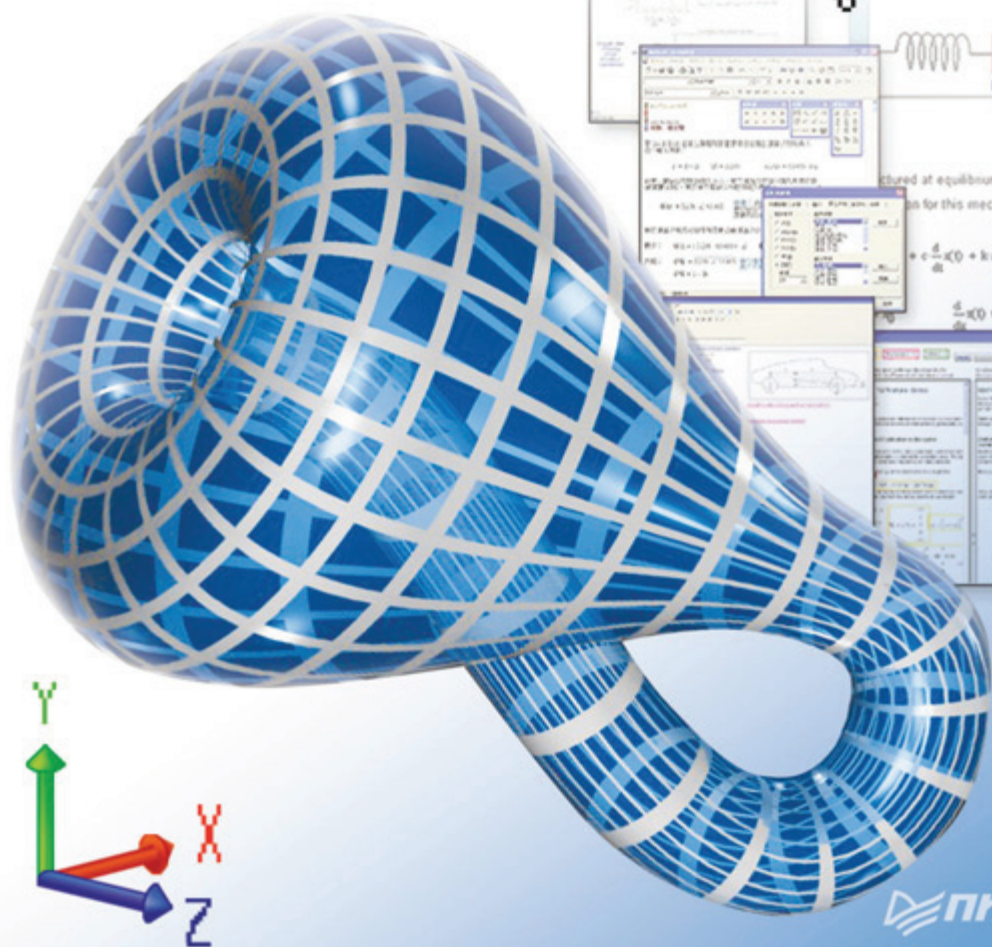


Евгений Макаров

ИНЖЕНЕРНЫЕ РАСЧЕТЫ В

Mathcad 15

УЧЕБНЫЙ КУРС



ПИТЕР®

ББК 004.42(075)
УДК 32.973.23-018.2я7
М15

Макаров Е.
М15 Инженерные расчеты в Mathcad 15: Учебный курс. — СПб.: Питер, 2011. — 400 с.: ил.

ISBN 978-5-459-00357-4

Книга предназначена для самостоятельного освоения работы в 15-й версии популярного математического пакета Mathcad. Издание имеет формат учебного курса и состоит из четырех частей. В первой части содержатся сведения, необходимые для начала работы в Mathcad и решения большинства практических задач. Вторая часть посвящена вычислительным возможностям программы, здесь на конкретных примерах рассматриваются особенности использования встроенных функций, основные приемы и способы вычислений. В третьей части рассматриваются возможности подготовки документов Mathcad, а четвертая содержит перечень примеров решения инженерных задач в Mathcad.

Важнейшее приложение книги размещено на интернет-сайте издательства — это электронная книга, встраиваемая в оболочку Mathcad. Она содержит примеры использования всех встроенных функций, программы решения 56 примеров, приведенных в четвертой части книги, и пояснения к ним. Электронная книга — это инструмент быстрого создания программ любой сложности.

Книга предназначена для всех пользователей Mathcad, как начинающих, так и опытных, и будет полезна студентам, инженерам и научным работникам.

ББК 004.42(075)
УДК 32.973.23-018.2я7

Все права защищены. Никакая часть данной книги не может быть воспроизведена в какой бы то ни было форме без письменного разрешения владельцев авторских прав.

Информация, содержащаяся в данной книге, получена из источников, рассматриваемых издательством как надежные. Тем не менее, имея в виду возможные человеческие или технические ошибки, издательство не может гарантировать абсолютную точность и полноту приводимых сведений и не несет ответственности за возможные ошибки, связанные с использованием книги.

Краткое оглавление

Предисловие	11
Электронная книга	17
Часть I. Начало работы с Mathcad	23
Глава 1. Построение выражений и графиков в Mathcad.	24
Глава 2. Некоторые возможности Mathcad	47
Часть II. Вычисления в Mathcad	85
Глава 3. Решение уравнений.	86
Глава 4. Работа с векторами и матрицами	105
Глава 5. Символьные вычисления	128
Глава 6. Решение дифференциальных уравнений	159
Глава 7. Обработка экспериментальных данных	192
Глава 8. Математическая статистика	217
Часть III. Дополнительные возможности Mathcad	225
Глава 9. Программирование	226
Глава 10. Анимация	255
Глава 11. Учет размерностей	261
Глава 12. Дополнительные встроенные функции Mathcad	275
Глава 13. Работа с Mathcad-документами.	300
Глава 14. Работа с текстом.	316
Глава 15. Работа с графиками	323
Глава 16. Справочная информация в Mathcad	345
Глава 17. Создание электронных книг	354
Глава 18. Работа с электронной книгой	365
Часть IV. Примеры инженерных расчетов в Mathcad	373

Оглавление

Предисловие	11
Mathcad и его возможности	11
Новое в Mathcad 15	13
Рекомендации по работе с книгой	15
От издательства	16
Электронная книга	17
ReadMe	17
Сведения об электронной книге	19
Назначение электронной книги.	21
Часть I. Начало работы с Mathcad	23
Глава 1. Построение выражений и графиков в Mathcad	24
1.1. Интерфейс Mathcad	24
1.1.1. Рабочее окно Mathcad	24
1.1.2. Главное меню	25
1.1.3. Панели инструментов	26
1.2. Построение выражений и их вычисление	30
1.3. «Цепкие» операторы	31
1.4. Редактирование объектов Mathcad	32
1.5. Стандартные функции	33
1.6. Числовые константы. Ввод греческих букв	34
1.7. Ввод текста	35
1.8. Функции пользователя	36
1.9. Дискретные переменные. Построение таблиц	37
1.10. Форматирование чисел	38
1.11. Построение плоского графика функции	39
1.12. Построение трехмерных графиков	44
Глава 2. Некоторые возможности Mathcad	47
2.1. Ступенчатые и разрывные функции и выражения. Условие в Mathcad.	47
2.2. Глобальное присвоение значений.	49
2.3. Символьные вычисления	50
2.4. Решение уравнений	52
2.4.1. Символьное решение	52
2.4.2. Численное решение (функция root)	52
2.5. Решение систем уравнений (функция find)	54
2.6. Приближенное решение систем уравнений (функция minerr)	56
2.7. Исследование функции на экстремум	57
2.8. Работа с матрицами	58
2.8.1. Создание матриц	58
2.8.2. Основные действия с матрицами	59
2.8.3. Решение матричных уравнений	60
2.9. Оператор векторизации.	61
2.10. Решение дифференциальных уравнений	63
2.11. Анализ экспериментальных данных.	67
2.11.1. Интерполяция	68
2.11.2. Регрессия	68

2.12. Элементы математической статистики	73
2.13. Обмен данными с другими программами	74
2.14. Учет размерностей в Mathcad	75
2.15. Преобразование функции в матрицу и матрицы в функцию.	79
2.16. Строковые функции	80
2.17. Программирование	80
2.18. Анимация	81
2.19. Отладка Mathcad-документов	81
2.20. «Горячие» клавиши	82

Часть II. Вычисления в Mathcad 85

Глава 3. Решение уравнений 86

3.1. Функция root	86
3.2. Решение уравнения с переменными параметрами	89
3.3. Нахождение корней полинома. Функция polyroots	91
3.4. Решение систем уравнений	93
3.5. Трудности при решении уравнений	98
3.6. Приближенное решение уравнений и систем уравнений	98
3.7. Исследование функции на экстремум.	100
3.8. Рекурсивные вычисления	102

Глава 4. Работа с векторами и матрицами 105

4.1. Создание матрицы	105
4.2. Нумерация элементов матрицы	107
4.3. Встроенная переменная ORIGIN	107
4.4. Определение параметров матрицы	108
4.5. Образование новых матриц из уже существующих.	109
4.6. Ввод больших массивов	110
4.7. Сортировка векторов и матриц	111
4.8. Матричные операторы.	111
4.8.1. Транспонирование матрицы	111
4.8.2. Вычисление определителя матрицы	112
4.8.3. Нахождение матрицы, обратной заданной	113
4.8.4. Сложение, вычитание и умножение матриц	113
4.8.5. Оператор векторизации.	115
4.8.6. Последовательное и параллельное вычисление элементов массива	116
4.9. Собственные числа и собственные векторы матриц	116
4.10. Норма квадратной матрицы	119
4.11. Число обусловленности квадратной матрицы	120
4.12. Матричные разложения	120
4.12.1. Разложение Холецкого	121
4.12.2. QR-разложение	121
4.12.3. LU-разложение.	121
4.12.4. Сингулярное разложение	122
4.13. Вывод векторов и матриц	122
4.14. Составные массивы.	124
4.15. Работа с комплексными числами	125

Глава 5. Символьные вычисления 128

5.1. Оптимизация численных вычислений	128
5.2. Использование меню символьных вычислений Symbolics	129
5.2.1. Символьная оценка выражения.	130

5.2.2. Упрощение выражений	131
5.2.3. Раскрытие выражений	132
5.2.4. Разложение на множители	133
5.2.5. Приведение подобных слагаемых.	133
5.2.6. Коэффициенты полинома.	134
5.2.7. Вычисление пределов.	134
5.2.8. Замена переменной	135
5.2.9. Вычисление производной или интеграла.	135
5.2.10. Символьное решение уравнений и неравенств	137
5.2.11. Разложение в ряд Тейлора	137
5.2.12. Разложение на простейшие дроби.	138
5.2.13. Символьные операции с матрицами.	138
5.2.14. Вычисление интегральных преобразований.	140
5.3. Использование символьной панели инструментов	143
5.3.1. Символьные операции без использования ключевых слов	144
5.3.2. Символьные операции с использованием ключевых слов	144
5.3.3. Символьное решение уравнений	148
5.3.4. Использование нескольких команд одновременно.	148
5.3.5. Символьные операции с матрицами	150
5.4. Новые возможности символьных вычислений в Mathcad 13–15.	151
5.4.1. Новые ключевые слова	151
5.4.2. Новые слова-модификаторы	153
5.4.3. Символьная векторизация выражений	155
5.4.4. Новые функции символьных вычислений	155
5.5. Встроенные функции и переменные	156
5.6. Особенности символьной обработки	156
Глава 6. Решение дифференциальных уравнений	159
6.1. Вычислительный блок Given–Odesolve.	159
6.1.1. Решение ОДУ с начальными и граничными условиями.	160
6.1.2. Объединение ОДУ с помощью операторов условия	160
6.1.3. Решение ОДУ, заданных неявно	162
6.1.4. Решение систем ОДУ.	162
6.1.5. Работа блока Given–Odesolve.	163
6.1.6. Новые функции решения ОДУ в Mathcad 14	165
6.1.7. Решение ОДУ с переменными параметрами	167
6.2. Использование функций rkfixed, Rkadapt, Bulstoer.	167
6.2.1. Решение дифференциального уравнения первого порядка	169
6.2.2. Дифференциальные уравнения более высокого порядка	170
6.2.3. Системы дифференциальных уравнений	172
6.3. Жесткие системы дифференциальных уравнений	173
6.3.1. Функция Radau в Mathcad 14	177
6.4. Нахождение решения только в конечной точке	178
6.5. Двухточечные краевые задачи.	179
6.6. Решение дифференциальных уравнений в частных производных	184
6.6.1. Функции решения параболических и гиперболических уравнений	184
6.6.2. Решение эллиптических уравнений (Лапласа и Пуассона)	188
Глава 7. Обработка экспериментальных данных	192
7.1. Интерполяция	192
7.1.1. Линейная интерполяция	192
7.1.2. Кубическая сплайн-интерполяция	193

7.1.3. В-сплайн-интерполяция	194
7.1.4. Линейное предсказание	196
7.1.5. Интерполяция функции двух переменных	197
7.2. Функции регрессии	200
7.2.1. Одномерная регрессия	201
7.2.2. Двухмерная регрессия	203
7.2.3. Обобщенная регрессия	204
7.3. Функции сглаживания	208
7.4. Дискретные преобразования.	210
7.4.1. Преобразование Фурье	210
7.4.2. Двухмерное преобразование Фурье	213
7.4.3. Волновое преобразование.	215
Глава 8. Математическая статистика	217
8.1. Характеристики выборки данных и связи двух массивов	217
8.2. Функции распределения вероятностей	219
8.3. Генераторы случайных чисел	221
8.4. Построение гистограмм	222
8.5. Расчет доверительного интервала	224
Часть III. Дополнительные возможности Mathcad	225
Глава 9. Программирование	226
9.1. Создание программ	226
9.1.1. Ввод строк в программу	226
9.1.2. Локальное присвоение значений	227
9.2. Условный оператор if	229
9.3. Операторы цикла	231
9.3.1. Оператор while	231
9.3.2. Оператор for	232
9.3.3. Операторы break, continue, return	234
9.4. Вывод результатов расчета из программы	235
9.5. Подпрограммы-функции	237
9.6. Интегрирование функции методом трапеций.	238
9.7. Решение уравнений и систем уравнений в программе	239
9.8. Нахождение экстремума функции в программе	240
9.9. Учет размерностей в программе.	242
9.10. Решение дифференциальных уравнений внутри программы	244
9.11. Отладка программ	247
9.12. Возможности ускорения работы Mathcad	252
Глава 10. Анимация	255
10.1. Создание анимации.	255
10.2. Галерея анимационных клипов	259
Глава 11. Учет размерностей	261
11.1. Выбор системы единиц измерения.	261
11.2. Ввод пользовательских единиц измерения	263
11.3. Изменение масштаба результатов расчета	264
11.4. Учет внесистемных размерностей	267
11.5. Размерность элементов массива	267
11.6. Учет размерностей в цикле.	269

11.7. Построение графиков с учетом размерностей	269
11.8. Учет размерностей в программе	270
11.9. Ограничения на проведение расчетов с учетом размерностей	270
11.10. Статическая проверка размерностей	271
11.11. Квазиразмерности	272
Глава 12. Дополнительные встроенные функции Mathcad	275
12.1. Функции для работы с комплексными числами.	275
12.2. Тип выражения	276
12.3. Доступ к файлам	277
12.4. Функции поиска значений матрицы	280
12.5. Преобразование функций и матриц	282
12.5.1. Преобразование функции в матрицу	282
12.5.2. Преобразование матрицы в функцию	283
12.6. Функции преобразования координат	286
12.7. Функции условия	290
12.8. Функции округления чисел	294
12.9. Функции комбинаторики и теории чисел	294
12.10. Функции тригонометрические, гиперболические, логарифмические и экспоненциальные.	294
12.11. Специальные функции	295
12.11.1. Функции Бесселя.	295
12.11.2. Другие специальные функции	297
12.11.3. Дополнительные неактивные функции	297
12.12. Строковые функции.	298
Глава 13. Работа с Mathcad-документами	300
13.1. Построение математических выражений	300
13.1.1. Стили записи выражений	300
13.2. Аннотация файлов и областей файлов.	305
13.3. Защита информации в Mathcad	307
13.4. Закрытые зоны в Mathcad	309
13.5. Создание вычислительных комплексов	310
13.5.1. Использование ссылок на другие файлы	311
13.5.2. Использование гиперссылок.	312
13.6. Новые служебные функции Mathcad 13.	314
13.6.1. Уменьшение размера файлов	314
13.6.2. Переключатель версий Mathcad	315
13.6.3. Автосохранение	315
Глава 14. Работа с текстом	316
14.1. Изменение стиля текстовых областей	316
14.1.1. Установка свойств абзаца	316
14.1.2. Изменение стиля отдельной текстовой области	318
14.1.3. Списки и таблицы в тексте	319
14.2. Использование гиперссылок	320
14.2.1. Переход на метку	320
14.2.2. Всплывающий документ.	322
Глава 15. Работа с графиками	323
15.1. Построение двухмерного плоского графика	323
15.2. Построение полярных графиков	327

15.3. Построение трехмерных графиков	328
15.3.1. Построение графика по массиву значений функции	328
15.3.2. Быстрое построение графика	329
15.3.3. Создание массива значений функции	332
15.3.4. Поверхности тел вращения	334
15.3.5. Построение сложных фигур	336
15.3.6. Построение многогранников.	338
15.3.7. Быстрое изменение типа графика	339
15.3.8. График векторного поля.	340
15.4. Особенности форматирования графиков в новых версиях Mathcad	340
15.4.1. Редактирование двухмерных графиков	340
15.4.2. Форматирование двухмерных графиков	342
15.4.3. Функции logspace и logpts.	343
Глава 16. Справочная информация в Mathcad	345
16.1. Меню Help (Помощь)	345
16.2. Перевод справочной системы Mathcad	347
16.3. Электронная книга qscheet.	348
16.3.1. Обзор и учебник	350
16.3.2. Примеры и справочные таблицы	350
16.4. Mathcad и Интернет	352
16.4.1. Техническая поддержка Mathcad	353
Глава 17. Создание электронных книг	354
17.1. Создание Mathcad-файлов для электронной книги	355
17.2. Подготовка содержания электронной книги.	356
17.3. Создание системы ссылок между файлами	357
17.4. Создание НВК-файла	357
17.5. Вложенные папки	359
17.6. Проверка электронной книги	360
17.7. Создание поисковых индексов.	361
Глава 18. Работа с электронной книгой	365
18.1. Обработка экспериментальных данных	365
18.2. Обработка цифрового сигнала	368
18.3. Оптимизация нагрузки на балку.	369
Часть IV. Примеры инженерных расчетов в Mathcad	373
Пример 1. Проверка статистических гипотез	375
Пример 2. Расчет электрической цепи переменного тока.	375
Пример 3. Переходные процессы при замыкании-размыкании электрической цепи.	376
Пример 4. Кинематический анализ кулачкового механизма	376
Пример 5. Движение двух машин по мосту.	377
Пример 6. Расчет моментов инерции сечения, заданного аналитически	378
Пример 7. Расчет моментов инерции сечения, заданного набором точек на контуре	378
Пример 8. Расчет геометрических характеристик составного сечения	378
Пример 9. Определение внутренних усилий при растяжении стержня	379
Пример 10. Определение внутренних усилий при кручении стержня	379
Пример 11. Определение внутренних усилий при изгибе консольной балки	380
Пример 12. Определение внутренних усилий при изгибе балки на двух опорах	380

Пример 13. Растяжение статически-определимого стержня	381
Пример 14. Растяжение статически-неопределимого стержня	382
Пример 15. Растяжение-сжатие статически неопределимой шарнирно-стержневой системы	382
Пример 16. Расчет простейших соединений	382
Пример 17. Расчет статически определимого стержня на кручение	382
Пример 18. Расчет статически неопределимого стержня на кручение	383
Пример 19. Расчет на изгиб консольной балки	383
Пример 20. Расчет на изгиб балки на двух опорах	384
Пример 21. Расчет на изгиб балок на упругом основании	384
Пример 22. Учет пластических деформаций при растяжении-сжатии	385
Пример 23. Учет упругопластических деформаций при изгибе	385
Пример 24. Учет пластических деформаций при кручении	385
Пример 25. Расчет прямого стержня на сложное сопротивление	385
Пример 26. Расчет пространственного стержня на сложное сопротивление	386
Пример 27. Определение перемещений систем из нескольких элементов	387
Пример 28. Раскрытие статической неопределимости плоской рамы	387
Пример 29. Расчет один раз статически неопределимой системы	388
Пример 30. Расчет пространственной статически неопределимой рамы	388
Пример 31. Расчет многопролетной балки с одной заделкой на изгиб	389
Пример 32. Расчет многопролетной балки на изгиб	389
Пример 33. Расчет на устойчивость сжатого стержня	389
Пример 34. Определение критических нагрузок на сжатый стержень энергетическим методом	390
Пример 35. Продольно-поперечный изгиб балок	390
Пример 36. Расчет консольной балки на удар и колебания	391
Пример 37. Расчет балки на двух опорах на удар и колебания	391
Пример 38. Расчет сложной системы на удар и устойчивость	391
Пример 39. Статистическая обработка результатов эксперимента	392
Пример 40. Расчет болта на усталостную прочность	392
Пример 41. Расчет долговечности детали при нестационарном нагружении	392
Пример 42. Расчет долговечности полосы с трещиной	393
Пример 43. Построение диаграммы истинных напряжений	393
Пример 44. Расчет тонкостенной оболочки	394
Пример 45. Оптимизация расчета кронштейна	395
Пример 46. Оптимальный расчет сферической крышки	395
Пример 47. Подбор нагрузок по перемещениям балки	396
Пример 48. Подбор нагрузок на балку, удовлетворяющих граничным условиям	396
Пример 49. Поиск оптимальных размеров сечения балки	396
Пример 50. Динамический расчет балки матричным методом перемещений	396
Пример 51. Динамический расчет плоской рамы методом конечных элементов	397
Пример 52. Расчет пространственной стержневой системы	397
Пример 53. Расчет кольца методом конечных элементов	397
Пример 54. Решение уравнения Пуассона при кручении стержня	398
Пример 55. Решение плоской упругопластической задачи методом конечных элементов	398
Пример 56. Использование прямоугольного квадратичного элемента в методе конечных элементов	399

Предисловие

Основное назначение этой книги — помочь пользователям быстро создавать свои расчетные программы любой сложности. Для удобства читателей книга разбита на 4 части.

Часть I предназначена для тех, кто впервые сталкивается с Mathcad. Здесь содержатся основные сведения, необходимые и достаточные для решения большинства задач.

В части II содержатся подробные сведения обо всех возможностях Mathcad, на конкретных примерах рассматриваются особенности использования встроенных функций, основные приемы и способы вычислений.

В части III рассматриваются возможности подготовки документов Mathcad.

Часть IV книги содержит 56 примеров решения инженерных задач в Mathcad.

ВНИМАНИЕ

Неотъемлемой частью книги является электронная книга, встраиваемая в оболочку Mathcad, которую можно скачать с сайта издательства www.piter.com или с сайта автора книги www.emakarov.nsknet.ru. В электронной книге вы найдете Mathcad-программы использования всех встроенных функций и всех описанных в книге примеров.

При написании книги я отошел от традиционного описания меню и панелей. Интерфейс Mathcad ничем не отличается от традиционных Windows-приложений. Знакомство с меню происходит непосредственно во время работы. Пояснения к отдельным пунктам меню появляются в тексте по мере необходимости.

Mathcad и его возможности

Mathcad создан разработчиками как инструмент работы расчетчиков-инженеров. Он не предназначен для профессиональных математиков — для них есть другие системы, действующие в областях символьной математики и математической статистики.

Время от времени выходят новые версии Mathcad, но они почти не добавляют новых функций, новых расчетных возможностей. Зато добавляют новые ограничения, новые сложности в использовании Mathcad.

На мой взгляд, лучшей версией был Mathcad 11. Последующие программы в связи с переходом на Net-технологии стали гораздо более медленными, и дополнились множеством ограничений. Последние версии Mathcad, 14 и 15, по своим вычислительным возможностям абсолютно одинаковы. Особенно приятно, что они «русифицированы» — вы увидите меню на русском языке. В Mathcad 15 даже включено руководство пользователя на русском языке, но все статьи в разделе Help (Помощь) по-прежнему написаны на английском языке.

На одном компьютере одновременно могут работать две версии программы: Mathcad 11 и Mathcad 14 (или 15), Mathcad 13 и Mathcad 15. Но обратите внимание на то, что версии Mathcad 14 и Mathcad 15 не будут работать вместе.

ВНИМАНИЕ

При установке двух версий на компьютер вначале ставится более старая версия, например Mathcad 11, а затем новая. В противном случае со сбоями будут работать обе версии.

Mathcad в современном виде не предназначен и для программирования сложных задач — для этого есть система Matlab, языки программирования Fortran и C++. Он создавался как мощный микрокалькулятор, позволяющий легко справляться с рутинными задачами инженерной практики, ежедневно встречающимися в работе: решение алгебраических или дифференциальных уравнений с постоянными и переменными параметрами; анализ функций, поиск их экстремумов; численное и аналитическое дифференцирование и интегрирование; вывод таблиц и графиков при анализе найденных решений.

Главными достоинствами Mathcad и его колоссальным преимуществом перед другими расчетными средствами являются легкость и наглядность программирования задачи, отображение сложных математических выражений в том виде, в каком они обычно записываются на листе бумаги, то есть отсутствие специального языка программирования, простота использования, возможность создания средствами Mathcad высококачественных технических отчетов с таблицами, графиками и текстом.

Mathcad завоевал популярность во всем мире. Им пользуются в работе свыше 0,5 млн человек. Уже в первых версиях Mathcad появилась дискретная переменная, позволяющая одновременно находить значения функций для целого ряда значений переменной и на основе полученных результатов строить таблицы и графики без применения операторов программирования. При построении графиков поверхностей можно создавать почти произведения искусства. Доведена до логического завершения великолепная функция решения дифференциальных уравнений *Odesolve*.

На этом фоне в полном забвении находится раздел программирования в Mathcad. По-видимому, этот раздел изначально был задуман для создания несложных программных модулей, необходимых для многократного вычисления небольших расчетных блоков из нескольких операторов.

На основании личного опыта я хочу в этой книге показать, что даже в таком в общем-то зачаточном состоянии программирования Mathcad позволяет решать задачи абсолютно любой сложности.

Кроме мелких недоработок в Mathcad есть два существенных недостатка, ограничивающих его применение в качестве средства программирования:

1. Отсутствие встроенных средств отладки программ, которые есть в других языках программирования.
2. Недостаточная скорость расчетов.

ПРИМЕЧАНИЕ

В Mathcad 13 появились средства отладки программ: панель Debug, функции trace и pause, — но реальной помощи в отладке программ они не оказывают.

Некоторые рекомендации по ускорению расчетов в Mathcad приведены в главе 9. На протяжении ряда лет я использую Mathcad для проведения сложных инженерных расчетов. Я преподаю студентам ряд дисциплин механики с решением всех задач в Mathcad. В математически сложных курсах, таких как теория пластичности, численное решение любой, даже самой простой, задачи можно получить, только решив ее на компьютере. Удобнее всего это делать в Mathcad.

Главные достоинства Mathcad — **простота и удобство** использования — практически исчезают при составлении сравнительно сложных программ. Сплошь и рядом возникают ситуации, когда Mathcad оказывается не в состоянии решить элементарные примеры. Вроде все просто и записано верно, а расчет не получается. После длительных размышлений ошибка всегда обнаруживается и чаще всего оказывается примитивной, но на отладку документа затрачивается масса времени.

Пользователи Mathcad — студенты и инженеры — не могут позволить себе несколько дней корпеть над отладкой программ. Выходом из положения становится **сборка собственных программ из готовых и отлаженных фрагментов**. В этом вам поможет электронная книга (скачайте ее с сайта издательства www.piter.com).

Система вложенных оглавлений со ссылками позволяет практически мгновенно найти нужный фрагмент, скопировать его в буфер обмена и вставить в свой рабочий документ. Остается подправить вставленные фрагменты в соответствии со своими исходными данными и обозначениями, связать их между собой по смыслу — и документ готов. Такой метод «электронных кубиков» позволяет многократно уменьшить затраты времени на создание и отладку расчетных программ, возвращая Mathcad простоту и удобство использования.

Новое в Mathcad 15

После того как в 2006 году фирма PTC приобрела фирму Mathsoft, изменилось стратегическое направление развития Mathcad.

Mathcad не является главным продуктом фирмы PTC. Основные продукты фирмы — пакеты Pro/ENGINEER, Windchill и Arbortext, предназначенные для автоматического проектирования конструкций и сопровождения проектов, создания технической документации, обмена информацией и объединения различных проектов.

Фирма PTC сразу занялась интеграцией своих продуктов с Mathcad, оставив без внимания математическое развитие пакета Mathcad, и решила создать принципиально новый пакет Mathcad Prime, сильно отличающийся от традиционного прежде всего необычным интерфейсом и набором функций. Первый выпуск Mathcad Prime вызвал резко отрицательный отклик пользователей. Тогда PTC решила на время вернуться к старому варианту Mathcad, выпустив в 2010 году

Mathcad 15, не добавив в него практически ничего нового по сравнению с Mathcad 14.

Вот что сказано о Mathcad 15 в рекламном проспекте фирмы PTC:

Описание: Mathcad® 15.0 включает в себя 25 функциональных обновлений, улучшенные расчетные библиотеки и расширенную интеграцию с решениями сторонних производителей, например с последней версией Microsoft Excel. Традиционно Mathcad® 15.0 превосходно поддерживает работу систем автоматизированного проектирования, например Pro/ENGINEER®, а также программное решение для управления данными от PTC — Windchill® для крупных компаний и такое же решение для сектора среднего и малого бизнеса — Windchill ProductPoint®. Что позволяет пользователям лучше управлять критически важными инженерными данными, упрощает возможность взаимодействия команды разработчиков и повторного использования информации об изделии.

