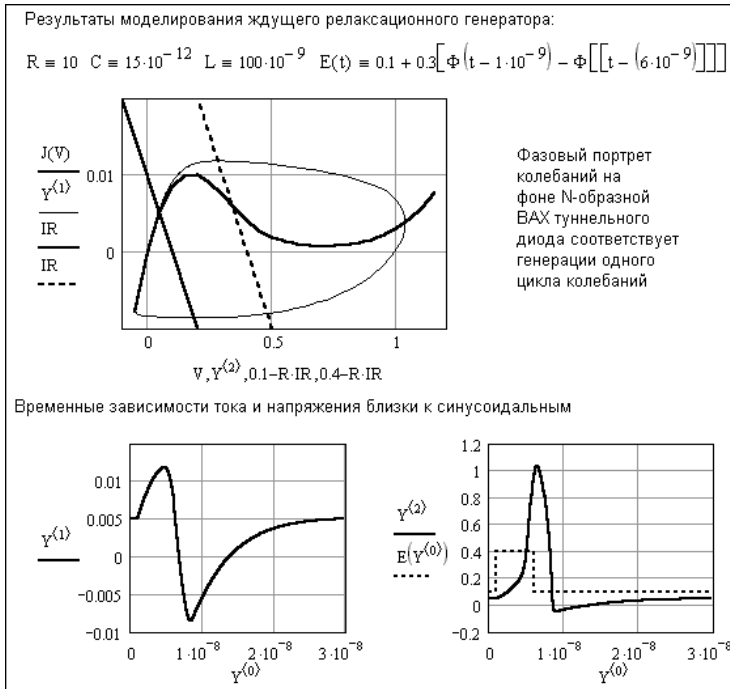


Рис. 12.73. Моделирование ждущего релаксационного генератора (одновибратора)

Как нетрудно заметить, в данном случае в ответ на входной (запускающий) импульс цепь генерирует один цикл релаксационных колебаний, после чего возвращается в исходное стабильное состояние. Разумеется, по завершении переходных процессов ее можно запускать снова и снова.

12.10.5. Моделирование триггера с раздельным запуском

В свое время большие надежды возлагались на триггеры, построенные на туннельных диодах. Потрясала их простота — резистор и туннельный диод (элементы L и C можно считать паразитными). Хотя эти надежды не оправдались (обычный триггер оказался более стабильным в работе, да и проще в интегральном исполнении), интересно проанализировать работу нашей цепи в триггерном режиме. Напомним, что триггер это устройство с двумя стабильными состояниями равновесия, в которые оно переходит под воздействием запускающих входных импульсов.

Для работы цепи в режиме триггера надо выбрать напряжение питания E и величину резистора R так, чтобы линия нагрузки пересекала ВАХ в трех точ-

ках (рис. 12.74). Две точки на восходящих участках устойчивы, а средняя на падающем участке неустойчива. Все, что надо для переключения триггера, — добавить положительный импульс к постоянной составляющей $E(t)$ для переброса триггера в состояние с высоким напряжением на диоде или добавить отрицательный импульс для переброса в состояние с низким напряжением на диоде. Это и демонстрирует рис. 12.74.

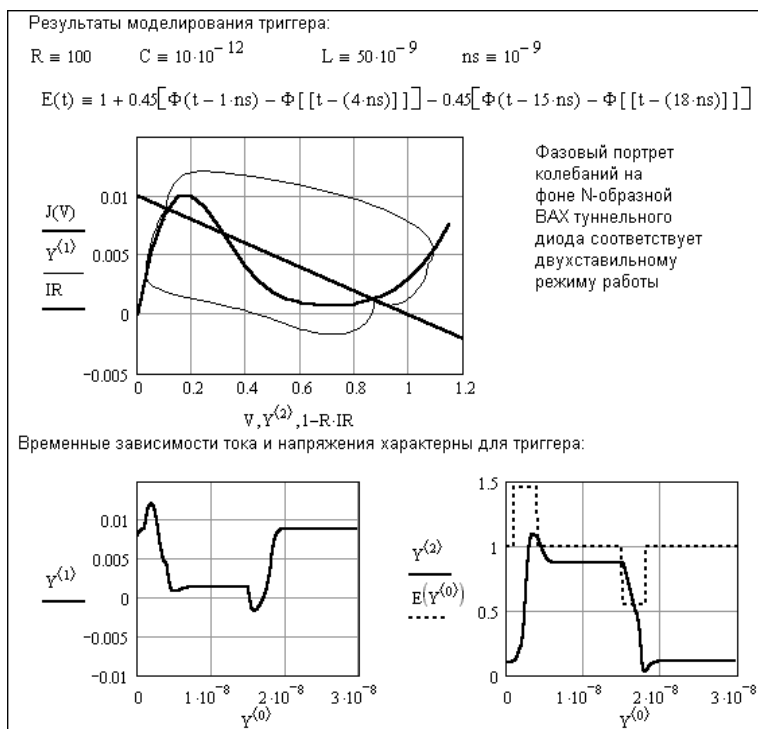


Рис. 12.74. Моделирование триггера на туннельном диоде

Фазовый портрет и временные зависимости напряжения и тока в данном случае свидетельствуют о сложности и специфичности переходных процессов и, в то же время, о высокой степени их визуализации. Именно это и характерно для моделирования цепей на специальных приборах, таких как туннельные диоды.

12.11. О моделировании электронных устройств

Приведенные выше примеры наглядно показывают, что даже незначительное изменение параметров одной и той же цепи на туннельном диоде способно резко (и даже качественно) изменить режимы ее работы. При этом цепь может выполнять практически любую функцию электронного устройства: нелинейного преобразователя, усилителя и генератора гармонических колебаний, автоколебательного и ждущего релаксатора и двухстабильного устройства — триггера.

Насколько разумно применять Mathcad, да и другие системы компьютерной математики, для моделирования электронных устройств? Ведь есть системы схе-

мотехнического моделирования (Pspice, Design Center, MicroCap, WorkBench и др.), которые имеют обширные библиотеки компонентов, обеспечивают визуально-ориентированное программирование задач моделирования, позволяют создавать сложные схемы, автоматически составляют и решают системы уравнений состояния и позволяют использовать множество виртуальных регистрирующих устройств, таких как осциллографы, анализаторы спектра, цифровые измерители, графопостроители и так далее.

Ответ на этот вопрос довольно прост — системы компьютерной математики надо применять тогда, когда пользователя интересуют физические и математические детали моделирования, а также когда в схемах используются специфические приборы, основанные на необычных физических явлениях. Туннельный диод лишь один из многих таких приборов. Можно упомянуть и массу других приборов — Λ -диоды и транзисторы, лавинные диоды и транзисторы, лавинно-пролетные диоды, диоды Ганна и так далее. В других же случаях, безусловно, лучше использовать специальные программы схемотехнического моделирования.

Справочный раздел

Приложение 1. Сравнительные данные версий Mathcad 8 — Mathcad 12

Ниже представлены сравнительные данные различных реализаций систем Mathcad 8—12 в том виде, как они представлены на Интернет-сайте разработчика этих систем, т. е. в форме англоязычных таблиц.

Табл. П1. Операционные системы для различных версий систем Mathcad

MICROSOFT STANDARDS AND OS SUPPORT	8	2000	2001	2001i	11	12
Microsoft® Windows 95	*	*	*			
Microsoft® Windows NT 4 (Service Pack 6 or higher)	*	*	*	*	*	
Microsoft® Windows 98 (service Pack 2)	*	*	*	*	*	
Microsoft® Windows 2000 support		*	*	*	*	*
Microsoft® Windows XP support				*	*	*
Microsoft® Windows ME support			*	*	*	
Microsoft® Office 2000 compatibility			*	*	*	*
Microsoft® Windows XP/ Office XP support				*	*	*
ActiveX support		*	*	*	*	*

Табл. П2. Поддержка данных и файлов различных программных систем

DATA EXCHANGE FEATURES	8	2000	2001	2001i	11	12
Data filters for .mat files, Excel files, Lotus 1-2-3, ASCII & others	*	*	*	*	*	*
Mixed format import, cut and paste (strings, complex, engineering notation, real numbers, Excel) to matrices					*	*
Real-time data acquisition from National Instruments® and Measurement Computing analog boards			*	*	*	*
Microsoft® Access, FoxPro and SQL-supported databases			*	*	*	*
WAV file read, write and information functions			*	*	*	*
File I/O component for multiple file formats, importing data as strings, and selecting specified columns or rows					*	*
Support for UserEFI strings					*	*
Improved APPENDPRN function					*	*
Binary file read and write					*	*
Enhanced Excel data exchange and integration					*	*
Data Wizard component with preview, text, binary, and Excel import settings						*
READFILE function for Excel, delimited, and fixed-width data						*

Табл. ПЗ. Данные о ресурсах, включенных в различные версии систем Mathcad

RESOURCES	8	2000	2001	2001i	11	12
References tables, key formulas and constants	*	*	*	*	*	*
The Collaboratory TM , the Mathcad Web-based forum	*	*	*	*	*	*
Technical support	*	*	*	*	*	*
Online tutorials and discipline-specific examples	*	*	*	*	*	*
Easy-to-use online Help with Search and Index		*	*	*	*	*
Adaptable QuickSheets demonstrating standard analyses and tasks		*	*	*	*	*
Online Developer's Reference			*	*	*	*
Mathcad Access Account to www.mathcad.com			*	*	*	*
FLEXlm license management (Enterprise only)					*	*
Completely rewritten, reorganized and reformatted Help, with more links to Quicksheets						*

Табл. П4. Данные об интеграции Mathcad с другими программами

CONNECTIVITY WITH OTHER APPS	8	2000	2001	2001i	11	12
Support for Mathsoft Designate Server						*
Support for MS SharePoint Server					*	*
Mathsoft® Axum 6	*	*	*	*	*	
MathWorks MATLAB® 4	*	*	*	*	*	*
MathWorks MATLAB® 5			*	*	*	*
Autodesk® AutoCAD 2000		*	*	*	*	*
Autodesk® AutoCAD 2000i/2002				*	*	*
Microsoft® Excel 95/98		*	*	*	*	*
Microsoft® Excel			*	*	*	*
Microsoft® Excel XP					*	*
Mathsoft® Axum 7				*	*	
Intergraph® SmartSketch 4				*	*	*

Табл. П5. Данные о пакетах расширения для систем Mathcad

ADDITIONAL TOOLS (FOR PURCHASE)	8	2000	2001	2001i	11	12
EE, ME or Civil Engineering Libraries	*	*	*	*	*	*
Wavelets Extension Pack	*	*	*	*	*	*
Signal Processing Extension Pack	*	*	*	*	*	*
Image Processing Extension Pack	*	*	*	*	*	*
Volume Licensing (Enterprise Only)	*	*	*	*	*	*
Premium Technical Support Plan (Enterprise Only)	*	*	*	*	*	*
Software Assurance (Enterprise Only)					*	*
Extension Pack for Data Analysis					*	*
Essential Mathcad On-site Training Courses (Enterprise Only)					*	*
Web-based Training Courses		*	*	*	*	*
VisSim Plus					*	*
SmartSketch 4 (full version)					*	*

Табл. П6. Данные о форматах и их представлении на экране дисплея

MATH FORMATS & DISPLAY	8	2000	2001	2001i	11	12
SI, MKS, U.S, and CGS unit-balancing with user-defined units	*	*	*	*	*	*
Extended built-in unit set and user-defined default units						*
Real, imaginary & complex number support	*	*	*	*	*	*
Decimal, binary, octal and hexadecimal formats	*	*	*	*	*	*
Math operator display options for equals, partial derivatives, multiplication, assignments, and more		*	*	*	*	*
Engineering and Scientific Notation Display		*	*	*	*	*

Табл. П7. Встроенные операторы

BUILT-IN OPERATORS	8	2000	2001	2001i	11	12
Over 35 Arithmetic, Vector and Matrix Operators	*	*	*	*	*	*
1st and nth Derivative, definite and indefinite integration, and limits	*	*	*	*	*	*
9 Evaluation Operators	*	*	*	*	*	*
10 Boolean Operators		*	*	*	*	*
User-Defined Operators		*	*	*	*	*
Programming operators (loops, assignments, etc.)	*	*	*	*	*	*
Namespace operator						*

Табл. П8. Встроенные функции

BUILT-IN FUNCTIONS	8	2000	2001	2001i	11	12
80+ core mathematical functions and 10 discrete transform functions	*	*	*	*	*	*
110+ statistics, probability and data analysis functions	*	*	*	*	*	*
18 Differential equation and partial differential equation solvers	*	*	*	*	*	*
Root Finding Functions	*	*	*	*	*	*
47 matrix creation, lookup, and characteristics functions	*	*	*	*	*	*
28 File Access Functions	*	*	*	*	*	*
14 Expression-Type and String Functions	*	*	*	*	*	*
18 Finance Functions		*	*	*	*	*
Complex arguments for Bessel/Hankel and truncation functions					*	*
Sinc function					*	*
Unlimited number arguments through statistical functions		*	*	*	*	*
Enhanced data fitting functions					*	*
1D and 2D Correlation						*
Logarithmically-spaced point generators						*
until function for iteration	*					*

Табл. П9. Решатели уравнений и систем уравнений

SOLVING CAPABILITIES	8	2000	2001	2001i	11	12
Define and evaluate variables and functions numerically or symbolically	*	*	*	*	*	*
Linear and Nonlinear System Solving and Optimization solve blocks	*	*	*	*	*	*
Solve systems with hundreds of variables and constraints, in matrix or individual-equation notation		*	*	*	*	*
Fast ODE solver for stiff systems and differential algebraic systems (Radau)				*	*	*
Systems of ODEs in solve blocks			*	*	*	*
1D PDE Solvers in Solve blocks					*	*
Live symbolic (algebraic) evaluation, including expansion, factorization, transforms	*	*	*	*	*	*
Multiple root finding algorithms, including symbolic methods		*	*	*	*	*
Local functions in programs						*

Табл. П10. Данные о языках программирования, поддерживаемых в Mathcad

APPLICATION CUSTOMIZATION AND EXTENSIBILITY	8	2000	2001	2001i	11	12
User-defined functions created in C, C++, or FORTRAN	*	*	*	*	*	*
Embed, link, and automate any OLE-compliant application or ActiveX control in Mathcad using VBScript or Jscript™	*	*	*	*	*	*
Use OLE Automation & Visual Basic(R) to develop complete solutions incorporating Mathcad computations	*	*	*	*	*	*
Software Development Kit (SDK) and wizard for building custom components using C++			*	*	*	*
Mathsoft Custom Controls component			*	*	*	*
Scriptable object component, for reuse and deployment of components and their settings			*	*	*	*
Automation interface for access to application, worksheets, windows, and regions				*	*	*
Mathcad Automation access from scriptable objects					*	*

Табл. П11. Возможности графической визуализации

GRAPHING AND VISUALIZATION	8	2000	2001	2001i	11	12
Bar charts, x-y, polar, vector, contour, scatter, and surface plots	*	*	*	*	*	*
Animation capabilities	*	*	*	*	*	*
Trace & Zoom	*	*	*	*	*	*
2D QuickPlot™ and plot annotation capabilities	*	*	*	*	*	*
Open GL™ 3-D graph rendering and properties formatting	*	*	*	*	*	*
3D QuickPlot™		*	*	*	*	*
CreateMesh and CreateSpace functions		*	*	*	*	*
Mapping functions to convert data for one plot type to another			*	*	*	*
Image Viewer with support for BMP, GIF, JPG, PCX, TARGA, PGM, TIFF			*	*	*	*
Image manipulation functionality (zoom/pan/crop, brightness/contrast, rotate/flip/transpose, etc.)			*	*	*	*
3D axis labels				*	*	*
2nd Y-axis for 2D plots						*
Positionable legends for 2D plots						*
Grid and marker color selectors						*

Табл. П12. Возможности редактирования документов

DOCUMENT/TEXT EDITING FEATURES	8	2000	2001	2001i	11	12
Customizable Spell Checker with Technical Terms	*	*	*	*	*	*
Document Templates and Style Sheets	*	*	*	*	*	*
Hyperlinking	*	*	*	*	*	*
Advanced Find and Replace in text and math	*	*	*	*	*	*
Region Layering, Bordering and Alignment Controls		*	*	*	*	*
Document ruler with indenting support		*	*	*	*	*
Text and equation Highlighting, Bordering and Color Selection		*	*	*	*	*
Header and Footer formatting		*	*	*	*	*
Hidden, Collapsed, and Locked Areas		*	*	*	*	*
Live Math within Text Regions		*	*	*	*	*
Indexing and and Hyperlink checking for E-books			*	*	*	*
Document protection				*	*	*
File encryption				*	*	*
Region nudging and alignment guides					*	*
Metadata, automatic audit-trail, and annotations						*
Autofilled Delete lines						*

Табл. П13. Форматы файлов и поддержка WEB

FILE-FORMATS, PUBLISHING & WEB SUPPORT	8	2000	2001	2001i	11	12
Save as XML						*
Compressed images and files						*
Calculated results saved in XML						*
Save as HTML	*	*	*	*	*	*
Read HTML written by Mathcad (round-trip HTML)			*	*	*	*
Save files in older file formats			*	*	*	*
Data in HTML/MathML files				*	*	*
XHTML and XSL:FO transforms from Mathcad to Web and PDF formats						*
HTTP File Open support					*	*
Save to RTF retaining region positioning for MS Word					*	*

Табл. П14. Средства автоматизации вычислений и контроля

USABILITY FEATURES	8	2000	2001	2001i	11	12
Automatic unit tracking and conversion	*	*	*	*	*	*
Automatic recalculation	*	*	*	*	*	*
Customizable toolbars and interface	*	*	*	*	*	*
Easy-to-use equation editor	*	*	*	*	*	*
Error Tracing					*	*
Function dialog box with functions arranged in categories		*	*	*	*	*
Algorithm AutoSelect for integration, optimization, and ODE solve blocks		*	*	*	*	*
Print Current Page				*	*	*
User configurable script security warnings				*	*	*
Multi-region property settings				*	*	*
Multi-region Undo					*	*
Redefinition warnings					*	*
New menu choices and Toolbars for Data I/O, Controls, Resources, and Tools					*	*
Mathcad User's Guide on Start menu					*	*
Web controls (scriptless, save state)						*

Приложение 2. Ввод встроенных операторов

В этом приложении описана процедура ввода с клавиатуры ряда встроенных операторов. Приводятся только те операторы, для ввода которых требуется нажатие клавиш с отличными от обозначений операторов символами. Напоминаем, что альтернативным вариантом ввода является применение палитр математических символов. Ввиду визуальной очевидности такого ввода мы не приводим здесь описание всех имеющихся операторов.

В приведенном ниже списке операторов, вводимых с клавиатуры, используются следующие обозначения:

- A и B — массивы векторов или матриц;
- u и v — векторы с действительными или комплексными элементами;
- M — квадратная матрица;
- z и w — действительные или комплексные числа;
- x и y — действительные числа;
- m и n — целые числа;
- i — диапазон переменных;
- t — любое имя переменной;
- f — функция;
- X и Y — переменные или выражения любого типа.

Оператор	Обозначение	Клавиши	Описание
Круглые скобки	(X)	'	Изменение приоритета операций
Нижний индекс	A_n	[Задание индексированной переменной
Верхний индекс	$A_{<n>}$	Ctrl+6	Выбор n -го столбца из массива A
Векторизация	\vec{X}	Ctrl+- (дефис)	Векторизация – выполнение заданной операции для всех элементов вектора или матрицы X
Факториал	$n!$!	Факториал целого неотрицательного числа n
Сопряженное комплексное число	\bar{X}	"	Сопряженное комплексное число
Транспонирование	A^T	Ctrl+1	Транспонирование матрицы A
Возведение в степень	z^w	^	Возведение числа z в степень w
Возведение в степень	M^n	^	Возведение в n -ю степень квадратной матрицы M (при $n=-1$ получение обратной матрицы)
Отрицание	$-X$	-	Умножение X на -1
Сумма вектора	Σv	Ctrl+4	Сумма элементов вектора v (возвращается скалярное значение)

Оператор	Обозначение	Клавиши	Описание
Квадратный корень	\sqrt{z}	\	Квадратный корень
Корень n-й степени	$\sqrt[n]{z}$	Ctrl+\	Корень n-й степени из числа z
Модуль комплексного числа	$ z $		$ \sqrt{\operatorname{Re}(z)^2 + \operatorname{Im}(z)^2} $
Размер вектора	$ v $		$\sqrt{v \cdot v}$, если все элементы в векторе v являются реальными, $\sqrt{v \cdot \bar{v}}$, если элементы в векторе v являются комплексными
Детерминант матрицы	$ M $		Определитель квадратной матрицы M
Деление	X/z	/	Деление выражения X на скаляр z, не равный нулю (если X является массивом, то на z делится каждый элемент массива)
Умножение	$X \cdot Y$	*	Произведение X и Y, если X и Y являются скалярами. Умножение каждого элемента Y на X, если Y является массивом, а X – скаляром. Скалярное произведение, если X и Y – векторы одинакового размера. Умножение матриц, если X и Y являются матрицами совместимых размеров
Векторное произведение	$u \cdot v$	Ctrl+8	Векторное произведение векторов u и v
Суммирование для конечного ряда	$\sum_{i=m}^n X$	Ctrl+Shift+4	Сумма членов X для $i=m, m+1, \dots, n$, причем X может быть любым выражением
Произведение для конечного ряда	$\prod_{i=m}^n X$	Ctrl+Shift+3	Произведение членов X для $i=m, m+1, \dots, n$, где X может быть любым выражением
Суммирование для бесконечного ряда	$\sum_i X$	\$	Сумма членов X бесконечного ряда
Произведение для бесконечного ряда	$\prod_i X$	#	Произведение членов X бесконечного ряда
Предел функции в заданной точке	$\lim_{x \rightarrow a} f(x)$	Ctrl+L	Предел функции $f(x)$ при x, стремящемся к a (выполняется только в режиме символьных вычислений)

Оператор	Обозначение	Клавиши	Описание
Предел функции слева от заданной точки	$\lim_{x \rightarrow a^-} f(x)$	Ctrl+B	Предел функции $f(x)$ при x , приближающемся к a слева (выполняется только в режиме символьных вычислений)
Предел функции справа от заданной точки	$\lim_{x \rightarrow a^+} f(x)$	Ctrl+A	Предел функции $f(x)$ при x , приближающемся к a справа (выполняется только в режиме символьных вычислений)
Определенный интеграл	$\int_a^b f(t)dt$	&	Определенный интеграл от подынтегральной функции $f(t)$ с нижним (a) и верхним (b) пределами интегрирования
Неопределенный интеграл	$\int f(t)dt$	Ctrl+I	Вычисление в символьном виде неопределенного интеграла от подынтегральной функции $f(t)$
Производная заданной функции по переменной t	$\frac{d}{dt}f(t)$?	Первая производная функции $f(t)$ по переменной t
n -я производная заданной функции по переменной t	$\frac{d^n}{dt^n}f(t)$	Ctrl+?	n -я производная функции $f(t)$ по переменной t
Сложение	$X+Y$	+	Скалярное, векторное или матричное сложение X и Y
Вычитание	$X-Y$	-	Скалярное, векторное или матричное вычитание Y из X
Перевод строки	X $+Y$	Ctrl+Enter	Перенос части выражения на следующую строку
Больше, чем	$x>y$	>	1, если $x>y$, иначе 0
Меньше, чем	$x<y$	<	1, если $x<y$, иначе 0
Больше или равно, чем	$x\geq y$	Ctrl+0	1, если $x\geq y$, иначе 0
Меньше или равно, чем	$x\leq y$	Ctrl+9	1, если $x\leq y$, иначе 0
Не равно	$z\neq w$	Ctrl+3	1, если $z\neq w$, иначе 0
Равно	$z=w$	Ctrl+=	1, если $z=w$, иначе 0
Логическое отрицание	$b \neg$	Ctrl+Shift+1	Инверсное значение булевого операнда b (только для Mathcad 2001 Professional и выше)

Оператор	Обозначение	Клавиши	Описание
Логическое умножение	$b1 \wedge b2$	Ctrl+Shift+7	Логическая 1, если $b1$ и $b2$ имеют значение логической единицы, иначе логический 0 (только для Mathcad 2001 Professional и выше)
Логическое сложение	$b1 \vee b2$	Ctrl+Shift+6	Логическая 1, если $b1$ или $b2$ имеют значение логической единицы, иначе логический 0 (только для Mathcad 2001 Professional и выше)
Логическое исключающее ИЛИ	$b1 \otimes b2$	Ctrl+Shift+5	Логический 0, если $b1=b2$ ($0=0$ или $1=1$), иначе логическая 1 (только для Mathcad 2001 Professional и выше)

Приложение 3. Системные переменные Mathcad

Переменная	Описание
$\pi = 3,14159\dots$	Число π . В численных расчетах Mathcad использует значение π с учетом 15 значащих цифр. В символьных вычислениях π выводится как символ. Для ввода π используется комбинация клавиш Shift+Ctrl+p)
$e = 2,71828\dots$	Основание натуральных логарифмов. В численных расчетах Mathcad 2001 использует значение e с учетом 15 значащих цифр. В символьных вычислениях e выводится как символ
∞	Бесконечность. В численных расчетах это предельно большое число (10^{307}). В символьных вычислениях выводится как знак бесконечности. Для ввода ∞ используется сочетание клавиш Shift+Ctrl+Z
% = 0,01	Процент – величина, равная 0.01. Используйте ее в выражениях, подобных $10 \cdot * \%$, или как масштабирующий множитель в поле, отводимом для единиц размерности
i или j	Множитель для мнимой части комплексного числа, равный корню квадратному из минус единицы
TOL = 10^{-3}	Допустимая погрешность для различных численных алгоритмов (интегрирования, решения уравнений и т. д.)
CTOL = 10^{-3}	Погрешность для условий ограничения при решении оптимизационных задач с применением функций Maximize, Minimize, Finf и Minerr
ORIGIN = 0	Начало массива. Определяет индекс первого элемента массива
PRNCOLWIDTH = 8	Ширина столбца, используемая при записи файлов функцией WRITEPRN
tblPRNPRECISION = 4	Число значащих цифр, используемых при записи файлов функцией WRITEPRN
FRAME = 0	Используется в качестве счетчика при создании анимации
CWD	Текущая папка в виде строки (только для Mathcad Professional)
inN	Переменная ввода для связи Mathcad через MathConnex с другими программными средствами. Здесь N – номер входа N – целое число, по умолчанию равно 0 (только для Mathcad Professional и при условии применения MathConnex в виде отдельной программы)
outN	Переменная выхода для связи Mathcad через MathConnex с другими программными средствами. Здесь N – номер выхода N – целое число, по умолчанию равно 0 (только для Mathcad Professional и при условии применения MathConnex в виде отдельной программы)
NaN	Нефизическое значение (только в Mathcad 12)

Приложение 4. Встроенные функции и ключевые слова

В этом приложении дан полный список всех функций Mathcad 2001i/11. В приведенных ниже функциях для систем класса Mathcad используются следующие обозначения:

- x и y — вещественные числа;
- z — вещественное либо комплексное число;
- m, n, i, j и k — целые числа;
- v, u и все имена, начинающиеся с v — векторы;
- A и B — матрицы либо векторы;
- M и N — квадратные матрицы;
- F — вектор-функция;
- `file` — либо имя файла, либо файловая переменная, присоединенная к имени файла.

Все углы в тригонометрических функциях выражены в радианах. Многозначные функции и функции с комплексным аргументом всегда возвращают главное значение. Имена приведенных функций нечувствительны к шрифту, но чувствительны к регистру — их следует вводить с клавиатуры в точности, как они приведены. Все функции возвращают указанное для них значение.

Функция или ключевое слово	Описание
<code>acos(z)</code>	Аркосинус
<code>acosh(z)</code>	Гиперболический аркосинус
<code>acot(x)</code>	Аркотангенс
<code>acoth(x)</code>	Гиперболический аркотангенс
<code>acsc(x)</code>	Аркосеканс
<code>acsch(x)</code>	Гиперболический аркосеканс
<code>Ai(x)</code>	Функция Эйри первого рода
<code>angle(x, y)</code>	Угол между положительным направлением оси x и радиус-вектором точки (x, y)
<code>APPENDPRN(file):=M</code>	Добавляет матрицу M к существующему на диске файлу <code>file.prn</code>
<code>arg(z)</code>	Аргумент комплексного числа z (в радианах)
<code>asec(x)</code>	Арксеканс
<code>asech(x)</code>	Гиперболический арксеканс
<code>asin(z)</code>	Арсинус
<code>asinh(z)</code>	Гиперболический арксинус
<code>assume</code>	Ключевое слово режима автоматических символьных преобразований, указывающее на отмену присваивания значений переменным

Функция или ключевое слово	Описание
<code>atan(z)</code>	Арктангенс
<code>atan2(x, y)</code>	Угол между осью x и отрезком прямой с конечными; точками $(0, 0)$ и (x, y) , причем x и y должны быть действительными значениями
<code>atanh(z)</code>	Обратный гиперболический тангенс
<code>augment(A, B)</code>	Объединение двух матриц с одинаковым числом строк (объединение идет бок о бок)
<code>bei(n, x)</code>	Мнимая часть функции Бесселя–Кельвина порядка n
<code>ber(n, x)</code>	Действительная часть функции Бесселя–Кельвина порядка n
<code>Bi(x)</code>	Функция Эйри второго рода
<code>bspline(vx, vy, u, n)</code>	Вектор коэффициентов B-сплайна степени n (1, 2 или 3) для данных, представленных векторами vx и vy , и вектора u , имеющего $n-1$ элементов
<code>bulstoer(v, x1, x2, acc, D, k, s)</code>	Матрица решения системы обыкновенных дифференциальных уравнений на интервале от $x1$ до $x2$ методом Булирша–Штера (используется метод решения с переменным шагом), правая часть которых записана в символьном векторе D , с заданными в векторе v начальными условиями. Параметры k и s задают максимальное число промежуточных точек, на которых ищется решение, и минимально допустимый интервал между ними
<code>Bulstoer(v, x1, x2, n, D)</code>	Матрица решения системы обыкновенных дифференциальных уравнений методом Булирша–Штера (используется метод решения с постоянным шагом), правая часть которых (в виде первых производных неизвестных функций) записана в векторе D , а начальные условия – в векторе v , и при решении на интервале от $x1$ до $x2$ для n точек решения, не считая начальной точки
<code>bvalfit(v1, v2, x1, x2, xi, D, L1, L2, s)</code>	Начальные условия для краевой задачи, заданной в векторах F , $v1$ и $v2$ на интервале от $x1$ до $x2$, где решение известно в некоторой промежуточной точке xi . $L1$ – вектор, чьи n элементов соответствуют величинам n неизвестных функций в $x1$. Некоторые из этих величин могут быть константами, определенными вами из начальных условий. $L2$, как и $L1$, – вектор, чьи n элементов соответствуют величинам n неизвестных функций в $x2$
<code>ceil(x)</code>	Наименьшее целое, не превышающее x
<code>cfft(A)</code>	Быстрое преобразование Фурье для массива комплексных чисел A (возвращает массив такого же размера, что и A)
<code>CFFT(A)</code>	То же, что и предыдущее, но в иной норме
<code>cholesky(M)</code>	Возвращает треугольную матрицу L для треугольного разложения симметричной матрицы M методом Холесского, то есть $M=L \cdot L^T$
<code>cnorm(x)</code>	Интеграл от $-\infty$ до x от функции стандартного нормального распределения

Функция или ключевое слово	Описание
<code>cnper (rate, pv, fv)</code>	Возвращает число составных периодов для получения будущего значения вклада fv при заданных текущем значении вклада pv и проценте начислений $rate$
<code>cols (A)</code>	Число столбцов в матрице A
<code>combin (n, k)</code>	Возвращает число сочетаний k из n , где $n > k$
<code>complex</code>	Ключевое слово режима автоматических символьных преобразований, указывающее на необходимость выполнения операций в комплексной форме
<code>concat (S1, S2, ...)</code>	Строковая переменная, полученная объединением строковых переменных или констант $S1, S2, \dots$
<code>cond1 (M)</code>	Число обусловленности для матрицы M , вычисленное в норме $L1$
<code>cond2 (M)</code>	Число обусловленности для матрицы M , вычисленное в норме $L2$
<code>conde (M)</code>	Число обусловленности для матрицы M , вычисленное в норме евклидова пространства
<code>condi (M)</code>	Число обусловленности матрицы, основанное на равномерной норме
<code>corr (vx, vy)</code>	Коэффициент корреляции двух векторов – vx и vy
<code>tlparcorrel (x, y)</code>	Вычисление одномерной корреляции между векторами x и y (только в Mathcad 12)
<code>correl2d (A, K)</code>	Вычисление двумерной корреляции между матрицей-прототипом A и матрицей-окном K (только в Mathcad 12)
<code>cos (z)</code>	Косинус
<code>cosh (z)</code>	Гиперболический косинус
<code>cot (z)</code>	Котангенс
<code>coth (z)</code>	Гиперболический котангенс
<code>crate (nper, pv, fv)</code>	Возвращает фиксированный процент начислений на период, необходимый для прироста вклада от настоящего значения pv до будущего значения fv при заданном числе составных периодов $nper$ (см. детали в разделе 8.8)
<code>CreateMesh (F, s0, s1, t0, t1, sgrid, tgrid, fmap)</code>	Функция для создания трех массивов параметрически заданной поверхности с двумя параметрами
<code>CreateSpace (F, t0, t1, tgrid, fmap)</code>	Функция создания массивов поверхности, задаваемой в параметрической форме с одним параметром
<code>csc (z)</code>	Косеканс
<code>csch (z)</code>	Гиперболический косеканс
<code>csgn (z)</code>	Функция знака комплексного числа (возвращает либо 0, если $z=0$, либо 1, если $Re(z) > 0$ или если $Re(z) = 0$ и $Im(z) > 0$, либо -1 в остальных случаях)