Новый Mathcad 15 — это

- функции для расчетов по планированию экспериментов (design of experiments (DoE)) — добавлено 25 новых функций, помогающих сократить время, затраченное на натурные эксперименты, за счет понимания общей тенденции при испытаниях. Приложения DoE помогают найти критические факторы и оптимальные условия при испытаниях сложных процессов. Содержатся шаблоны для проведения нескольких экспериментов, при наличии нескольких уровней эксперимента (режимов тестирования) и различных условий;
- углубленная интеграция с базой данных KnovelMath — быстрый доступ к базе по инженерным и техническим стандартам сокращает время проведения сложных расчетов;
- интеграция с программным обеспечением Kornucopia, выпускаемым компанией Bodie Technology, предназначенным для сокращения времени и усилий, затрачиваемых на проведение анализа. Предоставляя шаблоны для расчетов в Mathcad, этот модуль позволяет использовать наработанные процессы для всеобъемлющей оценки данных натурных экспериментов и результатов расчетов;
- первая интеграция с базой данных Truenumbers — Truenumbers от True Engineering Technology предоставляет разработчикам доступ к различным справочным материалам и данным. Результаты из Mathcad просто передаются в различные форматы документов, что значительно облегчает передачу данных по ключевой цепочке сотрудников;
- поддержка операционной системы Microsoft Windows 7;
- поддержка Microsoft Excel 2007.

С точки зрения инженерных расчетов новые функции планирования эксперимента нужны лишь узким специалистам и не представляют интереса для большинства пользователей Mathcad. Чтобы ознакомиться с новыми функциями, в стандартном меню Mathcad выберите Вставка функций ▶ План эксперимента (рис. 0.1).

Интеграция с базами данных, анонсированная в рекламном проекте, представляет собой выход на сайты, содержащие эти базы данных. Выйти на них можно просто из оболочки Mathcad (рис. 0.2), выбрав в стандартном меню Mathcad Инструменты ▶ Knovel, Kornucopia или Truenumbers.

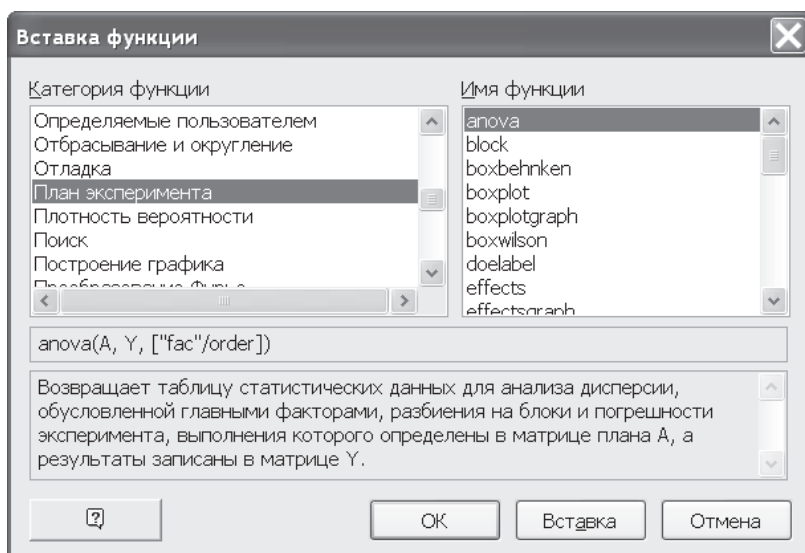


Рис. 0.1. Функции планирования эксперимента в Mathcad 15

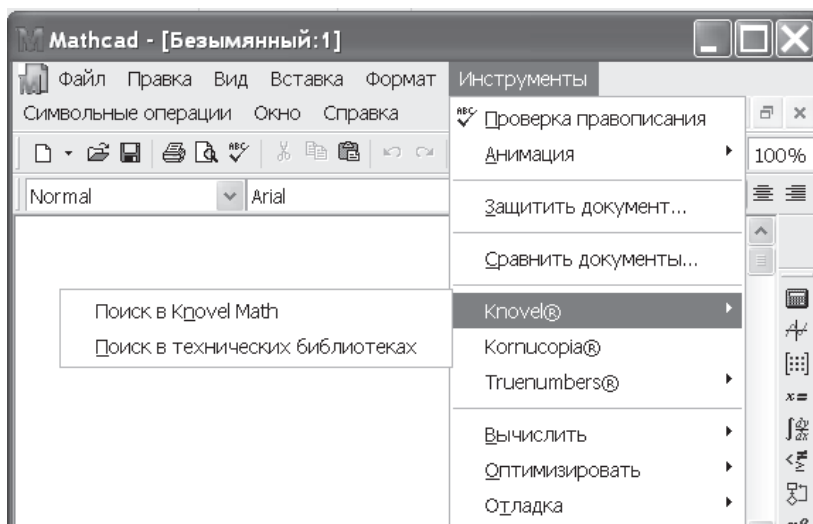


Рис. 0.2. Интеграция с новыми базами данных в Mathcad 15

Других новшеств в Mathcad 15 нет, к тому же сам Mathcad 15 идентифицирует себя как Mathcad 14.

Рекомендации по работе с книгой

Изучать Mathcad, только читая книгу, бесполезно. Осваивать Mathcad надо, сидя за компьютером. Рекомендую следующий стиль работы.

Прочитайте первую часть книги, затем садитесь за компьютер и шаг за шагом осваивайте основные приемы работы в Mathcad. Сведений, содержащихся в первой части, достаточно для выполнения подавляющего большинства расчетов. Внимательно изучите их и приступайте к изучению Mathcad.

Остальные главы пока полистайте, чтобы иметь общее представление, о чем там идет речь. При решении конкретных задач, когда сведений, приведенных в двух главах первой части, становится недостаточно, обращайтесь к последующим главам или поищите похожую задачу (или кусок задачи) в электронной книге.

Нужный вам фрагмент программы скопируйте из электронной книги, вставьте в свой расчет и подправьте в соответствии с вашими потребностями.

В Mathcad все операции можно выполнить, нажав соответствующую кнопку на математической панели или клавишу на клавиатуре. Рекомендуется запомнить наиболее часто употребляемые клавиши («горячие» клавиши), чтобы не обращаться постоянно к математической панели. Это сэкономит вам много времени при наборе программы. Сочетание клавиш для выполнения какой-либо операции появляется на экране рядом с курсором при наведении курсора на значок данной операции в математическом меню.

Отзывы о данной книге прошу оставлять на сайте автора www.emakarov.nsknet.ru или присылать по электронной почте emakarov@inbox.ru.

От издательства

Ваши замечания, предложения и вопросы отправляйте по адресу электронной почты comp@piter.com (издательство «Питер», компьютерная редакция). Мы будем рады узнать ваше мнение! Подробную информацию о наших книгах вы найдете на веб-сайте издательства www.piter.com.

Электронная книга

Электронная книга, которую можно скачать с сайта издательства www.piter.com или с сайта автора книги www.emakarov.nsknet.ru, является важнейшим дополнением книги печатной.

Скачав папку «Электронная книга», прежде всего прочитайте файл **ReadMe**. Здесь вы найдете сведения об установке электронной книги. Ввиду важности этой процедуры продублируем файл **ReadMe** полностью.

ReadMe

В данной папке находятся:

1. Файл **ReadMe** — сведения об установке электронной книги в оболочку Mathcad.
2. Электронная книга **Расчеты в Mathcad**, встраиваемая в оболочку Mathcad, которая содержит свыше ста Mathcad-файлов. В электронной книге приведены все примеры, упоминаемые на страницах печатного издания, и 56 программ решения задач в Mathcad,
3. **Примеры инженерных расчетов в Mathcad** — подробные пояснения (в формате *pdf*) ко всем 56 примерам, приведенным в электронной книге.
4. Сведения об авторе книги (в формате *pdf*).

В папке находятся две одинаковые электронные книги, сохраненные в разных версиях Mathcad.

- 1) **Mathcad 11\Primer** — файлы с расширением *mcd*, сохранены в формате версии Mathcad 2000 и работают во всех версиях, включая Mathcad 15.
- 2) **Mathcad14\Primer-xmcd** — файлы с расширением *xmcd*, сохранены в формате версии Mathcad 14 и могут быть открыты в Mathcad 14 или 15.

В Mathcad 13 или 14 при открытии файлов с расширением *mcd* у некоторых пользователей не отображается русский шрифт (эта проблема связана не с дефектом электронной книги, а с операционной системой Windows). При запуске файлов с расширением *xmcd* русский текст отображается нормально.

Резюме: Для работы в Mathcad 14 или 15 установите электронную книгу **Primer-xmcd**. Для работы в старых версиях Mathcad, включая Mathcad 13, установите электронную книгу **Primer** с расширением файлов *mcd*.

Установка электронной книги

□ Для установки электронной книги надо скопировать папку **primer** и файл **primer.hbk** в подкаталог установки

Mathcad_XXX\Handbook (XXX — номер версии Mathcad).

В Mathcad 12 и 13 подкаталог **Handbook** отсутствует. Создайте его сами.

В ранних версиях Mathcad скопируйте каталог **primer** и файл **primert.hbk** в корневой каталог Mathcad.

- ❑ После копирования указанных файлов при запуске Mathcad электронная книга появляется в стандартном меню.
- ❑ Для открытия электронной книги:
 - в стандартной версии Mathcad выберите меню Help ► E-books ► Расчеты в Mathcad;
 - в ранних версиях Mathcad выберите меню Help ► E-books ► Open book и укажите путь к файлу primer.hbk.

В Mathcad 12 не работают большинство программ, созданных в предыдущих версиях.

В Mathcad 13–15 практически все программы работают правильно, за исключением нескольких операторов.

В Mathcad 11 все программы работают идеально.

В более ранних версиях проблем не возникает, если в программе не используются функции, добавленные в более поздних версиях.

Для работы с русским текстом в Mathcad 11 и более ранних версиях скопируйте файл Normal.mct в каталог Mathcad/Template, заменив стоящий там файл.

В Mathcad 12–15 скопируйте файл Normal.xmct в указанный выше каталог.

После этого при запуске Mathcad будет открываться русифицированный шаблон документа.

Если в электронной книге вместо русского текста напечатана абракадабра, запустите файл RUSFONTS.exe из папки Русский шрифт и перезагрузитесь.

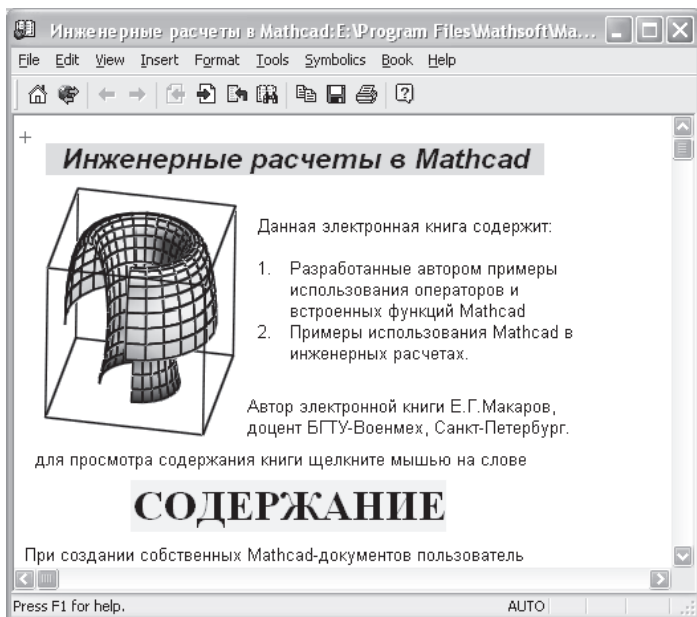


Рис. К.1. Титульная страница электронной книги

Сведения об электронной книге

При выборе в стандартном меню Mathcad Help /E-books / Расчеты в Mathcad на экране монитора появляется титульная страница электронной книги Расчеты в Mathcad, показанная на рис. К.1.

В электронной книге можно найти все примеры, представленные на страницах печатного издания. Основное оглавление электронной книги содержит названия глав (рис. К.2). Щелчок левой кнопкой мыши на названии главы вызывает открытие Mathcad-файла данной главы.

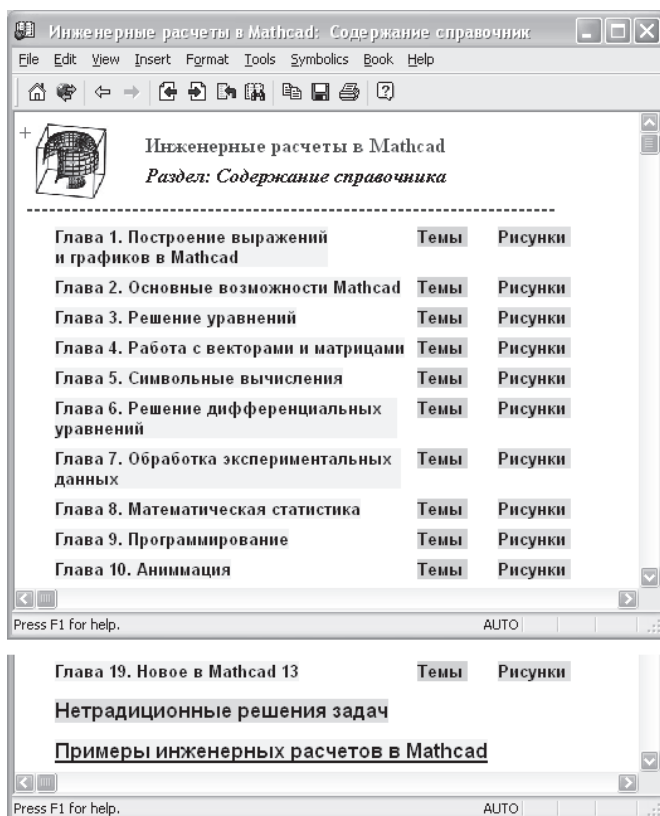


Рис. К.2. Основное оглавление электронной книги (начало и конец)

Рядом с названием главы расположены две кнопки, **Темы** и **Рисунки**. Выбор одной из них вызывает появление вложенного оглавления с перечнем тем, рассмотренных в этой главе (рис. К.3), или списка рисунков этой главы. Каждый заголовок связан ссылкой с соответствующим фрагментом Mathcad-документа.

В нижней части рис. К.2 показаны последние строки основного оглавления электронной книги. Обратите внимание на два важных раздела:

- Нетрадиционные решения задач;
- Примеры инженерных расчетов.

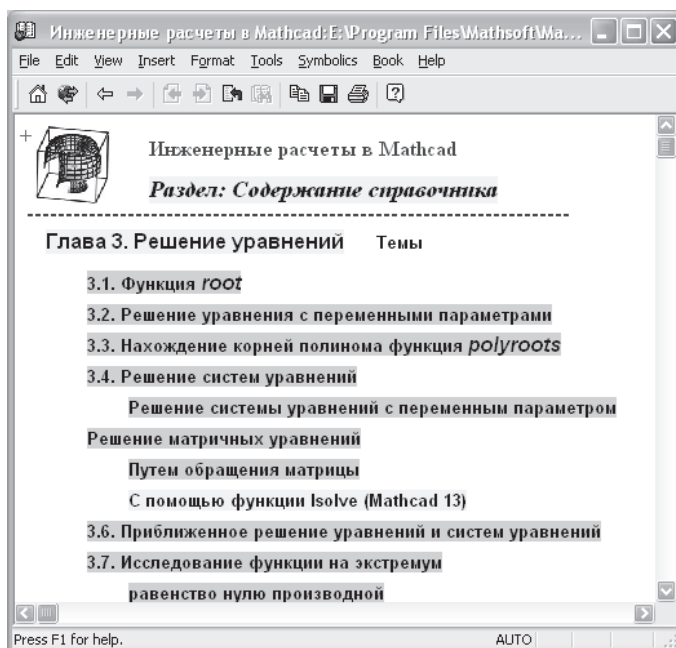


Рис. К.3. Оглавление Темы для главы 3

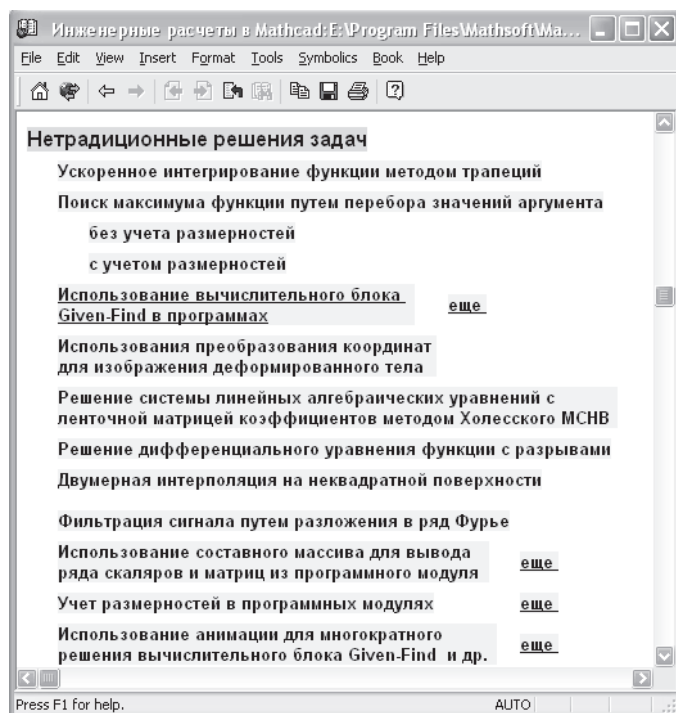


Рис. К.4. Авторские «изюминки» в электронной книге

Раздел Примеры инженерных расчетов содержит ссылки на 56 примеров, размещенных в электронной книге.

Раздел Нетрадиционные решения задач представляет собой перечень оригинальных авторских решений (со ссылками на Mathcad-файлы), позволяющих упростить и ускорить решение особо сложных задач (рис. К.4).

Система вложенных оглавлений позволяет быстро найти интересующий вас фрагмент электронной книги, скопировать его в буфер обмена и вставить в свой рабочий документ.

Подробные пояснения ко всем 56 примерам инженерных расчетов, приведенным в электронной книге, можно найти в файле Примеры-все.pdf (Примеры инженерных расчетов в Mathcad). Для его открытия на компьютере пользователя должна быть установлена программа Acrobat Reader, FoxitReader или STDUViewer. Одна из страниц этого файла показана на рис. К.5. В левой части окна программы выберите пункт Закладки (Bookmarks), чтобы увидеть перечень всех примеров. Щелчок левой кнопкой мыши на названии любого примера вызывает появление соответствующей страницы документа на экране компьютера.

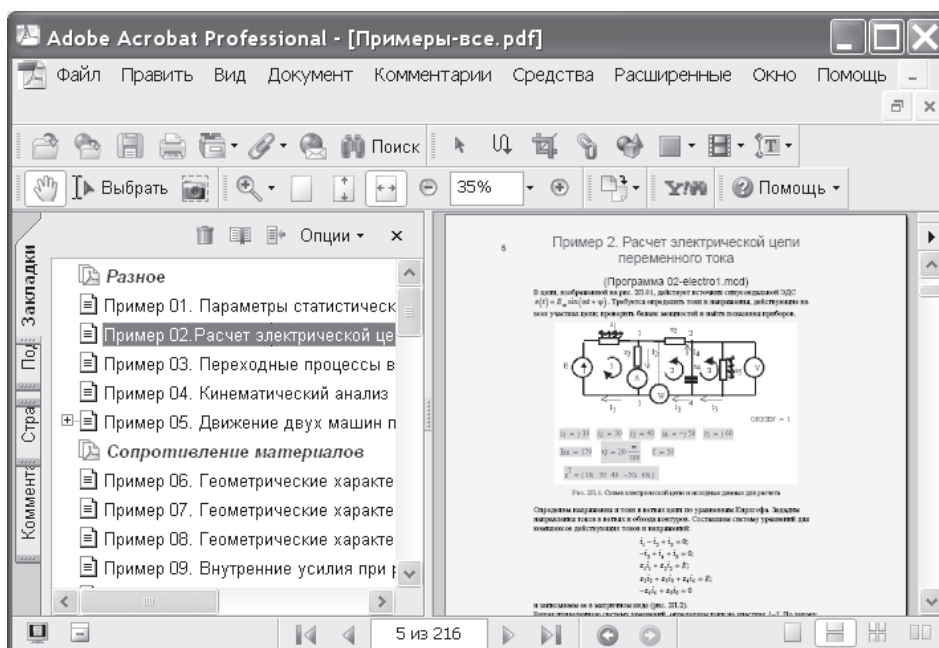


Рис. К.5. Примеры инженерных расчетов

Назначение электронной книги

Основное назначение электронной книги — дать возможность пользователю собирать собственные программы из готовых фрагментов.

Написать программу расчета в Mathcad сравнительно просто, но на отладку программы уходит значительно больше времени, чем на ее написание, что лишает

Mathcad его основного преимущества перед другими вычислительными системами.

Когда пользователь создает свою программу «с нуля», ему приходится выискивать в книгах примеры использования необходимых функций Mathcad, а затем отлаживать написанную программу. Все это требует огромных усилий и больших затрат времени. Иной раз на то, чтобы исправить одну ошибку, уходит несколько дней.

Ни один нормальный человек не в состоянии запомнить правила использования встроенных функций и операторов Mathcad, да это и не требуется. В созданной электронной книге подробно и с большим количеством примеров описываются все возможности Mathcad. Описание почти каждой встроенной функции сопровождается электронными ссылками на примеры инженерных расчетов.

Еще одно затруднение, с которым сталкиваются пользователи Mathcad, — это то, что вся официальная документация Mathcad (файлы помощи и примеры) написана на английском языке. Но далеко не все пользователи в совершенстве владеют иностранным языком и могут разобраться во всех тонкостях представленных материалов. Недаром российские сайты по Mathcad буквально заполнены просьбами помочь отыскать русскоязычный Help к Mathcad. Предлагаемая электронная книга является практически первым русскоязычным полным электронным справочником по пакету Mathcad.

Разработку программы надо начинать с создания алгоритма решения задачи в виде последовательности связанных друг с другом формул, проще всего сделать этот шаг на бумаге. Затем с помощью системы активных ссылок отыщите в электронной книге подходящие примеры инженерных расчетов и примеры использования встроенных функций. Нужные фрагменты электронной книги следует скопировать в свой рабочий документ, объединить их в единую программу и ввести свои исходные данные. Документ готов.

Составление программы методом «электронных кубиков» позволяет многократно сократить время, затраченное на изготовление программы, обеспечивает высокий уровень программирования. Mathcad действительно становится **простым и удобным** средством решения инженерных расчетных задач.

Часть I

Начало работы с Mathcad

В этой части книги приводятся основные сведения, необходимые для работы с Mathcad. Используя их, можно решить примерно 95 % всех задач. Внимательно изучите их и приступайте к работе с Mathcad.

Глава 1

Построение выражений и графиков в Mathcad

Перед началом работы в Mathcad ознакомимся с рабочим окном и содержанием панелей управления. Далее описываются панели и окна Mathcad 14. В других версиях Mathcad внешний вид и содержание панелей и окон немного отличаются от описываемых ниже.

1.1. Интерфейс Mathcad

Интерфейс Mathcad по своей структуре аналогичен интерфейсу других Windows-приложений.

1.1.1. Рабочее окно Mathcad

При открытии файла Mathcad.exe на экране появляется рабочее окно Mathcad с главным меню и пятью панелями: Standard (Стандартная), Formatting (Форматирование), Math (Математическая), Controls (Контроль) и Resources (Документация) (рис. 1.1).

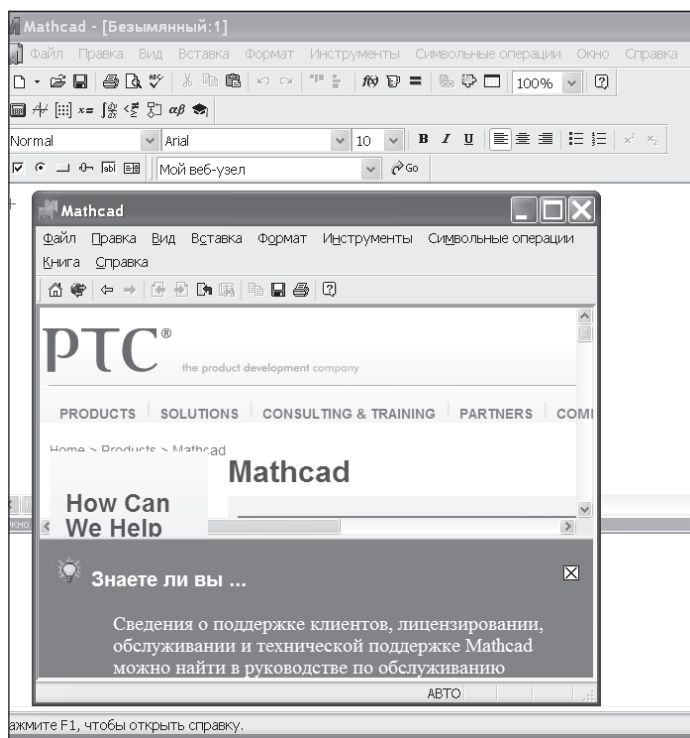


Рис. 1.1. Вид рабочего окна Mathcad после загрузки

Автоматически загружается файл Untitled 1 (Безымянный 1), представляющий собой шаблон Normal (Обычный) рабочего документа Mathcad, называемого Worksheet (Рабочий лист). Кроме того, автоматически загружаются окна Tip of the day (Совет дня) и Mathcad Resource (Документация Mathcad). Перед началом работы их надо закрыть. В окне Tip of the day следует снять флажок Show tip at startup (Показывать совет при загрузке) и щелкнуть на кнопке Close.

1.1.2. Главное меню

Главное меню Mathcad занимает верхнюю строку рабочего окна. Оно не зря называется главным. Все необходимые действия можно выполнить, следуя пунктам этого меню и последовательно открывающихся окон. Вид главного меню показан на рис. 1.2, *вверху*. Щелчок мышью на любом пункте меню открывает подменю с перечнем команд:

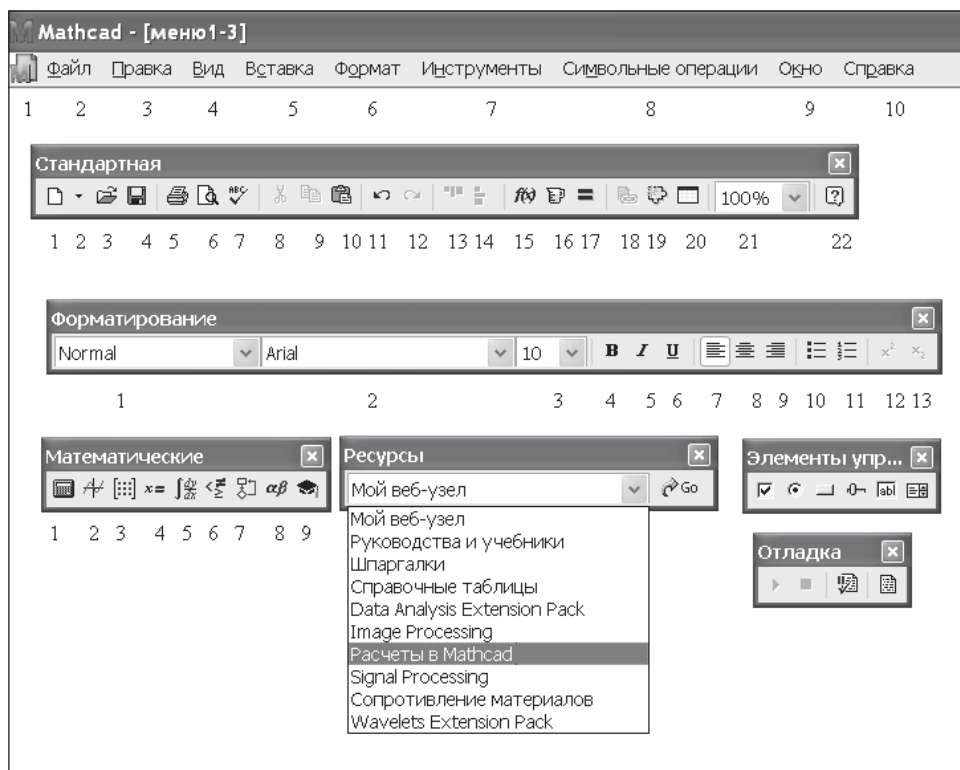


Рис. 1.2. Главное меню и панели Mathcad

1. Управление рабочим окном Mathcad.
2. File (Файл) — команды, связанные с созданием, открытием, сохранением, пересылкой по электронной почте и печатью на принтере файлов с документами.
3. Edit (Правка) — команды, относящиеся к правке текста (копирование, вставка, удаление фрагментов и т. д.).

4. View (Вид) — команды, управляющие внешним видом документа в рабочем окне Mathcad, а также команды создания файлов анимации.
5. Insert (Вставка) — команды вставки различных объектов в документ.
6. Format (Формат) — команды форматирования текста, формул и графиков.
7. Tools (Инструменты) — команды управления вычислительным процессом.
8. Symbolics (Символьные вычисления) — команды символьных вычислений.
9. Window (Окно) — команды, позволяющие управлять расположением окон с различными документами на экране.
10. Help (Помощь) — команды вызова справочной информации.

Щелчок мышью на пункте подменю вызывает появление соответствующего диалогового окна.

Выбрав в главном меню (рис. 1.3) Tools ► Preference (Инструменты ► Предпочтения), можно сделать работу в Mathcad более удобной.

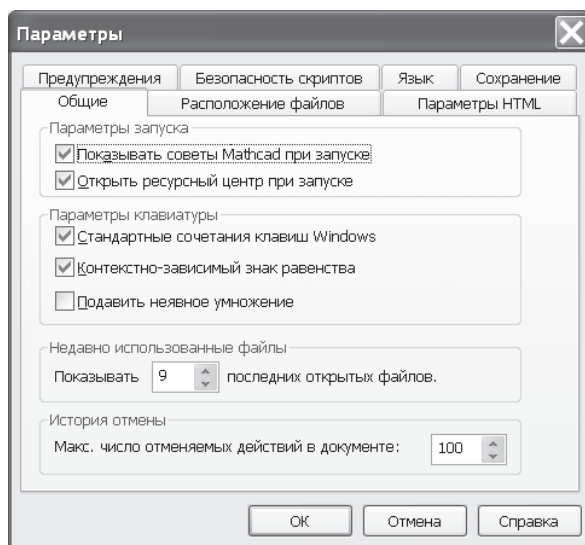


Рис. 1.3. Окно Preferences в Mathcad 15

1.1.3. Панели инструментов

Панели инструментов служат для быстрого выполнения наиболее часто применяемых команд. Вид панелей показан на рис. 1.2:

- ❑ Главное меню Mathcad (на рисунке вверху) осуществляет полное управление всеми процессами, кроме того, позволяет выполнить все команды, объединенные в остальные меню.
- ❑ Standard (Стандартная) служит для выполнения действий с файлами, редактирования документов, вставки объектов и т. д.