Функция или ключевое слово	Описание
<code>csort (A, n)</code>	Перестановка строк матрицы A таким образом, чтобы отсортированным в порядке возрастания значений элементов оказался n -й столбец
<code>cspline (vx, vy)</code>	Вектор коэффициентов вторых производных кубического сплайна, построенного по векторам v_x и v_y
<code>cumint (rate, nper, pv, start, end[, type])</code>	Совокупный процент, который платится по займу (см. детали в разделе 8.8)
<code>cumprn (rate, nper, pv, start, end[, type])</code>	Совокупная сумма, которая платится по займу (см. детали в разделе 8.8)
<code>cvar (X, Y)</code>	Коэффициент ковариации X и Y
<code>cyl2xyz (r, t, z)</code>	Функция преобразования цилиндрических координат точки в прямоугольные координаты
<code>dbeta (x, s1, s2)</code>	Плотность вероятности для b-распределения ($s_1, s_2 > 0$ – параметры формы, $0 < x < 1$)
<code>dbinom (k, n, p)</code>	Биномиальное распределение, возвращает значение вероятности $P(x=k)$, где n и k – целые числа, причем $0 \leq k \leq n$ и $0 \leq p \leq 1$, k – случайная величина для биномиального распределения
<code>dcauchy (x, l, s)</code>	Плотность вероятности для распределения Коши (l – параметр разложения, $s > 0$ – параметр масштаба)
<code>dchisq (x, d)</code>	Плотность вероятности для хи-квадрат-распределения ($x, d > 0$, причем d – число степеней свободы)
<code>dexp (x, r)</code>	Плотность вероятности для экспоненциального распределения ($r, x > 0$)
<code>dF (x, d1, d2)</code>	Плотность вероятности для распределения Фишера ($d_1, d_2 > 0$ – числа степеней свободы, $x > 0$)
<code>dgamma (x, s)</code>	Плотность вероятности для гамма-распределения
<code>dgeom (k, p)</code>	Вероятность $P(x=k)$, где k – случайная величина, для геометрического распределения (k – целое неотрицательное число), $0 < p \leq 1$ – вероятность успеха в отдельном испытании
<code>dhypergeom (m, a, b, n)</code>	Гипергеометрическая функция
<code>diag (v)</code>	Диагональная матрица, элементы главной диагонали которой равны элементам вектора v
<code>dlnorm (x, m, s)</code>	Плотность вероятности для логарифмического нормального распределения (μ – натуральный логарифм среднего значения, $\sigma > 0$ – натуральный логарифм среднеквадратичного отклонения, $x > 0$)
<code>dlogis (x, l, s)</code>	Плотность вероятности для логистического распределения (l – параметр разложения, $s > 0$ – параметр масштаба)

Функция или ключевое слово	Описание
<code>dnbinom(k, n, p)</code>	Вероятность $P(x=k)$, где k – случайная величина, для отрицательного биномиального распределения ($n>0$ и $k>0$ – целые числа, $0<p\leq 1$)
<code>dnorm(x, μ, σ)</code>	Плотность вероятности для нормального распределения (μ – среднее значение, $\sigma>0$ – среднеквадратичное отклонение)
<code>dpois(k, λ)</code>	Вероятность $P(x=k)$, где k – случайная величина, для распределения Пуассона ($\lambda>0$, k – целое неотрицательное число)
<code>dt(x, d)</code>	Плотность вероятности для распределения Стьюдента ($d>0$ – число степеней свободы, x – вещественное число)
<code>dunif(x, a, b)</code>	Плотность вероятности для равномерного распределения (a и b – граничные точки интервала, причем $a<b$ и $a\leq x\leq b$)
<code>dweibull(x, s)</code>	Плотность вероятности для распределения Вейбулла ($s>0$ – параметр формы)
<code>eff(rate, nper)</code>	Эффективная ежегодная сумма, которая платится по займу (см. детали в разделе 8.8)
<code>eigenvals(M)</code>	Собственные значения матрицы M
<code>eigenvec(M, z)</code>	Нормированный собственный вектор матрицы M , соответствующий ее собственному значению z
<code>eigenvecs(M)</code>	Матрица, столбцами которой являются собственные векторы матрицы M , при этом порядок расположения собственных векторов соответствует порядку собственных значений, возвращаемых функцией <code>eigenvals</code>)
<code>erf(x)</code>	Функция ошибок
<code>erfc(x)</code>	Дополнительная функция ошибок $1-\text{erf}(x)$
<code>errors(S)</code>	Задание сообщения об ошибке S . Используется в программных модулях
<code>exp(z)</code>	Значение e (основание натурального логарифма) в степени z
<code>expand</code>	Ключевое слово режима автоматических символьных преобразований, задающее расширение выражений
<code>expfit(vx, vy, vg)</code>	Возвращает вектор, содержащий коэффициенты (a , b и c) аппроксимирующего выражения вида $a \cdot e^{(b \cdot x) + c}$, которое лучшим образом приближается к точкам, координаты которых хранятся в векторах vx и vy , а вектор vg содержит первое приближение к решению
<code>factor</code>	Ключевое слово режима автоматических символьных преобразований, задающее разложение (факторизацию) выражений
<code>fft(v)</code>	Быстрое преобразование Фурье для данных, представленных в виде вещественных чисел в векторе v с 2^n элементами, где n – целое число (возвращает вектор размера $2^{n-1}+1$)

Функция или ключевое слово	Описание
<code>FFT(v)</code>	То же, что и <code>fft(v)</code> , но с иной нормировкой
<code>fhyper(a, b, c, x)</code>	Гипергеометрическая функция Гаусса в точке x с параметрами a , b и c
<code>Find(var1, var2, ...)</code>	Значения <code>var1</code> , <code>var2</code> , ..., дающие точные решения системы уравнений в блоке, объявленном директивой <code>Given</code> (число возвращаемых значений равно числу аргументов), который, помимо решаемой системы уравнений, может содержать условия ограничения
<code>float</code>	Ключевое слово режима автоматических символьных преобразований, задающее вывод результатов в виде чисел с плавающей точкой
<code>floor(x)</code>	Наибольшее целое число, меньшее или равное действительного x
<code>fv(rate, nper, pmt [, [pv] [, t ype]])</code>	Будущее значение вклада (см. детали в разделе 8.8)
<code>sal20fvadj(prin, v)</code>	Будущее значение вклада при ряде составных процентных ставок (см. детали в разделе 8.8)
<code>fv(rate, v)</code>	Будущее значение вклада при фиксированных процентных ставках (см. детали в разделе 8.8)
<code>gcd(v)</code>	Целое число, которое является наибольшим общим делителем для всех элементов вектора v , содержащего не менее двух элементов типа <code>real</code> или двух целых неотрицательных чисел
<code>genfit(vx, vy, vg, F)</code>	Вектор, содержащий параметры, которые делают функцию от x и n , заданную в векторе F и имеющую параметры u_0, u_1, \dots, u_{n-1} , наилучшим образом приближающей данные в векторах vx и vy (F является функцией, которая возвращает вектор из $n+1$ элемента, содержащий F и ее частные производные по всем n параметрам, vx и vy должны быть одинакового размера, vg – вектор n элементов, содержащий приближительные значения для n параметров)
<code>geninv(A)</code>	Левая обратная к матрице A , $L \cdot A = E$, где E – единичная матрица размерности $n \times n$, L – прямоугольная матрица размерности $n \times m$, A – прямоугольная матрица размерности $m \times n$
<code>genvals(M, N)</code>	Вектор обобщенных собственных значений v_i матрицы M , соответствующий решению уравнения $M \cdot x = v_i \cdot N \cdot x$ (M и N – матрицы с действительными элементами)
<code>genvecs(M, N)</code>	Матрица, содержащая нормированные собственные векторы, принадлежащие собственным значениям вектора v , возвращаемого <code>genvals</code> , причем n -й столбец этой матрицы является собственным вектором x , удовлетворяющим собственному значению уравнения $M \cdot x = v_n \cdot N \cdot x$, причем матрицы M и N содержат действительные значения
<code>GETWAVINFO(file)</code>	Создает 4-элементный вектор с параметрами файла

Функция или ключевое слово	Описание
Given	Ключевое слово, открывающее блок решения систем уравнений (в котором обычно используются функции Find, Minerr, Maximize и Minimize)
gmean (M)	Возвращает среднее геометрическое элементов матрицы M (элементы матрицы M должны иметь значения, большие нуля)
H1 (m, z)	Функция Ганкеля первого рода, выражаемая через функции Бесселя как $H_1(m, z) = J_n(m, z) + iY_n(m, z)$
H2 (m, z)	Функция Ганкеля второго рода, выражаемая через функции Бесселя как $H_2(m, z) = J_n(m, z) - iY_n(m, z)$
Her (n, x)	Полином Эрмита степени n с аргументом x
hist (intervals, data)	Возвращает вектор с числом точек из data, попавших в соответствующий интервал с границами, заданными вектором intervals (служит для построения гистограмм)
histogram (n, A)	Новая Функция, возвращающая матрицу с двумя столбцами для построения гистограмм
hlookup (z, A, r)	Функция просмотра верхней строки матрицы A на предмет поиска z с возвратом значения, содержащегося в строке r найденного столбца
hmean (M)	Среднее гармоническое элементов матрицы M, элементы которой должны иметь значения больше нуля
I0 (z)	Модифицированная функция Бесселя первого рода нулевого порядка
I1 (z)	Модифицированная функция Бесселя первого рода первого порядка
ibeta (a, x, y)	Неполная бета-функция для x и y с параметром a
icfft (A)	Обратное преобразование Фурье, соответствующее cfft (возвращается массив такого же размера, как и у аргумента A)
ICFFT (A)	Быстрое обратное преобразование Фурье, соответствующее CFFT
identity (n)	Создается единичная квадратная матрица размерности n×n
if (cond, x, y)	Условное выражение, которое возвращает выражение x, если условие cond больше 0, и выражение y в остальных случаях
iffft (v)	Обратное преобразование Фурье, соответствующее fft (вектор v имеет размерность $1+2^{n-1}$, где n – целое число, возвращается вектор с размерностью 2^n)
IFFT (v)	Быстрое обратное преобразование Фурье, соответствующее FFT
Im (z)	Мнимая часть комплексного числа z
In (m, z)	Модифицированная функция Бесселя первого рода m-го порядка
intercept (x, y)	Для векторов x и y возвращает параметр a линейной регрессии $y=a+b \cdot x$

Функция или ключевое слово	Описание
<code>interp(vs, vx, vy, x)</code>	Значение сплайна в точке x по исходным векторам vx и vy и коэффициентам (вторым производным) сплайна vs
<code>ipmt(rate, per, nper, pv[, [, fv][, type]])</code>	Процентная ставка на период
<code>irr(v[, guess])</code>	Внутренний процент возврата для ряда денежных вливаний
<code>isArray(x)</code>	Возвращает 1, если x – матрица или вектор, иначе возвращает 0
<code>isNaN(x)</code>	Возвращает 1, если x – NaN, иначе возвращает 0
<code>isScalar(x)</code>	Возвращает 1, если x – вещественный или комплексный скаляр, иначе возвращает 0
<code>isString(x)</code>	Возвращает 1, если x – строка, иначе возвращает 0
<code>iwave(v)</code>	Обратное вейвлет-преобразование относительно преобразования <code>wave.v</code> . v – вектор размерности 2^n
<code>J0(z)</code>	Функция Бесселя первого рода нулевого порядка
<code>i0J1(z)</code>	Функция Бесселя первого рода первого порядка
<code>Jac(n, a, b, x)</code>	Полином Якоби степени n в точке x с параметрами a и b
<code>Jn(m, z)</code>	Функция Бесселя m -го порядка ($0 < m < 100$)
<code>js(n, x)</code>	Сферическая функция Бесселя первого рода порядка n ($-200 \leq n$) в точке x ($x > 0$)
<code>K0(z)</code>	Модифицированная функция Бесселя второго рода нулевого порядка
<code>K1(z)</code>	Модифицированная функция Бесселя второго рода первого порядка
<code>Kn(m, z)</code>	Модифицированная функция Бесселя второго рода m -го порядка ($0 < m < 100$)
<code>ksmooth(vx, vy, b)</code>	n -мерный вектор возвращенных средних vx , вычисленных на основе распределения Гаусса, vx и vy – n -мерные векторы действительных чисел, параметр b задает ширину окна сглаживания
<code>kurt(A)</code>	Возвращает значение эксцесса (см. выражение в конце таблицы)
<code>Lag(n, x)</code>	Полином Лагерра степени n в точке x
<code>last(v)</code>	Индекс последнего элемента вектора v
<code>lcm(v)</code>	Целое положительное число, которое является наименьшим общим кратным для всех элементов вектора v , имеющего не менее двух элементов типа <code>real</code> или двух целых неотрицательных чисел
<code>Leg(n, x)</code>	Полином Лежандра степени n в точке x

Функция или ключевое слово	Описание
<code>lgsfit(vx, vy, vg)</code>	Возвращает вектор, содержащий коэффициенты (a , b и c) для аппроксимирующего выражения $a / (1 + b \cdot e^{-c \cdot x})$, которое наилучшим образом приближается к точкам, координаты которых хранятся в векторах vx и vy , а вектор vg содержит первое приближение к решению
<code>length(v)</code>	Число элементов в векторе v
<code>line(vx, vy)</code>	Возвращает вектор с параметрами линейной регрессии
<code>linfit(vx, vy, F)</code>	Вектор коэффициентов линейной аппроксимации методом наименьших квадратов по функциям, хранящимся в символьном векторе F , при котором среднеквадратичная погрешность приближения «облака» исходных точек, координаты которых хранятся в векторах vx и vy , оказывается минимальной
<code>linterp(vx, vy, x)</code>	Значение в точке x , вычисленное при линейной интерполяции данных с точками, координаты которых хранятся в векторах vx и vy
<code>Literally</code>	Ключевое слово режима символьной оптимизации
<code>ln(z)</code>	Натуральный логарифм
<code>lnfit(vs, vy)</code>	Функция для логарифмической регрессии вида $a \cdot \ln(x) + b$ – возвращает вектор с параметрами a и b
<code>LoadColormap(file)</code>	Возвращает массив цветковых подмассивов для файла <code>file</code> (файл рисунка), находящегося в папке <code>CMAPS</code>
<code>loess(vx, vy, span)</code>	Вектор, используемый функцией <code>interp</code> для определения набора многочленов второй степени, которые наилучшим образом аппроксимируют часть данных из векторов vx и vy , причем параметр <code>span</code> определяет размер части аппроксимируемых данных
<code>log(z)</code>	Десятичный логарифм
<code>log(z, b)</code>	Логарифм z по основанию b
<code>logfit(vx, vy)</code>	Возвращает вектор, содержащий коэффициенты (a , b и c) аппроксимирующего выражения $a \cdot \ln(x + b) + c$, которое наилучшим образом приближается к точкам, координаты которых хранятся в векторах vx и vy , причем начального приближения не требуется)
<code>lookup(z, A, B)</code>	Функция просмотра вектора или матрицы A на предмет нахождения z с возвратом значения <code>value(s)</code> в позиции <code>position(s)</code> , заданной в матрице B
<code>lsolve(M, v)</code>	Вектор неизвестных, дающих решение системы линейных алгебраических уравнений вида $M \cdot x = v$
<code>0lspline(vx, vy)</code>	Вектор коэффициентов (вторых производных) линейного сплайна, построенного по векторам vx и vy
<code>lu(M)</code>	Треугольное разложение матрицы M : $P \cdot M = L \cdot U$, где L и U – соответственно нижняя и верхняя треугольные матрицы, причем все четыре матрицы квадратные и одного порядка