- ❑ **Formatting** (Форматирование) предназначена для форматирования текста и формул.
- ❑ **Math** (Математика) служит для вставки математических символов и операторов в документы.
- ❑ **Resources** (Дополнительные ресурсы) содержит список электронных книг, включенных в оболочку Mathcad.
- ❑ **Controls** (Контроль) содержит кнопки для дополнительного контроля работы Mathcad-документа.
- ❑ **Debug** (Отладка) появилась в Mathcad 13, служит для трассировки выполнения программ.

При наведении указателя мыши (курсора) на любую из кнопок рядом с ней появляется всплывающая подсказка — короткий текст, поясняющий назначение кнопки.

Содержание панелей (кроме главного меню) можно корректировать. При наведении курсора на панель и нажатии правой кнопки мыши появляется контекстное меню с двумя пунктами, **Hide** (Скрыть панель) и **Customize** (Выбрать нужные кнопки). Выбрав **Customize**, можно удалить старые и добавить на панель новые кнопки из предлагаемого списка.

Кратко рассмотрим типичное содержание панелей, показанных на рис. 1.2.

Стандартная панель

Стандартная панель Mathcad содержит перечисленные далее пункты:

1. Создание документа на основе шаблона **Normal** (Обычный).
2. Перечень предлагаемых шаблонов документов.
3. Открытие файла.
4. Сохранение файла.
5. Печать файла.
6. Просмотр печати (вид готового документа).
7. Проверка орфографии (только англоязычного текста).
8. Вырезание объекта.
9. Копирование объекта.
10. Вставка объекта.
11. Отмена предыдущего действия (только при вводе текста или формул).
12. Повтор отмененного действия.
13. Выравнивание выделенной группы объектов по горизонтали.
14. Выравнивание выделенной группы объектов по вертикали.
15. Список встроенных функций и их вставка.
16. Список размерностей и их вставка.
17. Пересчет документа.

18. Вставка гиперссылки.
19. Component Wizard (Вставка компонентов) — вставка в рабочий документ окна другого приложения.
20. Масштаб и его изменение.
21. Открытие окна Mathcad Help (Справочная система Mathcad).

Ряд кнопок стандартного меню, показанного на рис. 1.2, мною уже удален. Это кнопки вызова конкретных компонентов других программ, о которых в данной книге не говорится.

Панель форматирования

Панель форматирования Mathcad содержит следующие пункты:

1. Стиль набора текста и формул.
2. Шрифт, применяемый для набора текста и формул.
3. Размер шрифта.
4. Полужирный шрифт.
5. Курсив.
6. Подчеркнутый шрифт.
7. Выравнивание текста по левому краю.
8. Выравнивание текста по центру.
9. Выравнивание текста по правому краю.
10. Маркированный список.
11. Нумерованный список.

Математическая панель

Математическая панель со всеми подчиненными панелями инструментов показана на рис. 1.4. Щелчок мышью на любом из значков вызывает вставку на место курсора в рабочем документе соответствующего этому значку символа или шаблона выполнения математической операции.

Рассмотрим панели инструментов подробнее:

1. Calculator (Калькулятор) — вставка шаблонов основных математических операций, цифр, знаков арифметических операций.
2. Graph (График) — вставка шаблонов графиков.
3. Matrix (Матрица) — вставка шаблонов матриц и матричных операций.
4. Evaluation (Оценка) — операторы присвоения значений и вывода результатов расчета.
5. Calculus (Вычисления) — вставка шаблонов дифференцирования, интегрирования, суммирования.
6. Boolean (Булевы операторы) — вставка логических (булевых) операторов.
7. Programming (Программирование) — операторы, необходимые для создания программных модулей.
8. Greek (Греческие буквы).

9. Symbolics (Символика) — вставка операторов символьных вычислений.

Панель ресурсов

В выпадающем меню панели Resources (см. рис. 1.2) содержится список электронных книг, включенных в пакет Mathcad:

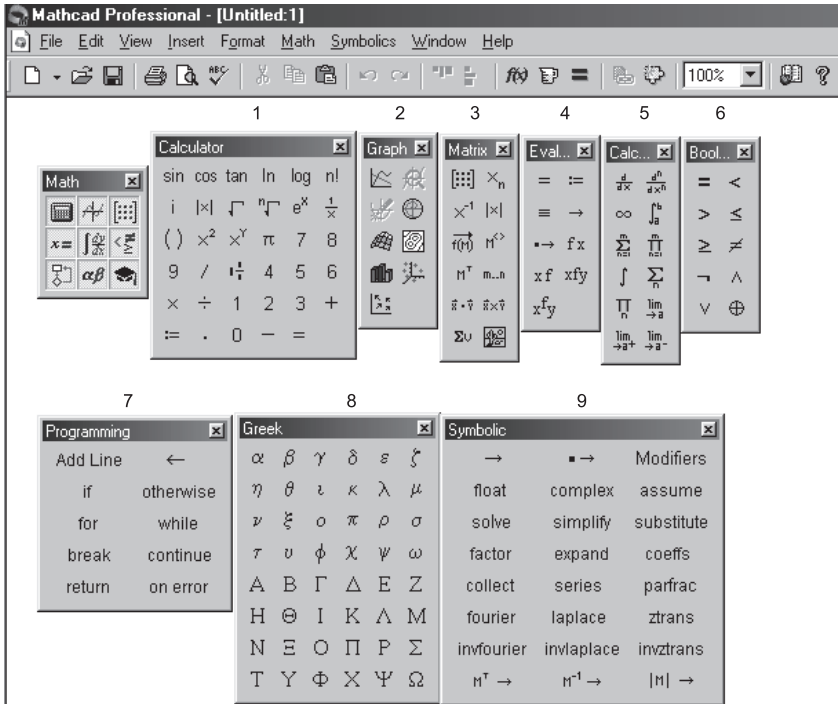


Рис. 1.4. Математическая панель

- Tutorials — обучение работе с Mathcad.
- QuickSheets — примеры использования Mathcad.
- Reference Tables — справочные таблицы.
- Data Analysis Extension Pack — пакет для работы с наборами данных.
- Images Processing — обработка изображений в Mathcad.
- Signal Processing — обработка цифровых сигналов в Mathcad.
- Wavelets Extension Pack — волновые преобразования массивов данных в Mathcad.
- Расчеты в Mathcad — электронная книга.

При нажатии кнопки Go открывается окно справочной информации Mathcad.

Панель контроля

Панель Controls содержит кнопки для дополнительного контроля над работой Mathcad-документа. Эти кнопки предназначены для украшения документа, используются в качестве средства диалога с пользователем, для управления работой подключенных к документу других файлов. Их использование, как правило, требует написания пользовательских DDL-файлов. При разработке Mathcad-

документов нет необходимости использовать такие средства контроля. В данной книге они не рассматриваются. Подробные сведения о них содержатся в меню Help (Помощь), раздел Developer's Reference (Руководство для разработчиков).

Панель Debug (Отладка)

Панель Debug, появившаяся в Mathcad 13, содержит кнопки вызова на экран окна отладки программ и включения (остановки) процесса трассировки (вывода промежуточных результатов расчета на экран) (см. рис. 9.27).

При наведении курсора на кнопку панели инструментов появляется всплывающая подсказка с названием операции и сочетанием клавиш, нажатие которых эквивалентно щелчку на данной кнопке.

1.2. Построение выражений и их вычисление

Перед началом работы курсор на экране имеет вид крестика. В момент ввода выражения курсор приобретает вид синего уголка, обрамляющего вводимое выражение. В рабочем документе введите какие-либо буквенные выражения и присвойте им численные значения. Имя выражения (все, что стоит слева от оператора присваивания) может состоять из латинских, русских, греческих и других букв и цифр, знаков подчеркивания (), штриха (´), символа процента (%), знака бесконечности, вводимых с клавиатуры.

Имена переменных и функций не могут начинаться с цифры, знака подчеркивания, штриха, символа процента, не могут включать в себя пробелы. Символ бесконечности может быть только первым символом в имени.

ВНИМАНИЕ

Mathcad воспринимает прописные и строчные буквы как различные идентификаторы, то же касается букв, изображенных различными шрифтами, — это разные имена.

Mathcad не делает различий между именами переменных и функций. Если вначале определить функцию $f(x)$, а затем — переменную f , окажется невозможно использовать $f(x)$ в расчетах где-либо после определения f .

ПРИМЕЧАНИЕ

Mathcad 12 не допускает переопределения значений функций. Mathcad 13–15 его разрешают, но предупреждают о нем появлением волнистой линии под именем функции. В окне Preference (Параметры) (см. рис. 1.3), выбрав вкладку Warnings (Предупреждения), можно такие предупреждения отключить.

Некоторые имена уже используются Mathcad для встроенных констант, единиц измерения и функций. Имена можно переопределить, но следует учитывать, что это уничтожит их встроенные значения и этими константами и функциями пользоваться будет нельзя.

Оператор присваивания ($:=$) в первый раз выберите на математической панели Calculator (Калькулятор). Познакомьтесь с содержанием панели. При наведении стрелки указателя мыши на какую-либо кнопку появляется всплывающая подсказка с названием оператора, вызываемого при щелчке на этой кнопке, и указа-

нием клавиши или сочетания клавиш, нажатие которых вызывает то же действие, что и щелчок на кнопке. В дальнейшем оператор присваивания вы будете набирать с клавиатуры, нажав клавишу «двоеточие» (:). **Запоминайте «горячие» клавиши!**

Все операции в Mathcad продублированы. Их можно ввести:

- выбрав соответствующий пункт меню;
- нажав соответствующую клавишу (сочетание клавиш) на клавиатуре.

Набрав вычисляемое выражение, нажмите клавишу (=) — появится численный результат (рис. 1.5).

Попробуйте набрать $\frac{x+y}{4}$ — у вас получится $x + \frac{y}{4}$. Для правильной записи надо нажать клавишу пробела так, чтобы уголок курсора охватывал нужную часть выражения (в нашем случае $x+y$). Другой способ — взять выражение в скобки ($x+y$). В некоторых случаях это просто необходимо.

$$x := 5 \quad y := 3 \quad a := 10 \quad (5+3) \cdot 10 = 80 \quad (x+y) \cdot a = 80$$

$$\frac{x+y}{a} = 0.8 \quad \int_0^a x^3 dx = 2.5 \times 10^3 \quad \frac{d}{dx} x^3 = 75$$

Рис. 1.5. Примеры построения математических выражений

1.3. «Цепкие» операторы

С первым захватом «цепкого» оператора вы уже познакомились на примере дроби. «Цепкие» операторы — это возведение в степень, извлечение корня, знаменатель дроби (рис. 1.6). Чтобы «вырваться из объятий» «цепкого» оператора, надо выделить клавишей пробела или клавишей → (стрелка вправо) нужную часть выражения, тогда следующая операция будет относиться ко всему выделенному выражению.

<p>надо записать так</p> $2^{x+a} = 42$ $\frac{x+y}{\ln(a)} \cdot a + 2 = 116.612$ $\sqrt{(x+y)} \cdot \frac{2}{a} - 8 = -7.434$	<p>забыл нажимать пробелы</p> $2^{x+a} = 3.277 \times 10^4$ $x^{2+y} \frac{0.3}{\ln(a)+2} = 127.696$ $\sqrt{x+y} \cdot \frac{2}{a-8} = 2.828$
----------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------

Рис. 1.6. Примеры построения выражений с «цепкими» операторами

На первых порах «цепкие» операторы раздражают. Вы начинаете печатать быстрее и не всегда вспоминаете о выделении нужной части выражения. В результате вместо, например, $x^2 + y$ на экране появляется x^{2+y} . Поупражняйтесь во введении выражений и получении их результатов (рис. 1.7).

Для возведения числа в степень используйте клавишу ^.

вычислить самостоятельно

$$24 + 6^{0.12} \quad b^a \quad \frac{(a+b) \cdot (a-b)}{a \cdot (c-b)} + 2 \cdot \frac{a \cdot b^2}{c^3 + 2}$$

$$\sqrt{12} + \frac{32 \cdot 0.12}{\sqrt{26} - 3^2} \quad \int_0^c e^a \cdot a^b da \quad \frac{4}{da} \frac{b}{a}$$

Рис. 1.7. Самостоятельная работа. Построение выражений и их вычисление

1.4. Редактирование объектов Mathcad

Редактирование введенных выражений производится обычным для всех Windows-приложений способом:

- ❑ Уголок курсора перемещается по экрану клавишами со стрелками или щелчком левой кнопки мыши в нужном месте экрана.
- ❑ Для выделения уголком курсора одного символа надо установить синий уголок курсора так, чтобы он охватывал нужный символ слева или справа.
- ❑ Для расширения выделения на часть выражения или на все выражение целиком следует использовать клавиши со стрелками или клавишу пробела. Для выхода из «цепкого» оператора предпочтительно пользоваться клавишей пробела. Уголок курсора должен охватывать все выражение или всю его часть, над которой надо выполнить какие-либо действия.
- ❑ Для выделения части выражения или всего выражения надо щелкнуть мышью в начале или в конце выделяемой части выражения и переместить курсор до другого края, не отпуская левую кнопку мыши. Можно использовать комбинацию клавиш Shift+← или Shift+→. Выделенная часть выражения имеет черный фон. Выделение в Mathcad используется для вырезания или копирования части выражения, изменения шрифта, а также для выполнения символьных вычислений с частями выражений.
- ❑ Для выделения объекта или группы объектов (любых — математических, текстовых или графических) надо щелкнуть мышью на свободном месте рабочего документа и растянуть пунктирный прямоугольник выделения так, чтобы он охватил нужные вам объекты. Один объект при этом будет выделен синим уголком курсора, а группа объектов — обведена пунктирной рамкой.

Все три вида выделения выражений показаны на рис. 1.8.

Если надо удалить, вырезать или скопировать в буфер обмена выделенную часть выражения, выделенный объект целиком или группу выделенных объектов, выполните следующие действия:

- ❑ для удаления (безвозвратного) нажмите клавишу Delete или Backspace;
- ❑ для вырезания в буфер обмена нажмите кнопку Cut (Вырезать) с изображением ножниц на стандартной панели Mathcad. При нажатии кнопки Cut объект исчезает из рабочего документа, но переносится в буфер;

- ❑ для копирования в буфер обмена нажмите кнопку Copy (Копировать) с изображением двух страниц на стандартной панели Mathcad. При нажатии кнопки Copy объект остается в рабочем документе и копируется в буфер;
- ❑ для вставки объекта из буфера обмена установите крестообразный курсор в то место, куда вы хотите вставить содержимое буфера обмена, и нажмите кнопку Paste (Вставить) на стандартной панели Mathcad.

Поскольку вырезать, копировать, вставлять объекты приходится часто, полезно запомнить сочетания клавиш, нажатие которых вызывает эти действия:

- ❑ Cut (Вырезать) — Ctrl+x;
- ❑ Copy (Копировать) — Ctrl+c;
- ❑ Paste (Вставить) — Ctrl+v;
- ❑ Undo (Отмена предыдущего действия) — Ctrl+z.

Эти действия выполняются сочетаниями клавиш, расположенных по соседству, и поэтому легко запоминаются. Как и во всех Windows-приложениях, выделенный объект или выделенную группу объектов можно перетащить или скопировать с помощью мыши. Для этого следует:

- ❑ подвести мышь к выделенному объекту или группе объектов, чтобы появилась черная ладошка;
- ❑ при нажатой левой кнопке мыши перетащить курсор в то место, куда надо переместить объекты;
- ❑ если нужно скопировать выделенные объекты, после появления черной ладошки нужно нажать левую кнопку мыши, затем клавишу Ctrl и, удерживая их нажатыми, перетащить курсор в нужное место.

СОВЕТ

При работе с Mathcad постоянно пользуйтесь правой кнопкой мыши. В возникающем при этом контекстном меню появляются, как правило, самые нужные в данный момент пункты, в частности, почти всегда присутствуют пункты Cut (Вырезать), Copy (Копировать), Paste (Вставить).

1.5. Стандартные функции

Mathcad содержит свыше 200 встроенных функций. На стандартной панели нажмите кнопку $f(x)$. Вы увидите список всех встроенных функций. Просмотрите группы функций (левый список). Щелкнув мышью на любой из групп функций, вы увидите справа перечень функций, входящих в эту группу. В этой части книги мы рассмотрим не все функции, а только те, которые используются в приведенных примерах. Познакомиться с остальными функциями можно в части II книги, а также нажав кнопку $f(x)$ на стандартной панели.

Начнем с двух групп: Log and Exponential (Логарифмические и экспоненциальные) и Trigonometric (Тригонометрические). Присмотритесь к написанию этих функций (см. рис. 1.8), которое не всегда совпадает с привычной математической записью. Названия функций можно вводить со стандартной панели из

раскрытого окна функций $f(x)$, выделив название функции и нажав кнопку Insert (Вставить), или набрав имя функции на клавиатуре в точности так, как оно записано в окне функций.

Наберите и вычислите несколько разных функций.

ЧИСЛОВЫЕ КОНСТАНТЫ $e = 2.718$ $\pi = 3.142$ ∞

$\frac{\pi \cdot e^2}{4} = 5.803$ $\ln(e^2) = 2$ $\int_0^{\infty} 2^{-x} dx = 1.443$

СТАНДАРТНЫЕ ФУНКЦИИ $x := 5$ $a := 10$

$\ln(e^2) = 2$ $\log(x) = 0.699$ $\exp\left(\frac{a}{\pi}\right) = 24.121$

$\sin\left(\frac{\pi \cdot e^2}{4}\right) = -0.462$ $\sinh\left(\frac{a}{e}\right) = 19.787$

$\tan(a) = 0.648$ $\cot(a) = 1.542$ $\operatorname{atan}(a) = 1.471$

Рис. 1.8. Примеры использования встроенных констант и функций

1.6. Числовые константы. Ввод греческих букв

Введенная с клавиатуры латинская буква e внутри математического выражения означает основание натурального логарифма (см. рис. 1.8) $e = 2,718$. Это значение можно отменить и присвоить e другое значение, используя знак локального присваивания $:=$, например: $e:=2$.

Знак бесконечности ∞ можно вставить с математической панели **Calculus** (со знаком интеграла).

Часто используемое в выражениях число π можно набрать с математической панели **Calculator** (Калькулятор), где есть кнопка π , или нажать аналогичную кнопку панели греческих букв.

СОВЕТ

Есть и более простой способ ввода греческих букв. Надо с клавиатуры набрать латинский аналог греческой буквы, как правило, первую букву в латинском названии греческой буквы. Введя аналог греческой буквы, следует нажать комбинацию клавиш **Ctrl+G**. На экране появится греческая буква — прописная или строчная в зависимости от того, в верхнем или в нижнем регистре был набран ее латинский аналог.

Греческие буквы:	α β χ δ ε η γ λ μ ν ω ϕ π ψ ρ σ τ θ ξ ζ
Их аналоги:	a b c d e h g l m n w f p y r s t q x z

Наберите несколько выражений с греческими буквами, например: $\delta:=2$, $\Delta:=2$, $\sin(\alpha)$ и т. д.

ВНИМАНИЕ

Mathcad вычисляет выражения слева направо и сверху вниз.

Если какой-либо константе или переменной не присвоено никакого значения левее и выше ее положения на экране, то она окрашена в красный цвет, что свидетельствует об ошибке.

1.7. Ввод текста

Для ввода текста в документ можно в главном меню выбрать команду **Insert** ▶ **Text Region** (Вставить ▶ Текстовую область), но лучше ввести с клавиатуры символ " (кавычка). При этом на экране появляется текстовая область, в которой можно печатать текст.

СОВЕТ

Еще лучше, сменив латинский шрифт на русский, печатать текст прямо в математической области. Когда напечатано первое слово, при нажатии клавиши пробела область с напечатанным словом автоматически из математической превращается в текстовую.

В текстовую область можно вставлять математическую область. Для этого в главном меню **Mathcad** следует выбрать команду **Insert** ▶ **Math. Region** (Вставить ▶ Математическую область). Вставленная математическая область участвует в вычислениях наравне с другими математическими выражениями.

ПРИМЕЧАНИЕ

Начиная с **Mathcad 13** при наборе текстовой области в контекстном меню (при нажатии правой кнопки мыши) присутствует пункт **Insert Math.Region** (Вставить математическую область).

При желании вставленную математическую область можно отключить (рис. 1.9). Для этого щелкните правой кнопкой мыши на вставленном выражении и в открывшемся контекстном меню выберите команду **Disable Evaluation** (Отключить вычисление). После этого вставленное математическое выражение будет играть роль иллюстрации.

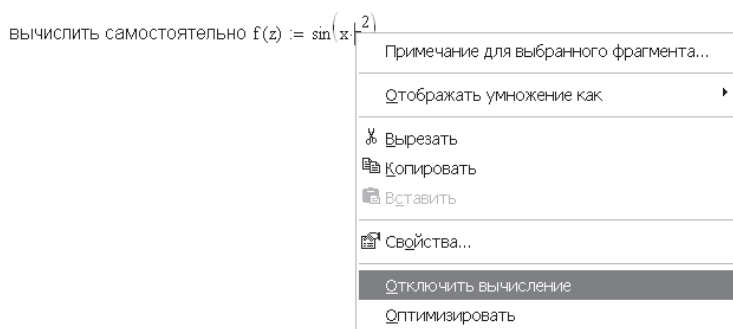


Рис. 1.9. Математическая область, вставленная в текст

Текстовой области в **Mathcad** присвоен стиль **Normal**. Чтобы настроить его, сделайте следующее:

1. В главном меню выберите команду **Style** ▶ **Normal** ▶ **Modify** ▶ **Font** (Стиль ▶ Обычный ▶ Изменить ▶ Шрифт).

2. В появившемся диалоговом окне выберите нужные вам шрифт (Font), форму шрифта (Style Font) и размер (Size).

Шрифты Mathcad плохо воспринимают кириллицу. В частности, привычный по работе в Word шрифт Times New Roman кириллицу не принимает. С кириллицей работают шрифты System и Ms Sans Serif.

СОВЕТ

В электронной книге содержатся файлы Normal.mct и Normal.xmcd с установленными шрифтами, работающими с кириллицей. Для Mathcad 12, 13, 14 файл Normal.xmcd надо скопировать в каталог Template папки, в которой установлен Mathcad. Если Mathcad самостоятельно выбирал путь для своей установки, то это C:\Program Files\Mathsoft\Mathcad 14\Template. Новый шаблон из электронной книги заменит старый, и впредь при каждом открытии Mathcad на экране будет появляться установленный вами приспособленный для работы с кириллицей шаблон документа. Для более ранних версий в каталог Template надо скопировать файл Normal.mct.

В Mathcad можно вводить и редактировать текст почти как в Word, но все-таки это не так удобно. О вводе текста речь пойдет в части III книги.

1.8. Функции пользователя

Удобство и эффективность расчетов в Mathcad определяется прежде всего возможностью и легкостью создания функций пользователя. При многократном использовании одного и того же выражения без функций пользователя просто не обойтись.

Вид функции пользователя (рис. 1.10):

слева *название функции (с параметрами в скобках)*,

справа, после оператора присваивания :=, *вычисляемое выражение*.

Переменные величины, входящие в правую часть, должны быть записаны в параметры после имени функции. Все величины из правой части, не входящие в параметры левой части, должны быть заданы численно левее и выше функции пользователя. В противном случае Mathcad указывает на ошибку, окрашивая не заданную величину в красный цвет. При выделении функции щелчком мыши появляется текст сообщения об ошибке This variable is not defined above (Эта переменная не определена ранее).

$$\begin{array}{l}
 x := 3 \quad a := 0.4 \quad f(z) := \sin(x \cdot z^a) \quad f1(z) := \int_0^z f(z) dz \\
 f2(z) := \frac{d}{dz} f(z) \quad f3(z) := \frac{d^3}{dz^3} f(z) \\
 \text{Здесь } x \text{ и } a \text{ - константы.}
 \end{array}$$

Рис. 1.10. Функции пользователя

Функция пользователя не вычисляется Mathcad, а принимается к сведению. Для вычисления функции надо задать численные значения всех параметров в имени

функции, набрать имя функции и нажать клавишу = или щелкнуть мышью на кнопке = на стандартной панели Mathcad либо в математическом меню на панели Calculator Toolbar (Калькулятор).

1.9. Дискретные переменные. Построение таблиц

Одно из лучших творений создателей Mathcad — это дискретная переменная, выполняющая роль оператора цикла. В ряде книг по Mathcad ее называют ранжированной переменной. Однако это не лучший перевод выражения «range variable». Выражение «дискретная переменная», использовавшееся в ранних книгах, более точно отражает суть понятия — диапазон изменения переменной: вместо непрерывной переменной используется ряд чисел, выстроенных в порядке возрастания или убывания. Простота использования этого оператора восхищает. Без использования дискретной переменной было бы очень сложно построить графики, вывести таблицы результатов расчета. Если в функцию пользователя подставить численное значение переменной, результатом расчета будет число. Дискретная переменная задает ряд значений переменной, для которых вычисляется функция пользователя. Этот ряд значений функции можно вывести в виде графика или таблицы. Определение дискретной переменной имеет вид $x := 0..5$, что означает задание ряда значений $x = 0, 1, 2, 3, 4, 5$. Пример:

$$f(x) := \sin(x) \cos(x), x := 0..5, f(x) = \text{ответ.}$$

Ответ выводится в виде вектора — столбца чисел (рис. 1.11) — или графика (рис. 1.12).

a := 10 b := 5 n := 5 поменяйте a,b,n

$$z := 0..5 \quad z1 := 0,0,2..1 \quad z2 := a, a + \frac{b-a}{n} .. b \quad z3 := b, b - \frac{b-a}{n} .. a$$

z = z) = z1 = f1(z1) = z2 = f2(z2) = z3 = f3(z3) =

0	0	0	0	10	0.094	5	-0.106
1	0.141	0.2	0.171	9	0.189	6	-0.064
2	0.729	0.4	0.362	8	0.283	7	-0.034
3	0.998	0.6	0.514	7	0.362	8	-0.014
4	0.872	0.8	0.617	6	0.406	9	-5.754·10 ⁻⁴
5	0.542	1	0.669	5	0.384	10	7.867·10 ⁻³

измените формат чисел

Рис. 1.11. Дискретные переменные. Числовые результаты выведены для функций, приведенных на рис. 1.10

Дискретная переменная может задавать как целые, так и дробные значения переменной, но обязательно равноотстоящие друг от друга, например:

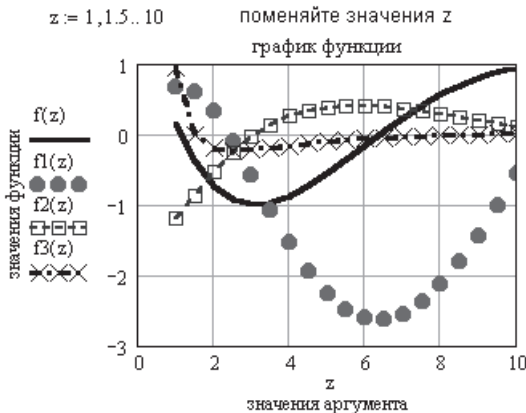
- $x := 0..5$ — ряд целых чисел от 0 до 5;
- $x := 1, 1,1..5$ — ряд дробных чисел, где 1 — первое число, 1,1 — второе число, 5 — последнее число. Интервал между числами $1,1 - 1 = 0,1$;

- $x := A, A - \frac{B-A}{n} .. B$ – ряд чисел, где A – первое, $A - \frac{B-A}{n}$ – второе, B – последнее число (A и B должны быть заданы заранее), n – число интервалов, на которые разбит отрезок от A до B . Такая форма записи удобна, когда рассматриваются разные варианты одного расчета и изменение констант A и B позволяет мгновенно пересчитать результаты и перестроить графики.

ВНИМАНИЕ

Двоеточие – знак диапазона нельзя набирать с клавиатуры, нажимая два раза клавишу «точка». Надо на клавиатуре нажать клавишу ; (точка с запятой) или в математической панели выбрать Matrix (значок матрицы) ▶ $m..n$.

Если константы, входящие в правую часть функции пользователя, не задаются непосредственно перед использованием функции, Mathcad берет их значения, использовавшиеся в последний раз перед данным расчетом. Проверьте, устраивают ли вас эти значения. Наберите имя параметра и нажмите клавишу =.



Самостоятельно сделайте такой же график и отформатируйте его до такого вида

Рис. 1.12. Графики функций, приведенных на рис. 1.10

1.10. Форматирование чисел

В Mathcad на результат расчета повлиять нельзя, но можно изменить формат вывода чисел. Mathcad вычисляет все выражения с точностью 20 знаков, но выводит на экран не все значащие цифры.

Установив указатель мыши на нужном численном результате расчета, сделайте двойной щелчок левой кнопкой мыши. Откроется окно форматирования чисел Result Format (Формат результата), открытое на пункте Number Format (Формат чисел). В этом окне можно выбрать следующие форматы:

- General (Общий) – принят по умолчанию. Числа отображаются с порядком. Количество знаков перед запятой определяется в пункте Exponential threshold (Порог экспоненты).
- Scientific (Научный) – числа отображаются только с порядком: $1,22 \cdot 10^5$.

- Decimal (Десятичный) — десятичное представление чисел с плавающей запятой: 12,2564.
- Engineering (Инженерный) — числа отображаются только с порядком, кратным 3: $1,22 \cdot 10^6$.
- Fraction (Дробь) — в виде правильной или неправильной дроби: $\frac{5}{3}$ или $1\frac{2}{3}$.

В дробном формате можно выбрать уровень точности (Level of accuracy) и смешанные числа (Use fixed number).

Кроме вида формата можно изменять количество знаков после запятой (Number of decimal pieces) и порог порядка (Exponential threshold). При превышении порога число отображается с порядком. Примеры вывода чисел в различных форматах приведены на рис. 1.13. Mathcad автоматически округляет числа до нуля, если они меньше установленного порога.