Функция или ключевое слово	Описание
<code>match(z, A)</code>	Новая функция поиска z в матрице A , возвращающая индекс позиции z в матрице
<code>matrix(m, n, f)</code>	Создается матрица, в которой (i, j) -й элемент равен $f(i, j)$, где $i=0, 1, \dots, m$ и $j=0, 1, \dots, n$, а $f(i, j)$ – некоторая функция
<code>max(A)</code>	Возвращает наибольший по значению элемент матрицы A
<code>Maximize(f, var1, var2, ...)</code>	Значения переменных $var1, var2, \dots$ с ограничительными условиями, при которых функция этих переменных f имеет максимум (используется в вычислительном блоке <i>Given</i>)
<code>mean(v)</code>	Среднее арифметическое значение элементов вектора v
<code>medfit(vx, vy)</code>	Возвращает вектор, содержащий коэффициенты $(a$ и $b)$ аппроксимирующего выражения вида $a + b \cdot x$, которое наилучшим образом приближается к точкам, координаты которых хранятся в векторах vx и vy (медиан-медиан-регрессия)
<code>median(X)</code>	Медиана элементов вектора или матрицы X
<code>medsmooth(vy, n)</code>	Возвращает вектор с m действительными числами, сглаживающий m -мерный вектор вещественных чисел vy методом скользящей медианы (параметр n задает ширину окна, по которому происходит сглаживание)
<code>mhyper(a, b, x)</code>	Конфлюэнтная гипергеометрическая функция в точке x с параметрами a и b
<code>min(A)</code>	Возвращает наименьший элемент в матрице A
<code>Minerr(x1, x2, ...)</code>	Значения $x1, x2, \dots$, дающие приближенные решения системы уравнений и приводящие к минимальной ошибке (используется в вычислительном блоке <i>Given</i>)
<code>Minimize(f, var1, var2, ...)</code>	Значения переменных $var1, var2, \dots$ с условиями ограничений, при которых функция этих переменных f имеет наименьшее значение (используется в вычислительном блоке <i>Given</i>)
<code>utomirr(v, fin_rate, rein_rate)</code>	Модифицированная внутренняя ставка возврата для ряда денежных взносов (см. детали в разделе 8.8)
<code>mod(n, k)</code>	Остаток от деления n на k (аргументы должны быть действительными, результат имеет такой же знак, что и n)
<code>mode(A)</code>	Возвращает наиболее часто встречающиеся повторяющиеся значения из вектора или матрицы A
<code>multigrid(M, n)</code>	Создает матрицу решения уравнения Пуассона, у которого решение равно нулю на границах
<code>nom(rate, nper)</code>	Номинальная процентная ставка
<code>norm1(M)</code>	L1– норма матрицы M
<code>norm2(M)</code>	L2– норма матрицы M

Функция или ключевое слово	Описание
<code>norme (M)</code>	Евклидова норма матрицы M
<code>normi (M)</code>	Неопределенная норма матрицы M
<code>nper (rate, pmt, pv[, [, fv[, t ype] [, type])</code>	Число периодов (см. детали в разделе 8.8)
<code>npv (rate, v)</code>	Нынешнее значение вклада (см. детали в разделе 8.8)
<code>num2str (z)</code>	Преобразует десятичное число z в строковое значение
<code>numol (x_endpts, xpts, t_endpts, tpts, num_pde, num_pae, pde_func, pinit, bc_func)</code>	Возвращает матрицу решений размером $xpts$ на $tpts$ для одномерного дифференциального уравнения в частных производных. Есть только в Mathcad 11, детали см. в разделе 7.6.8
<code>Odesolve (x, b[, steps])</code>	Возвращает решение дифференциальных уравнений, описанных в блоке <i>Given</i> , при заданных начальных условиях и конце интервала интегрирования b
<code>optimize</code>	Ключевое слово, включающее режим символьной оптимизации
<code>pbeta (x, s1, s2)</code>	Значение в точке x функции стандартного нормального распределения
<code>pbinom (k, n, p)</code>	Значение функции распределения биномиального закона для k успехов в серии из n испытаний
<code>pcauchy (x, l, s)</code>	Значение в точке x функции распределения Коши со шкалой параметров l и s
<code>pchisq (x, d)</code>	Значение в точке x кумулятивного хи-квадрат-распределения, в котором d – степень свободы
<code>Pdsolve (u, x, xrange, t, trange[, xpts] [, tpts])</code>	Возвращает функцию или вектор функций и t для решения систем дифференциальных уравнений в частных производных, как без ограничений, так и с ограничениями, заданными в виде алгебраических уравнений. Здесь: u – вектор из имен заданных функций, x – пространственная переменная, $xrange$ – двухэлементный вектор, задающий граничные значения переменной x , t – время, $trange$ – двухэлементный вектор, задающий действительные граничные значения времени, $xpts$ и $tpts$ – целые числа, задающие дискретность изменения x и t . Есть только в Mathcad 11
<code>permut (n, k)</code>	Возвращает число размещений из n элементов по k , причем n и k должны быть целыми неотрицательными числами
<code>pexp (x, r)</code>	Значение в точке x функции экспоненциального распределения
<code>pF (x, d1, d2)</code>	Значение в точке x функции распределения Фишера
<code>pgamma (x, s)</code>	Значение в точке x функции гамма-распределения
<code>pgeom (k, p)</code>	Значение в точке x функции геометрического распределения
<code>hypergeom (m, n, M, N)</code>	Кумулятивное распределение вероятности

Функция или ключевое слово	Описание
<code>plnorm(x, μ, σ)</code>	Значение в точке x функции логарифмического нормального распределения, в котором μ – логарифм среднего значения, $\sigma > 0$ – логарифм стандартного отклонения
<code>plogis(x, l, s)</code>	Значение в точке x функции последовательного распределения, где l – параметр положения, $s > 0$ – параметр масштаба
<code>pmt(rate, nper, pv[, [fv][, type]])</code>	Плата за период займа (см. детали в разделе 8.8)
<code>pnbinom(k, n, p)</code>	Значение в точке x функции отрицательного биномиального распределения, в котором $n > 0$ и $0 < p \leq 1$
<code>pnorm(x, μ, σ)</code>	Значение в точке x функции нормального распределения со средним значением μ и стандартным отклонением σ
<code>Polyhedron(S)</code>	Новая функция построения полиэдра по спецификации S
<code>PolyLookup(n)</code>	Новая функция, возвращающая спецификацию полиэдра с номером n
<code>polyroots(v)</code>	Корни многочлена степени n , коэффициенты которого находятся в векторе v , имеющем длину, равную $n + 1$
<code>Pol2xy(r, t)</code>	Преобразование полярных двумерных координат в прямоугольные
<code>ppmt(rate, per, nper, pv[, [fv][, type]])</code>	Плата на общую сумму данного периода (см. детали в разделе 8.8)
<code>ppois(k, λ)</code>	Значение для k функции распределения Пуассона
<code>predict(v, m, n)</code>	Вектор, содержащий равноотстоящие предсказанные (в ходе экстраполяции) значения n точек, вычисленные по последним m заданным в массиве v данным авторегрессионным методом Бурга
<code>pspline(vx, vy)</code>	Вектор коэффициентов (вторые производные) параболического сплайна, построенного по векторам vx и vy
<code>pspline(Mxy, Mz)</code>	Вектор вторых производных для данных Mxy и Mz , который является параметром функции <code>interp</code>
<code>pt(x, d)</code>	Значение в точке x функции распределения Стьюдента (d – степень свободы, $x > 0$ и $d > 0$)
<code>punif(x, a, b)</code>	Значение в точке x функции равномерного распределения (b и a – границы интервала, $a < b$)
<code>pv(rate, nper, pmt[, [fv][, type]])</code>	Вклад, который должен быть внесен в настоящее время при устойчивой плате и фиксированном проценте ставки (см. детали в разделе 8.8)
<code>pweibull(x, s)</code>	Значение в точке x функции распределения Вейбулла ($s > 0$)
<code>pwrfit(vx, vy, vg)</code>	Возвращает вектор, содержащий коэффициенты (a , b и c) аппроксимирующего выражения вида $a \cdot x^b + c$, которое наилучшим образом приближается к точкам, координаты которых хранятся в векторах vx и vy (вектор vg содержит первое приближение к решению)

Функция или ключевое слово	Описание
$q_{\text{beta}}(p, s_1, s_2)$	Квантили обратного бета-распределения с параметрами формы s_1 и s_2 ($0 \leq p \leq 1$ и $s_1, s_2 > 0$)
$q_{\text{binom}}(p, n, q)$	Количество успешных определений при n -м количестве испытаний при решении уравнения Бернулли при условии, что вероятность этого количества успешных определений есть p (q – вероятность успеха при однократном испытании, $0 \leq q \leq 1$ и $0 \leq p \leq 1$)
$q_{\text{cauchy}}(p, l, q)$	Квантили обратного распределения Коши со шкалой параметров l и s ($s > 0$ и $0 < p < 1$)
$q_{\text{chisq}}(p, d)$	Квантили обратного хи-квадрат распределения, при котором $d > 0$ является характеристикой степеней свободы ($0 \leq p < 1$)
$q_{\text{exp}}(p, r)$	Квантили обратного экспоненциального распределения, при котором $r > 0$ определяет частоту ($0 \leq p < 1$)
$q_F(p, d_1, d_2)$	Квантили обратного распределения Фишера, в котором d_1 и d_2 – степени свободы ($0 \leq p < 1$)
$q_{\text{gamma}}(p, s)$	Квантили обратного гамма-распределения, при котором $s > 0$ – параметры формы ($0 \leq p < 1$)
$q_{\text{geom}}(p, q)$	Квантили обратного геометрического распределения, где q определяет вероятность успеха однократного испытания ($0 < p < 1$ и $0 \leq q < 1$)
$q_{\text{hypergeom}}(p, n, M, N)$	Обратное кумулятивное распределение вероятности, при котором наименьшее целое k соответствует $q_{\text{hypergeom}}(k, a, b, n) \geq p$
$q_{\text{lnorm}}(p, \mu, \sigma)$	Квантили обратного логарифмического нормального распределения, при котором μ – логарифм среднего числа, $\sigma > 0$ – логарифм стандартного отклонения ($0 \leq p < 1$)
$q_{\text{logis}}(p, l, s)$	Квантили обратного последовательного распределения (l – параметр положения, $s > 0$ – параметр масштаба, $0 < p < 1$)
$q_{\text{nbinom}}(p, n, q)$	Квантили обратного отрицательного биномиального распределения с размером n и вероятностью ошибки q ($0 \leq q \leq 1$ и $0 \leq p \leq 1$)
$q_{\text{norm}}(p, \mu, \sigma)$	Квантили обратного нормального распределения со средним значением μ и стандартным отклонением σ ($0 < p < 1$ и $\sigma > 0$)
$q_{\text{pois}}(p, \lambda)$	Квантили обратного распределения Пуассона ($\lambda > 0$ и $0 \leq p \leq 1$)
$q_r(A)$	Разложение матрицы A , $A=Q \cdot R$, где Q – ортогональная матрица, а R – верхняя треугольная матрица
$q_t(p, d)$	Квантили обратного распределения Стьюдента, где d определяет степени свободы ($d > 0$ и $0 < p < 1$)
$q_{\text{unif}}(p, a, b)$	Квантили обратного равномерного распределения, где b и a – конечные значения интервала ($a < b$ и $0 \leq p \leq 1$)
$q_{\text{weibull}}(p, s)$	Квантили обратного распределения Вейбулла ($s > 0$ и $0 < p < 1$)

Функция или ключевое слово	Описание
<code>radau(y, x1, x2, acc, F, k, save)</code>	Возвращает вектор решения дифференциального уравнения новым методом stiffRADAU5 с погрешностью <code>acc</code> (только в Mathcad 11, детали см. в разделе 7.6.5)
<code>Radau(y, x1, x2, k, F)</code>	Возвращает вектор решения дифференциального уравнения новым методом stiffRADAU5 (только в Mathcad 11, детали см. в разделе 7.6.5)
<code>rank(A)</code>	Ранг квадратной матрицы <code>A</code>
<code>rate(nper, pmt, pv[, [fv], [type], [guess]])</code>	Процентная ставка за период ренты (см. детали в разделе 8.8)
<code>rbeta(m, s1, s2)</code>	Вектор <code>m</code> случайных чисел, имеющих бета-распределение (<code>s1, s2 > 0</code> являются параметрами формы)
<code>rbinom(m, n, p)</code>	Вектор <code>m</code> случайных чисел, имеющих биномиальное распределение ($0 \leq p \leq 1$, <code>n</code> – целое число, удовлетворяющее условию <code>n > 0</code>)
<code>rcauchy(m, l, s)</code>	Вектор <code>m</code> случайных чисел, имеющих распределение Коши (<code>l, s > 0</code> – параметры масштаба)
<code>rchisq(m, d)</code>	Вектор <code>m</code> случайных чисел, имеющих хи-квадрат-распределение (<code>d > 0</code> определяет степени свободы)
<code>Re(z)</code>	Действительная часть комплексного числа <code>z</code>
<code>READBIN(file, type[, [endian] [, cols] [, skip] [, maxrows]])</code>	Считывает бинарный файл со спецификацией, <code>type</code> . Файл может иметь тип <code>uint16</code> , <code>unit32</code> , <code>double</code> (64-разрядные числа), <code>float</code> (32-разрядные числа) и <code>byte</code> (8-разрядные числа). Необязательные параметры уточняют структуру считываемых бинарных данных. Есть только в Mathcad 11 (см. детали в разделе 6.3.3)
<code>READ_BLUE(file)</code>	Массив, соответствующий синему компоненту изображения, содержащегося в <code>file</code>
<code>READBMP(file)</code>	Массив, соответствующий черно-белому компоненту изображения, содержащегося в <code>file</code>
<code>READ_GREEN(file)</code>	Массив, соответствующий зеленому компоненту изображения, содержащегося в <code>file</code>
<code>READ_HLS(file)</code>	Массив, представляющий данные о цвете объекта в <code>file</code> (оттенки цвета, насыщенность и интенсивность)
<code>READ_HLS_HUE(file)</code>	Массив, представляющий данные об оттенках цвета для объекта в <code>file</code>
<code>READ_HLS_LIGHT(file)</code>	Массив, представляющий данные о яркости цвета для объекта в <code>file</code>
<code>READ_HLS_SAT(file)</code>	Массив, представляющий данные о насыщенности цвета для объекта в <code>file</code>
<code>READ_HSV(file)</code>	Массив, представляющий значения оттенков цвета, яркости и насыщенности для объекта в <code>file</code>

Функция или ключевое слово	Описание
<code>READ_HSV_HUE (file)</code>	Массив, представляющий значение оттенка цвета компонента в <code>file</code>
<code>READ_HSV_SAT (file)</code>	Массив, представляющий значение насыщенности цвета компонента в <code>file</code>
<code>READ_HSV_VALUE (file)</code>	Массив, представляющий значения интенсивности цвета для компонента в <code>file</code>
<code>READ_IMAGE (file)</code>	Матрица изображения из файла <code>file</code> , представляющая это изображение в форме черно-белого с полутонами
<code>READPRN (file)</code>	Присваивание матрице значений из файла с именем <code>file.prn</code>
<code>READ_RED (file)</code>	Массив, соответствующий красному цвету компонента в <code>file</code>
<code>READRGB (file)</code>	Массив, состоящий из трех подмассивов, которые представляют красный, зеленый и синий компоненты цветного изображения, находящегося в <code>file</code>
<code>READWAV (file)</code>	Новая функция считывания звукового файла и создания его матричного представления
<code>regress (Mxy, vz, n)</code>	Вектор, запрашиваемый функцией <code>interp</code> для вычисления многочлена n -й степени, который наилучшим образом приближает «облако» точек с координатами, хранящимися в <code>Mxy</code> и <code>vz</code> (<code>Mxy</code> – матрица размером $m \times 2$, содержащая координаты x и y , <code>vz</code> – m -мерный вектор, содержащий z -координаты, соответствующие m точкам, указанным в <code>Mxy</code>)
<code>relax (M1, M2, M3, M4, M5, A, U, x)</code>	Квадратная матрица решения уравнения Пуассона для спектрального радиуса x
<code>reverse (v)</code>	Вектор с обратным (начиная с конца) расположением элементов исходного вектора <code>v</code>
<code>rexp (m, r)</code>	Вектор m случайных чисел, имеющих экспоненциальное распределение ($r > 0$)
<code>rF (m, d1, d2)</code>	Вектор m случайных чисел, имеющих распределение Фишера ($d1, d2 > 0$ определяют степени свободы)
<code>rgamma (m, s)</code>	Вектор m случайных чисел, имеющих гамма-распределение ($s > 0$ – параметр формы)
<code>rgeom (m, p)</code>	Вектор m случайных чисел, имеющих геометрическое распределение ($0 < p \leq 1$)
<code>rhypergeom (k, n, M, N)</code>	Вектор k случайных чисел с гипергеометрическим распределением
<code>rkadapt (v, x1, x2, acc, n, D, k, s)</code>	Матрица, содержащая таблицу значений решения задачи Коши на интервале от x_1 до x_2 для системы обыкновенных дифференциальных уравнений, вычисленную методом Рунге–Кутты с переменным шагом и начальными условиями в векторе <code>v</code> , причем правые части системы записаны в <code>D</code> , n – число шагов, k – максимальное число промежуточных точек решения и s – минимально допустимый интервал между точками

Функция или ключевое слово	Описание
<code>Rkadapt(v, x1, x2, n, D)</code>	Матрица решения системы обыкновенных дифференциальных уравнений численным методом Рунге–Кутты на интервале от x_1 до x_2 с переменным шагом, при минимальном числе шагов n , причем правые части уравнений в символьной форме задаются в векторе D , а начальные условия – в векторе v
<code>rkfixed(v, x1, x2, n, D)</code>	Матрица решения системы обыкновенных дифференциальных уравнений методом Рунге–Кутты на интервале от x_1 до x_2 при фиксированном числе шагов n , причем правые части уравнений записаны в символьном векторе D , а начальные условия – в векторе v
<code>rlnorm(m, μ, σ)</code>	Вектор m случайных чисел, имеющих логарифмическое нормальное распределение (μ – логарифм среднего значения, $\sigma > 0$ – логарифм стандартного отклонения)
<code>rlogis(m, l, s)</code>	Вектор m случайных чисел, имеющих логистическое распределение (l – локализационный параметр и $s > 0$ – параметр масштаба)
<code>rnbinom(m, n, p)</code>	Вектор m случайных чисел, имеющих отрицательное биномиальное распределение ($0 < p \leq 1$, n – целое число, которое удовлетворяет условию $n > 0$)
<code>rnd(x)</code>	Функция генерации случайных чисел с равномерным распределением в интервале $[0, x]$
<code>rnorm(m, μ, σ)</code>	Вектор m случайных чисел с нормальным распределением
<code>root(expr, var)</code>	Значение переменной var (в пределах точности TOL), при котором выражение $expr$ равно нулю
<code>root(expr, var, [a, b])</code>	Значение переменной var (в пределах точности TOL), при котором в интервале изоляции корня $[a, b]$ значение выражения $expr$ равно нулю
<code>round(x, n)</code>	При $n > 0$ возвращает округленное значение x с точностью до n знаков после десятичной точки. При $n < 0$ возвращает округленное значение x с n цифрами слева от десятичной точки. При $n = 0$ возвращает округленное до ближайшего целого значение x (x – скаляр типа <code>real</code> или целое число)
<code>rows(A)</code>	Число строк матрицы A
<code>rpois(m, λ)</code>	Вектор m случайных чисел, имеющих распределение Пуассона ($\lambda > 0$)
<code>rref(A)</code>	Ступенчатый вид матрицы A
<code>rsort(A, n)</code>	Матрица A , отсортированная по строке n (перестановка столбцов матрицы A таким образом, чтобы отсортированной по возрастанию значений элементов оказалась n -я строка)
<code>rt(m, d)</code>	Вектор m случайных чисел, имеющих распределение Стьюдента ($d > 0$)
<code>runif(m, a, b)</code>	Вектор m случайных чисел, имеющих равномерное распределение (b и a – границы интервала, $a < b$)

Функция или ключевое слово	Описание
<code>rweibull(m, s)</code>	Вектор m случайных чисел, имеющих распределение Вейбулла ($s > 0$ является параметром формы)
<code>SaveColormap(file, M)</code>	Создает файл (с именем <code>file</code>) цветовой карты для значений матрицы M и возвращает число строк записанного файла
<code>SaveColormap(file, M)</code>	Создает файл цветовой карты с именем <code>file</code> используя изображение, хранящееся в матрице M , и возвращает число строк файла
<code>sbval(v, x1, x2, D, L, S)</code>	Установка начальных условий для краевой задачи, определенной в символьном векторе D , вектор v – начальные условия на интервале $[x1, x2]$, L – векторзначная функция <code>load(x1, v)</code> с вектором v , содержащим n начальных условий в точке $x1$, и S – векторзначная функция <code>score(x2, y)</code> с вектором из n элементов, представляющих разности между начальными условиями в точке $x2$ и значениями искомого решения в этих точках
<code>search(S, Subs, m)</code>	Стартовая позиция подстроки <code>Subs</code> в строке <code>S</code> при поиске, начиная с позиции <code>m</code> . При неуспешном поиске возвращает -1
<code>sec(z)</code>	Секанс
<code>sech(z)</code>	Гиперболический секанс
<code>series</code>	Ключевое слово режима автоматических символьных преобразований, задающее разложение в ряд
<code>Seed(x)</code>	Сбрасывает число, инициализирующее генератор случайных чисел, и возвращает значение предшествующего числа
<code>sign(x)</code>	Функция знака (возвращает либо 0 , если $x=0$, либо 1 , если x положительно, либо -1 , если x отрицательно)
<code>signum(z)</code>	Возвращает 0 , если $z=0$ и $z/ z $ в остальных случаях
<code>simplify</code>	Ключевое слово режима автоматических символьных преобразований, задающее упрощение выражения
<code>sin(z)</code>	Синус
<code>sinc(z)</code>	Возвращает $\sin(z)/z$
<code>sinfit(vx, vy, vg)</code>	Возвращает вектор, содержащий коэффициенты (a , b и c) аппроксимирующего выражения вида $a \cdot \sin(x + b) + c$, которое наилучшим образом приближается к точкам, координаты которых хранятся в векторах vx и vy (вектор vg содержит первое приближение к решению)
<code>sinh(z)</code>	Гиперболический синус
<code>SIUnitsOf(x)</code>	Возвращает единицу измерения x в системе SI
<code>Slope(x, y)</code>	Для векторов данных x и y возвращает коэффициент b линейной регрессии $a+b \cdot x$
<code>skew(A)</code>	Возвращает асимметрию из множества значений

Функция или ключевое слово	Описание
<code>slope (vx, vy)</code>	Значение параметра b (угловой коэффициент линии регрессии) линейной регрессии $y = a + b \cdot x$ для данных, заданных векторами vx и vy
<code>sort (v)</code>	Вектор v , отсортированный по убыванию
<code>sph2xyz (r, t, phi)</code>	Новая функция преобразования сферических координат точки в прямоугольные
<code>stack (A, B)</code>	Объединяет две матрицы A и B путем размещения A над B (матрицы A и B должны иметь одинаковое число столбцов)
<code>stderr (vx, vy)</code>	Возвращает стандартную ошибку линейной регрессии для точек, данные о которых содержатся в векторах vx и vy
<code>stdev (A)</code>	Стандартное отклонение элементов матрицы A
<code>Stdev (A)</code>	Стандартное отклонение элементов матрицы A в иной нормировке
<code>stiffb (v, x1, x2, acc, n, D, J, k, s)</code>	Матрица решений методом Булирша–Штера с переменным шагом жесткого дифференциального уравнения, записанного в D , с якобианом J , причем v – вектор начальных значений на интервале $[x1, x2]$
<code>Stiffb (v, x1, x2, n, D, J)</code>	Матрица решений методом Булирша–Штера с постоянным шагом жесткого дифференциального уравнения, записанного в D , с якобианом J , причем v – вектор начальных значений на интервале $[x1, x2]$
<code>stiffR (v, x1, x2, acc, n, D, J, k, s)</code>	Матрица решений методом Розенброка с переменным шагом жесткого дифференциального уравнения, записанного в D , с якобианом J , причем v – вектор начальных значений на интервале $[x1, x2]$
<code>StiffR (v, x1, x2, n, D, J)</code>	Матрица решений методом Розенброка с постоянным шагом жесткого дифференциального уравнения, записанного в D , с якобианом J , причем v – вектор начальных значений на интервале $[x1, x2]$
<code>str2num (S)</code>	Преобразование строкового представления числа (в любой форме) в реальное число
<code>str2vec (S)</code>	Преобразование в реальный вектор строки S с записями чисел в строковом формате
<code>strlen (S)</code>	Количество знаков в строке S
<code>submatrix (A, ir, jr, ic, jc)</code>	Блок матрицы A , состоящий из всех элементов, содержащихся в строках от ir до jr и столбцах от ic до jc ($ir \geq jr$ и $ic \geq jc$)
<code>substr (S, m, n)</code>	Подстрока, полученная из строки S выделением n знаков, начиная с позиции m в строке S
<code>supsmooth (vx, vy)</code>	n -мерный вектор, сглаживающий зависимость y от x , представленную точками с координатами, хранящимися в векторах vy и vx

Функция или ключевое слово	Описание
<code>svd(A)</code>	Сингулярное разложение матрицы A размерности $n \times m$: $A=U \cdot S \cdot V^T$, где U и V – ортогональные матрицы размерности $m \times m$ и $n \times n$ соответственно, S – диагональная матрица, на диагонали которой расположены сингулярные числа матрицы A
<code>svds(A)</code>	Вектор, содержащий сингулярные числа матрицы A размерности $m \times n$, где $m \geq n$
<code>tan(z)</code>	Тангенс
<code>tanh(z)</code>	Гиперболический тангенс
<code>Tcheb(n, x)</code>	Полином Чебышева первого рода степени n в точке x
<code>tr(M)</code>	След (сумма диагональных элементов) квадратной матрицы M
<code>trunc(x)</code>	Целая часть от действительного числа x
<code>Ucheb(n, x)</code>	Полином Чебышева второго рода степени n в точке x
<code>UnitsOf(x)</code>	Возвращает размерность x , если x – размерная переменная, иначе возвращает 1
<code>until(выражение1, выражение2)</code>	Возвращает выражение2 до тех пор, пока выражение1 равно или больше 0 (в новых версиях Mathcad эта функция не рекомендуется к применению)
<code>var(A)</code>	Дисперсия элементов матрицы A
<code>Var(A)</code>	Дисперсия элементов матрицы A в иной норме, чем <code>var</code>
<code>vec2str(v)</code>	Строковое представление вектора v
<code>vlookup(z, A, c)</code>	Новая функция поиска z в первом столбце матрицы с возвратом <code>value(s)</code> в строке <code>row(s)</code> и столбце, заданном параметром <code>c</code>
<code>wave(v)</code>	Дискретное волновое преобразование действительных чисел с использованием 4-коэффициентного волнового фильтра Добеши, причем вектор v должен содержать 2^n действительных значений, где n – целое число
<code>WRITEBIN(file, type)</code>	Записывает в файл матрицу бинарного файла (например, <code>WRITEBIN(file, type):=A</code>), файл может иметь тип <code>unit16</code> , <code>unit32</code> , <code>double</code> (64-разрядные числа), <code>float</code> (32-разрядные числа) и <code>byte</code> (8-разрядные числа)
<code>WRITEBMP(file)</code>	Создает файл формата BMP из оттенков серого
<code>WRITE_HLS(file)</code>	Создает матрицу, в которой представлена цветовая информация о форматах файлов BMP, GIF, JPG или TGA величинами оттенка, освещенности и насыщенности (HLS)
<code>WRITE_HSV(file)</code>	Создает матрицу, в которой представлена цветовая информация о форматах файлов BMP, GIF, JPG или TGA оттенками, насыщенностью и величиной (HSV)
<code>WRITEPRN(file)</code>	Запись матрицы в файл <code>file</code>

Функция или ключевое слово	Описание
WRITERGB (file)	Создает цветной файл формата BMP из матрицы, в которой изображение хранится в формате RGB
WRITEWAV (file, s, b)	Запись в формате WAV-файла данных матрицы M: = WRITEWAV (file, s, b) со скоростью s и разрядностью b
xyz2cyl (x, y, z)	Новая функция преобразования прямоугольных трехмерных координат точки в цилиндрические координаты
xyz2pol (x, y, z)	Новая функция преобразования прямоугольных двумерных координат точки в полярные координаты
xyz2sph (x, y, z)	Новая функция преобразования прямоугольных трехмерных координат точки в сферические координаты
Y0 (z)	Функция Бесселя второго рода нулевого порядка (x – действительное и положительное значение)
Y1 (z)	Функция Бесселя второго рода первого порядка (x – действительное и положительное значение)
Yn (m, z)	Функция Бесселя второго рода m-го порядка (x – действительное и положительное значение, m – от 0 до 100)
ys (n, z)	Сферическая функция Бесселя второго рода порядка n (n≥200) в точке x (x>0)
δ (x, y)	Символ Кронекера, равный 1, если x=y, и 0 в противном случае (x и y целые). В Mathcad эта функция удалена
ε (i, j, k)	Полностью асимметричный тензор размерности 3. i, j и k должны быть целыми числами от 0 до 2 (или между ORIGIN и ORIGIN+2, если ORIGIN≠0). Результат равен либо 0, если любые два аргумента равны, либо 1, если три аргумента являются четной перестановкой (0, 1, 2), либо -1, если три аргумента являются перестановкой (0, 1, 2), кратной 2 и не кратной 4
Γ (z)	Гамма-функция
F (x)	Функция Хевисайда, возвращающая 1, если x≥0, и 0 в остальных случаях

Приложение 5. Сообщения о наиболее характерных ошибках

Сообщение	Описание
All evaluations resulted in either an error or complex result	Вычисления приводят к ошибке или комплексному результату
A ""Find"" or ""Minerr"" must be preceded by a mathing ""Given""	Перед Find или Minerr должно быть слово Given
Arguments in function definitions must by name	Аргументы в определениях функции должны быть именами
Argument to large	Аргумент слишком велик
At least one limit must be infinity	Хотя бы один предел должен быть бесконечным
Array size mismatch	Несоответствие размера массива
Cannot be defined	Не может быть определено
Cannot take subscript	Не содержит индексов
Can't evaluate this function when is argument less than or equal to zero	Нельзя вычислить данную функцию, когда ее аргумент меньше или равен нулю
Can't converge to a solution	Не сходится к решению
Can't define the same variable more than once in the same expression	Невозможно определить одну и ту же переменную в выражении более одного раза
Can't determine what units the result of this operation should have	Невозможно определить, в каких единицах должен быть результат данной операции
Can't divide by zero	Деление на 0 невозможно
Can't evaluate this accurately at one or more of values you specified	Невозможно точно вычислить одно или более значений
Can't evaluate this expression. It may have resulted in an overflow or an infinite loop	Невозможно вычислить это выражение. Это может быть результатом переполнения или зацикливания
Can't find a solution	Невозможно найти решение
Can't use a range variable ib a solve block	Нельзя использовать ранжированную переменную в вычислительном блоке
Definition stack overflow	Переполнение стека определений
Degree of the polynomial must be between 1 and 99	Степень полинома должна быть от 1 до 99
Did not find solution	Решение не найдено
Dimension must be >4	Размерность должна быть больше 4
Dimension to non real power	Размерность массива не является целым числом
Discarding large (huge) result	Сброс большого результата (его можно разместить в буфере)
Domain error	Ошибка области определения

Приложение 5. Сообщения о наиболее характерных ошибках

Сообщение	Описание
Duplicate	Дублирование
End of file	Конец файла
End points cannot be the same	Конечные точки не могут быть одинаковыми
Equation too large	Слишком большое выражение
Error in constant	Ошибка в константе
Error in list	Ошибка в списке
Error in solve block	Ошибка в блоке
Expected array or list	Ожидается массив или список
Floating point error	Ошибка вычислений с плавающей точкой
Floats not handled	Вычисления с плавающей точкой не поддерживаются
File error	Ошибка в файле
File not found	Файл не найден
Found a number with a magnitude greater than 10^{307}	Найдено число со значением выше 10^{307}
Found a singularity while evaluating this expression. You may be dividing by zero	При вычислении этого выражения найдена сингулярность. Возможно, вы делите на 0
Illegal array operation	Неверная операция с массивом
Illegal context. Press <F1> for help	Неверный контекст. Нажмите клавишу F1 для справки
Illegal dimensions	Недопустимые размеры (матрицы)
Illegal factor	Неверный множитель
Illegal function name	Неверное имя функции
Illegal function syntax	Неправильный синтаксис функции
Illegal ORIGIN	Неверное значение переменной ORIGIN
Illegal range	Неправильный диапазон
Illegal tolerance	Некорректная точность аппроксимации
Incompatible units	Несовместимые единицы
Index out of bounds	Индекс вне границ
Integer too large	Целое число слишком велико
Integer too small	Целое число слишком мало
Interrupted	Вычисления прерваны
Invalid arguments	Недопустимые аргументы
Invalid format	Неверный формат
Invalid order	Неверный порядок

Сообщение	Описание
Invalid range	Недопустимый интервал
List too long	Слишком длинный входной список
Misplaced comma	Неуместная запятая
Missing operand	Пропущенный операнд
Missing operator	Пропущенный оператор
Must be 3-vector	Должно быть трехмерным вектором
Must be array	Должно быть массивом
Must be between two lock reions	Должно быть между двумя закрытыми областями
Must be dimensionless	Должно быть безразмерным
Must be function	Должно быть функцией
Must be increasing	Должно быть возрастающим
Must be integer	Должно быть целым
Must be less than the number of data points	Должно быть меньше, чем число точек данных
Must be nonzero	Должно быть ненулевым
Must be positive	Должно быть положительным
Must be range	Должно быть диапазоном
Must be real	Должно быть вещественным
Must be scalar	Должно быть скаляром
Must be square	Должна быть квадратной (о матрице)
Must be string	Должно быть строкой
Must be vector	Должно быть вектором
Nested solve block	Вложенный блок решения
No answer found; stack limit reached	Не найден ответ; лимит стека исчерпан
No closed form for ...	Не найдено замкнутой формы для ...
No matching Given	Нет соответствующей директивы Given
No solution found	Не найдено решение
No scalar value	Не является скаляром
Not a name	Не является именем
Not converging	Не сходится
Not enough memory for this operation	Недостаточно памяти для этой операции
Only one array allowed	Допустим только один массив
Only positive values are allowed here	Только положительные значения разрешены здесь
Overflow	Переполнение