Форматы чисел		для выражения $a := e^{10}$	
General (по умолчанию)	$a = 2.203 \times 10^4$		
Scientific	$a = 2.203 \times 10^4$	$a = 2.203 \times 10^4$	
Decimal	$a = 22026.5$		
Engineering	$a = 22.026 \times 10^3$	$a = 22.026E+003$	
Fraction	$a = \frac{11053739568}{501839}$	$a = 22026 \frac{233754}{501839}$	

Вывод значений функций с использованием дискретной переменной

z = 1..5	Формат чисел		
	десятичный f(z) =	научный f1(z) =	инженерный f2(z) =
1	0.14112	6.892·10 ⁻¹	-1.188·10 ⁰
2	-0.72905	3.201·10 ⁻¹	-541.892·10 ⁻³
3	-0.99838	-5.858·10 ⁻¹	-35.271·10 ⁻³
4	-0.8723	-1.545·10 ⁰	255.407·10 ⁻³
5	-0.5415	-2.263·10 ⁰	384.096·10 ⁻³

Рис. 1.13. Примеры форматирования результата численного расчета

Выбранные установки могут быть применены только к выделенному числу (выберите пункт ОК) или по умолчанию ко всем числам данного документа (выберите пункт Set as default (Применить по умолчанию)).

1.11. Построение плоского графика функции

Для построения плоского графика функции следует:

- установить крестообразный курсор в то место, где надо построить график;
- на математической панели щелкнуть мышью на кнопке Graph Toolbar ► X-Y Plot (Плоский график);

- ❑ в появившемся на месте курсора шаблоне плоского графика ввести на оси абсцисс имя аргумента, на оси ординат — имя функции;
- ❑ щелкнуть мышью вне шаблона графика. График построен для заданного диапазона изменения аргумента.

Если диапазон значений аргумента не задан, по умолчанию график будет построен в диапазоне значений аргумента от -10 до 10 .

Чтобы на одном шаблоне разместить несколько графиков, надо, набрав на оси ординат имя первой функции, нажать клавишу запятой (уголок курсора при этом обязательно должен находиться в конце имени функции). В появившемся месте ввода (черном квадратике) впишите имя второй функции и т. д.

Если две функции имеют разные аргументы, например, $f_1(x)$ и $f_2(y)$, то на оси ординат нужно ввести (через запятую) имена обеих функций, а на оси абсцисс (также через запятую) — имена обоих аргументов, x и y . Тогда первый график будет построен для первой функции по первому аргументу, а второй график — для второй функции по второму аргументу.

Если функций введено несколько, а аргументов — 2, то график первой функции строится по первому аргументу, а графики остальных функций — по второму.

Если ввести на осях ординат и абсцисс имена двух функций одного аргумента, то будет построен параметрический график функции (рис. 1.14).

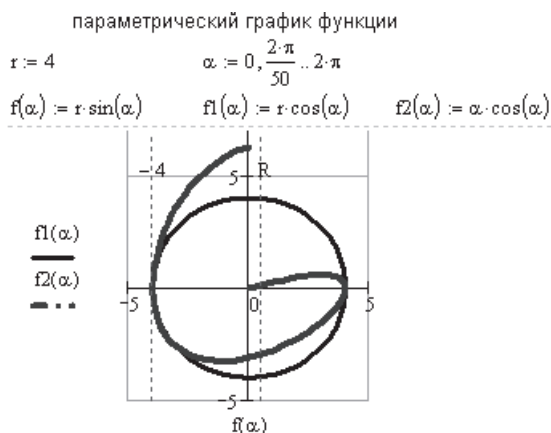


Рис. 1.14. Параметрический график функции

Чтобы отформатировать график, сделайте двойной щелчок мышью в поле графика — откроется окно форматирования графика (рис. 1.15). «Погуляйте» по окну. Выбирайте различные пункты меню (щелкнув на них мышью, а затем на кнопке Применить или ОК) и посмотрите, как изменится при этом вид графика.

В открывшемся окне (см. рис. 1.15) видны заголовки пяти вкладок:

1. X-Y Axes (Оси X–Y) — отформатировать оси координат: нанести сетку линий, проставить численные значения; ось абсцисс провести через ноль ординаты (Crossed — по центру) или по нижнему краю графика (Boxed — по краям),

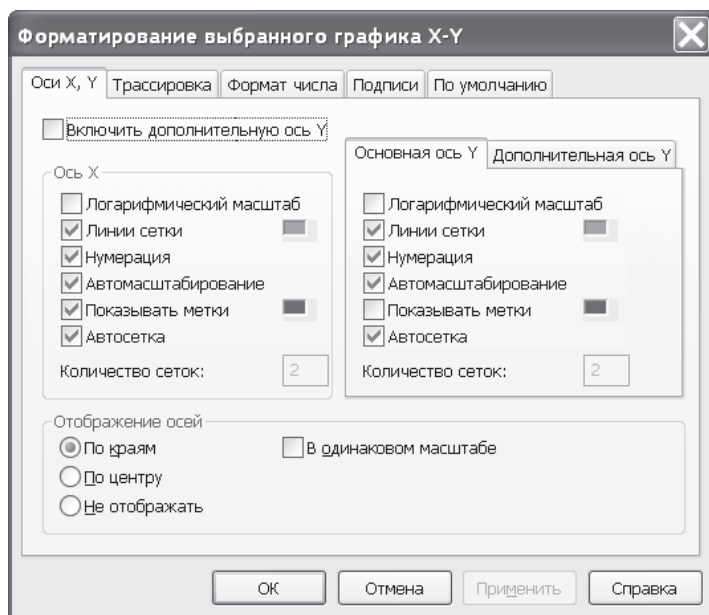


Рис. 1.15. Окно форматирования плоского графика

нанести метки на графике. Щелкните мышью на нужных пунктах подменю (отметьте их флажком):

- Log Scale — представить численные значения на осях в логарифмической шкале. По умолчанию численные значения наносятся в равномерной шкале;
- Grid Lines — нанести сетку линий;
- Numbered — нанести численные значения для каждой линии сетки;
- Auto Scale автоматически выбирает предельные численные значения на осях, большие максимальных вычисленных значений. Если этот пункт не отмечен, предельными будут максимальные вычисленные значения;
- Autogrid — количество линий сетки выбирается автоматически. Если пункт не отмечен, надо задать количество линий Number of Grids;
- Show Markers — нанести метки на графике. На каждой оси появляются два места ввода, в которые можно ввести численные значения (можно не вводить ничего или ввести одно число или буквенные обозначения констант). На графике появятся горизонтальные или вертикальные пунктирные линии, соответствующие указанному значению на оси. В конце этой линии появляется само число (см. рис. 1.14).

Имеется возможность нанести численные значения как слева, так и справа от графика. Если это вам надо, поставьте флажок на пункте Enable Secondary Y-axis (Включить дополнительную ось Y). С правой стороны графика появится Placeholder (Место ввода), куда надо вписать имя функции, выводимой на график.

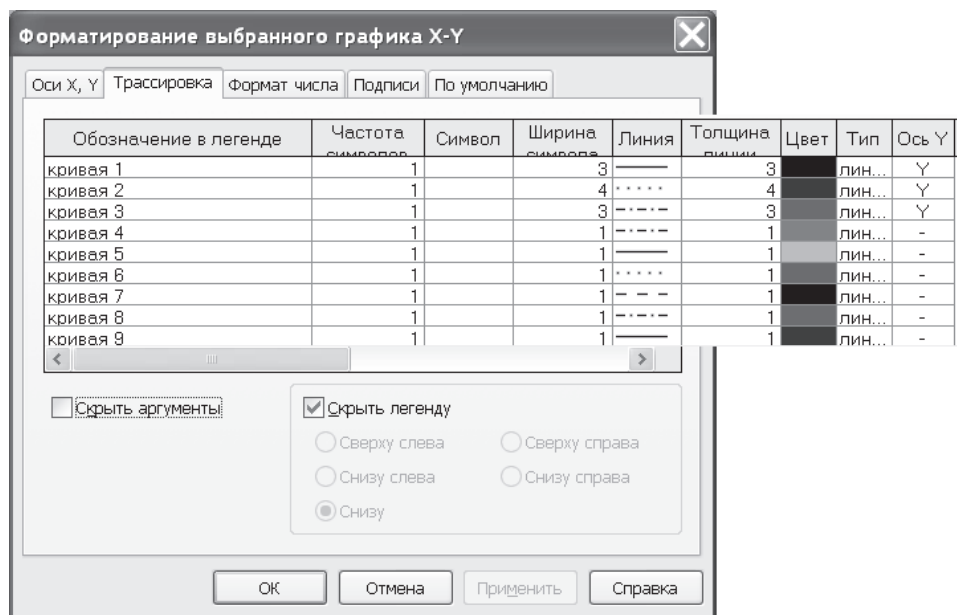


Рис. 1.16. Окно Trace (Трассировка) двумерного графика в Mathcad

- Trace (Трассировка) — отформатировать графики функций (рис. 1.16). Для каждого графика в отдельности задаются:
 - Symbol Frequency (Частота символов) — на графике можно отмечать не все расчетные точки, а через одну, через 2 и т. д.;
 - Symbol (Символ) — обозначение расчетных точек на графике (выбрать кружок, крестик, прямоугольник, ромб);
 - Symbol Weight (Ширина символа) — размер значка, которым обозначается расчетная точка;
 - Line (Линия) — вид линии: Solid — сплошная, Dot — пунктир, Dash — длинный пунктир, Dadot — штрих-пунктир;
 - Line Weight (Толщина линии) — толщина линии;
 - Color (Цвет) — цвет линии;
 - Type (Тип) — тип графика: Lines — линия, Points — точки, Bar или Solid-bar — столбики, Step — ступенчатый график и др.;
 - Y-Axis (Ось Y) — в этой колонке Mathcad сам проставляет Y или Y2 в зависимости от того, слева или справа будут выведены численные значения функции.
- Number Format (Формат чисел) — формат чисел, выводимых на графике.
- Labels (Заголовки) — вписать в область графика заголовки. В окне Title (Заголовок) нужно записать текст заголовка, выбрать его положение — вверх или вниз от графика. Указать, если надо, названия осей (Axis Label).

5. Defaults (По умолчанию) — вернуться к виду графика, принятому по умолчанию (Change to default), либо использовать по умолчанию для всех графиков данного документа (Use for default) выполненные вами изменения.

Выполните самостоятельную работу (рис. 1.17) и поработайте с графиками приведенных в примерах функций.

вычислить самостоятельно
и построить графики функций

$$c := 0.5 \quad b := 2$$

$$f(x) := \sin(x) \cdot \cos(x) \quad f(x) := e^{-c \cdot x} + b \cdot \sin(c \cdot x)$$

$$f1(x) := \int_c^b f(x) dx \quad f2(x) := \frac{d}{dx} f(x) \quad f3(x) := \frac{d^2}{dx^2} f(x)$$

Рис. 1.17. Самостоятельная работа с графиками

Чтобы изменить размеры графика, нужно:

- щелкнуть мышью в поле графика;
- подвести указатель мыши к одному из черных квадратиков на краю графика;
- при появлении двунаправленной стрелки нажать левую кнопку мыши и, удерживая ее, переместить край графика в нужное место.

Чтобы переместить график, нужно:

- щелкнуть мышью в поле графика;
- подвести указатель мыши к краю графика;
- при появлении черной ладошки нажать левую кнопку мыши и, удерживая ее, переместить график в нужное место.

ПРИМЕЧАНИЕ

Точно так же можно переместить любой объект документа: выражение, текст, график.

Описанный способ перемещения объектов — традиционный для Windows. Однако для больших документов он не всегда удобен, например, если надо перетащить выделенную группу объектов на другую страницу. В этом случае удобнее другой способ:

- Нажать левую кнопку мыши и, удерживая ее, двигать мышью, заключая в появляющуюся при этом пунктирную рамку объекты, которые надо переместить.
- На стандартной панели Mathcad щелкнуть мышью на кнопке Cut (Вырезать). При этом выделенная группа объектов будет удалена из документа и помещена в буфер обмена. Если нажать кнопку Copy (Копировать), выделенные объекты будут скопированы в буфер, оставаясь на своем месте.
- Подвести указатель мыши к месту, куда надо вставить вырезанные объекты, и щелкнуть левой кнопкой мыши. Появится крестообразный курсор.
- На стандартной панели Mathcad щелкнуть мышью на кнопке Paste (Вставить). Готово, выделенные объекты перемещены в заданное место.

СОВЕТ

На первых порах не торопитесь вырезать объекты. В описанной процедуре перемещения, вместо того чтобы вырезать объект в буфер, лучше скопировать его, а затем вставить из буфера в нужное место. Убедившись, что все вышло, как задумано, можно вернуться в исходную позицию и удалить теперь уже лишние объекты, то есть снова выделить их рамкой и нажать на клавиатуре клавишу **Delete** (удалить безвозвратно) либо вырезать в буфер.

Если объекты (выражения, графики) перекрывают друг друга, в контекстном меню появляются пункты **Bring to Front** (Выдвинуть на передний план) и **Send to Back** (Убрать на задний план). Использование этих пунктов позволяет экономить место в документе, специально накладывая края объектов один на другой.

Если вам надо раздвинуть наложившиеся друг на друга объекты, выделите их, перечеркнув мышью при нажатой левой кнопке. Все выделенные объекты будут обведены пунктирной рамкой. В главном меню Mathcad выберите **Format ▶ Separate Region** (Разделить области). При этом будут раздвинуты только выделенные области.

Если надо раздвинуть два объекта, подведите к одному из них курсор. При появлении черной ладонки нажмите левую кнопку мыши и, двигая мышью, переместите объект в нужное место.

ВНИМАНИЕ

Постарайтесь не нажимать кнопку **Separate Region**, не выделив предварительно объекты, так как Mathcad при этой команде раздвигает области сразу во всем документе. Для экономии места в документе объекты обычно располагают как можно ближе друг к другу. Mathcad раздвигает документы сверху вниз и при этом может нарушить нужную вам последовательность вычислений.

1.12. Построение трехмерных графиков

Для построения трехмерного графика

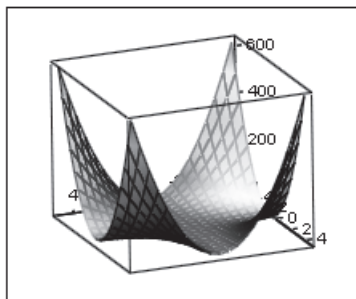
- наберите имя функции двух переменных, знак присвоения значения **:=** и выражение функции;
- установите курсор в то место, где вы хотите построить график;
- в математической панели щелкните мышью на кнопке **Graph Toolbar** (Панель графиков), изображающей график, затем на **Surface Plot** (График поверхности). На месте курсора появится шаблон трехмерного графика;
- в единственном поле ввода шаблона графика введите имя функции;
- щелкните мышью вне области шаблона. График построен (рис. 1.18, *слева*).

Кроме описанного способа ускоренного построения графика поверхности по функции двух переменных существует и часто применяется другой способ создания графика поверхности — с использованием массива численных значений функции. Результат такого построения показан на рис. 1.18, *справа*. Для построения 3D-графика нужно:

- ❑ с помощью дискретных переменных ввести значения обоих аргументов заданной функции;
- ❑ ввести массив, элементами которого являются значения функции, вычисленные при заданных значениях аргументов;
- ❑ установить курсор в то место, где вы хотите построить график;
- ❑ в математической панели щелкнуть мышью на кнопке, изображающей график, и выбрать трехмерный график. На месте курсора появится шаблон трехмерного графика;
- ❑ в единственном поле ввода шаблона графика ввести имя функции;
- ❑ щелкнуть мышью вне области шаблона. График построен.

Построение графика поверхности ускоренным путем (заданием функции)

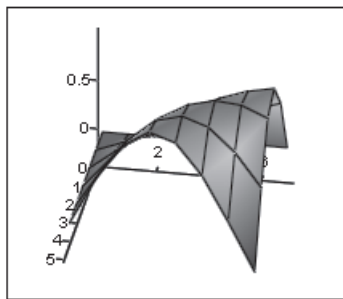
$$f(x,y) := (x^2 \cdot y^2)$$



f

Построение графика поверхности путем создания массива данных

$$\begin{aligned} n &:= 5 \quad m := 7 \quad i := 0..n \quad j := 0..m \\ X_{1,i} &:= 0.5 \cdot i \quad Y_{1,j} := 0.2 \cdot j \quad Z_{1,i,j} := \sin(X_{1,i} \cdot Y_{1,j}) \end{aligned}$$



Z

Рис. 1.18. Построение трехмерных графиков

График получился черно-белый, некрасивый, Mathcad позволяет сделать из него почти произведение искусства. Подробно о возможностях форматирования поговорим в главе 15, а пока выполните следующие действия:

- ❑ Сделайте двойной щелчок мышью в поле графика — появится окно форматирования графика (рис. 1.19).
- ❑ Выберите пункт Appearance ▶ Fill surface ▶ Colormap (Оформление ▶ Поверхность с заливкой ▶ Карта цветов). Щелкните мышью на кнопке Применить. График стал цветным.
- ❑ Выберите в окне форматирования пункт Lighting ▶ Enable Lighting ▶ On (Подсветка ▶ Необходима подсветка ▶ Включить подсветку). Выберите схему освещения (Lighting schema). В выпадающем меню находится 6 различных схем. Выберите любую и нажмите кнопку Применить или ОК. Полюбуйтесь на эффект освещения.

- ❑ Щелкните правой кнопкой мыши на графике — откроется контекстное меню, дающее дополнительные возможности улучшения графика. Выбирайте различные пункты и смотрите, как они влияют на график.

В окне форматирования на вкладке **General (Общие)** внизу перечислены 6 типов графиков (см. рис. 1.19):

- ❑ Surface plot — график поверхности;
- ❑ Contour plot — график линий уровня;
- ❑ Date points — на графике только расчетные точки;
- ❑ Vector Field plot — график векторного поля;
- ❑ Bar plot — график трехмерной гистограммы;
- ❑ Patch plot — площадки расчетных значений.

Щелкните мышью на каком-либо типе графика, а затем на кнопке **Применить**. Посмотрите все типы графиков, кроме векторной диаграммы. Ее построение рассматривается в главе 15.

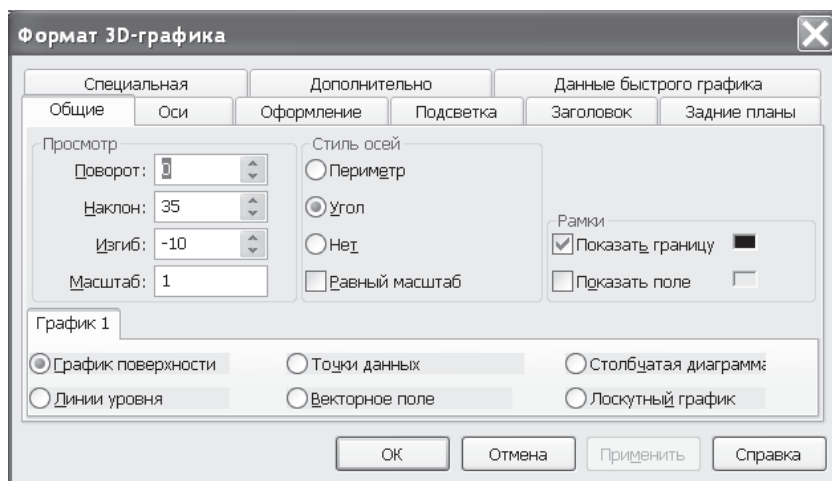


Рис. 1.19. Панель форматирования 3D-графиков

Дополнительные возможности работы с графиком:

- ❑ Вращение графика. Поместите курсор в область графика. Двигайте курсор при нажатой левой кнопке мыши.
- ❑ Масштабирование графика. Действия те же при нажатой клавише **Ctrl**.
- ❑ Анимация графика. Действия те же при нажатой клавише **Shift**. Для остановки вращения щелкните левой кнопкой мыши внутри поля графика.

Панель форматирования трехмерных графиков, приведенная на рис. 1.19, содержит 9 вкладок, обеспечивающих огромные возможности форматирования графиков.

Подробнее о построении и форматировании 3D-графиков рассказано в главе 15.

Глава 2

Некоторые возможности Mathcad

Научившись строить математические выражения, функции пользователя и графики, познакомимся с основными возможностями Mathcad, наиболее часто используемыми функциями и приемами работы.

2.1. Ступенчатые и разрывные функции и выражения. Условие в Mathcad

Используемые в расчетах функции не всегда бывают непрерывными. Часто при разных значениях аргумента функция описывается различными выражениями. Бывают функции ступенчатые или с разрывами. При вычислении производных или интегралов от таких функций приходится их брать по частям. Условный оператор позволяет записать такие функции в виде одного выражения, что упрощает расчеты и украшает Mathcad-документ.

В Mathcad существует три различных способа ввода условий:

- ❑ с помощью функции условия if;
- ❑ с помощью оператора if с панели программирования;
- ❑ с использованием булевых операторов.

Рассмотрим пример вычисления перемещения балки при изгибе с помощью интеграла Мора (рис. 2.1). На балке есть два участка, на которых изгибающий момент описывается различными функциями, $M_1(x)$ и $M_2(x)$.

Для использования функции условия if нужно:

- ❑ записать имя выражения и оператор присваивания (:=);
- ❑ на стандартной панели нажать кнопку $f(x)$ и в списке встроенных функций выбрать if, после чего нажать кнопку Insert (Вставить). Появится шаблон функции if с тремя местами ввода;
- ❑ заполнить места ввода.

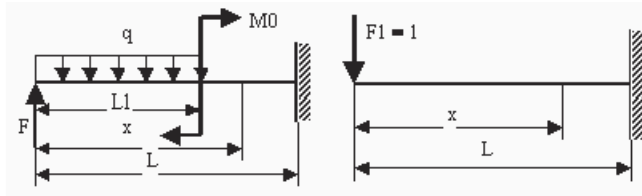
Обращение к функции:

$$\text{if}(\text{cond}, x, y),$$

где cond — условие типа $x \leq L_1$, x и y — значения, возвращаемые функцией. Если условие выполняется, то выражению присваивается значение x , если не выполняется, то значение y .

Чтобы записать условный оператор с панели программирования, следует:

- ❑ записать имя выражения и оператор присваивания (:=);
- ❑ вызвать панель программирования Programming Toolbar нажатием соответствующей кнопки математической панели и щелкнуть мышью на кнопке Add lines (Добавить линию);



Изгибающие моменты от внешних сил

$$M1(x) := F \cdot x - q \cdot \frac{x^2}{2} \quad M2(x) := F \cdot x - q \cdot L1 \cdot \left(x - \frac{L1}{2}\right) + M0$$

Изгибающий момент от единичной силы $MM(x) := x$

Условный оператор программирования if Функция условия if

$$M(x) := \begin{cases} M1(x) & \text{if } x \leq L1 \\ M2(x) & \text{otherwise} \end{cases} \quad M(x) := \text{if}(x \leq L1, M1(x), M2(x))$$

если то иначе

Применение булевых операторов $M(x) := M1(x) \cdot (x \leq L1) + M2(x) \cdot (x > L1)$

Все 3 выражения $M(x)$ эквивалентны

$F, q, M0, L1, L$ - заданы глобально около графика

поперечная сила прогиб балки $E := 2 \cdot 10^{11}$ $J := 2000 \cdot 10^{-8}$

$$Q(x) := \frac{d}{dx} M(x) \quad \Delta := \int_0^L \frac{M(x) \cdot MM(x)}{E \cdot J} dx \quad \Delta = 0.013$$

Рис. 2.1. Определение перемещения балки

- ❑ в верхнем поле ввода (черный квадратик) ввести выражение для изгибающего момента на первом участке;
- ❑ щелкнуть мышью на кнопке if на панели программирования (выражение для изгибающего момента при этом должно быть полностью выделено синим уголком или взято в скобки); появится место ввода, куда надо вписать условие, например, $x \leq L_1$ или $0 \leq x \leq L_1$;
- ❑ в нижнем поле ввода (у вертикальной черты) ввести изгибающий момент для второго участка и выделить его целиком синим уголком (клавишей пробела);
- ❑ щелкнуть мышью на кнопке Otherwise (Иначе) на панели программирования или выбрать if и вписать условие $x > L_1$. Функция $M(x)$ готова.

Использование логических (булевых) операторов состоит в умножении слагаемых заданного выражения на соответствующий логический оператор. Логические операторы вводятся с панели булевых операторов (кнопка Boolean Toolbar). Булевы операторы могут возвращать только 0 или 1. Если условие справедливо, то значением оператора является 1, если несправедливо, то 0. В математическом выражении умножение на логический оператор обращает соответствующее слагаемое в ноль или не меняет его значения. Пример применения всех трех форм записи условия показан на рис. 2.1.

Схемы грузового и единичного состояния балки, изображенные на рис. 2.1, выполнены автором в Corel Draw. Для вставки любого объекта, подготовленного в другом приложении, в Mathcad надо выбрать пункт меню Insert ► Object (Вставка ► Объект). Появится стандартное диалоговое окно Windows Вставка объекта. В этом окне надо выбрать команду Создать из файла ► Обзор, затем выбрать имя файла, в котором сохранен рисунок. Чтобы в дальнейшем можно было редактировать рисунок, не выходя из оболочки Mathcad, рекомендуется установить флажок Связь в диалоговом окне Вставка объекта. Рисунок появился в документе. Подробнее об использовании условия в Mathcad говорится в главе 12 (раздел 12.7).

2.2. Глобальное присвоение значений

Посмотрите на функцию $M(x)$ (рис. 2.2), она содержит константы — нагрузки и длины, P , q , M_0 и L_1 . До сих пор эти константы не были заданы. Но строгий Mathcad тем не менее не указывает на ошибку. Дело в том, что константы заданы глобально около графика.

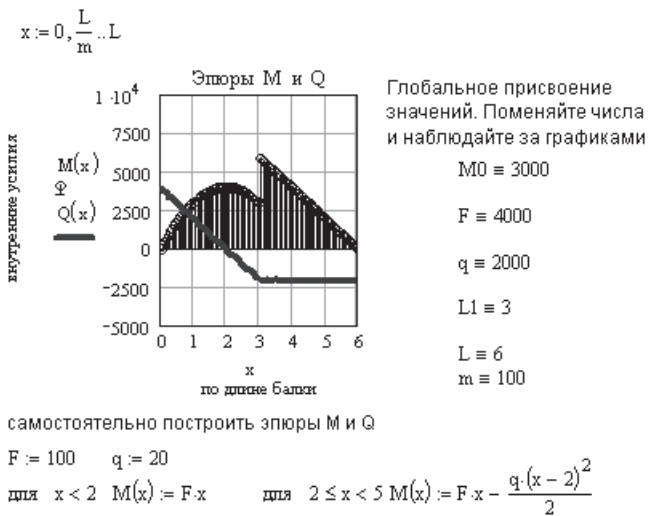


Рис. 2.2. Эпюры внутренних усилий для балки, изображенной на рис. 2.1

Чтобы присвоить некоторой константе глобальное значение, нужно:

- набрать с клавиатуры имя константы;
- в математической панели щелкнуть мышью на кнопке $x=$;
- щелкнуть на кнопке \equiv в открывшейся панели Evaluation (Вычисление) или нажать комбинацию клавиш Shift+~.

Обратите внимание: если локальный оператор присваивания ($:=$) действует вправо и вниз от места ввода, то глобальный оператор присваивания (\equiv) действует по всему документу, и вверх, и вниз. При открытии документа интерпретатор

Mathcad вначале просматривает весь документ сверху вниз, отыскивая глобальные операторы присваивания, затем при втором проходе выполняет локальные присваивания.

СОВЕТ

Следует избегать присвоения одному и тому же имени константы и глобального, и локального значений, так как при этом глобальное значение отменяется локальным.

Глобальные операторы присваивания значений константам удобно размещать вблизи таблиц или графиков, чтобы, изменяя присваиваемое значение, сразу видеть изменение результатов расчета.

В рассматриваемом примере поменяйте значения нагрузок P , q , M_0 , длин L_1 , L и число интервалов разбиения длины балки n . Самостоятельно постройте и отформатируйте графики для функции, заданной на рис. 2.2.

2.3. Символьные вычисления

Наряду с числовыми расчетами Mathcad может производить вычисления в символьном виде. Существует два способа символьных вычислений:

1. С использованием меню **Symbolics** (Символьные вычисления) из главного меню Mathcad.
2. С использованием панели **Symbolic** из математической панели.

Оба способа будут подробно рассмотрены в главе 5. Пример использования меню **Symbolics** приведен в следующем разделе (рис. 2.4), а пока ограничимся наиболее простым и часто используемым методом — применением символьного знака равенства (\rightarrow). В качестве примера рассмотрим вычисление неопределенного интеграла (рис. 2.3). Далее описан порядок символьных вычислений:

- ❑ В математической панели щелкнуть мышью на кнопке **Calculus Toolbar** (Панель вычислений) со значком интеграла.
- ❑ В открывшейся панели **Calculus** (Вычисления) выбрать, щелкнув мышью, шаблон неопределенного интеграла.
- ❑ Заполнить места ввода — вписать интегрируемое выражение или имя интегрируемой функции, а также имя переменной в шаблоне дифференциала.
- ❑ Ввести символьный знак равенства (\rightarrow). Это можно сделать с помощью двух панелей, вызываемых из математической: **Symbolic Toolbar** и **Evaluation Toolbar**. Можно ввести этот оператор и с клавиатуры, нажав комбинацию клавиш **Ctrl+** (точка). На экране появится результат символьного вычисления.