Приложение 5. Сообщения о наиболее характерных ошибках

Сообщение	Описание
Significance lost	Потеряны значащие цифры
Singularity	Деление на ноль
Singular matrix	Матрица сингулярная
Stack overflow	Переполнение стека
Syntax error	Синтаксическая ошибка
rightSubscript too large	Слишком большой индекс
The expression to the left of the equal sign cannot be defined	Выражение слева от знака равенства не может быть определено
The number of rows and/or columns in this arrays do not math	Число строк и/или столбцов в этих массивах не совпадает
The units in this expression do not match	Размерности в этом выражении не согласованы
Ther is an extra comma in this expression	В выражении лишняя запятая
This expression in incomplete. You must fill in the placeholders	Это выражение неполное. Вы должны заполнить места ввода
This function has too many arguments	Эта функция имеет слишком много аргументов
This function needs more arguments	Этой функции требуется больше аргументов
This operation can only be performed on a function (or array, number, string, ...)	Эта операция может быть произведена только над функцией (или массивом, числом, строкой, ...)
This subscript is too large	Этот нижний индекс слишком велик
This system of the equation has more unknowns than there are equations	Эта система уравнений имеет больше неизвестных, чем уравнений
This value must be a matrix	Это значение должно быть матрицей
This value must be a vector	Это значение должно быть вектором
This value must be a integer greater than 1	Это значение должно быть целым числом, большим 1
This variable or function is not defined above	Эта переменная или функция не определена выше
This variable must be a range variable	Эта переменная должна быть ранжированной переменной
Too few arguments	Слишком мало аргументов
Too few constraints	Слишком мало ограничений
Too few elements	Слишком мало элементов
Too few subscripts	Слишком мало индексов
Too large to display	Слишком велико, чтобы отобразить
Too many arguments	Слишком много аргументов
Too many constraints	Слишком много ограничений
Too many iterations	Слишком много итераций

Сообщение	Описание
Too many integration steps	Слишком много шагов интегрирования
Too many points	Слишком много точек
Too many subscripts	Слишком много индексов
Undefined	Не определено
Underflow	Потеря значащих цифр результата
Unequal array dimensions	Неравные размеры массивов
Unmatched parenthesis	Дисбаланс скобок
Value of subscript or superscript is too big (or too small) for this array	Значение нижнего или верхнего индекса слишком велико (или слишком мало) для данного массива
Vector elements cannot all be the same	Все элементы вектора не могут быть одинаковыми
Wrong size vector	Неверный размер вектора

ПРИМЕЧАНИЕ Со строчной маленькой буквы даны сообщения, которые могут относиться к разным объектам, например к функциям, операторам, выражениям и т. д.

Список литературы

1. Дьяконов В. П. Компьютерная математика. Теория и практика. М.: Нолидж, 2001.
2. Дьяконов В. П. Система MathCAD. Справочник. М.: Радио и связь, 1993.
3. Дьяконов В. П. Справочник по MathCAD PLUS 6.0 PRO. М.: СК Пресс, 1997.
4. Дьяконов В. П. Справочник по MathCAD PLUS 7.0 PRO. М.: СК Пресс, 1998.
5. Дьяконов В. П., Абраменкова И. В. Mathcad 7 в математике, в физике и в Internet. М.: Нолидж, 1998.
6. Дьяконов В. П., Абраменкова И. В. Mathcad 8 PRO в математике, в физике и в Internet. М.: Нолидж, 1999.
7. Дьяконов В. П. Mathcad 8/2000. Специальный справочник. СПб: Питер, 2000.
8. Дьяконов В. П. Mathcad 2000. Учебный курс. СПб: Питер, 2000.
9. Дьяконов В. П. Mathcad 2001. Учебный курс. СПб: Питер, 2001.
10. Дьяконов В. П. Mathcad 2001. Специальный справочник. СПб: Питер, 2002.
11. Дьяконов В. П. Mathcad 2001i/11. Энциклопедия. М.: СОЛОН-Пресс, 2004.
12. MathCAD 6.0 PLUS./Пер. с англ. руководства пользователя. М.: Филинь, 1996.
13. Очков В. Ф. Mathcad 7 Pro для студентов и инженеров. М.: Компьютер Пресс, 1998.
14. Очков В. Ф. Mathcad 8 Pro для студентов и инженеров. М.: Компьютер Пресс, 1999.
15. Плис А. И., Сливина Н. А. MATHCAD: математический практикум для экономистов и инженеров. М.: Финансы и статистика, 1999.
16. Кирьянов Д. Самоучитель MathCAD 11. СПб: БХВ-Петербург, 2003.
17. Кирьянов Д. Самоучитель MathCAD 12. СПб: БХВ-Петербург, 2004.
18. Херхагер М., Партоль Х. Mathcad 2000: Полное руководство. К.: БХВ, 2001.
19. Поршнева С. В. Компьютерное моделирование физических процессов с использованием пакета MathCAD. М.: Горячая линия — Телеком. 2002.
20. Глушаков С. В., Жакин И. А., Хачиров Т. С. Математическое моделирование. Учебный курс. Харьков: Фолио. М.: АСТ. 2001.
21. Гурский Д. А. Вычисления в Mathcad. Минск: Новое знание, 2003.
22. Каганов В. И. Радиотехника + компьютер + Mathcad. М.: Горячая линия — Телеком. 2001.
23. Mathcad 11. User's Guide. Mathsoft Engineering&Education Inc., 2003.
24. Дьяконов В. П. Maple 9 в математике, физике и образовании. М.: СОЛОН-Пресс, 2004.
25. Дьяконов В. П. Mathematica 4.1/4.2/5 в математических и научно-технических расчетах. М.: СОЛОН-Пресс, 2004.
26. Дьяконов В. П. MATLAB 6/6.1/6.5 + Simulink 4/5. Основы применения. М.: СОЛОН-Пресс, 2004.
27. Дьяконов В. П. MATLAB 6/6.1/6.5 + Simulink 4/5 в математике и моделировании. М.: СОЛОН-Пресс, 2004.
28. Фадеев А. К., Фадеева В. Н. Вычислительные методы линейной алгебры. Изд. 3-е, стереотипное. СПб.: Изд-во «Лань», 2002.
29. Справочник по специальным функциям/Под. ред. М. Абрамовича и И. Стиган. М.: Наука, Физматлит, 1979.
31. Бахвалов Н. С., Жидков Н. П., Кобельков Г. М. Численные методы. М.: Наука, Физматлит, 1987.

32. Г. Корн, Т. Корн. Справочник по математике для научных работников и инженеров. М.: Наука, Физматлит, 1973.
33. Х. Гулд, Я. Тобочник. Компьютерное моделирование в физике. М.: Мир, 1990, часть 1 и часть 2. Толстой Г. П. Ряды Фурье. М.: Физматлит, 1980.
34. Жуков А. И. Метод Фурье в вычислительной математике. М.: Физматлит, 1992.
35. Ланцош К. Практические методы прикладного анализа. М.: Физматлит, 1961.
36. Блейхут Р. Быстрые алгоритмы цифровой обработки сигналов. М.: Мир, 1989.
37. Марпл-мл. С. Л. Цифровой спектральный анализ и его приложения. М.: Мир. — 1990.
38. И. Добеши. Десять лекций по вейвлетам. М.: R&C Dynamics, 2001.
39. К. Чуи. Введение в вейвлеты. М.: Мир, 2001.
40. Столниц Э., ДеРоуз Т., Салезин Д. Вейвлеты в компьютерной графике. Теория и приложения. М. — Ижевск: РХД, 2002.
41. Дьяконов В. П. Вейвлеты. От теории к практике. Изд. 2-е, дополненное и переработанное. М.: СОЛОН-Пресс, 2004
42. Дьяконов В. П. Мой Word 95/97. М.: АСТ, 1998.
43. Новые информационные технологии. Учебное пособие. Под. ред. проф. В. П. Дьяконова. М.: СОЛОН-Пресс, 2004.
44. Дьяконов В. П. Internet. Настольная книга пользователя. Издание 5-е. М.: Солон-Р, 2004.
45. Дьяконов В. П., Максимчук А. А., Ремнев А. М., Смердов В. Ю. Энциклопедия устройств на полевых транзисторах. Под общей ред. проф. В. П. Дьяконова. М.: Солон-Р, 2002.
46. Зернов Н. В., Карпов В. Г. Теория радиотехнических цепей. Л.: Энергия, 1972.
47. Баскаков С. И. Радиотехнические цепи и сигналы. Учебник для вузов по специальности «Радиотехника». М.: Высшая школа, 2000.
48. Абиев Р. Ш. Вычислительная гидродинамика и теплообмен. Введение в метод конечных разностей: Учебное пособие. С.Пб. Изд-во НИИХимии СПбГУ. 2002.
49. Холоднов В. А., Дьяконов В. П., Иванова Е. Н., Кирьянова Л. С. Математическое моделирование и оптимизация химико-технологических процессов. Практическое руководство. С.Пб.: АНО НПО «Профессионал». 2003.

Содержание

Введение	3
Глава 1. Первое знакомство с Mathcad	5
1.1. Общие сведения о системе Mathcad	5
1.1.1. История создания и место системы Mathcad	5
1.1.2. Документы и рабочие листы Mathcad	7
1.1.3. Входной язык и язык реализации Mathcad	8
1.1.4. Традиционные средства программирования	9
1.1.5. Средства оптимизации вычислений	9
1.1.6. Средства расширения систем Mathcad	10
1.2. Начало работы с Mathcad	11
1.2.1. Запуск Mathcad	11
1.2.2. Основные возможности систем Mathcad	13
1.3. Основы работы с системой Mathcad	14
1.3.1. К чему сводится работа в среде Mathcad	14
1.3.2. Работа с текстовым редактором	15
1.3.3. Работа с формульным редактором	17
1.3.4. Подготовка и вычисление простых выражений	17
1.3.5. Операции вывода и присваивания	20
1.3.6. Применение шаблонов математических операторов и символов	22
1.3.7. Применение встроенных функций	24
1.3.8. Применение ранжированных переменных и векторов	25
1.3.9. Ввод матриц и векторов	26
1.3.10. Запрет вычисления математических выражений	27
1.3.11. Прерывание вычислений	28
1.4. Введение в графику Mathcad	28
1.4.1. Построение двумерного графика одной функции	28
1.4.2. Изменение размеров и перемещение графика	29
1.4.3. Построение графиков ряда функций	30
1.4.4. Виды графиков	31
1.4.5. Построение графиков поверхностей	31
1.4.6. Вращение трехмерного графика мышью	34
1.4.7. Построение на одном трехмерном графике ряда поверхностей	35
1.4.8. Одновременное построение разных типов трехмерных графиков	36
1.4.9. Трехмерные гистограммы и визуализация матриц	37
1.4.10. Двумерные графики с двумя осями Y в Mathcad 12	37
Глава 2. Интерфейс пользователя	40
2.1. Работа с окном документа	40
2.1.1. Окна документов систем Mathcad 2001i/11/12	40
2.1.2. Курсор ввода и линия раздела страниц	42
2.1.3. Строка заголовка	42
2.1.4. Строка и состав меню	43

2.1.5. Строка состояния	44
2.1.6. Полосы прокрутки.....	44
2.1.7. Всплывающие подсказки	44
2.1.8. Контекстное меню правой клавиши мыши	45
2.1.9. Представление операции умножения	45
2.2. Работа с панелью инструментов.....	45
2.2.1. Обзор панели инструментов	45
2.2.2. Файловые операции с панели инструментов.....	46
2.2.3. Печать и просмотр документов	48
2.2.4. Операции редактирования в панели инструментов	48
2.2.5. Управление размещением блоков	49
2.2.6. Операции с выражениями	49
2.2.7. Доступ к дополнительным возможностям Mathcad	50
2.2.8. Вставка переменной-таблицы в Mathcad 11/12	50
2.2.9. Управление ресурсами Mathcad 8 — Mathcad 2001i	50
2.3. Работа с панелью форматирования.....	51
2.3.1. Обзор средств панели форматирования	51
2.3.2. Кнопки переключения стилей документов	51
2.3.3. Выбор шрифтов для документа	51
2.3.4. Выравнивание текста.....	51
2.3.5. Создание маркированных и нумерованных списков	51
2.3.6. Кнопки задания индексов Mathcad 11/12	52
2.4. Применение палитр математических знаков	52
2.4.1. Вывод панели палитр математических знаков.....	52
2.4.2. Применение палитр математических знаков	52
2.5. Управление видом интерфейса.....	54
2.5.1. Обзор позиции View меню системы Mathcad 2001i.....	54
2.5.2. Обзор позиции View меню Mathcad 11/12	54
2.5.3. Отображение и скрытие элементов интерфейса.....	55
2.5.4. Обновление экрана.....	55
2.5.5. Задание предпочтений в Mathcad 2001i.....	56
2.5.6. Задание предпочтений в Mathcad 11/12	57
2.6. Управление окнами	58
2.6.1. Обзор позиции Window меню	58
2.6.2. Каскадное расположение окон.....	59
2.6.3. Расположение окон по горизонтали	59
2.6.4. Расположение окон по вертикали.....	60
2.6.5. Системное меню окна.....	61
2.7. Работа с Центром информационных ресурсов и справкой	63
2.7.1. Центр информационных ресурсов Mathcad 2001i	63
2.7.2. Центр информационных ресурсов Mathcad 11/12.....	64
2.7.3. Организация справочной системы Mathcad.....	64
2.7.4. Меню справки Mathcad 2001i	65
2.7.5. Работа со справкой Mathcad 2001i	66
2.7.6. Работа с электронными учебниками	69
2.7.7. «Быстрые шпаргалки» и справочные таблицы.....	69

2.7.8. Работа с электронными книгами и расширениями Mathcad	72
2.7.9. Работа с информационными ресурсами Mathcad 11/12.....	75
2.7.10. Оперативная справка по объектам документов	77
2.8. Некоторые особенности интерфейса Mathcad 12.....	77
2.8.1. Окно записи файлов в новых форматах	77
2.8.2. Окно свойств файлов	77
2.8.3. Отправка документов по электронной почте.....	78
2.8.4. Настройка состава кнопок панелей инструментов.....	79
2.8.5. Выбор инструментов контроля.....	80
Глава 3. Подготовка документов	81
3.1. Операции с файлами документов	81
3.1.1. Типы файлов в Mathcad.....	81
3.1.2. Меню файловых операций File	82
3.1.3. Создание нового документа.....	82
3.1.4. Открытие существующего документа	84
3.1.5. Запись документа на диск без переименования	85
3.1.6. Запись документа на диск с переименованием	86
3.1.7. Заккрытие документа	86
3.1.8. Отправка документа по электронной почте.....	87
3.1.9. Выход из системы.....	87
3.2. Печать и просмотр документов	87
3.2.1. Установка параметров страницы.....	87
3.2.2. Выбор принтера	88
3.2.3. Предварительный просмотр.....	89
3.2.4. Печать документов	89
3.2.5. Выделение и показ всех блоков	90
3.2.6. Изменение масштаба документа	91
3.3. Ввод с помощью клавиатуры.....	92
3.3.1. Формы курсора ввода.....	92
3.3.2. Выделение объектов с помощью клавиш	93
3.3.3. Применение управляющих клавиш	93
3.3.4. Применение клавиш редактирования.....	95
3.3.5. Клавишное управления документами и окнами.....	95
3.3.6. Клавиши для создания объектов.....	95
3.4. Редактирование	96
3.4.1. Применение команд позиции Edit меню	96
3.4.2. Основные приемы редактирования	97
3.4.3. Отмена и повторение операций редактирования	98
3.4.4. Выделение объектов и перенос их в буфер обмена.....	98
3.4.5. Выделение всех объектов	100
3.4.6. Копирование объекта в буфер обмена.....	100
3.4.7. Вставка объекта из буфера обмена.....	101
3.4.8. Специальная вставка	101
3.4.9. Удаление выделенных объектов	103
3.4.10. Поиск подстроки.....	103
3.4.11. Замена подстроки.....	104

3.4.12. Переход к заданной странице документа	106
3.4.13. Проверка орфографии	106
3.4.14. Редактирование связей	107
3.4.15. Редактирование внедренных объектов.....	107
3.5. Вставка в документ объектов.....	108
3.5.1. Подменю вставки Insert.....	108
3.5.2. Вставка шаблона матриц и векторов	109
3.5.3. Вставка функций.....	110
3.5.4. Вставка единиц измерения размерных величин	110
3.5.5. Вставка шаблона импортируемого рисунка	111
3.5.6. Вставка текстового комментария	112
3.5.7. Вставка математических выражений в текстовый комментарий	113
3.5.8. Вставка линии разрыва страницы	113
3.5.9. Вставка гипертекстовой ссылки	113
3.5.10. Вставка гипермедиассылки	116
3.5.11. Вставка объектов.....	117
3.6. Особенности вставок в системе Mathcad 11/12	119
3.6.1. Позиция Insert меню Mathcad 11/12	119
3.6.2. Примеры вставки элементов контроля.....	119
3.7. Управление вычислениями в документах.....	119
3.7.1. Позиция Math меню Mathcad 2001i	119
3.7.2. Ручные вычисления	120
3.7.3. Автоматические вычисления	121
3.7.4. Оптимизация вычислений	121
3.7.5. Запрет вычисления выражений	123
3.8. Установка опций вычислений	123
3.8.1. Обзор опций вычислений	123
3.8.2. Установка значений системных переменных.....	124
3.8.3. Опции вычислений	125
3.8.4. Опции отображения результатов вычислений	125
3.8.5. Выбор системы единиц	126
3.8.6. Форматирование размерных единиц.....	126
3.9. Ошибки в документах и отладка вычислений	127
3.9.1. Виды ошибок в документах Mathcad	127
3.9.2. Индикация ошибок	127
3.9.3. Трассировка ошибок.....	128
3.10. Работа с командами позиции Tools меню Mathcad 11/12.....	130
3.10.1. Обзор команд позиции Tools меню Mathcad 11/12	130
3.10.2. Защита документов Mathcad 11/12.....	130
3.10.3. Опции вычислений в документах Mathcad 11/12	131
Глава 4. Графика	132
4.1. Виды графиков.....	132
4.2. Двумерный график	133
4.2.1. Вставка шаблона двумерного графика.....	133
4.2.2. Особенности построения графиков функции одной переменной	133

4.2.3. Простейшие приемы форматирования двумерных графиков.....	135
4.2.4. Графики с параметрическим заданием функций.....	136
4.2.5. Построение графиков ряда функций на одном рисунке	136
4.2.6. Полулогарифмические и логарифмические графики	137
4.2.7. Построение графиков в полярной системе координат	138
4.3. Трехмерный график поверхности.....	139
4.3.1. Построение поверхностей по матрице аппликат их точек	140
4.3.2. Построение параметрически заданных поверхностей.....	142
4.3.3. Построение трехмерных фигур с вырезом	142
4.3.4. Построение трехмерных графиков без задания матрицы	143
4.3.5. Построение графика поверхности, заданной в векторной параметрической форме	144
4.3.6. Применение новой функции CreateMesh.....	145
4.3.7. Построение объемной фигуры, образованной вращением кривой.....	146
4.3.8. Построение полиэдров	146
4.3.9. Построение на одном графике нескольких трехмерных объектов.....	149
4.4. Контурный трехмерный график	151
4.4.1. Стандартный способ построения контурных графиков.....	151
4.4.2. Построение контурных графиков без явного задания матрицы	151
4.5. Точечный трехмерный график	152
4.5.1. Определение точечного графика	152
4.5.2. Построение точечного графика с заданием матрицы аппликат точек	152
4.5.3. Построение точечного графика с заданием только функции поверхности	153
4.5.4. Применение функции CreateSpace.....	154
4.6. Трехмерная гистограмма	155
4.6.1. Обычное построение гистограмм	155
4.6.2. Построение трехмерных гистограмм с заданием только функции поверхности	155
4.7. Трехмерный график в векторном представлении	156
4.7.1. Обычное построение графиков векторного поля	157
4.7.2. Построение графика векторного поля, заданного в параметрической форме	157
4.8. Специальные приемы построения трехмерных графиков	158
4.8.1. Построение трехмерных графиков мастером	158
4.8.2. Трехмерный «лоскутный» график	160
4.8.3. Два пересекающихся в пространстве тора.....	161
4.8.4. Представление функций двух переменных графиками векторного поля.....	161
4.9. Техника анимации (оживления) графиков.....	162
4.9.1. Принципы анимации графиков	162
4.9.2. Подготовка к анимации	163
4.9.3. Создание кадров изображения	164
4.9.4. Воспроизведение анимированного рисунка.....	165
4.9.5. Вызов проигрывателя	166
Глава 5. Форматирование объектов.....	169

5.1. Понятие о форматировании объектов документов.....	169
5.1.1. Для чего нужно форматирование	169
5.1.2. Позиция Format меню.....	169
5.1.3. Форматирование математических выражений	170
5.2. Форматирование результатов вычислений	171
5.2.1. Установки формата действительных чисел	171
5.2.2. Представление комплексных чисел	172
5.2.3. Представление размерных величин.....	172
5.2.4. Представление результатов численных расчетов	174
5.2.5. Представление массивов	174
5.3. Форматирование текста.....	175
5.3.1. Изменение параметров шрифта	175
5.3.2. Форматирование абзацев	176
5.3.3. Установка позиций табуляции.....	177
5.3.4. Задание стиля текстовых комментариев.....	178
5.3.5. Надписи на переднем и заднем плане	179
5.4. Установка свойств объектов	179
5.4.1. Установка свойств математического выражения	179
5.4.2. Установка свойств оптимизации	181
5.5. Форматирование двумерных графиков.....	181
5.5.1. Форматирование графиков в прямоугольной системе координат	181
5.5.2. Форматирование осей.....	182
5.5.3. Форматирование линий графиков	183
5.5.4. Форматирование надписей на графиках.....	185
5.5.5. Задание параметров по умолчанию.....	186
5.5.6. Примеры форматирования двумерных графиков	186
5.5.7. Трассировка графиков.....	186
5.5.8. Просмотр участков двумерных графиков	188
5.6. Форматирование графиков в полярной системе координат.....	189
5.6.1. Установка форматов графиков в полярной системе координат.....	189
5.6.2. Трассировка графика в полярной системе координат	190
5.6.3. Просмотр участков двумерных графиков в полярной системе координат.....	191
5.7. Форматирование трехмерных графиков	192
5.7.1. Окно установки форматов трехмерных графиков	192
5.7.2. Общие параметры трехмерных графиков	193
5.7.3. Параметры отображения осей трехмерных графиков	194
5.7.4. Параметры внешнего вида трехмерных графиков	195
5.7.5. Параметры надписей трехмерных графиков	196
5.7.6. Параметры освещения трехмерных графиков.....	196
5.7.7. Параметры форматирования граней	197
5.7.8. Специальные эффекты трехмерных графиков.....	198
5.7.9. Дополнительные параметры трехмерных графиков	199
5.7.10. Параметры быстрого построения графиков	200
5.8. Форматирование электронных документов.....	201

5.8.1. Форматирование цветов.....	203
5.8.2. Разделение областей документов.....	203
5.8.3. Выравнивание областей.....	204
5.8.4. Блокировка области.....	205
5.8.5. Открытие заблокированной области.....	207
5.8.6. Скрытие области.....	207
5.8.7. Открытие скрытой области.....	208
5.8.8. Создание колонтитулов.....	209
5.8.9. Замена латинских букв на греческие.....	210
5.9. Форматирование изображений и рисунков.....	211
5.9.1. Подготовка к форматированию рисунков.....	211
5.9.2. Повороты рисунков.....	212
5.9.3. Масштабирование рисунков и операции с буфером.....	212
5.9.4. Управление передаточной характеристикой по яркости.....	212
5.9.5. Управление перемещением и выделением фрагментов изображения.....	213
5.9.6. Определение координат пиксела.....	213
5.9.7. Использование для форматирования рисунков контекстного меню.....	214
Глава 6. Обычные вычисления и математический анализ.....	217
6.1. Входной язык системы Mathcad.....	217
6.1.1. Алфавит входного языка системы Mathcad.....	217
6.1.2. Применение чисел и констант.....	217
6.1.3. Новая константа NaN в Mathcad 12.....	218
6.1.4. Переменные.....	219
6.1.5. Операторы и операнды.....	219
6.1.6. Встроенные функции.....	220
6.1.7. Математические выражения и приоритет операций.....	220
6.2. Применение переменных.....	221
6.2.1. Присваивание переменным значений.....	221
6.2.2. Локальное и глобальное присваивание.....	221
6.2.3. Применение размерных переменных.....	222
6.2.4. Задание ранжированных переменных.....	223
6.2.5. Применение ранжированных переменных.....	224
6.2.6. Применение таблиц вывода.....	225
6.3. Работа с файловыми данными.....	226
6.3.1. Файлы как данные.....	226
6.3.2. Применение функций для работы с файлами.....	226
6.3.3. Новые функции для работы с бинарными файлами в Mathcad 11/12.....	228
6.4. Вычисление математических выражений.....	229
6.4.1. Операторы и операнды.....	229
6.4.2. Арифметические операторы.....	229
6.4.3. Применение операторов отношения.....	230
6.4.4. Применение логических операторов.....	231
6.5. Решение типовых задач математического анализа.....	231
6.5.1. Простые примеры на применение операторов математического анализа.....	231

6.5.2. Вычисление сумм.....	233
6.5.3. Вычисление произведений.....	234
6.5.4. Вычисление пределов.....	234
6.5.5. Определение производной и полного дифференциала.....	235
6.5.6. Вычисление и визуализация производных.....	237
6.5.7. Определение интегралов.....	238
6.5.8. Вычисление однократных интегралов.....	239
6.5.9. Вычисление кратных интегралов.....	241
6.5.10. «Подводные камни» интегрирования.....	241
6.5.11. Задание и графическое представление градиента функции.....	244
6.6. Вычисление элементарных функций.....	245
6.6.1. Вычисление типовых элементарных функций.....	245
6.6.2. Функции комплексного аргумента.....	246
6.6.3. Функции комбинаторики и теории чисел.....	246
6.7. Вычисление специальных математических и иных функций.....	248
6.7.1. Вычисление функций Бесселя.....	248
6.7.2. Вычисление гамма-функции.....	249
6.7.3. Дополнительные специальные функции.....	249
6.7.4. Альтернативные функции с новой нормировкой в Mathcad 11/12.....	250
6.7.5. Дополнительные неактивные функции.....	250
6.7.6. Задание функций пользователя.....	251
6.7.7. Числовые функции с условиями сравнения.....	253
6.7.8. Функция условных выражений.....	253
6.7.9. Рекурсивные функции.....	254
6.7.10. Функции контроля типа переменных.....	255
6.7.11. Функции обработки строк.....	255
6.7.12. Вычисление различных специальных математических функций.....	257
6.8. Решение задач линейной алгебры.....	258
6.8.1. Массивы, векторы и матрицы.....	258
6.8.2. Использование индексированных переменных.....	259
6.8.3. Ввод элементов векторов и матриц.....	260
6.8.4. Задание векторов и матриц.....	260
6.8.5. Векторные и матричные операции.....	261
6.8.6. Операция векторизации.....	262
6.8.7. Векторные и матричные функции.....	263
6.8.8. Функции, возвращающие специальные характеристики матриц.....	264
6.8.9. Примеры применения векторных и матричных операторов и функций.....	264
6.8.10. Дополнительные матричные функции.....	265
6.8.11. Функции сортировки для векторов и матриц.....	267
6.8.12. Примеры применения дополнительных векторных и матричных функций.....	268
6.8.13. Решение систем линейных уравнений.....	268
Глава 7. Численные методы.....	270
7.1. Решение нелинейных и алгебраических уравнений.....	270
7.1.1. Решение нелинейных уравнений вида $F(x) = 0$	270
7.1.2. Вычисление корней степенного многочлена.....	271

7.1.3. Функция <code>root</code> в составе функции пользователя	271
7.1.4. Решение систем нелинейных уравнений с блоком <code>Given</code>	272
7.1.5. Примеры решения нелинейных уравнений	273
7.2. Реализация итерационных вычислений.....	275
7.2.1. Рекуррентное вычисление чисел Фибоначчи.....	275
7.2.2. Функция <code>until</code> и ее применение.....	276
7.2.3. Пример решения системы рекуррентных уравнений	277
7.3. Решение задач оптимизации.....	277
7.3.1. Поиск минимума функции Розенброка с помощью функции <code>Minerr</code>	278
7.3.2. Поиск экстремумов функции Розенброка с помощью функций <code>Maximize</code> и <code>Minimize</code>	278
7.3.3. Решение задач линейного программирования	281
7.4. Интегральные преобразования Фурье	282
7.4.1. Спектральный анализ и синтез на основе рядов Фурье	282
7.4.2. Быстрые прямое и обратное преобразования Фурье.....	283
7.4.3. Примеры преобразований Фурье	284
7.4.4. Альтернативные преобразования Фурье	288
7.4.5. Двумерное преобразование Фурье	288
7.5. Дискретные волновые (вейвлет) преобразования.....	288
7.5.1. Вейвлеты как новый базис представления сигналов.....	288
7.5.2. Особенности вейвлетов	291
7.5.3. Вейвлеты типа «мексиканская шляпа»	292
7.5.4. Прямое и обратное непрерывные вейвлет-преобразования	294
7.5.5. Примеры непрерывных вейвлет-преобразований и построение их спектрограмм	294
7.5.6. Встроенные в <code>Mathcad</code> функции вейвлет-преобразований.....	297
7.5.7. Простые примеры дискретных вейвлет-преобразований.....	297
7.6. Решение обыкновенных дифференциальных уравнений.....	300
7.6.1. Представление дифференциальных уравнений.....	300
7.6.2. Решение систем обыкновенных дифференциальных уравнений	301
7.6.3. Решение дифференциальных уравнений с помощью функции <code>odesolve</code>	303
7.6.4. Решения жестких систем дифференциальных уравнений	303
7.6.5. Новая функция <code>Radau</code> в <code>Mathcad 2001i/11/12</code>	306
7.6.6. Решение двухточечных краевых задач	307
7.6.7. Решение дифференциальных уравнений Пуассона и Лапласа	309
7.6.8. Новые функции <code>Mathcad 11/12</code> для решения ОДУ в частных производных	310
7.6.9. Анализ колебаний струны в одномерном случае.....	310
7.6.10. Анализ колебаний поверхности.....	310
7.6.11. Анимация колебания поверхности.....	312
7.6.12. Решение дифференциальных уравнений с комплексными параметрами	313
7.7. Реализация некоторых специальных численных методов	314
7.7.1. Вычисление ортогональных полиномов по рекуррентным формулам	314
7.7.2. Численное дифференцирование.....	316

7.7.3. Численное вычисление трех производных аналитической функции.....	317
7.7.4. Численное дифференцирование зашумленной функции.....	319
7.7.5. Интегрирование таблично заданных функций	320
7.7.6. Вычисление площади под таблично заданной кривой.....	321
7.7.7. Быстрое вычисление интегралов по формулам Уэддла и Ньютона—Котесса	321
7.7.8. Интерполяция по частным формулам Лагранжа	323
7.7.9. Поиск глобального максимума многоэкстремальной функции	323
7.7.10. Поиск локальных экстремумов в массиве	325
7.7.11. Анализ сложной функции	325
7.7.12. Расчет и построение асимптот функции $f(x)$	327
7.8. Численное решение ОДУ по классическим формулам	327
7.8.1. Решение ДУ модифицированным методом Эйлера	328
7.8.2. Решение ОДУ первого порядка методом Рунге—Кутта	330
7.8.3. Решение ДУ вида $y''=F(x,y,z)$ методом Рунге—Кутта	331
7.8.4. Решение ДУ вида $y''(t) + Ay'(t) + By(t) = C(t)$	331
7.8.5. Решение системы из двух ДУ методом Рунге—Кутта	333
7.8.6. Решение дифференциального уравнения $dy/dx = f(x,y)$ методом Рунге—Кутта—Фельберга	334
7.9. О реализации в Mathcad вариационных методов	336
7.9.1. Особенности решения задач механики вариационными методами.....	336
7.9.2. Решение задачи на прогиб струны.....	336
7.9.3. Решение задачи на прогиб струны в среде Mathcad	337
Глава 8. Обработка данных, статистика и финансовые расчеты.....	339
8.1. Интерполяция, экстраполяция и аппроксимация	339
8.1.1. Одномерная линейная интерполяция и экстраполяция	339
8.1.2. Одномерная сплайновая интерполяция и экстраполяция	339
8.1.3. Примеры линейной и сплайновой интерполяции.....	340
8.1.4. Одномерная В-сплайновая интерполяция и экстраполяция	341
8.1.5. Двумерная линейная и сплайновая интерполяция.....	341
8.2. Аппроксимация функций и данных.....	343
8.2.1. Приближение функций рядом Тейлора.....	343
8.2.2. Интерполяция и экстраполяция функций по Лагранжу.....	345
8.2.3. Полиномиальная аппроксимация	345
8.3. Статистическая обработка данных.....	346
8.3.1. Эксперименты, события и другие понятия статистики	346
8.3.2. Решение задач комбинаторики.....	348
8.3.3. Дискретные и непрерывные случайные величины.....	349
8.3.4. Законы распределения и статистические функции Mathcad.....	350
8.3.5. Функции вычисления плотности распределения вероятности.....	353
8.3.6. Функции распределения	354
8.3.7. Квантили распределения.....	354
8.3.8. Функции создания случайных чисел с различными законами распределения.....	355
8.3.9. Новые функции корреляции в Mathcad 12	356
8.4. Регрессия	357