Аналогично можно выполнить все операции, предусмотренные на панели **Calculus**, а именно: вычисление производной любого заданного порядка, опреде-

$$f(z) := \sin(g \cdot z) \cdot \cos\left(\frac{z}{p}\right)$$

p := 2 p := p отмена
предыдущего
численного
значения p

Вместо p1 поставьте p,
оно будет заменено числом

$$\frac{d}{dv} (\sin(v) \cdot \cos(v)) \rightarrow \cos(v)^2 - \sin(v)^2$$

$$\int f(z) dz \rightarrow \frac{-1}{2} \cdot \frac{\cos\left(\frac{g \cdot p + 1}{p} \cdot z\right)}{g \cdot p + 1} \cdot p - \frac{1}{2} \cdot \frac{\cos\left(\frac{g \cdot p - 1}{p} \cdot z\right)}{g \cdot p - 1} \cdot p$$

$$\sum_{n=1}^{\infty} \frac{1}{n^2} \rightarrow \frac{1}{6} \cdot \pi^2 \quad \lim_{t \rightarrow \infty} \frac{\sin(3 \cdot t)^2}{\ln(1 + 2 \cdot t)^2} \rightarrow 0$$

самостоятельно проинтегрируйте и
продифференцируйте функции

$$\tan(t)^2 \quad (\tan(t) + \cot(t))^2 \quad e^{3 \cdot t} \cdot 3^t$$

Вычислите сумму и произведение членов
ряда при n, стремящемся к бесконечности

$$\frac{2 \cdot n + 1}{n^2 \cdot (n + 1)^2}$$

Вычислить пределы функций при t,
стремящемся к нулю

$$\frac{\sin(t) - \operatorname{atan}(t)}{t^3}$$

Рис. 2.3. Символьное вычисление производной, интеграла, сумм, произведений, пределов

ленного и неопределенного интегралов, суммы или произведения ряда, предела выражения. Решите самостоятельно приведенные на рис. 2.3 примеры.

Недостаток использования символьного знака равенства заключается в том, что величины, которым были ранее присвоены численные значения, сохраняют их и при символьном вычислении, то есть вместо символьного вычисления получается численное.

СОВЕТ

Чтобы получить символьное вычисление вместо численного, необходимо отменить предыдущие численные значения всех параметров, присвоив каждому параметру значение самого себя, например: $x := x$ (см. рис. 2.3).

Достоинство использования символьного знака равенства в том, что найденное решение пересчитывается автоматически при изменении выражения или входящих в него величин и участвует в последующих расчетах.

2.4. Решение уравнений

Mathcad позволяет решить любое алгебраическое, а также многие дифференциальные и интегральные уравнения. Произвольно, «с потолка» взятые дифференциальные и интегральные уравнения и системы уравнений вообще могут не иметь решений, и Mathcad не в силах сотворить чудо.

Для примера возьмем квадратное уравнение и найдем его решение вначале символьным, затем численным способом.

2.4.1. Символьное решение

Для символьного решения уравнения нужно:

- ❑ набрать решаемое уравнение и синим уголком курсора выделить переменную, относительно которой нужно решить уравнение;
- ❑ в главном меню выбрать команду **Symbolics** ▶ **Variable** ▶ **Solve** (Символьные вычисления ▶ Переменная ▶ Решить). Появится ответ (рис. 2.4).

Решить уравнение
символьно

С помощью меню
Symbolic

msgMapleSolve

$$2h^2 + h - bb = 0$$

$$\left[\begin{array}{c} \frac{-1}{4} + \frac{1}{4} \cdot (1 + 8 \cdot bb)^{\frac{1}{2}} \\ \frac{-1}{4} - \frac{1}{4} \cdot (1 + 8 \cdot bb)^{\frac{1}{2}} \end{array} \right] \cdot \frac{1}{2}$$

Символьная оценка функции Root

$$\text{root}\{2 \cdot h^2 + h - bb, h\} \rightarrow \left[\begin{array}{c} \frac{-1}{4} + \frac{1}{4} \cdot (1 + 8 \cdot bb)^{\frac{1}{2}} \\ \frac{-1}{4} - \frac{1}{4} \cdot (1 + 8 \cdot bb)^{\frac{1}{2}} \end{array} \right] \cdot \frac{1}{2}$$

Рис. 2.4. Символьное решение уравнения

Недостаток использования меню **Symbolics** заключается в том, что найденное решение не пересчитывается автоматически при изменении выражения или входящих в него величин и не участвует в последующих расчетах.

Достоинством использования меню **Symbolics** является то, что ранее принятые численные значения величин не учитываются в символьных расчетах.

Если выделенное выражение не имеет символьного решения (а большинство уравнений не имеет символьного решения), то Mathcad сообщает об ошибке: «No solution was found» («Решение не найдено»).

В Mathcad есть и другие возможности символьного решения уравнений, например, с использованием функции **root** (см. рис. 2.4, *внизу*). О прочих возможностях символьных вычислений говорится в главе 5.

2.4.2. Численное решение (функция root)

Рассмотрим одну из ряда функций, позволяющих решать алгебраические уравнения, — функцию **root**.

Обращение к функции:

$$\text{root}(f(x), x),$$

где $f(x) = 0$. Возвращает значение x , при котором функция $f(x) = 0$.

Функция `root` решает уравнения итерационным методом секущих и поэтому требует задания перед собой начальных значений. Кроме того, функция `root`, выполняя вычисления методом спуска, находит и выводит только один корень, ближайший к начальному приближению.

Прежде чем решать уравнение, желательно построить график функции $f(x)$ (рис. 2.5). На графике видно, пересекает ли кривая $f(x)$ ось абсцисс, то есть имеет ли действительные корни. Если точки пересечения кривой с осью есть, нужно выбирать начальное приближение поближе к значению корня. Если корней несколько, для нахождения каждого корня надо задавать свое начальное приближение.

```
нахождение двух корней уравнения
x := 1  x0 := root(F(x), x)  x0 = 1.149
x := 5  x0 := root(F(x), x)  x0 = 4.351
Проверка  F(x0) = 0
```

```
Нахождение экстремума
x := 1  начальное приближение
```

```
x1 := root( $\frac{d}{dx} F(x), x$ )  x1 = 2.75
F(x1) = -5.125  экстремум
```

Самостоятельно решить уравнения
численно, символично и найти экстремум

$$x^3 - 3 \cdot x = 0 \quad \ln(e^x - 2) - 3 \cdot x = 0$$

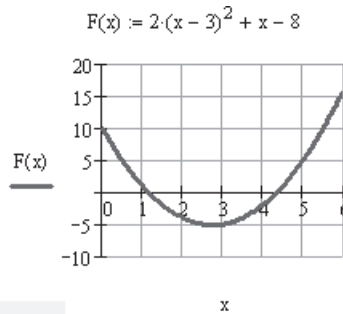


Рис. 2.5. Численное решение уравнения

Если же точек пересечения нет, то корни уравнения могут быть только мнимыми числами. Для их нахождения нужно задавать начальное приближение в комплексной форме. В результате расчета корни уравнения появляются в комплексном виде.

ВНИМАНИЕ

Для ввода мнимой единицы надо на клавиатуре набрать $1i$ или $1j$. При выходе из области мнимого числа единица исчезает, и мы видим комплексное число в обычном виде.

С помощью функции `root` можно найти и экстремум функции, приравняв производную к нулю (см. рис. 2.5). Функции

$$\text{root}\left(\frac{df(x)}{dx}, x\right)$$

должно предшествовать начальное приближение (по умолчанию $\frac{df(x)}{dx} = 0$).

Для нахождения экстремума функции следует:

- ❑ задать начальное приближение поближе к экстремуму;
- ❑ записать выражение с функцией `root`, включив в качестве функции, которая должна быть равна нулю, производную по заданной переменной;
- ❑ вычислить значение заданной функции от найденного корня уравнения $\frac{df(x)}{dx} = 0$. Экстремум найден (см. рис. 2.5).

Функция `root` в старых версиях (до Mathcad 12) позволяет решить уравнение и в символьном виде. При этом задавать начальное приближение не требуется, нужно лишь ввести выражение, содержащее функцию `root`, и выбрать символьный знак равенства, нажав комбинацию клавиш `Ctrl+` (точка). Если символьное решение существует, появится ответ, содержащий сразу все корни уравнения, а не один, как при численном решении уравнения (см. рис. 2.4).

2.5. Решение систем уравнений (функция `find`)

В Mathcad системы уравнений решаются с помощью вычислительного блока `Given–find`. Так как системы уравнений решаются итерационным методом, перед решением необходимо задать начальные приближения для всех неизвестных.

Чтобы решить систему алгебраических уравнений, нужно:

- ❑ задать начальные приближения для всех неизвестных, входящих в систему;
- ❑ напечатать ключевое слово `Given` (Дано). Убедитесь, что при печати вы не находитесь в текстовой области. Если нажать клавишу пробела, то математическое выражение становится текстовой областью и слово `Given` перестает восприниматься как ключевое;
- ❑ ввести уравнения и неравенства, входящие в систему, правее и ниже ключевого слова `Given`. Между левой и правой частями уравнения должен стоять знак равенства. Это не знак присвоения значения, а знак логического равенства. Для его ввода используйте комбинацию клавиш `Ctrl+=` или выберите его на панели `Boolean` (Булевы операторы);
- ❑ введите любое выражение, содержащее функцию `find`. При печати слов `Given` и `find` можно использовать любой шрифт, прописные и строчные буквы.

Обращение к функции:

$$\text{find}(x, y, z, \dots),$$

где x, y, z — неизвестные. Количество неизвестных должно быть равно количеству уравнений. Возвращает значения неизвестных x, y, z , обращающих уравнения в тождества.

Функция `find` может решать и одно уравнение с одним неизвестным как частный случай системы уравнений. Для системы из нескольких уравнений функция `find` выводит решение в виде вектора.

Пример решения системы уравнений приведен на рис. 2.6. Заданная система уравнений состоит из уравнения окружности и уравнения прямой. Если прямая и окружность пересекаются (рис. 2.8), то для нахождения двух точек пересечения нужно задавать два разных начальных приближения. Если окружность и прямая не пересекаются, действительных корней нет. Для нахождения мнимых корней следует задавать начальное приближение в виде комплексного числа.

окружность и прямая пересекаются - действительные решения есть

$x := 1$ $y := 0$ начальное приближение

Given $x^2 + y^2 = 36$ $x + y = 2$ $F := \text{Find}(x, y)$ $F = \begin{pmatrix} 5.123 \\ -3.123 \end{pmatrix}$

Для вывода второго корня надо задаться другим начальным приближением $x = -1$ $y = 0$

Если окружность и прямая не пересекаются - действительных решений нет

$x := 1$ $y := 1$ начальное приближение

Given $x^2 + y^2 = 36$ $x + y = 10$ $F := \text{Find}(x, y)$ $F = \blacksquare$

$x := i$ $y := 0$ начальное приближение
комплексное число (введите $x := 1i$)

Given $x^2 + y^2 = 36$ $x + y = 10$ $F := \text{Find}(x, y)$ $F = \begin{pmatrix} 5 - 2.646i \\ 5 + 2.646i \end{pmatrix}$

Эта переменная не определена.

Рис. 2.6. Решение системы уравнений

Функция `find`, так же как и функция `root`, позволяет решать системы уравнений и в численном, и в символьном виде (рис. 2.7).

ПРИМЕЧАНИЕ

В Mathcad 13–15 функция `root` требует другой формы обращения с 4 параметрами (раздел 5.3.4).

Given - начало ввода системы уравнений

$x^2 + y^2 = aa$ $x + y = bb$

$\text{Find}(x, y) \rightarrow \begin{bmatrix} \frac{1}{2} \cdot bb - \frac{1}{2} \cdot \left(-bb^2 + 2 \cdot aa \right)^{\frac{1}{2}} & \frac{1}{2} \cdot bb + \frac{1}{2} \cdot \left(-bb^2 + 2 \cdot aa \right)^{\frac{1}{2}} \\ \frac{1}{2} \cdot bb + \frac{1}{2} \cdot \left(-bb^2 + 2 \cdot aa \right)^{\frac{1}{2}} & \frac{1}{2} \cdot bb - \frac{1}{2} \cdot \left(-bb^2 + 2 \cdot aa \right)^{\frac{1}{2}} \end{bmatrix}$

Рис. 2.7. Символьное решение системы уравнений

Если система уравнений не имеет точного решения, для приближенного решения в Mathcad существует функция `minerr`.

2.7. Исследование функции на экстремум

Отметим четыре пути поиска экстремума:

1. Для непрерывной функции используем равенство нулю производной от заданной функции. В этой процедуре используют функцию `root`.
2. Для функции с переломами используем функцию `minerr`. Для этого по графику выбираем число заведомо большее (или меньшее) экстремального значения функции и записываем его в качестве ограничения в блоке `Given–minerr`. Функция `minerr` возвращает значение аргумента, при котором расхождение между заданным числом и значением функции минимально. Возвращаемый результат зависит от выбора начального приближения.
3. Для непрерывных функций удобно использовать функции `maximize` и `minimize` (они вводятся аналогично функции `find`). Ключевое слово `Given` обычно можно опускать — оно необходимо лишь при наличии ограничений.
4. Для ступенчатых функций целесообразно использовать функцию автора F_{\max} , реализующую простой метод перебора значений функции (функция F_{\max} описана в разделе 9.8). Другого средства определения максимума ступенчатой функции в Mathcad нет.

СОВЕТ

При анализе конкретного уравнения рекомендуется внимательно изучить график функции, на котором хорошо видны области нахождения экстремумов.

Все четыре способа определения экстремумов функции имеют свои плюсы и минусы. Творческий подход к их выбору почти всегда позволяет правильно найти экстремумы функции.

На рис. 2.9 в качестве функции рассмотрена кубическая парабола и показан ее график с двумя экстремумами (максимумом и минимумом). На рис. 2.10–2.12 показано определение этих экстремумов разными способами. Результаты расчета, естественно, совпадают.

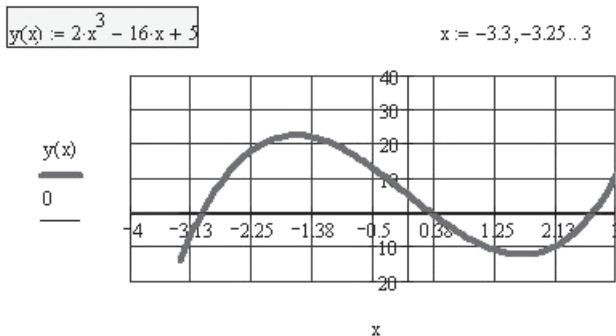


Рис. 2.9. Заданная функция и ее график

$$\begin{array}{l}
 x := -3 \quad \text{начальное значение} \quad x_{\max} := \text{root}\left(\frac{d}{dx}y(x), x\right) \\
 x_{\max} = -1.633 \quad y(x_{\max}) = 22.419 \quad \text{максимум} \\
 x := 3 \quad \text{другое начальное приближение} \quad x_{\min} := \text{root}\left(\frac{d}{dx}y(x), x\right) \\
 x_{\min} = 1.633 \quad y(x_{\min}) = -12.419 \quad \text{минимум}
 \end{array}$$

Рис. 2.10. Нахождение экстремума функции путем приравнивания к нулю ее первой производной

$$\begin{array}{l}
 x := -3 \quad \text{начальное приближение} \\
 \text{Given } y(x) = 30 \quad x_{\max} := \text{Minerr}(x) \\
 x_{\max} = -1.633 \quad y(x_{\max}) = 22.419 \quad \text{максимум} \\
 x := 3 \quad \text{начальное приближение} \\
 \text{Given } y(x) = -20 \quad x_{\min} := \text{Minerr}(x) \\
 x_{\min} = 1.633 \quad y(x_{\min}) = -12.419 \quad \text{минимум}
 \end{array}$$

Рис. 2.11. Нахождение экстремума функции с использованием функции `minerr`

$$\begin{array}{l}
 \text{ключевое слово given необходимо} \\
 \text{только в случае наличия ограничений} \\
 x := -3 \quad \text{начальное значение} \quad x_{\max} := \text{Maximize}(y, x) \\
 x_{\max} = -1.633 \quad y(x_{\max}) = 22.419 \quad \text{максимум} \\
 x := 3 \quad \text{Given } x > 0 \quad x_{\min} := \text{Minimize}(y, x) \\
 x_{\min} = 1.633 \quad y(x_{\min}) = -12.419 \quad \text{минимум}
 \end{array}$$

Рис. 2.12. Нахождение экстремума функции с помощью функций `minimize` и `maximize`

Подробнее о нахождении экстремума функции говорится в разделе 3.7.

2.8. Работа с матрицами

Преимущества Mathcad особенно явно видны при работе с матрицами. Операции с матрицами трудоемки и, как правило, требуют компьютерного программирования. В системе Mathcad мы видим традиционную запись матричных выражений, как на листе бумаги, но уже с готовым численным или символьным ответом.

2.8.1. Создание матриц

Чтобы определить вектор или матрицу, следует:

- записать имя матрицы, ввести оператор присваивания (`:=`);
- в математическом меню выбрать кнопку с изображением матрицы. Откроется панель *Matrix* (Матрица), на которой нужно вновь выбрать кнопку с изображением матрицы. На этот раз откроется диалоговое окно, в котором надо вве-

сти число строк и число столбцов матрицы и нажать кнопку ОК. На экране появится шаблон матрицы. То же действие вызывается нажатием комбинации клавиш $\text{Ctrl}+m$;

- каждое место ввода в шаблоне заполнить числами или буквенными выражениями. Матрица готова.

С помощью шаблона можно ввести матрицу, содержащую не более 100 элементов. Как ввести матрицы больших размеров, рассказано в главе 4 (раздел 4.1).

Вектор — это матрица, состоящая из одного столбца.

Доступ к любому элементу матрицы можно получить, задав имя матрицы с двумя индексами. Первый индекс обозначает номер строки, второй — номер столбца. Произвольный элемент вектора задается одним индексом.

Для набора нижнего индекса можно открыть панель *Vector and Matrix Toolbar* (Матрицы) нажатием соответствующей кнопки на математической панели, после чего щелкнуть на кнопке X_n (Subscript), но лучше использовать клавишу [(открывающая квадратная скобка), так как при работе с матрицами ставить нижний индекс приходится очень часто.

Нумерация элементов массива

Нумерация элементов массива (вектора или матрицы) может начинаться с 0, 1 или с любого другого числа (положительного или отрицательного). Порядком нумерации элементов массива управляет встроенная переменная `ORIGIN`. По умолчанию `ORIGIN = 0`. Это означает, что первый элемент массива имеет номер 0.

Чтобы нумерация членов векторов и матриц начиналась, как обычно принимается в математике, с 1, нужно перед вводом матрицы, а лучше в начале документа, напечатать `ORIGIN:=1` (все буквы прописные).

На рис. 2.13, *вверху* показано создание элементов матрицы D по формуле с использованием нижних индексов. По умолчанию `ORIGIN = 0`, поэтому $D_{0,0} = 10$. После ввода `ORIGIN:=1` элемент $D_{0,0}$ не имеет смысла, а $D_{1,1} = 10$.

2.8.2. Основные действия с матрицами

Mathcad позволяет выполнять с матрицами основные арифметические действия: сложение, вычитание, умножение, а также операции транспонирования, обращения, вычисления определителя матрицы, нахождения собственных чисел и собственных векторов и т. д. Примеры численного и символьного выполнения этих операций приведены на рис. 2.13–2.15.

СОВЕТ

Во время работы с матрицами внимательно следите за размерами матриц. При появлении сообщения о несоответствии размеров матриц (см. рис. 2.13) напишите в сторонке имя матрицы и нажмите клавишу `=`. Если размер матрицы отличается от того, что вы вводили, обнулите матрицу перед очередным оператором присваивания, написав, например, `A:=0`.

ORIGIN = 0 (по умолчанию) $i := 0..2$ $j := 0..4$

Формирование матрицы

$D_{i,j} := 10 - i - j$

$$D = \begin{pmatrix} 10 & 9 & 8 & 7 & 6 \\ 9 & 8 & 7 & 6 & 5 \\ 8 & 7 & 6 & 5 & 4 \end{pmatrix}$$

транспонирование матрицы

$D := D^T$

$$D = \begin{pmatrix} 10 & 9 & 8 \\ 9 & 8 & 7 \\ 8 & 7 & 6 \\ 7 & 6 & 5 \\ 6 & 5 & 4 \end{pmatrix}$$

Влияние ORIGIN на нумерацию элементов матрицы

ORIGIN := 0 $D_{0,0} = 10$

ORIGIN := 1 $D_{0,0} = \blacksquare$ $D_{1,1} = 10$

Индекс массива является недопустимым для этого массива.

$$B := \begin{pmatrix} 3 & 4 & 5 \\ 4 & 5 & 1 \\ 5 & 1 & 2 \\ 1 & 2 & 3 \\ 2 & 3 & 4 \end{pmatrix}$$

$$B + D = \begin{pmatrix} 13 & 13 & 13 \\ 13 & 13 & 8 \\ 13 & 8 & 8 \\ 8 & 8 & 8 \\ 8 & 8 & 8 \end{pmatrix}$$

$$B - D = \begin{pmatrix} -7 & -5 & -3 \\ -5 & -3 & -6 \\ -3 & -6 & -4 \\ -6 & -4 & -2 \\ -4 & -2 & 0 \end{pmatrix}$$

умножение матриц

$B \cdot D = \blacksquare \blacksquare$ перемножение матриц

Размерности массива не совпадают.

Рис. 2.13. Арифметические действия с матрицами

$$v := \begin{pmatrix} 12 \\ 11 \\ 10 \\ 9 \\ 8 \end{pmatrix} \quad v \cdot v^T = \begin{pmatrix} 144 & 132 & 120 & 108 & 96 \\ 132 & 121 & 110 & 99 & 88 \\ 120 & 110 & 100 & 90 & 80 \\ 108 & 99 & 90 & 81 & 72 \\ 96 & 88 & 80 & 72 & 64 \end{pmatrix} \quad \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \cdot (1 \ 2 \ 3) = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & 9 \end{pmatrix}$$

столбец x строку = матрица

$$v^T \cdot v = (510) \quad (1 \ 2 \ 3) \cdot \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} = (14)$$

строка x столбец = число

Рис. 2.14. Произведение векторов

2.8.3. Решение матричных уравнений

Матричные уравнения представляют собой, как правило, систему линейных алгебраических уравнений $A \cdot X = B$ и решаются путем обращения матрицы коэффициентов $X = A^{-1} \cdot B$ (рис. 2.16).

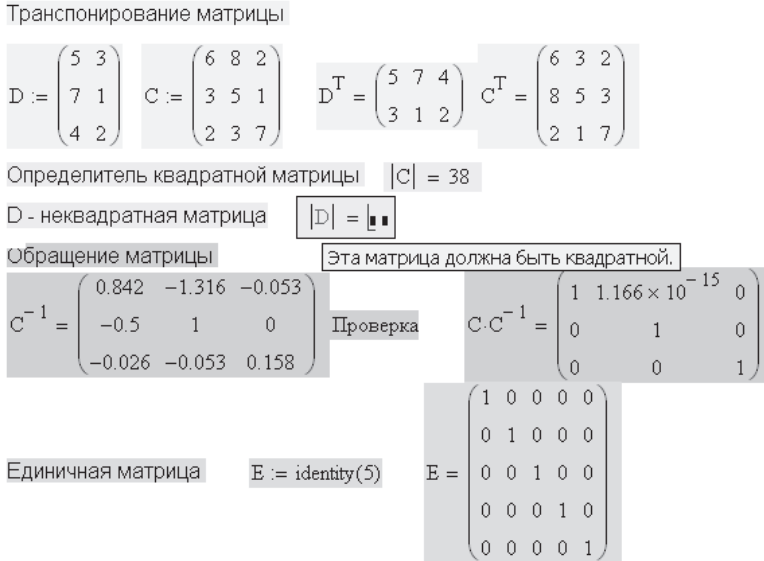


Рис. 2.15. Операции с матрицами

$$A := \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \\ 2 & 1 & 3 \end{pmatrix} \quad B := \begin{pmatrix} 10 \\ 20 \\ 50 \end{pmatrix} \quad X := A^{-1} \cdot B \quad X = \begin{pmatrix} 17.5 \\ -22.5 \\ 12.5 \end{pmatrix}$$

Рис. 2.16. Решение системы алгебраических линейных уравнений путем обращения матрицы коэффициентов

Символьные операции с матрицами можно выполнять с помощью меню **Symbolics** (Символьные операции), а также символьного знака равенства. В примерах, приведенных на рис. 2.17, используется только символьный знак равенства \rightarrow . Подробно о символьных операциях с матрицами рассказывается в главе 5, разделы 5.2.13 и 5.3.

При выполнении символьных операций с матрицами необходимо помнить, что если какому-либо символу ранее присвоено численное значение, то при использовании символьного знака равенства этот символ участвует в символьных расчетах как число. Если символу ранее присвоено значение вектора или матрицы, то символьные вычисления с его участием становятся невозможными. В этом случае нужно использовать для символьных вычислений меню **Symbolics**.

2.9. Оператор векторизации

Mathcad допускает вводить в качестве аргумента функции не только числа, но и векторы. При этом вычисляется значение функции для всех элементов вектора.

$$\begin{array}{l}
 A := \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad B := \begin{pmatrix} ee \\ ff \end{pmatrix} \quad X := \begin{pmatrix} s \\ t \end{pmatrix} \quad B \cdot X^T \rightarrow \begin{pmatrix} ee \cdot s & ee \cdot t \\ ff \cdot s & ff \cdot t \end{pmatrix} \\
 \text{транспонирование} \qquad \qquad \qquad \text{произведение векторов и матриц} \\
 A^T \rightarrow \begin{pmatrix} a & c \\ b & d \end{pmatrix} \quad A \cdot X \rightarrow \begin{pmatrix} a \cdot s + b \cdot t \\ c \cdot s + d \cdot t \end{pmatrix} \quad B^T \cdot X \rightarrow ee \cdot s + ff \cdot t \\
 \text{обратная матрица} \\
 A^{-1} \rightarrow \begin{bmatrix} \frac{d}{(a \cdot d - b \cdot c)} & \frac{-b}{(a \cdot d - b \cdot c)} \\ \frac{-c}{(a \cdot d - b \cdot c)} & \frac{a}{(a \cdot d - b \cdot c)} \end{bmatrix} \quad \text{определитель} \\
 \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad |A| \rightarrow a \cdot d - b \cdot c \\
 \text{собственные} \\
 \text{числа} \\
 \text{eigenvals}(A) \rightarrow \begin{bmatrix} \frac{1}{2} \cdot a + \frac{1}{2} \cdot d + \frac{1}{2} \cdot \left(a^2 - 2 \cdot a \cdot d + d^2 + 4 \cdot b \cdot c \right)^{\frac{1}{2}} \\ \frac{1}{2} \cdot a + \frac{1}{2} \cdot d - \frac{1}{2} \cdot \left(a^2 - 2 \cdot a \cdot d + d^2 + 4 \cdot b \cdot c \right)^{\frac{1}{2}} \end{bmatrix}
 \end{array}$$

Рис. 2.17. Символьные операции с матрицами

Если аргумент функции — матрица, то, чтобы вычислить значения функции для всех элементов матрицы, надо использовать оператор векторизации (рис. 2.18).

Для использования оператора векторизации нужно:

- ❑ ввести выражение или функцию;
- ❑ выделить синим уголком необходимую часть выражения (чаще всего выражение целиком);
- ❑ на математической панели щелкнуть на кнопке Vector and Matrix Toolbar (Матрицы), а в открывшейся панели — на кнопке Vectorize ($\overline{f(M)}$) (Векторизация). Над выделенной частью выражения появится стрелка — символ операции векторизации;
- ❑ нажать клавишу =.

Оператор векторизации изменяет смысл векторной или матричной операции. Векторизация означает выполнение однотипной операции, предписанной выражением, со всеми элементами массива. Например, \sqrt{A} — операция невозможная, если A — вектор или матрица. Как уже сказано, аргумент функции может быть вектором, и функция, как и в случае использования дискретной переменной, вычисляется для всех элементов вектора. Если аргумент функции — матрица, необходимо применение оператора векторизации, чтобы выполнить то же самое действие, то есть вычислить функцию для всех элементов матрицы. В нашем случае это нахождение корня квадратного из каждого элемента матрицы A . Перемножение матриц $A \cdot B$ — это матричное произведение, а $\overline{A \cdot B}$ — это попарное произве-

дение элементов матриц A и B с одинаковыми индексами. Все массивы под знаком векторизации должны быть одного размера, так как операция над всеми массивами производится поэлементно. Примеры использования векторов или матриц в качестве аргументов функции приведены на рис. 2.18.



Рис. 2.18. Операция векторизации

2.10. Решение дифференциальных уравнений

Математически решение дифференциальных уравнений — очень сложная проблема. Mathcad не в состоянии решить без дополнительных упрощений многие дифференциальные уравнения и их системы. Все, что Mathcad может сделать с ними, подробно описано в главе 6. Здесь же рассмотрим использование лишь функции `Odesolve`.

Имя функции `Odesolve` можно писать и с прописной, и со строчной буквы. Алгоритм функции `Odesolve` использует большинство имеющихся в Mathcad функций решения дифференциальных уравнений, фактически заменяя их. `Odesolve` может решать и системы дифференциальных уравнений. В контекстном меню есть возможность выбора метода решения дифференциальных уравнений.

Функция `Odesolve` позволяет записывать уравнение в блоке решения в привычном виде, как обычно записывают уравнение на листе бумаги.

Обращение к функции `Odesolve` требует записи вычислительного блока, который содержит три части.

1. Ключевое слово `Given`.
2. Дифференциальное уравнение и начальные или граничные условия для него или система дифференциальных уравнений и ее условия.
3. Функция `Odesolve(x, xk, n)`, где x — имя переменной, относительно которой решается уравнение; xk — конец интервала интегрирования. Начало интервала интегрирования указано ранее, в начальных условиях; n — необязательный внутренний параметр, определяющий число шагов интегрирования,

используемый при интерполяции решения, то есть при переходе от матрицы численных значений к функции. Параметр n не является обязательным. Его можно удалить, предоставив Mathcad возможность самому выбирать число шагов интегрирования. По умолчанию Mathcad использует $n = 1000$.

ВНИМАНИЕ

Появление других математических выражений в вычислительном блоке между словами **Given** и **Odesolve** недопустимо. Текстовую область внутри вычислительного блока размещать можно. Граничные условия можно задавать лишь в двух точках, одна из которых — начало интервала интегрирования.

Примеры использования функции **Odesolve** приведены на рис. 2.19–2.22.

(исходное уравнение)

$$\frac{d^3}{dx^3}y(x) + x^2 \cdot \frac{d}{dx}y(x) + x \cdot y(x) = e^x \cdot \cos(x)$$

Дифференциальное уравнение с начальными условиями

$$\begin{aligned} \text{Given} \quad & y'''(x) + x^2 \cdot y'(x) + x \cdot y(x) = e^x \cdot \cos(x) \\ & y(0) = -8 \quad y'(0) = 3 \quad y''(0) = 3 \\ y := & \text{Odesolve}(x, 5) \quad x := 0, 0.1..5 \end{aligned}$$

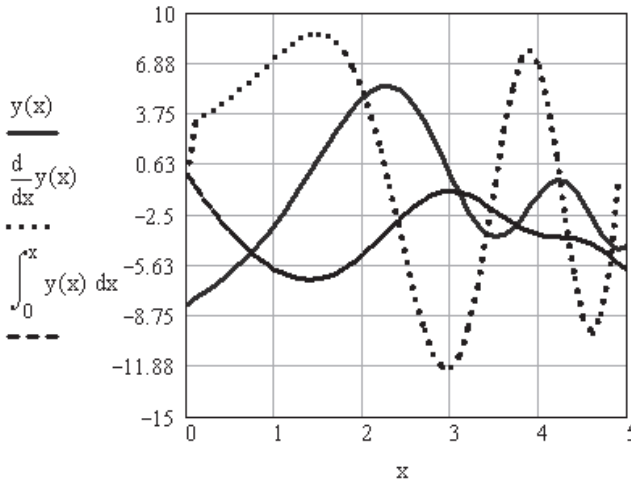


Рис. 2.19. Решение дифференциального уравнения с начальными условиями

дано уравнение $4 \frac{d^2}{dt^2} x(t) + x(t) = t$

граничные условия $x(C) = A$ $x(D) = B$

$A := 4$ $B := 10$ $C := -3$ $D := 6$

Поменяйте граничные условия A,B,C,D

решение уравнения

Given $4 \frac{d^2}{dt^2} x(t) + x(t) = t$

$x(C) = A$ $x(D) = B$ $x := \text{Odesolve}(t, D + 2)$

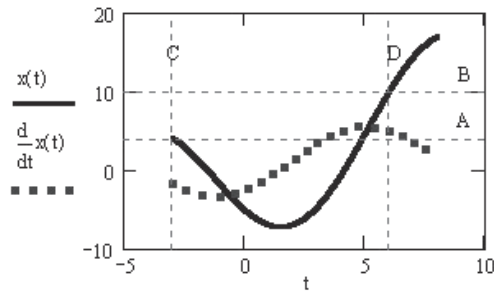


Рис. 2.20. Решение дифференциального уравнения с граничными условиями

ПРИМЕЧАНИЕ

Исходное уравнение можно записывать с использованием как оператора дифференцирования, так и штриха (рис. 2.21). Граничные условия следует записывать только со штрихом. Для набора штриха используйте комбинацию клавиш **Ctrl+F7**.

Функция `Odesolve` возвращает решение дифференциального уравнения в виде функции, а не в виде массива, как все остальные функции, описанные в главе 6, поэтому найденное решение можно интегрировать и дифференцировать (см. рис. 2.19–2.21), а также использовать в последующих расчетах как функцию пользователя.

Функция `Odesolve` решает дифференциальные уравнения как с начальными условиями, когда все условия заданы в начале интервала интегрирования, так и с граничными условиями, заданными в двух точках. Из этих двух точек одна обязательно является началом интервала интегрирования, другая произвольная, но ее аргумент больше, чем в начальной точке. Решение уравнения с начальными условиями показано на рис. 2.19, уравнения с граничными условиями — на рис. 2.20. Решение дифференциального уравнения высокого порядка (4-го) продемонстрировано на рис. 2.21.

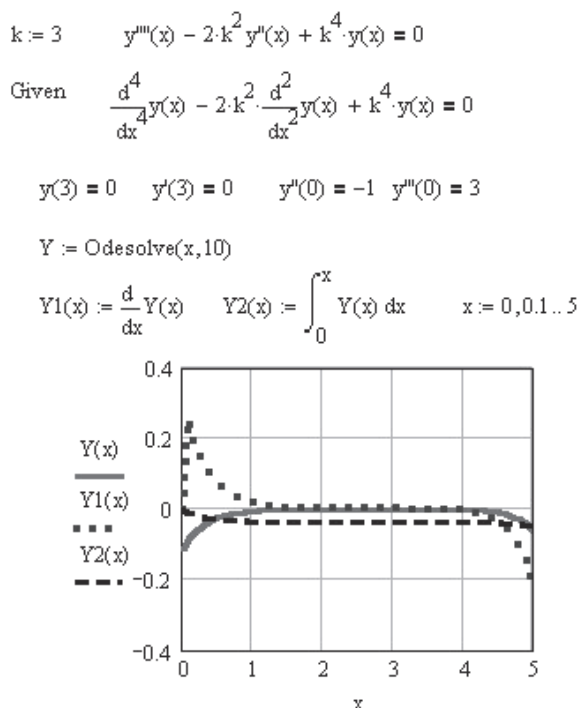


Рис. 2.21. Решение дифференциального уравнения 4-го порядка

$$\text{Given} \quad \begin{array}{ll} 2x(t) + 1y(t) - y(t) + z(t) = 0 & x(0) = 0 \\ 4y'(t) + y(t) = x(t) + z(t) & y'(0) = -1 \\ \frac{d}{dt} z(t) - z(t) = x(t) + y(t) & z(0) = -1 \end{array}$$

+
Текстовая область
между словами
given и odesolve
допустима

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} := \text{odesolve} \left[\begin{pmatrix} x \\ y \\ z \end{pmatrix}, t, 10 \right]$$

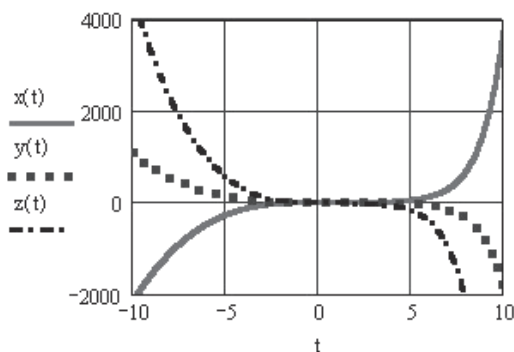


Рис. 2.22. Решение системы дифференциальных уравнений с начальными условиями

Решение системы дифференциальных уравнений приведено на рис. 2.22. При этом обращение к функции `Odesolve` изменилось. Для решения одного уравнения:

$$\text{odesolve}(x, xk, n),$$

для решения системы дифференциальных уравнений:

$$\text{odesolve}(\text{(вектор имен неизвестных)}, x, xk, n).$$

В примере, приведенном на рис. 2.22, при решении системы трех уравнений вектор имен неизвестных содержит три имени.

Потренируйтесь в использовании функции `Odesolve` (рис. 2.23).

Решить самостоятельно

$$100 \cdot \frac{d^2}{dt^2} x(t) + 10 \cdot \frac{d}{dt} x(t) + 101 \cdot x(t) = 50 \cdot \cos\left(\frac{1}{4} \cdot t\right)$$

$$x(0) = 0 \quad x'(0) = 1$$

$$y' = \frac{y}{\ln(y) \cdot \cos(x)} \quad y(0) = 0$$

$$y''' \cdot (x - 1) - y'' = 0 \quad y(2) = 2 \quad y'(2) = 1 \quad y''(2) = 1$$

$$y''' - 2 \cdot y'' + y' = 0 \quad \text{поменяйте граничные условия}$$

Решить систему ОДУ

$$u'(t) - 1 \cdot u(t) + 5 \cdot v(t) = 0$$

$$v'(t) - 5 \cdot u(t) = 1 \cdot v(t)$$

при $u(0) = -1 \quad v(0) = 1$

Рис. 2.23. Примеры для самостоятельной работы

2.11. Анализ экспериментальных данных

При проведении различных экспериментов обычно требуется массив экспериментальных данных представить в виде функции, которую можно использовать в дальнейших расчетах. Если кривая, описываемая этой функцией, должна проходить через все экспериментальные точки, операция получения промежуточных точек и расчетной функции называется *интерполяцией*.

Если кривая, описываемая этой функцией, не должна проходить через все экспериментальные точки и является аппроксимацией (усреднением) исходных данных, операция получения промежуточных точек и расчетной функции называется *регрессией*.

Если необходимо уменьшить разброс данных или исключить некоторую систематическую погрешность, например, в виде наложенных колебаний, используют *сглаживание* данных или фильтрацию спектра колебаний данных.

Подробно о работе с массивами опытных данных рассказано в главе 7. Здесь же поговорим только о наиболее удобных способах интерполяции и регрессии.

2.11.1. Интерполяция

В Mathcad имеется несколько функций интерполяции, различающихся способом «соединения» точек данных (прямой линией или различными кривыми). В этой главе поговорим только о кубической сплайн-интерполяции, при которой экспериментальные точки соединяются отрезками кубических полиномов. В процессе интерполяции одновременно используются две функции, `interp` и `cspline`.

Обращение к функциям:

$$\text{interp}(s, x, y, t); \text{cspline}(x, y),$$

где x — вектор значений аргумента, элементы которого расположены в порядке возрастания; y — вектор значений функции того же размера; s — вектор вторых производных, создаваемый функцией `cspline`, которая обеспечивает равенство вторых производных на границах стыковки полиномов, то есть в экспериментальных точках; t — значение аргумента, при котором вычисляется интерполирующая функция.

Координаты экспериментальных точек не могут быть комплексными числами. Примеры интерполяции приведены на рис. 2.24.

```

Y := (38 15 5.5 23 51 79 105 126 144 156 164 166 162 152 137 117 93 66)
Y := YT    n := 17    i := 0..n    φi := i ·  $\frac{\pi}{9}$     число замеров    n + 1 = 18

YY(x) := interp(cspline(φ, Y), φ, Y, x)    YY(x) - функция, ее можно интегрировать

j := 0..nn    φ1j :=  $\frac{2 \cdot \pi}{nn} \cdot j$     nn - новое число точек, задано глобально
    около графиков. Поменяйте его

Y1 := interp(cspline(φ, Y), φ, Y, φ1)    Y1 - массив -интегрировать нельзя

```

Рис. 2.24. Интерполяция исходных данных

При записи функции интерполяции удобно две функции объединять в одну запись:

$$YY(t) := \text{interp}(\text{cspline}(X, Y), X, Y, t).$$

В результате интерполяции можно получить функцию или массив данных с любым количеством точек интерполяции (см. рис. 2.24). Если результатом является функция, ее можно интегрировать, дифференцировать, использовать в функциях пользователя.

На рис. 2.25 число точек nn массива после интерполяции задано глобально. Для удобства наблюдения за изменением результатов расчета $nn \equiv 5$ помещено около графика. Поменяйте nn .

2.11.2. Регрессия

Смысл регрессии состоит в подборе функции, аппроксимирующей экспериментальные данные. Регрессия сводится к подбору коэффициентов в той или иной аналитической зависимости.

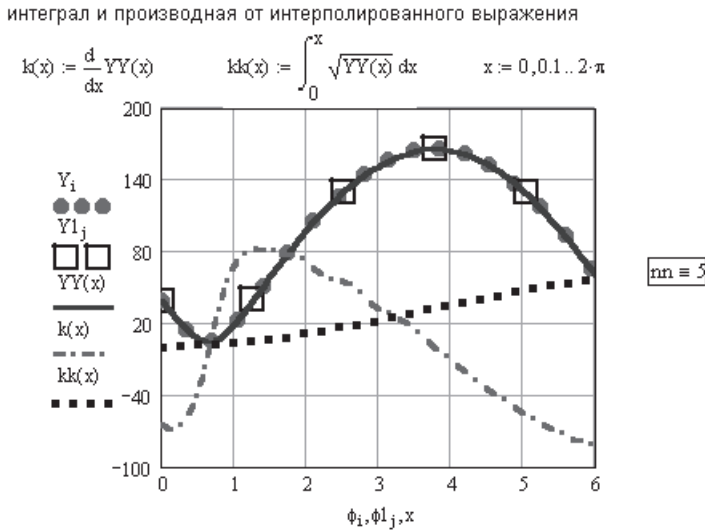


Рис. 2.25. Интерполяция исходных данных (продолжение)

В Mathcad имеется несколько встроенных функций регрессии двух типов:

- позволяющих увидеть аналитическую зависимость, то есть возвращающих набор аппроксимирующих коэффициентов;
- не позволяющих увидеть аналитическую зависимость.

Рассмотрим две функции, которые не выводят коэффициентов и аппроксимируют массив данных одним степенным полиномом или отрезками нескольких полиномов.

В Mathcad регрессия с использованием одного полинома реализуется комбинацией встроенных функций регрессии и интерполяции

$$\text{interp}(s, x, y, t); \text{regress}(x, y, n),$$

где x — вектор значений аргумента, элементы которого расположены в порядке возрастания; y — вектор значений функции того же размера; s — вектор коэффициентов для построения аппроксимирующего полинома, создаваемый функцией `regress`; t — значение аргумента, при котором вычисляется интерполирующая функция; n — степень аппроксимирующего полинома.

Степень аппроксимирующего полинома может быть любой. Практика показывает, что полинома 5-й степени достаточно для аппроксимации почти любой кривой.

Обращение к указанным функциям:

$$s := \text{regress}(X, Y, n),$$

$$YY(t) := \text{interp}(s, X, Y, t)$$

или

$$YY(t) := \text{interp}(\text{regress}(X, Y, n), X, Y, t).$$

Регрессия с использованием нескольких отрезков полинома реализуется комбинацией встроенных функций регрессии и интерполяции $\text{interp}(s, x, y, t)$ и $\text{loess}(x, y, \text{span})$, где $s := \text{loess}(X, Y, \text{span})$ — вектор коэффициентов для построения аппроксимирующего полинома 2-й степени, требуемый функцией interp ; $\text{span} > 0$ — параметр, определяющий размер отрезков полиномов.

Параметр span задает степень сглаженности данных. На практике $0,2 < \text{span} < 2$. При $\text{span} = 2$ результат аппроксимации тот же, что и при аппроксимации одной параболой. При $\text{span} = 0,2$ аппроксимирующая кривая почти точно описывает любой набор данных.

Примеры использования полиномиальной регрессии приведены на рис. 2.26.

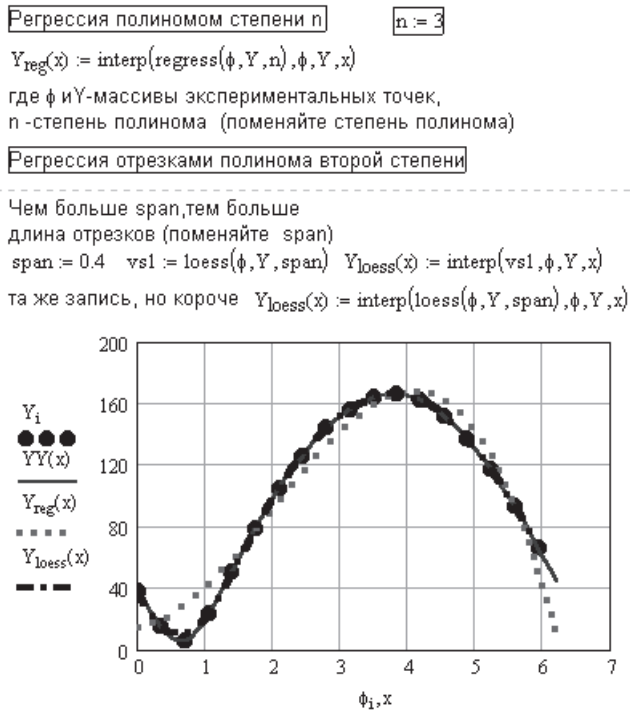


Рис. 2.26. Регрессия исходных данных (см. рис. 2.24) без получения аналитической зависимости

В Mathcad имеется восемь встроенных функций для получения аналитического выражения аппроксимирующей функции. Однако при их использовании необходимо знать форму аналитического выражения. Функции, использующие любой произвольный вид аппроксимирующей функции, будут рассмотрены в главе 7. Здесь же рассмотрим только наиболее простые в применении функции, каждая из которых строит аппроксимирующую функцию лишь определенного вида.

Используйте соответствующий вид регрессии, если хорошо представляете себе, какой зависимостью описывается ваш массив данных. Когда вид регрессии плохо соответствует набору данных, ее результат часто оказывается неудовлетворительным и зависит от выбора начальных приближений.

Из восьми встроенных функций пять требуют предварительного задания вектора начальных приближений:

- `expfit(X, Y, g)` – регрессия экспонентой $f(t) = a \cdot e^{bt} + c$;
- `sinfit(X, Y, g)` – регрессия синусоидой $f(t) = a \cdot \sin(t + b) + c$;
- `pwrfit(X, Y, g)` – регрессия степенной зависимостью $f(t) = a \cdot t^b + c$;
- `lgsfit(X, Y, g)` – регрессия логистической функцией $a(e) = a/(1 + b \cdot e^{-ct})$;
- `logfit(X, Y, g)` – регрессия логарифмической функцией $f(t) = a \cdot \ln(t + b) + c$.

В этих функциях x – вектор значений аргумента, элементы которого расположены в порядке возрастания; y – вектор значений функции того же размера; g – вектор начальных приближений коэффициентов a , b и c ; t – значение аргумента, при котором вычисляется интерполирующая функция.

Примеры использования этих функций приведены на рис. 2.27 и 2.28.

Эти функции требуют начальных приближений

$$\begin{array}{l} \text{expfit -- } y = a \cdot e^{b \cdot x} + c \\ Y1(x) := C_0 \cdot e^{C_1 \cdot x} + C_2 \end{array} \quad c := \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad \begin{array}{l} C := \text{expfit}(\phi, Y, c) \\ \text{corr}(Y1(\phi), Y) = 0.60716588 \end{array}$$

$$\begin{array}{l} \text{sinfit -- } y = a \cdot \sin(x + b) + c \\ Y2(x) := C_0 \cdot \sin(x + C_1) + C_2 \end{array} \quad \begin{array}{l} C := \text{sinfit}(\phi, Y, c) \\ \text{corr}(Y2(\phi), Y) = 0.98538802 \end{array}$$

$$\begin{array}{l} \text{pwrfit -- } y = a \cdot x^b + c \\ Y3(x) := C_0 \cdot x^{C_1} + C_2 \end{array} \quad c2 := \begin{pmatrix} 10 \\ 0.1 \\ 10 \end{pmatrix} \quad \begin{array}{l} C := \text{pwrfit}(\phi, Y, c2) \\ \text{corr}(Y3(\phi), Y) = 0.68961301 \end{array}$$

$$\begin{array}{l} \text{logfit -- } y = a \cdot \ln(x + b) + c \\ Y4(x) := C_0 \cdot \ln(x + C_1) + C_2 \end{array} \quad c1 := \begin{pmatrix} 10 \\ 10 \\ 10 \end{pmatrix} \quad \begin{array}{l} C := \text{logfit}(\phi, Y, c1) \\ \text{corr}(Y4(\phi), Y) = -0.61951307 \end{array}$$

$$\begin{array}{l} \text{lgsfit -- } y = \frac{a}{1 + b \cdot e^{-c \cdot x}} \\ Y5(x) := \frac{C_0}{1 + C_1 \cdot e^{-C_2 \cdot x}} \end{array} \quad c3 := \begin{pmatrix} 10 \\ 1 \\ 1 \end{pmatrix} \quad \begin{array}{l} C := \text{lgsfit}(\phi, Y, c3) \\ \text{corr}(Y5(\phi), Y) = 0.88153268 \end{array}$$

Рис. 2.27. Регрессия исходных данных (см. рис. 2.24) с выводом аналитической зависимости

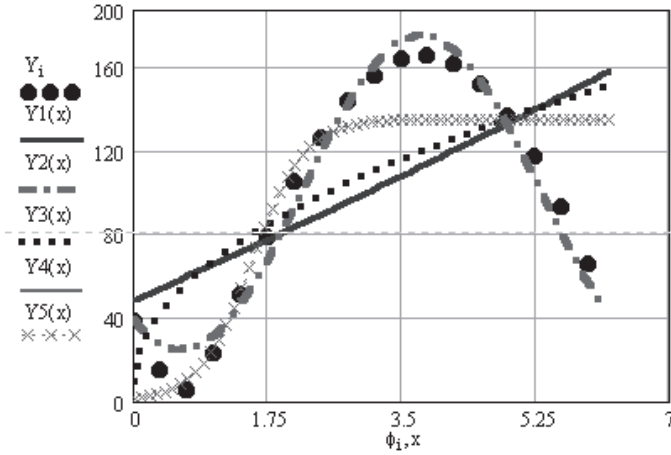


Рис. 2.28. Результаты регрессии исходных данных (см. рис. 2.24) функциями, приведенными на рис. 2.27

Функции, не требующие начальных приближений:

- ❑ `line(X, Y)` – регрессия прямой линией, использующая минимизацию суммы квадратов ошибок $f(t) = a + b \cdot t$;
- ❑ `medfit(X, Y)` – регрессия прямой линией, использующая медиан-медианную линейную регрессию $f(t) = a + b \cdot t$. Функции `line` и `medfit` дают близкие результаты, слегка различающиеся наклоном прямых линий;
- ❑ `Infit(X, Y)` – регрессия логарифмической функцией $f(t) = a \cdot \ln(t) + b$.

Примеры использования этих функций приведены на рис. 2.29 и 2.30.

Эти функции не требуют начальных приближений

$$\begin{aligned}
 \text{medfit } a+b \cdot x \quad C := \text{medfit}(\phi, Y) \quad C &= \begin{pmatrix} 34.146 \\ 23.038 \end{pmatrix} & Y5(x) &:= C_0 + C_1 \cdot x \\
 & & \text{corr}(Y5(\phi), Y) &= 0.62220544 \\
 \text{line } a+b \cdot x \quad C := \text{line}(\phi, Y) \quad C &= \begin{pmatrix} 44.968 \\ 18.538 \end{pmatrix} & Y6(x) &:= C_0 + C_1 \cdot x \\
 & & \text{corr}(Y6(\phi), Y) &= 0.62220544 \\
 \phi_0 &:= 0.01 \\
 \text{Infit } a \cdot \ln(x) + b \quad C := \text{lnfit}(\phi, Y) \quad C &= \begin{pmatrix} 22.194 \\ 86.4 \end{pmatrix} & Y7(x) &:= C_0 \cdot \ln(x) + C_1 \\
 & & \text{corr}(Y7(\phi), Y) &= 0.60503411
 \end{aligned}$$

Рис. 2.29. Регрессия исходных данных (см. рис. 2.24) с выводом аналитической зависимости

Во всех примерах, приведенных на рис. 2.27–2.30, использованы одни и те же массивы данных, характер распределения которых, естественно, не соответствует использованным видам регрессии. Для оценки связи между массивом данных и значениями аппроксимирующей функции подсчитан коэффициент корреляции `corr`. Опытные данные неплохо аппроксимируются синусоидой (коэффици-

ент корреляции 0,98). В остальных случаях связь между величинами плохая (коэффициент корреляции примерно 0,6).

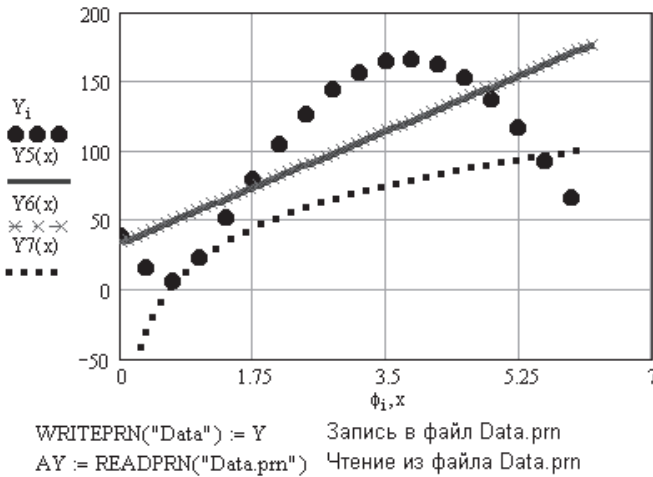


Рис. 2.30. Результаты регрессии исходных данных (см. рис. 2.24) функциями, приведенными на рис. 2.29

2.12. Элементы математической статистики

В Mathcad встроено много функций для решения задач математической статистики. Подробно они рассматриваются в главе 8. Здесь же остановимся лишь на функциях оценки параметров выборки данных. Все перечисленные далее функции можно применять и для векторов, и для прямоугольных матриц. Примеры вычисления средних арифметических и средних геометрических значений, медианы, средних квадратических отклонений и дисперсий показаны на рис. 2.31.

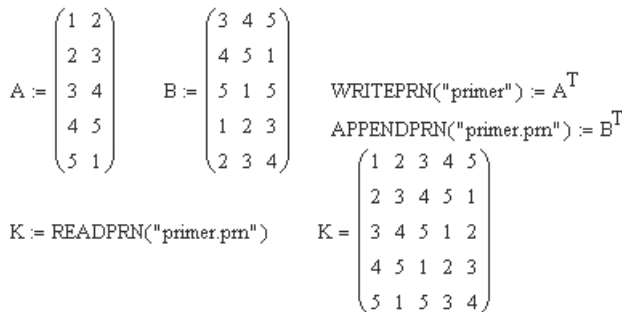


Рис. 2.31. Оценка параметров выборки данных AY

ВНИМАНИЕ

Функции, начинающиеся с прописной и со строчной буквы, — это разные функции.

2.13. Обмен данными с другими программами

В нижней части рис. 2.30 продемонстрирована еще одна очень важная возможность Mathcad — запись числовых данных в файл и считывание их из файла. Появляется возможность передачи данных из одной программы в другую и даже создания вычислительных комплексов из нескольких программ, исполняемых в различных компьютерных системах. Например, данные из Excel передаются в Fortran, из Fortran — в Mathcad и, если нужно, обратно в Fortran.

В Mathcad существует три функции, позволяющие считать числовые данные из других файлов или записать числовые данные в другой файл:

- ❑ APPENDPRN("file") — добавляет числовые данные в существующий файл. Здесь file — имя файла или путь к файлу, если он находится в другом каталоге;
- ❑ READPRN("file") — считывает числовые данные из файла;
- ❑ WRITEPRN("file") — записывает числовые данные в файл.

Все три функции работают только с файлами, в которых числа образуют прямоугольную матрицу (вектор чисел — частный случай прямоугольной матрицы). Пример обмена данными показан на рис. 2.32. Функция WRITEPRN записывает массив A в файл primer, который сама же и создает с расширением .prn. Функция APPENDPRN добавляет в этот же файл массив B . Функция READPRN считывает данные из файла primer.

$$\begin{array}{l}
 A := \begin{pmatrix} 1 & 2 \\ 2 & 3 \\ 3 & 4 \\ 4 & 5 \\ 5 & 1 \end{pmatrix} \quad B := \begin{pmatrix} 3 & 4 & 5 \\ 4 & 5 & 1 \\ 5 & 1 & 5 \\ 1 & 2 & 3 \\ 2 & 3 & 4 \end{pmatrix} \quad \begin{array}{l} \text{WRITEPRN("primer")} := A^T \\ \text{APPENDPRN("primer.prn")} := B^T \end{array} \\
 K := \text{READPRN("primer.prn")} \quad K = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 4 & 5 & 1 \\ 3 & 4 & 5 & 1 & 2 \\ 4 & 5 & 1 & 2 & 3 \\ 5 & 1 & 5 & 3 & 4 \end{pmatrix}
 \end{array}$$

Рис. 2.32. Запись данных в файл и чтение из файла

Для записи прямоугольной матрицы или вектора в отдельный файл необходимо выполнить следующие действия:

- ❑ В стандартном меню выбрать кнопку $f(x)$. Появится окно Insert function (Вставить функцию).
- ❑ Выбрать группу функций File Access (Доступ к файлам).
- ❑ Выбрать имя функции WRITEPRN.
- ❑ В появившийся шаблон вписать имя файла, затем оператор присваивания := и имя числового массива. Массив будет записан в файл с указанным вами именем и расширением .prn и помещен в тот же каталог, в котором находится и рабочий файл.

ВНИМАНИЕ

Перед тем как вписать имя файла, не забудьте ввести символ " (кавычки).

Для передачи в Mathcad данных из какого-либо файла, содержащего прямоугольную матрицу числовых данных, выполните следующие действия:

- ❑ Наберите имя, которое присваиваете массиву, считываемому из файла, и знак присвоения значения :=.
- ❑ В стандартном меню выберите кнопку $f(x)$. Появится окно Insert function.
- ❑ Выберите группу функций File Access.
- ❑ Выберите имя функции READPRN.
- ❑ В появившийся шаблон впишите имя файла с расширением, если оно есть, не забывая перед этим ввести символ " (кавычки).

Подробнее об обмене данными говорится в разделе 12.3.

ПРИМЕЧАНИЕ

В Mathcad 13 появилась функция Readfile, позволяющая считывать данные из любых файлов. Подробно о ней написано в разделе 12.3.

2.14. Учет размерностей в Mathcad

Mathcad позволяет вести расчеты с учетом размерностей. Для этого при вводе исходных данных достаточно умножить число на стандартную размерность. Теперь любые действия с введенными таким образом величинами выполняются с учетом размерностей.

Перед началом работы с размерными величинами надо установить систему единиц, в которой вы будете работать:

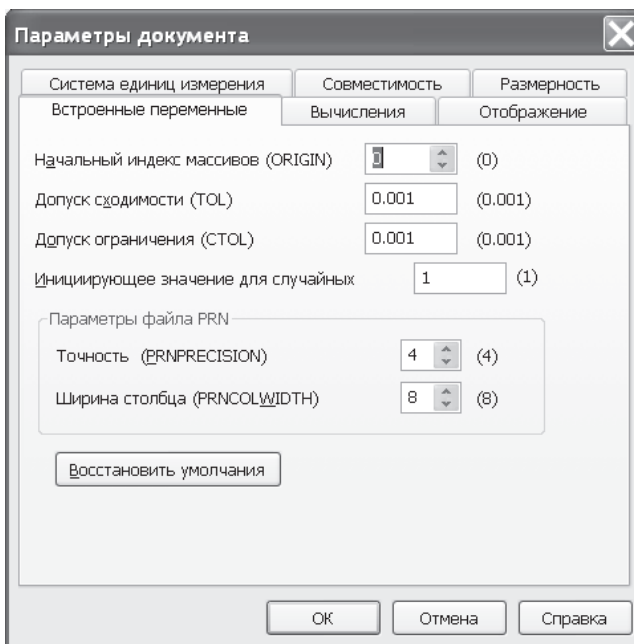


Рис. 2.33. Установка системы единиц

- ❑ В главном меню Mathcad выберите Tools ► Worksheet Options ► Unit System (Параметры документа ► Система единиц измерения ► Встроенные переменные), как показано на рис. 2.33.
- ❑ В открывшемся диалоговом окне выберите при помощи переключателя систему единиц, например SI (International). Щелкните на кнопке ОК. Если в диалоговом окне выбрать None (Нет), то никакие размерности в расчете не учитываются.