8.4.1. Линейная регрессия.....	357
8.4.2. Реализация линейной регрессии общего вида.....	358
8.4.3. Реализация одномерной полиномиальной регрессии.....	359
8.4.4. Проведение многомерной регрессии.....	361
8.4.5. Проведение нелинейной регрессии общего вида.....	361
8.4.6. Новые функции для проведения регрессии в Mathcad 2000/2001.....	362
8.4.7. Пример выполнения экспоненциальной регрессии.....	363
8.4.8. Пример выполнения синусоидальной регрессии.....	363
8.5. Сглаживание данных.....	363
8.5.1. Функции сглаживания данных.....	363
8.5.2. Линейное сглаживание по пяти точкам.....	365
8.5.3. Нелинейное сглаживание по семи точкам.....	365
8.5.4. Вейвлет- и вейвлет-медианное сглаживание.....	367
8.5.5. Комбинированное сглаживание вейвлетами и фильтром «скользящее среднее».....	368
8.5.6. Вейвлет-сплайновое сглаживание.....	368
8.6. Предсказание зависимостей.....	370
8.6.1. Предсказание методом Бурга «чистых» зависимостей.....	370
8.6.2. Предсказание зашумленных зависимостей.....	371
8.7. Аппроксимация, сглаживание и экстраполяция рядами Фурье.....	372
8.7.1. Приближение данных рядом Фурье.....	372
8.7.2. Улучшение сходимости приближения рядом Фурье.....	373
8.7.3. Эффективное приближение данных рядом Фурье.....	374
8.8. Финансово-экономические расчеты.....	375
8.8.1. Функции финансово-экономических расчетов.....	375
8.8.2. Примеры финансово-экономических расчетов.....	376
Глава 9. Символьные вычисления.....	379
9.1. Роль и средства символьных (аналитических) вычислений.....	379
9.1.1. Роль и значение символьных вычислений.....	379
9.1.2. Возможности символьного процессора.....	379
9.2. Символьные вычисления в командном режиме.....	380
9.2.1. Обзор меню символьных операций Symbolics.....	380
9.2.2. Символьные операции с выделенными выражениями.....	382
9.2.3. Символьные операции с выделенными переменными.....	382
9.2.4. Символьные операции с выделенными матрицами.....	382
9.2.5. Символьные операции интегральных преобразований.....	383
9.2.6. Выделение объектов символьных операций.....	383
9.2.7. Команды подменю Evaluate.....	384
9.2.8. Команда упрощения Symbolically.....	384
9.2.9. Команда вычислений с плавающей точкой Floating Point.....	385
9.2.10. Команда задания комплексных вычислений Complex.....	385
9.3. Техника и практика символьных вычислений.....	386
9.3.1. Упрощение математических выражений — команда Simplify.....	386
9.3.2. Символьное вычисление производных.....	387
9.3.3. Символьное вычисление интегралов.....	388

9.3.4. Символьное вычисление сумм и произведений	389
9.3.5. Замечания по поводу упрощения выражений	390
9.3.6. Расширение выражений — команда Expand	391
9.3.7. Разложение выражений на простые множители — команда Factor.....	391
9.3.8. Разложение по подвыражениям — команда Collect	393
9.3.9. Вычисление коэффициентов полиномов — команда Polynomial Coefficients.....	393
9.4. Операции относительно переменной.....	394
9.4.1. Символьное дифференцирование выражений — команда Differentiate ...	394
9.4.2. Символьное интегрирование — команда Integrate	395
9.4.3. Символьное решение уравнений — команда Solve	396
9.4.4. Осуществление подстановок — команда Suvstitute.....	397
9.4.5. Разложение выражений в ряд Тейлора — команда Expand to Series	398
9.4.6. Разложение на правильные дроби — команда Convert to Partial Fraction	400
9.4.7. Матричные операции	400
9.4.8. Интегральные преобразования	401
9.5. Настройка символьных операций	403
9.5.1. Задание стиля символьных операций — команда Evaluation Style.....	403
9.5.2. Интерпретация результатов символьных операций.....	404
9.6. Символьные операции с оператором символьного вывода	405
9.6.1. Оператор символьного вывода \rightarrow	405
9.6.2. Директивы символьного оператора.....	406
9.6.3. Решение систем линейных уравнений в символьном виде	407
9.6.4. Решение систем нелинейных уравнений и неравенств в символьном виде	408
9.6.5. Символьное решение систем уравнений, заданных в векторном виде ...	409
9.6.6. Типовые символьные операции	409
9.6.7. Символьное интегрирование	412
9.6.8. Символьное вычисление пределов функций.....	413
9.6.9. Символьное дифференцирование	414
9.6.10. Символьные преобразования Фурье, Лапласа и Z-преобразование	415
9.7. Символьное решение дифференциальных уравнений.....	416
9.7.1. Средства символьного решения дифференциальных уравнений	416
9.7.2. Применение преобразования Лапласа для решения ДУ.....	416
9.7.3. Решение задачи Коши для линейного неоднородного ДУ	417
9.7.4. Общее решение неоднородного ДУ первого порядка	417
9.7.5. Нахождение всех решений ДУ первого порядка	418
9.7.6. Решение задачи Коши для ДУ в полных дифференциалах	418
9.7.7. Нахождение частного решения ДУ третьего порядка	420
9.7.8. Фундаментальная система уравнений и общее решение неоднородного ДУ четвертого порядка.....	420
Глава 10. Программирование и интеграция.....	422
10.1. Создание простейших программных конструкций	422
10.1.1. Задание операторов пользователя	422
10.1.2. Задание и применение функций пользователя	423
10.2. Модульное программирование	424

10.2.1. Задание программных модулей.....	424
10.2.2. Инструкция добавления линий в модуль Add Line	425
10.2.3. Оператор внутреннего присваивания.....	425
10.2.4. Условная инструкция if.....	425
10.2.5. Инструкция организации цикла for	425
10.2.6. Инструкция организации цикла while	426
10.2.7. Инструкция otherwise.....	426
10.2.8. Инструкция прерывания break	426
10.2.9. Инструкция continue.....	426
10.2.10. Инструкция return	426
10.2.11. Инструкция on error и функция error	426
10.3. Примеры программирования в среде Mathcad	427
10.3.1. Простейшие примеры создания программных модулей	427
10.3.2. Обработка ошибок в программных модулях	428
10.3.3. Модуль построения точек в пространстве.....	428
10.3.4. Модуль фурье-анализа.....	429
10.3.5. Рекурсивная генерация простых чисел	430
10.3.6. Программа моделирования аттрактора Лоренца.....	430
10.3.7. Построение фрактала «кукуруза»	431
10.3.8. Заключительные замечания по программированию.....	432
10.4. Обзор программных средств, интегрируемых с Mathcad	433
10.4.1. Текстовые процессоры Word.....	433
10.4.2. Электронные таблицы Excel	434
10.4.3. Графическая система Axum.....	434
10.4.4. Матричная математическая система MATLAB	434
10.4.5. Система символьной математики Maple.....	435
10.5. Интеграция Mathcad с другими программными средствами.....	435
10.5.1. Интеграция Mathcad с текстовым процессором Word	435
10.5.2. Интеграция Mathcad с табличным процессором Excel	439
10.5.3. Вставка в Mathcad объектов класса ActiveX.....	441
10.5.4. Интеграция Mathcad с графическим пакетом Axum	442
10.5.5. Интеграция Mathcad с пакетом Visio	443
10.5.6. Интеграция Mathcad с пакетом SmartSketch LE.....	444
10.5.7. Интеграция Mathcad с матричной лабораторией MATLAB.....	447
10.5.8. Интеграция Mathcad с пакетом VisSim	448
Глава 11. Математическое моделирование	450
11.1. Основные понятия моделирования.....	450
11.1.1. Назначение и цели моделирования.....	450
11.1.2. Основные виды моделей и их свойства	450
11.1.3. Основные принципы моделирования	451
11.1.4. Технология моделирования.....	452
11.1.5. Основные методы решения задач моделирования.....	452
11.1.6. Контроль правильности модели	454
11.2. Задача моделирования полета камня	455
11.2.1. Постановка задачи моделирования	455
11.2.2. Построение математической модели.....	456

11.2.3. Решение аналитическим методом	456
11.2.4. Реализация математической модели «Бросок камня» в среде Mathcad	457
11.2.5. Уточнение модели «Бросок камня» и ее реализация.....	458
11.3. Моделирование процессов на основе известных формул.....	459
11.3.1. Моделирование изменения параметров атмосферы	459
11.3.2. Моделирование преодоления самолетом звукового барьера	460
11.3.3. Стрельба по летящей утке	461
11.4. Моделирование на основе конечно-разностных методов.....	462
11.4.1. Имитация броуновского движения частиц в плоскости	462
11.4.2. Имитация броуновского движения частиц в пространстве	463
11.4.3. Моделирование диффузии	464
11.4.4. Моделирование торможения автомобиля.....	465
11.4.5. Моделирование падения парашютиста	465
11.4.6. Моделирование эффекта Доплера.....	466
11.4.7. Обтекание крыла самолета воздухом	469
11.5. Моделирование колебательных систем.....	469
11.5.1. Анализ линейной колебательной системы	469
11.5.2. Анализ нелинейной колебательной системы Ван—дер—Поля.....	470
11.5.3. Моделирование системы Дафинга с внешним воздействием.....	471
11.5.4. Хаос и моделирование аттрактора Лоренца	472
11.6. Моделирование фундаментальных физических явлений.....	473
11.6.1. Моделирование математического маятника с анимацией	473
11.6.2. Моделирование рассеивания альфа-частиц.....	476
11.6.3. Разделение изотопов.....	478
11.6.4. Движение частиц в магнитном поле	479
11.6.5. Моделирование дифракции на щели	481
11.6.6. Интерференция света, излучаемого малым отверстием.....	482
11.7. Моделирование биологических и экономических систем	484
11.7.1. Модель системы «хищник—жертва» Лотки—Вольтерра.....	484
11.7.2. Модель системы «хищник—жертва» с логистической поправкой.....	485
11.7.3. Модель системы «хищник—жертва» Холлинга—Тэннера	485
11.7.4. Моделирование замкнутой экономической системы	487
11.8. Моделирование экономических процессов.....	487
11.8.1. Моделирование закона Мура на основе нелинейной регрессии.....	487
11.8.2. Закон Мура и метод предсказания Бурга	493
11.8.3. Оптимальные экономико-математические модели.....	494
11.8.4. Решение задач максимизации объема продукции	495
11.8.5. Решение задач минимизации ресурсов.....	496
11.8.6. Решение транспортной задачи.....	496
11.8.7. Задачи целочисленного программирования с булевыми переменными.....	499
11.9. Сетевые модели в оптимизации управленческих решений	501
11.9.1. Задача поиска кратчайшего пути.....	501
11.9.2. Задача о распределении потоков в сетях	502
Глава 12. Обработка сигналов и расчет электронных устройств.....	506
12.1. Создание, моделирование и представление сигналов	506

12.1.1. Моделирование ступени и прямоугольного импульса	506
12.1.2. Синусоидальный (гармонический) сигнал	507
12.1.3. Квантование и преобразование синусоидального сигнала	507
12.1.4. Модуляция синусоидальных сигналов	508
12.1.5. Создание сигналов комбинациями элементарных функций	509
12.1.6. Выборки сигнала и теорема Котельникова	509
12.1.7. Понятие о случайных сигналах	511
12.2. Основы спектрального метода моделирования сигналов и линейных систем.....	511
12.2.1. Схема спектрального метода.....	511
12.2.2. Программа точного вычисления коэффициентов Фурье	512
12.2.3. Спектральный анализ и синтез пилообразного импульса	513
12.2.4. Спектральный анализ и синтез меандра.....	514
12.2.5. Спектральный анализ и синтез короткого прямоугольного импульса ..	514
12.2.6. Спектральный анализ и синтез «выпрямленной» синусоиды	515
12.2.7. Спектральный анализ и синтез радиоимпульса	516
12.3. Эффект Гиббса и борьба с ним	517
12.3.1. Математическая природа эффекта Гиббса	517
12.3.2. Подавление эффекта Гиббса с помощью сигма-множителей.....	519
12.3.3. Подавление эффекта Гиббса с помощью множителей Фейера	519
12.3.4. Улучшенное моделирование сигналов с применением их интерполяции	520
12.4. Специальные приложения спектрального анализа и синтеза сигналов	522
12.4.1. Ограничения и недостатки преобразования Фурье	522
12.4.2. Кратковременное (оконное) преобразование Фурье	524
12.4.3. Фильтрация зашумленных аналоговых сигналов.....	525
12.4.4. Цифровая фильтрация с помощью БПФ	526
12.4.5. Фильтрация пространственного образа с применением БПФ	528
12.5. Инженерные методы спектрального анализа	529
12.5.1. Спектральный анализ методом Берга	529
12.5.2. Спектральный анализ методом пяти ординат	530
12.5.3. Спектральный анализ методом двенадцати ординат	532
12.6. Вейвлет-обработка сигналов	532
12.6.1. Пример вейвлет-разложения и реконструкции прямоугольного импульса.....	532
12.6.2. Пример вейвлет-разложения и реконструкции прямоугольного импульса с линейно нарастающей вершиной	535
12.6.3. Вейвлет-разложение и реконструкция радиоимпульса	536
12.6.4. Сравнение вейвлет- и фурье-анализа и синтеза	536
12.7. Обработка рисунков.....	538
12.7.1. Общие замечания по обработке рисунков.....	538
12.7.2. Обработка монохромных изображений.....	538
12.7.3. Обработка цветных изображений	539
12.7.4. Функции для работы с файлами и матрицами рисунков.....	540
12.7.5. Пример вейвлет-обработки и компрессии изображения	541
12.8. Основы применения операторного метода.....	543

12.8.1. Вводные замечания к операторному методу	543
12.8.2. Основные характеристики линейных систем	544
12.8.3. Программа построения диаграмм Боде	545
12.8.4. Построение переходной характеристики на основе БПФ	545
12.9. Примеры электро- и радиотехнических расчетов	548
12.9.1. Моделирование поля двух электрических зарядов	548
12.9.2. Расчет АЧХ и ФЧХ импеданса параллельного LCR-контура	549
12.9.3. Построение АЧХ и ФЧХ импеданса последовательного LCR-контура	550
12.9.4. Реакция резонансного контура на синусоидальное воздействие	551
12.9.5. Реакция LCR-контура на прямоугольные импульсы — меандр	552
12.9.6. Расчет микрополосковых линий	554
12.9.7. Расчет основных параметров каскодного усилителя	555
12.9.8. Расчет на постоянном токе двухтранзисторной цепи	558
12.9.9. Расчет универсального активного фильтра	561
12.9.10. Расчет нелинейной диодной цепи	562
12.10. Моделирование нелинейной цепи на туннельном диоде	563
12.10.1. Подготовка к моделированию цепи	563
12.10.2. Моделирование режима почти синусоидальных колебаний	564
12.10.3. Моделирование нелинейного усилителя	566
12.10.4. Моделирование ждущего релаксационного генератора	567
12.10.5. Моделирование триггера с отдельным запуском	567
12.11. О моделировании электронных устройств	568
Справочный раздел	570
Приложение 1. Сравнительные данные версий Mathcad 8 — Mathcad 12	570
Приложение 2. Ввод встроенных операторов	583
Приложение 3. Системные переменные Mathcad	587
Приложение 4. Встроенные функции и ключевые слова	588
Приложение 5. Сообщения о наиболее характерных ошибках	608
Список литературы	613

Серия «Библиотека студента»

Владимир Павлович Дьяконов

Mathcad 8—12 для студентов

Ответственный за выпуск

В. Митин

Макет и верстка

А. Иванова

Обложка

Е. Жбанов

ООО «СОЛОН-Пресс»

123242, г. Москва, а/я 20

Телефоны:

(095) 254-44-10, (095) 252-36-96, (095) 252-25-21

E-mail: Solon-R@coba.ru

По вопросам приобретения обращаться:

ООО «Альянс-книга»

Тел: (095) 258-91-94, 258-91-95, www.abook.ru

ООО «СОЛОН-Пресс»

127051, г. Москва, М. Сухаревская пл., д. 6, стр. 1 (пом. ТАРП ЦАО)

Формат 70×100/16. Объем 39,5 п. л. Тираж 1000 экз.

Отпечатано в ООО «Аделия»

142605, Московская обл., г. Орехово-Зуево,

ул. Красноармейская, д. 1

Заказ №