В дальнейшем в начале документа при вводе исходных данных для расчета их надо записывать, умножая число на стандартное обозначение размерности (можно ввести его с клавиатуры либо выбрать в стандартном меню Mathcad кнопку Insert (Вставить) (рис. 2.34), щелкнув мышью на соответствующем обозначении).

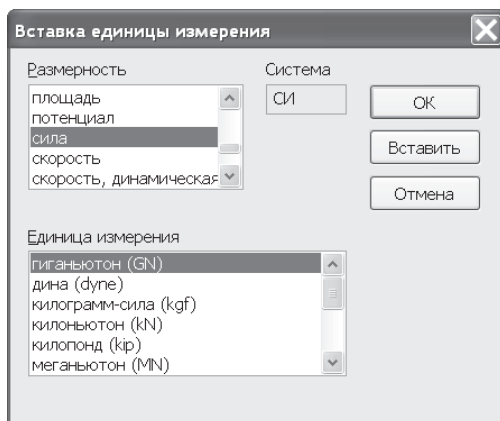


Рис. 2.34. Окно выбора стандартных размерностей

В принципе можно придумать и использовать свои, абсолютно любые размерности. Тогда в начале Mathcad-документа надо ввести свои производные размерности, например размерности на русском языке, присвоив их значениям размерностей, указанных в стандартном меню Mathcad (рис. 2.35).

При выводе результатов расчета после нажатия клавиши = появляется число и рядом с ним — место ввода единицы размерности. Если число уже имеет размерность, то рядом с размерностью появляется еще одно место ввода. Сюда можно поместить любую переменную, константу или выражение.

ВНИМАНИЕ

Не забывайте, что всякий раз, когда что-нибудь находится в месте ввода единицы измерения, Mathcad изменяет полученный результат так, чтобы произведение числа и выражения, стоящего на месте размерности, давало правильное значение.

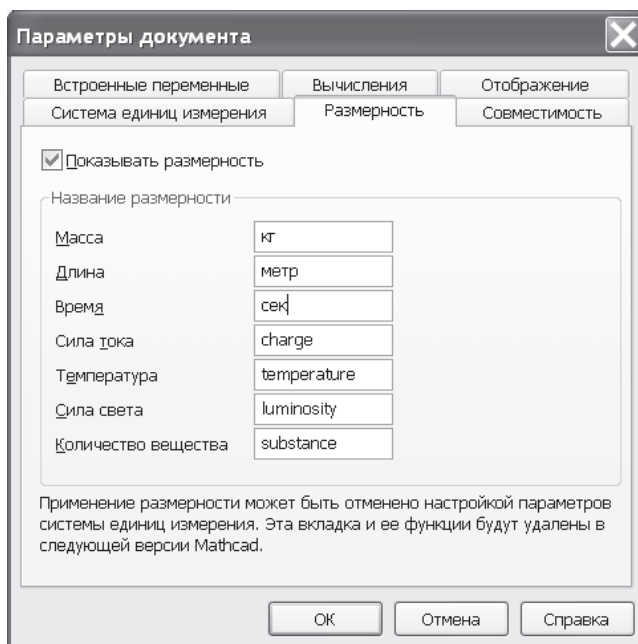


Рис. 2.35. Окно установки своих размерностей

Можно использовать место ввода, чтобы, например, выразить, значение угла в долях числа π или вывести числовой коэффициент при каком-то буквенном выражении.

Размерность элементов массива вводится так же, как и размерность чисел, путем умножения выражения на стандартную размерность на английском языке или производную размерность, введенную ранее.

Дискретная переменная может быть размерной. Для этого в ней необходимо умножить числа на соответствующую размерность.

Графики всегда строятся в стандартных размерностях, приведенных в List of built-in-units (Список встроенных функций). Ввести пользовательские размерности на графиках нельзя. Для изменения масштаба графика разделите на нем названия аргумента и функции на размерность (рис. 2.36).

Ограничения на проведение расчетов с учетом размерностей:

- При расчетах с матрицами все элементы используемых в расчетах матриц должны иметь одинаковую размерность.
- Ряд встроенных функций не может работать с размерными величинами. Это функции регрессии, сглаживания, функция Odesolve (решение дифференциальных уравнений), вычисления логарифма числа (функции ln и log). Для того чтобы преодолеть это ограничение, нужно сделать безразмерными аргументы функций, разделив размерную величину на ее размерность.

- В сложных расчетах при использовании многих функций и нескольких дискретных переменных учет размерностей может оказаться сложным. В таких случаях размерность (для справки) можно указать рядом с результатом расчета в виде текстового комментария.

Подробные сведения об учете размерностей в расчетах можно найти в главе 11.

ПРИМЕЧАНИЕ

В Mathcad 13 введена статическая проверка размерностей, накладывающая гораздо больше ограничений на учет размерностей. Из-за этого ряд программ, работающих в предыдущих версиях, не действует в этой версии. Подробнее о статической проверке размерностей написано в разделе 11.10.

Квазиразмерности

По сути дела процесс присвоения числу размерности состоит в умножении числа на некоторый числовой коэффициент. Буквенному обозначению размерности достаточно присвоить численное значение, не умножая его на стандартную размерность.

Ввод производных размерностей

метр := m сек := s мин := 60·сек час := 3600·s град := deg

Размерные массивы $\Phi := (10 \ 55 \ 200 \ 340 \ 400 \ 420)^T \cdot \text{град}$ угол поворота

$T := (0 \ 3 \ 7 \ 10 \ 15 \ 17)^T \cdot \text{мин}$ время

Размерная функция $\Phi 1(t) := \text{interp}(\text{lspline}(T, \Phi), T, \Phi, t)$ $\Phi 2(t) := \frac{\Phi 1(t)}{t}$

Размерная дискретная переменная $t := 1 \cdot \text{сек}, 2 \cdot \text{мин}, 20 \cdot \text{мин}$ $\Phi 2(t) =$

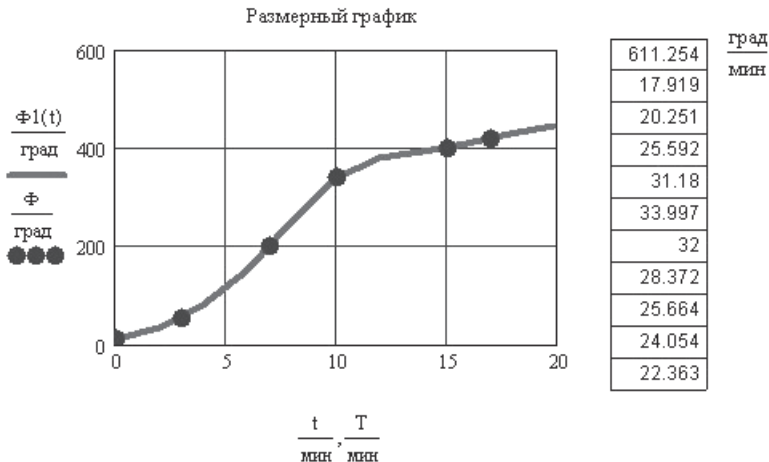


Рис. 2.36. Различные действия с размерными величинами

ПРИМЕЧАНИЕ

Стандартный ввод размерности — производной размерности присваивается число, умноженное на стандартную размерность.

Ввод квазиразмерности — производной размерности присваивается число.

Использование квазиразмерностей позволяет избежать всех ограничений, также позволяет использовать размерности в программах, в матричных расчетах. Пример использования квазиразмерностей показан на рис. 11.19.

При использовании квазиразмерностей Mathcad уже не следит за правильностью использования размерностей и вся ответственность за результаты расчета ложится на расчетчика, который сам должен следить за логикой расчета и вписывать в поле ввода ответа правильную размерность.

2.15. Преобразование функции в матрицу и матрицы в функцию

Преобразование функции в матрицу и матрицы в функцию необходимо в связи с особенностями встроенных функций Mathcad. Это делается, например, если одни функции возвращают матрицу, а вам надо иметь функцию для последующего интегрирования или дифференцирования, или, наоборот, Mathcad выводит функцию, а вам нужна матрица для последующих матричных преобразований.

Заданы векторы аргумента X и функции Y

$$X := (-3 \ 1 \ 5 \ 6 \ 9 \ 10)^T \quad Y := (2 \ 5 \ -3 \ 4 \ 9 \ 2)^T$$

$F_{\text{interp}}(x) := \text{interp}(\text{cspline}(X, Y), X, Y, x)$ интерполяция

$F_{\text{regress}}(x) := \text{interp}(\text{regress}(X, Y, 4), X, Y, x)$ регрессия полиномом 4-й степени

$$x := X_1 .. X_{\text{rows}(X)}$$

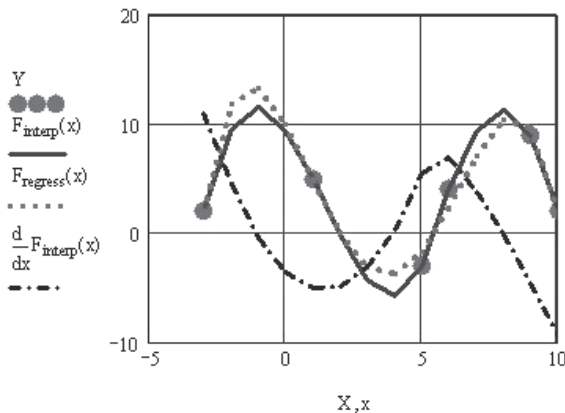


Рис. 2.37. Преобразование вектора данных в функцию

Для преобразования массива данных в функцию используют интерполяцию или регрессию. На рис. 2.37 показано преобразование вектора в функцию. Там же показана возможность дифференцирования полученной функции.

Для получения массива данных с помощью заданной функции надо задать векторы значений аргументов функции и, подставив в функцию значения аргументов, получить массив (рис. 2.38).

$$\begin{aligned}
 & \text{ORIGIN} := 1 \\
 & x := (3 \ 5 \ 9)^T \quad y := (2 \ -20 \ 7 \ 15)^T \\
 & i := 1..rows(x) \quad j := 1..rows(y) \\
 & M_{i,j} := \frac{\ln\left[\frac{x_i^2 + y_j^2}{x_i + y_j}\right]}{x_i + y_j} \quad M = \begin{pmatrix} 0.513 & -0.354 & 0.406 & 0.303 \\ 0.481 & -0.403 & 0.359 & 0.276 \\ 0.404 & -0.561 & 0.304 & 0.238 \end{pmatrix}
 \end{aligned}$$

Рис. 2.38. Получение массива значений с помощью заданной функции

2.16. Строковые функции

Строковые функции в Mathcad позволяют выполнять действия с фрагментами текста, формировать из них необходимую реплику (рис. 2.39). На том же рисунке показана функция CWD, позволяющая узнать путь к рабочему файлу. Эта функция полезна, когда на диске находится несколько копий файла и важно не перепутать, с какой из них идет работа.

```

путь к открытому файлу   CWD = "G:\Быстрый старт\"
n := 6  m := 4  C := "должно быть равно "
A := "Число строк "  B := "Число столбцов "
D1 := num2str(n)  D2 := num2str(m)

Z := | concat(A,C,D1) if n > m
      | concat(A,C,D2) if m > n
      | 1 otherwise

Z = "Число строк должно быть равно 6"

```

Рис. 2.39. Формирование реплики с помощью строковых функций

Подробнее о строковых функциях читайте в разделе 12.12.

2.17. Программирование

Раздел «Программирование» занимает особое место в Mathcad. При начальном обучении этот раздел совершенно не нужен. Огромные возможности Mathcad

позволяют решать подавляющее число задач без использования программирования, да к тому же, как правило, несколькими способами.

Но есть класс задач, которые невозможно решить без программирования. Это задачи, в которых часть документа из нескольких или многих операторов надо выполнить многократно. В таких случаях документ должен состоять из отдельных подпрограмм, объединенных в единую «головную» программу.

Использование раздела «Программирование» позволяет написать в Mathcad программы любой сложности. Подробные сведения о создании программ содержатся в главе 9.

2.18. Анимация

Mathcad предусматривает возможность анимации графиков и результатов вычислений путем создания AVI-файлов и вставки их в Mathcad-документ. Производятся AVI-файлы встроенными в Windows средствами. Анимация не упрощает и не улучшает расчеты, а лишь украшает их презентацию. Поэтому на первых шагах обучения Mathcad обращаться к ней не следует. Подробные сведения о создании анимации к расчетам содержатся в главе 10.

2.19. Отладка Mathcad-документов

Mathcad-документ — это набор исходных данных для расчета, расчетных формул, выведенных на экран результатов расчета в виде чисел, таблиц и графиков. Каждый из перечисленных объектов занимает одну математическую область (Math Region). Между ними в произвольном порядке располагаются текстовые области (Text Region). Все математические объекты взаимодействуют друг с другом. Напомним, что Mathcad вычисляет выражения сверху вниз и слева направо последовательно друг за другом и не переходит к вычислению следующего объекта, не закончив работу с предыдущим. Текстовые области никак не влияют на математические области.

Если в каком-либо выражении есть ошибка, величина, содержащая ошибку, отображается красным цветом. Щелкните мышью на объекте с ошибкой. Под объектом появляется сообщение об ошибке. При нажатии клавиши F1 на экране появляется часть раздела справки (на английском языке) (рис. 2.40).

Щелчок мышью на любом из сообщений открывает расшифровку ошибки аналогично нажатию клавиши F1.

В Mathcad есть функция Trace Errors (Трассировка ошибок), позволяющая проследить всю цепочку ошибочных вычислений, то есть найти место, где была допущена ошибка. Щелкните правой кнопкой мыши на выражении с ошибкой. Если ошибка возникла не в этом выражении, то в контекстном меню появляется

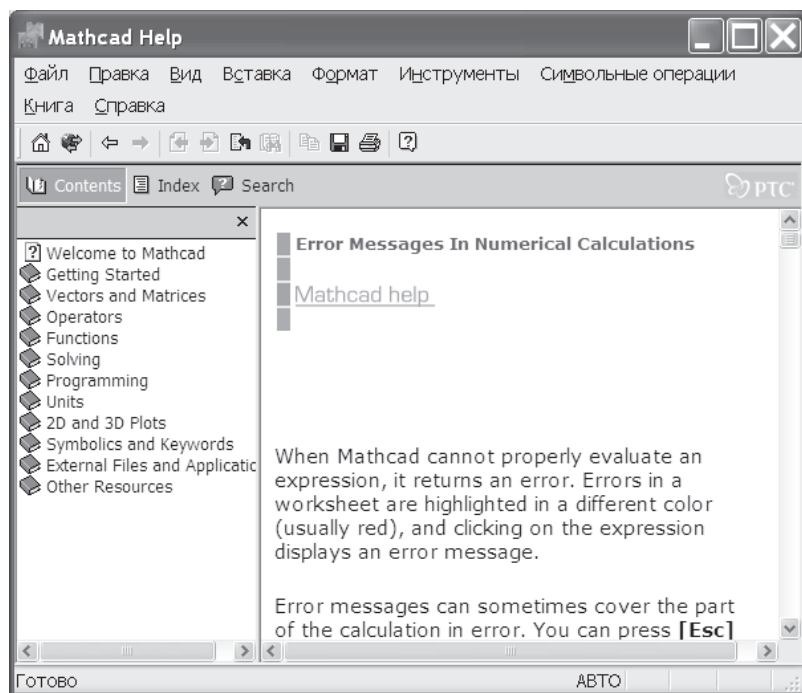


Рис. 2.40. Описание возможных ошибок в меню Help Mathcad 15

пункт Trace Errors (Трассировка ошибок). Выберите этот пункт и, следуя указаниям, найдите выражение, из которого исходит ошибка. На практике эта функция не слишком полезна, так как в простых случаях и без нее понятно, откуда взялась ошибка, а в сложных программах Mathcad сам этого не понимает. В таких случаях в диалоговом окне все кнопки, кроме Close (Закреть), заблокированы (затемнены).

Многие ошибки устраняются легко. Прочитайте сообщение об ошибке, и станет ясно, что делать. Но встречаются ошибки, на устранение которых уходит несколько дней глубоких раздумий. В таких случаях надо просмотреть численные значения всех входящих в ошибочное выражение величин. Возможно, заданные вами значения где-то были заменены другими.

СОВЕТ

Если правильно записанное выражение выводит неправильный числовой ответ, попробуйте «обнулить» имя выражения и значения ряда параметров перед их последним вычислением, так как возможно наложение значений друг на друга.

2.20. «Горячие» клавиши

В Mathcad есть операции, многократно повторяющиеся при наборе программы. Для экономии времени работы желательно запомнить сочетания клавиш, нажатие которых вызывает выполнение таких операций. Ниже приводится таблица

таких сочетаний клавиш («Горячие» клавиши). Полезно распечатать такую таблицу и держать ее на столе перед компьютером в первое время работы с Mathcad. Для удобства распечатки эта таблица включена в папку Интернет-файлов в виде файла гор-клав.doc.

Клавиша	Результат нажатия клавиши (или их сочетания)
Ctrl+R	Освежение экрана (уборка мусора с экрана)
Enter	Вставка чистой строки
Delete	Удаление чистой строки
*, /, +, -	Умножение, деление, сложение, вычитание
^	Возведение в степень
\	Извлечение квадратного корня
	$ M $ – модуль числа, длина вектора, определитель матрицы, если M – число, вектор или матрица
[Нижний индекс у элемента вектора или матрицы. Для ввода нижнего индекса-комментария надо ввести имя переменной, нажать клавишу . (точка), набрать комментарий: M_x или $M_{\text{главное}}$
Латинский эквивалент греческой буквы, далее Ctrl+G	Ввод греческих букв. Греческие буквы/Латинские эквиваленты: $\alpha/a \quad \beta/b \quad \chi/c \quad \delta/d \quad \varepsilon/e \quad \eta/h \quad \gamma/g \quad \lambda/l \quad \mu/m \quad \nu/n \quad \omega/w \quad \phi/f \quad \pi/p \quad \psi/y$ $\rho/r \quad \tau/t \quad \theta/q \quad \xi/x \quad \zeta/z$
e	Основание натурального логарифма
Ctrl+Enter	Сложение с переносом строки (перенос части длинной строки)
: (двоеточие)	$a:=b$ – присвоение значения (локальное), действует направо и вниз от места присвоения (отменяет глобальное присвоение ($a\equiv b$))
~	$a\equiv b$ – присвоение значения (глобальное), действует по всему документу (обычно помещают вблизи графика или таблицы)
Ctrl+=	$a=b$ – жирный знак равенства, (логическое равенство, а не знак присвоения), используется после слова Given
; (точка с запятой)	$x:=1..n$ или $x:=1,1.1..n$ – диапазон чисел дискретной переменной x , где 1 – первое значение, 1.1 – второе значение, n – последнее значение. Разность первого и второго значений определяет шаг выбираемых значений x . Нельзя набирать две точки с клавиатуры!
Ctrl+m	Вставить матрицу
=	Вывод результата численного расчета
Ctrl+. (точка)	→ вывод результата символьного расчета
{	← в программах – локальное присвоение
]	Вертикальная черта – add line – добавка строки в программу
Пробел+ ←, или →	Выделение выражения или его части

Mathcad вычисляет выражения слева направо и сверху вниз.

Для выделения объекта его надо перечеркнуть курсором при нажатой левой кнопке мыши. Далее выделенный объект можно

- переместить при нажатой левой кнопке мыши;
- удалить, нажав клавишу Del;
- переместить в буфер, выбрав «ножницы» на панели инструментов, клавиши Ctrl+x;
- скопировать в буфер, выбрав Copy на панели инструментов, клавиши Ctrl+c;
- вставить из буфера, выбрав Paste на панели инструментов (портфель), клавиши Ctrl+v.

Те же действия можно выполнить, нажав правую кнопку мыши и выбрав в контекстном меню команды Cut, Copy или Paste соответственно.

Часть II

Вычисления в Mathcad

В этой части книги содержатся подробные сведения обо всех возможностях Mathcad, на конкретных примерах рассматриваются особенности использования встроенных функций, основные приемы и способы вычислений.

Глава 3

Решение уравнений

Встроенные функции Mathcad в состоянии решить любое алгебраическое уравнение. Для решения одного уравнения с одним неизвестным можно использовать функции `root`, `polyroot` или `find`. Mathcad решает уравнения или системы итерационным методом, поэтому перед решением необходимо задать начальное приближение для всех корней.

3.1. Функция `root`

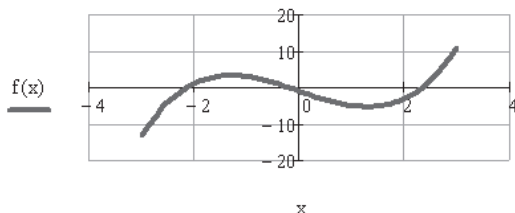
Функция `root` используется для решения одного уравнения с одним неизвестным.

Обращение к функции:

`root(f(x))`, где $f(x)$ — выражение, равное нулю; x — аргумент, варьируя который, система ищет значение, обращающее функцию в нуль (рис. 3.1).

```
Уравнение      x3 - 5·x - 1 = 0
начальное приближение  x := 0
решение      root(x3 - 5·x - 1, x) = -0.202
или          f(x) := x3 - 5·x - 1  x := 0  root(f(x), x) = -0.202

Другие корни   x := 10  x0 := root(f(x), x)  x0 = 2.33
                x := -5  x0 := root(f(x), x)  x0 = -2.128
x := -3, -2.9..3
```



Задан интервал поиска корней

```
root(f(x), x, -5, -1) = -2.128  root(f(x), x, 1, -1) = -0.202
```

```
root(f(x), x, -3, 1) = -0.202  root(f(x), x, 1, 5) = 2.33
```

Значения функции в конечных точках должны иметь противоположные знаки.

Рис. 3.1. Использование функции `root`

Функция $f(x)$ и аргумент x должны быть скалярами, то есть результат вычисления функции — число, а не вектор или матрица. Функция `root` использует итерационный метод секущих. Корень уравнения — ближайшее к начальному приближению значение x , обращающее функцию $f(x)$ в нуль. Если корней несколько,

для отыскания каждого корня необходимо задавать свое начальное приближение.

СОВЕТ

Перед началом решения желательно построить график функции, чтобы проверить, есть ли корни, то есть пересекает ли график ось абсцисс. Начальное приближение лучше всего выбрать по графику поближе к значению корня. Итерационные методы весьма чувствительны к выбору начального приближения.

Mathcad позволяет вместо начального приближения задавать диапазон значений аргумента, в котором лежит значение искомого корня (см. рис. 3.1). В этом случае обращение к функции root должно иметь 4 параметра:

$$\text{root}(f(x), x, a, b),$$

где a и b — границы интервала, в котором лежит корень уравнения. Внутри интервала не должно быть больше одного корня, так как Mathcad выводит на экран лишь один корень, лежащий внутри интервала.

ВНИМАНИЕ

Значения функции на границах интервала должны быть разного знака, иначе, возможно, корень не будет найден (см. последнюю строчку на рис. 3.1).

Если уравнение не имеет действительных корней, то есть функция $f(x)$ нигде не равна нулю, то Mathcad выводит комплексное число. В старых версиях при отсутствии действительного корня надо было вводить начальное приближение в комплексной форме (рис. 3.2).

Если функция имеет мнимый корень,
начальное приближение задается комплексным числом

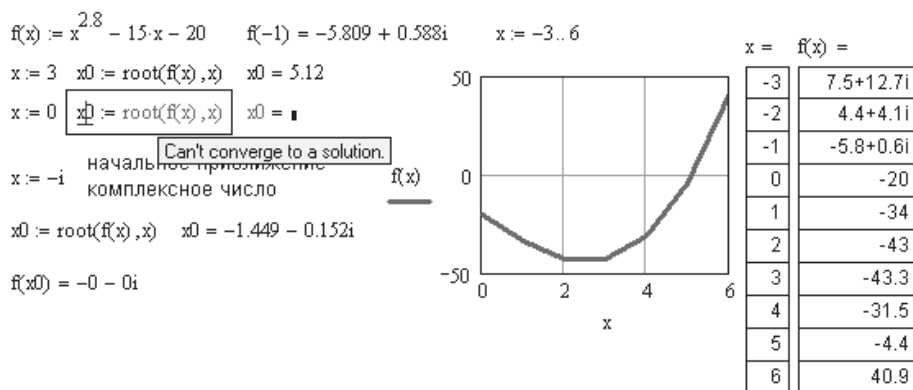


Рис. 3.2. Решение уравнения с комплексными корнями в Mathcad 11

ПРИМЕЧАНИЕ

Для ввода мнимой единицы надо ввести с клавиатуры i или $1j$.

Подробнее об использовании комплексных чисел написано в конце главы 4 (раздел 4.15). Если уравнение имеет несколько корней, то для их нахождения можно использовать разложение функции $f(x)$ на простые множители:

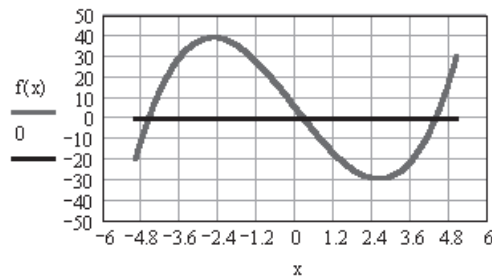
$$f(x) = (x - x_1)(x - x_2) \cdots (x - x_n),$$

где x_1, x_2, \dots, x_n — корни уравнения. Начальное приближение можно задать только для первого корня. В качестве функции $f(x)$ нужно взять

$$h(x)_i = h(x)_{i-1} / (x - x_i),$$

где $h(x)_0 = f(x)$, $h(x)_1 = h(x)_0 / (x - x_1)$ и т. д. (рис. 3.3).

$f(x) := x^3 - 20x + 5$ у этой функции 3 корня
 $x := -5, -4.9..5$ диапазон значений x для вывода графика



```
x := 1          x1 := root(f(x), x)      x1 = 0.2507995
h1(x) := f(x) / (x - x1)  x2 := root(h1(x), x)      x2 = 4.341
h2(x) := h1(x) / (x - x2) x3 := root(h2(x), x)  x3 = -4.593
```

Рис. 3.3. Определение трех корней уравнения

Если функция $f(x)$ имеет малый наклон вблизи искомого корня, то функция $\text{root}(f(x), x)$ может сходиться к значению, довольно далеко отстоящему от корня. В таком случае для уточнения корня необходимо уменьшить значение погрешности вычислений, задаваемое встроенной переменной TOL. Для этого

- в стандартном меню Mathcad выберите команду Tools ▶ Worksheet Options ▶ Built-In Variables (Инструменты ▶ Параметры документов ▶ Встроенные переменные);
- в открывшемся окне поменяйте значение Convergence Tolerance (TOL) (Погрешность сходимости).

Чем меньше константа TOL, тем ближе к нулю будет значение функции при найденном корне уравнения, но тем больше будет время вычисления корня.

Для повышения точности расчета корня можно заменить $f(x)$ на

$$g(x) = f(x) / \frac{d(f(x))}{dx}.$$

Корень можно найти и по графику, увеличив масштаб (рис. 3.4). Для этого необходимо:

- выделить график, щелкнув левой кнопкой мыши внутри графика;
- в главном меню Mathcad выбрать команду Format ▶ Graph ▶ Zoom (Формат ▶ График ▶ Масштаб);
- при нажатой левой кнопке мыши обвести пунктирной линией область графика вблизи искомого корня, которую надо увеличить;
- в открытом окне X-Y Zoom (Масштаб по осям X-Y) нажать кнопку Zoom.

Прямо с графика можно передать в буфер обмена численное значение корня. Для этого выполните следующие действия:

- Выделите график, щелкнув левой кнопкой мыши внутри графика.
- В главном меню Mathcad выберите команду Format ▶ Graph ▶ Trace (Формат ▶ График ▶ Трассировка).
- Щелкните левой кнопкой мыши внутри графика — появится перекрестье осей.
- Двигая мышь при нажатой левой кнопке, установите перекрестье на пересечение графика с осью абсцисс. При этом численные значения координат перекрестья появляются в открытом окне X-Y Trace (Трассировка X и Y).
- Правильно выбрав положение перекрестья, нажмите кнопки Copy X и Copy Y — численные значения будут помещены в буфер (см. рис. 3.4).
- Вне поля графика запишите имя, которое хотите дать корню, и оператор присваивания :=. Нажмите кнопку Paste (Вставить) в стандартном меню Mathcad или в контекстном меню, открываемом при нажатии правой кнопки мыши.

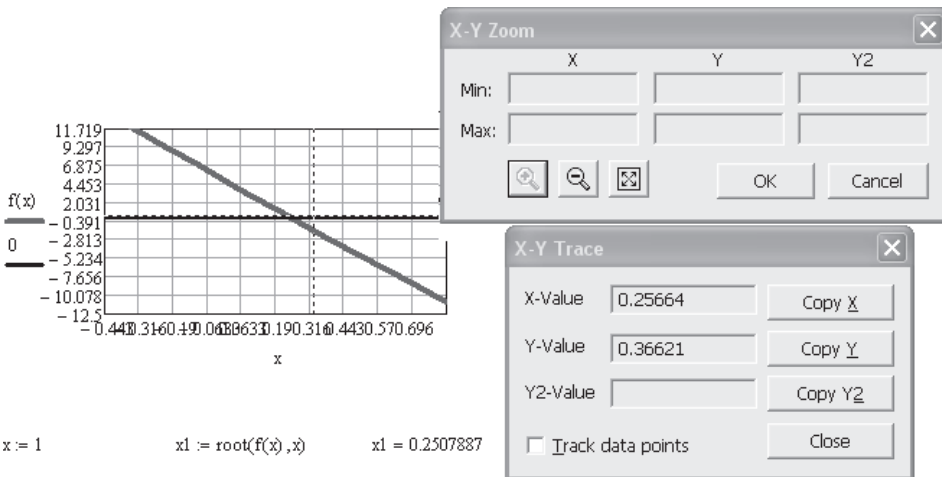


Рис. 3.4. Определение корня уравнения по графику

СОВЕТ

При работе с Mathcad постоянно пользуйтесь правой кнопкой мыши (в контекстном меню каждый раз появляются новые, наиболее нужные в данный момент функции). Щелкните правой кнопкой мыши на графике: в открывшемся контекстном меню есть пункты Zoom (Масштаб) и Trace (Трассировка).

В окне X-Y Trace есть пункт Track Data Points (Отмечать расчетные точки). Если установить этот флажок, при перемещении мыши пунктирное перекрестье на графике будет перемещаться скачками, отмечая расчетные значения функции. Если флажок снять, движение перекрестья становится плавным.

3.2. Решение уравнения с переменными параметрами

Если нужно многократно решать уравнение при изменении одного или нескольких его параметров, то необходимо создать собственную функцию, включающую функции root. Такая функция принимается Mathcad «к сведению». Для ее вычисления надо задать значения (диапазоны значений) параметров, указанных в названии функции. Например, в функции $f(a, x) = ax - \text{tg}(ax)$, варьируя параметр a , находим соответствующий каждому значению a корень уравнения x (обращающий $f(a, x)$ в нуль). Результаты расчета можно вывести в виде вектора решений x_0 или графика $x_0(a)$ (рис. 3.5). Начальное приближение при этом задается лишь один раз. Результат предыдущего вычисления является начальным приближением для последующего.

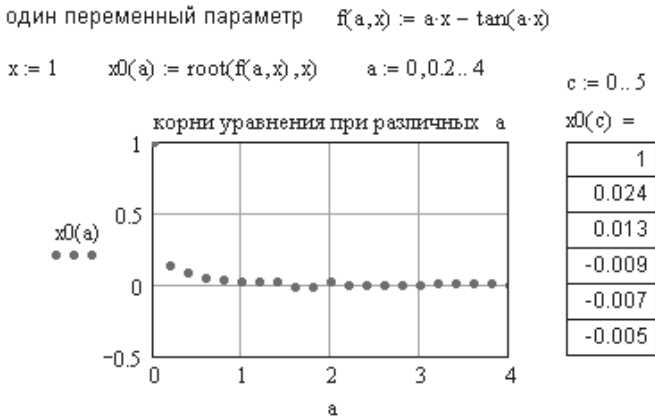


Рис. 3.5. Решение уравнения с одним переменным параметром

Решение уравнения $f(b, c, x) = x^2 + bx - c = 0$, приведенное на рис. 3.6, зависит от численных значений параметров b и c . Задавая значение одного из параметров в виде константы, а другого — в виде дискретной переменной, с помощью функции root можно найти ряд решений уравнения, соответствующих заданным зна-

чениям параметров b и c . В таблице на рис. 3.6 выведены значения корня уравнения x_0 при $c = 4$ и b — ряде значений, определяемых дискретной переменной b . На графике b — ряды значений. Так как функция `root` выводит значение одного корня, а квадратное уравнение имеет два корня, поменяйте значение начального приближения для вывода таблицы и графика, соответствующих второму корню.

```
f(b,c,x) := x^2 + b·x - c      b := 0,2..10
x := 1    поменяйте знак начального приближения
          (ведь у квадратного уравнения два корня),
x0(b,c) := root(f(b,c,x),x)  x0 - решение уравнения
Вывод результатов расчета
```

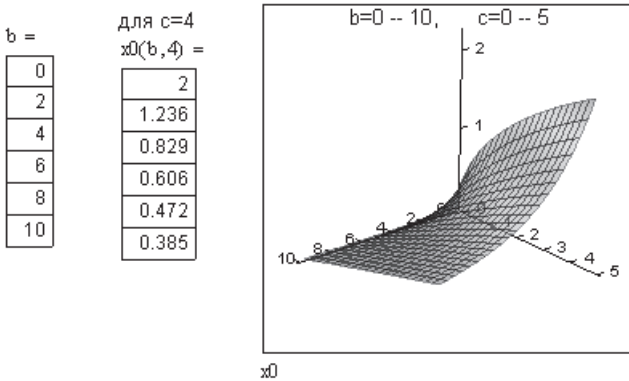


Рис. 3.6. Решение уравнения с двумя переменными параметрами

При построении графика поверхности $x_0(b,c)$ значения параметров $b = 0 - 10$ и $c = 0 - 5$ заданы с помощью меню 3-D Plot Format ▶ Quick Plot Data (Формат 3D-графика ▶ Данные быстрого графика), открывающегося после двойного щелчка мышью в поле графика.

3.3. Нахождение корней полинома. Функция polyroots

Для нахождения корней полинома можно использовать функцию `polyroots(K)`, которая определяет все корни полинома одновременно. Здесь K — вектор коэффициентов полинома, начиная со свободного члена (рис. 3.7). Нулевые коэффициенты опускать нельзя. Если полином имеет N корней (с учетом кратности), то вектор K включает в себя $N + 1$ коэффициент. Начальное приближение вводить не надо.

СОВЕТ

Если исходный полином записан не в развернутом виде, а, например, как произведение полиномов, в этом случае коэффициенты можно определить, используя меню Symbolics (Символьные операции), как показано в главе 5 и на рис. 3.8.

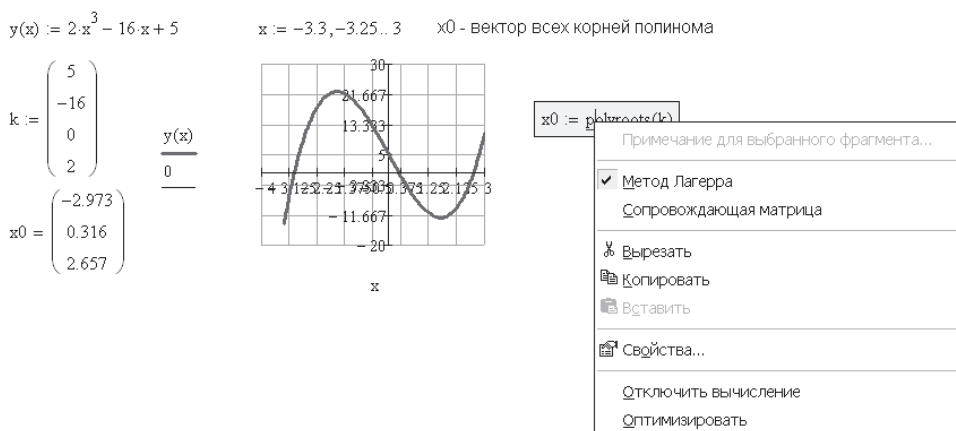


Рис. 3.7. Нахождение корней полинома с помощью функции polyroots

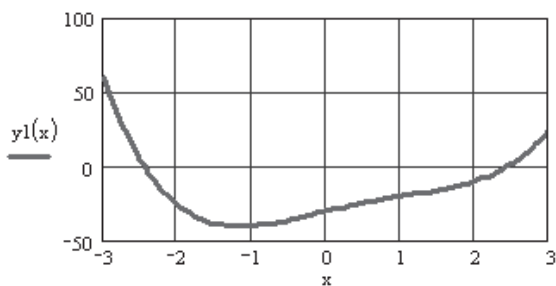
$$y1(x) := (x^2 - 2x + 5) \cdot (x^2 - 6) \quad \text{исходное выражение}$$

для определения коэффициентов используем меню

- выделите x в выражении
- в меню **Symbolic** нажмите **Polynomial Coefficient**

$$(x^2 - 2x + 5) \cdot (x^2 - 6) \text{ has coefficients } \begin{pmatrix} -30 \\ 12 \\ -1 \\ -2 \\ 1 \end{pmatrix} \quad k := \begin{pmatrix} -30 \\ 12 \\ -1 \\ -2 \\ 1 \end{pmatrix}$$

$x := -3, -2.95 \dots 3$



$$x0 := \text{polyroots}(k) \quad x0 = \begin{pmatrix} -2.449 \\ 1 - 2i \\ 1 + 2i \\ 2.449 \end{pmatrix}$$

Рис. 3.8. Нахождение корней полинома при неявном задании коэффициентов

Для функции polyroots можно выбрать один из двух численных методов: метод полиномов Лагерра (он установлен по умолчанию) или метод сопровождающей матрицы (см. рис. 3.7).

Для смены метода сделайте описанные далее операции:

1. Щелкните правой кнопкой мыши на слове `polyroots`, вызвав контекстное меню.
2. В открывшемся меню выберите `LaGerre` (Метод Лагерра) или `Companion Matrix` (Сопровождающая матрица).
3. Щелкните мышью вне поля функции — произойдет пересчет корней в соответствии с выбранным методом.

Чтобы оставить выбор за `Mathcad`, в контекстном меню установите флажок `AutoSelect` (Автоматический выбор).

Графики функций, изображенные на рис. 3.7 и 3.8, построены для проверки правильности определения корней.

3.4. Решение систем уравнений

Для решения систем уравнений нужно использовать вычислительный блок следующего вида:

1. Начальные приближения для всех переменных.
2. Ключевое слово `Given` (Дано).
3. Система уравнений (при записи уравнений надо использовать жирный знак равенства (клавиши `Ctrl+=`), так как это не знак присвоения значения, а оператор отношения).
4. Ограничения на поиск решения в виде неравенств, если они есть.
5. Выражение, содержащее функцию `find`, с неизвестными в качестве параметров.

Результат расчета — вектор решения системы. Вычислительный блок позволяет решать системы, содержащие от 1 до 200 уравнений.

`Mathcad` допускает использование двухсторонних неравенств типа $a \leq x \leq b$. Операторы \leq и \geq выбираются на математической панели.

Решение, выданное функцией `find`, желательно проверить, подставив в уравнения найденные корни, так как в зависимости от начального приближения `Mathcad` может вывести корни, не имеющие физического смысла.

На рис. 3.9 показана проверка решения системы трех уравнений путем подстановки корней в уравнения и путем построения графиков уравнений и определения корней как точек пересечения поверхностей. На графике видна точка Z пересечения трех поверхностей, координаты которой являются решением системы, обращающим все уравнения в тождества.

При обычном ускоренном построении графика поверхности значения аргументов выбираются `Mathcad` автоматически в диапазоне от -5 до 5 , что в нашем примере приводит к делению на нуль и невозможности построения графика. Пределы значений координат можно изменить (см. главу 15). Для этого дважды щелкните мышью в поле графика и в открывшемся окне выберите страницу `Quick Plot Data` (Данные быстрого графика), на которой введите необходимые пределы изменения переменных.

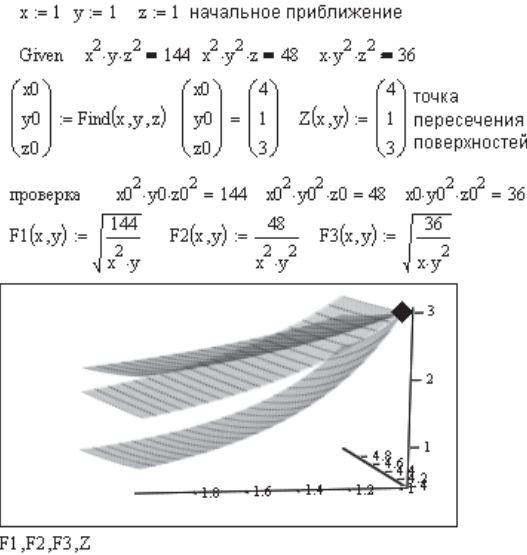


Рис. 3.9. Решение системы уравнений с помощью функции find

Функция find реализует градиентные численные методы и предлагает на выбор три метода. Щелкните правой кнопкой мыши на названии функции find. В открывшемся контекстном меню и его подменю выберите нужный метод (рис. 3.10):

- Linear (Линейный метод) — метод касательной;
- Nonlinear (Нелинейный метод).

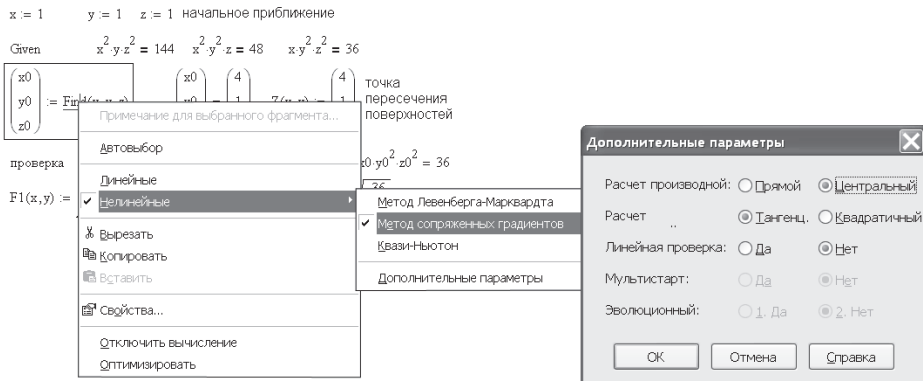


Рис. 3.10. Выбор метода решения системы уравнений

Нелинейных методов три:

- Conjugate Gradient (Метод сопряженных градиентов);
- Quasi-Newton (Квази-ньютоновский метод);
- Levenberg-Marquart (Метод Левенберга).

Кроме выбора самого метода, нажав кнопку **Advanced Options** (Дополнительные параметры), можно выбрать:

- оценку производной конечными разностями (**Derivative Estimation**):
 - Forward** (Вперед) — правая двухточечная схема;
 - Central** (Центральная) — трехточечная симметричная схема;
- оценку переменной (**Variable Estimation**):
 - Tangent** (Касательная) — касательная — прямая линия,
 - Quadratic** (Квадратичная) — касательная — парабола;
- проверку линейности:
 - Yes** (Да);
 - No** (Нет).

Если вы уверены, что нелинейности всех функций, входящих в уравнения, мало влияют на значения их частных производных, выберите при проверке нелинейности пункт **No**. В этом случае производные будут приняты постоянными и не будут вычисляться на каждом шаге, что уменьшит время расчета.

К выбору метода расчета стоит обращаться, если вы хорошо разбираетесь в численных методах и тогда, когда **Mathcad** не может найти решение. В большинстве же случаев лучше доверить выбор метода **Mathcad**, отметив пункт **AutoSelect** (Автоматический выбор).

$$f_1(a, b, x, y) := x^3 \cdot \cos\left(\frac{b}{3} \cdot y\right) - a \cdot b \quad f_2(a, b, x, y) := y^3 - b \sin\left(\frac{a}{3} \cdot x\right)$$

$x := 1 \quad y := 1$ начальные приближения

Given $f_1(a, b, x, y) = 0 \quad f_2(a, b, x, y) = 0$

$z(a, b) := \text{Find}(x, y) \quad x_0(a, b) := z(a, b)_0 \quad y_0(a, b) := z(a, b)_1$

один из параметров делаем константой, другой варьируем

$b := 3$

$a := 0..4$ для таблицы

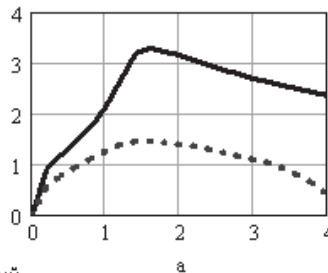
$x_0(a, b) = \quad y_0(a, b) =$

0
2.115
3.138
2.693
2.344

0
1.248
1.375
1.092
0.37

$x_0(a, b)$
—
 $y_0(a, b)$
.....

$a := 0, 0.2..4$ для графика



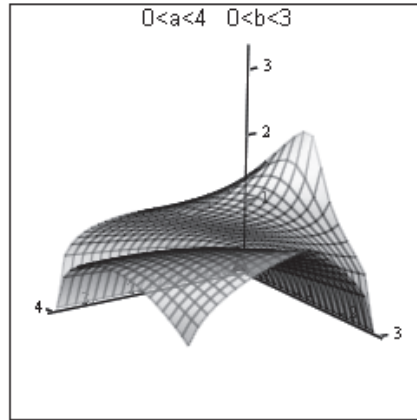
это корни x и y системы уравнений для заданных значений a и b

Рис. 3.11. Решение системы уравнений с переменными параметрами

Градиентные методы, реализованные функцией `find`, требуют многократного вычисления производных. При работе с достаточно гладкими функциями они обеспечивают быстрый и надежный поиск корня.

Как и функция `root`, функция `find` может использоваться в функциях пользователя для нахождения корней системы уравнений при переменных значениях параметров, перечисленных в названии функции пользователя (рис. 3.11 и 3.12). Для системы уравнений решение выводится в виде массива, каждый столбец которого соответствует вектору решения для одной переменной.

Меняем оба параметра, а и b
 Диапазон изменения координат a и b задан на странице Quick Plot Data окна форматирования



$x0, y0$

Рис. 3.12. Поверхность решений системы уравнений (см. рис. 3.11)

Решение системы алгебраических линейных уравнений $A \cdot X = B$ путем обращения матрицы A

$$A := \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \\ 2 & 1 & 3 \end{pmatrix} \quad B := \begin{pmatrix} 10 \\ 20 \\ 30 \end{pmatrix} \quad X := A^{-1} \cdot B \quad X = \begin{pmatrix} 10.833 \\ -9.167 \\ 5.833 \end{pmatrix}$$

Блок решения Given-find
 начальное приближение $X := \begin{pmatrix} 10 \\ 10 \\ 10 \end{pmatrix}$

Given $A \cdot X = B$ $X := \text{Find}(X)$ $X = \begin{pmatrix} 10.833 \\ -9.167 \\ 5.833 \end{pmatrix}$

$\text{root}(A \cdot X - B, X) = \blacksquare$

Функция `root` с матрицами работать не может

Рис. 3.13. Решение системы уравнений в матричном виде

Mathcad позволяет решать системы линейных алгебраических уравнений (СЛАУ) не только в скалярной, но и в матричной форме, при этом начальные условия и ограничения задаются в виде векторов (рис. 3.13).

Для решения СЛАУ предназначена также функция `lsolve`. Она использует более быстрый алгоритм и может работать с неквадратными матрицами, то есть решать недоопределенные и переопределенные системы уравнений с действительными и комплексными числами в матрицах (рис. 3.14).

Обращение к этой функции:

`lsolve(A,B)`, где A —прямоугольная матрица числовых коэффициентов уравнений (действительных или комплексных), B — вектор (или матрица) правых частей уравнений (свободных членов), имеющий то же число строк, что и матрица A .

$$\begin{array}{l}
 A := \begin{pmatrix} 2.3 & 9.8 \\ 24 & 1 \\ 6 & 12 \end{pmatrix} \quad B := \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \quad x := \text{lsolve}(A, B) \quad x = \begin{pmatrix} 0.08 \\ 0.159 \end{pmatrix} \\
 A := \begin{pmatrix} 2.3 & 6.7 & 9.8 \\ 24 & 42 & 1 \\ 6 & 0 & 12 \end{pmatrix} \quad B := \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \quad x := \text{lsolve}(A, B) \quad x = \begin{pmatrix} 0.272 \\ -0.111 \\ 0.114 \end{pmatrix} \\
 A := \begin{pmatrix} 2.3 & 6.7 & 2.8 & 9.8 \\ 24 & 42 & 7 & 1 \\ 6 & 0 & 8 & 12 \end{pmatrix} \quad B := \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \quad x := \text{lsolve}(A, B) \quad x = \begin{pmatrix} 0.156 \\ -0.069 \\ 0.155 \\ 0.068 \end{pmatrix} \\
 A := \begin{pmatrix} 2.3 & 6.7 & 2.8 & 9.8 \\ 24 & 42 & 7 \cdot i & 1 \\ 6 \cdot i & 0 & 8 & 12 \end{pmatrix} \quad B := \begin{pmatrix} 1 \cdot i \\ 2 \\ -3 \cdot i \end{pmatrix} \quad x := \text{lsolve}(A, B) \quad x = \begin{pmatrix} -0.361 - 0.165i \\ 0.194 + 0.105i \\ -0.091 - 0.364i \\ -0.022 + 0.173i \end{pmatrix} \\
 +
 \end{array}$$

Рис. 3.14. Применение функции `lsolve`

ПРИМЕЧАНИЕ

Решение систем уравнений в символьном виде возможно с помощью функции `find` (см. главу 5).

ВНИМАНИЕ

Блоки решения уравнений не могут быть вложены друг в друга. Каждый блок может иметь только одно ключевое слово `Given` и одно имя функции `find`. Внутри блока решения нельзя использовать оператор присваивания вида $x:=1$, а нужно использовать знак логического равенства $=$ (жирный знак равенства).

3.5. Трудности при решении уравнений

Не допускается присутствие между словами Given и find

- других вычисляемых выражений;
- дискретных переменных ($x:=0..20$);
- знака «не равно».

Если Mathcad не может найти решение, то следует:

- признать, что решения не существует;
- уяснить, что корни — мнимые числа, и ввести начальные значения в мнимой форме;
- поискать другое начальное приближение;
- увеличить или уменьшить точность расчетов. Для этого в главном меню Mathcad выберите команду Math ▶ Options ▶ Built-In Variables (Математика ▶ Параметры документа ▶ Встроенные переменные). Уменьшите допустимую погрешность вычислений (Convergence Tolerance) TOL. По умолчанию TOL = 0,001;
- изменить ограничения в блоке Given–find;
- вместо функции find использовать minerr;
- внимательно изучить график функции (возможно попадание в локальный максимум или минимум функции), поискать другое приближение или добавить ограничения на поиск решения;
- посмотреть, не достигнута ли точка, из которой не может быть получено более точное решение, например, когда график функции только касается оси аргументов, не пересекая ее.

3.6. Приближенное решение уравнений и систем уравнений

Если уравнение не имеет точного решения, например, график функции не пересекается с осью аргументов (нет вещественных корней), можно найти значение аргумента, при котором невязка будет минимальной (для случая двух уравнений — минимальное расхождение между двумя кривыми). Для этого используется функция minerr.

Обращение к minerr аналогично обращению к функции find. Функция find дает точное решение, а minerr — приближенное. Функция minerr использует тот же вычислительный алгоритм, что и функция find. При нажатии правой кнопки мыши открывается контекстное меню, позволяющее выбрать метод вычислений.

Используется функция minerr аналогично функции root. Замена в вычислительном блоке Given–find функции find на minerr дает то же решение, если оно существует. Если решения нет, то функция root указывает на ошибку, а minerr находит

минимум невязки, то есть возвращает значение аргумента, соответствующее минимальному расхождению между заданным значением y и функцией $y(x)$.

На рис. 3.15 показано определение координат точек пересечения прямой линии с эллипсом. Точек пересечения две. Одна точка найдена на рисунке. Найдите вторую точку пересечения, взяв другое начальное приближение.

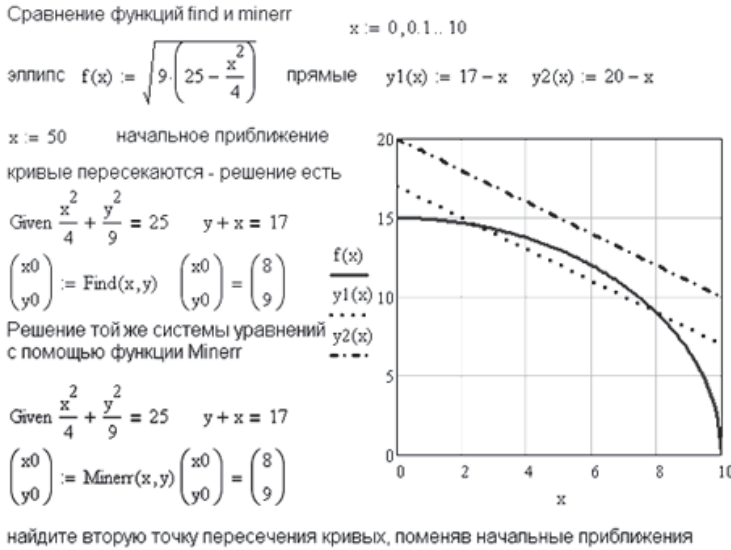


Рис. 3.15. Приближенное решение системы уравнений

При использовании уравнения прямой $x + y = 17$, которая пересекается с эллипсом, обе функции, find и minerr, дают одинаковые результаты.

При использовании уравнения прямой $x + y = 20$, которая не пересекается с эллипсом, функция find выдает ошибку. Функция minerr выдает координаты точки эллипса, наиболее близкой к прямой линии (рис. 3.16).

Функцию minerr удобно использовать для нахождения экстремумов функции.

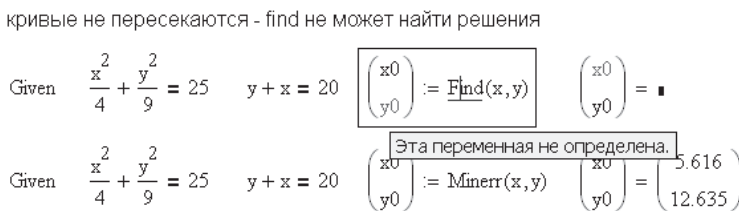


Рис. 3.16. Приближенное решение системы уравнений (продолжение)

При использовании функции `minerr` для решения системы уравнений необходимо проверять полученные результаты. Нередко решение оказывается ошибочным, так как система уравнений имеет несколько корней и Mathcad может предложить корень, не имеющий физического смысла или просто бесполезный. Желательно как можно точнее указывать начальные приближения к решению.

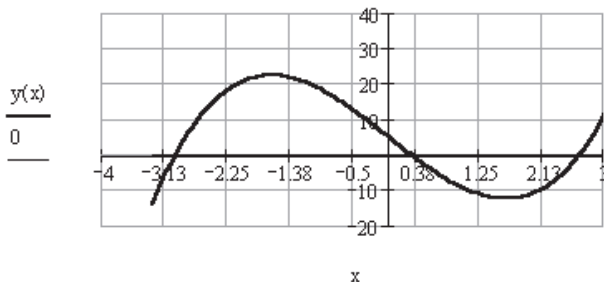
3.7. Исследование функции на экстремум

Поиск экстремума функции включает в себя задачи нахождения локального и глобального экстремумов. Mathcad с помощью встроенных функций решает задачу нахождения только локального экстремума. Для поиска глобального экстремума необходимо вычислить все локальные экстремумы и выбрать из них наибольший (наименьший).

Отметим три пути поиска экстремума:

- Для непрерывной функции используем равенство нулю производной от функции (рис. 3.17).

исходная функция $y(x) := 2 \cdot x^3 - 16 \cdot x + 5$ $x := -3.3, -3.25..3$



$$\begin{array}{lll}
 x := -3 & \text{начальное значение} & x_{\max} := \text{root}\left(\frac{d}{dx}y(x), x\right) \\
 x_{\max} = -1.633 & y(x_{\max}) = 22.419 & \text{максимум} \\
 x := 3 & \text{другое начальное приближение} & x_{\min} := \text{root}\left(\frac{d}{dx}y(x), x\right) \\
 x_{\min} = 1.633 & y(x_{\min}) = -12.419 & \text{минимум}
 \end{array}$$

Рис. 3.17. Нахождение экстремумов функции путем приравнивания производной к нулю

- Для функции с переломами используем функцию `minerr` (рис. 3.18). Для этого по графику выбираем число, заведомо большее (или меньшее) экстремального значения функции, и записываем его в качестве ограничения в блоке `Given`. Функция `minerr` возвращает значение аргумента, при котором расхождение между заданным числом и значением функции минимально, на том участке, где выбрано начальное приближение.

поиск экстремума с помощью функции Minerr

$$y(x) := (x-1)^2 \cdot (x < 4) + (6-x) \cdot (x \geq 4)$$

$x := 0, 0.1..10$
начальное приближение $x := 3$

(свое для каждого экстремума)

Given $y(x) = 10$ $x_{\max} := \text{MinErr}(x)$

$x_{\max} = 4$ $y(x_{\max}) = 9$ максимум
функции

$x := 3$ начальное приближение

Given $y(x) = -10$ $x_{\min} := \text{MinErr}(x)$

$x_{\min} = 1$ $y(x_{\min}) = 0$ минимум
функции

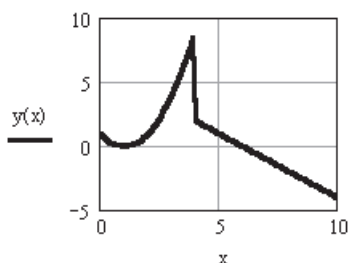


Рис. 3.18. Поиск экстремума ступенчатой функции

ПРИМЕЧАНИЕ

Для определения максимума ступенчатой функции удобно использовать программу Fmax, приведенную в разделе 9.7 (рис. 9.14–9.17).

- Для непрерывных функций удобно использовать функции maximize и minimize (они вводятся аналогично функции find) (рис. 3.19). Ключевое слово Given можно опустить — оно необходимо лишь при наличии ограничений.

$$y(x) := 2 \cdot x^3 - 16 \cdot x + 5 \quad \text{начальное значение} \quad x := -2$$

$$x_{\max} := \text{Maximize}(y, x) \quad x_{\max} = -1.633 \quad y(x_{\max}) = 22.419$$

$$x := -5 \text{ Given } x > 0 \quad x_{\min} := \text{Minimize}(y, x) \quad x_{\min} = 1.633 \quad y(x_{\min}) = -12.419$$

Рис. 3.19. Использование функций maximize и minimize

$$f(x, y) := 2 + 2 \cdot x + 2 \cdot y - x^2 - (3y)^2 \quad \text{поменяйте}$$

$$f(x, y) := (y^3 - x^2) - (1 - x)^3 \quad \text{функции}$$

местами

поиск минимума (максимума) функции двух переменных

Результат зависит от начального приближения,

Поменяйте условия на $x := -2$ $y := 0.5$

$$x := 2 \quad y := 2 \text{ Given } \frac{d}{dx} f(x, y) = 0 \quad \frac{d}{dy} f(x, y) = 0$$

$$\begin{pmatrix} x \\ y \end{pmatrix} := \text{MinErr}(f, x, y) \quad \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 2.215 \\ 0.003 \end{pmatrix} \quad f(x, y) = -3.113$$

то же самое с функциями Maximize и Minimize

$$P := \text{Maximize}(f, x, y) \quad P = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad f(P_0, P_1) = -0.369$$

$$Q := \text{Minimize}(f, x, y) \quad Q = \begin{pmatrix} 2.2153 \\ 0.0015 \end{pmatrix} \quad f(Q_0, Q_1) = -3.113 \quad f$$

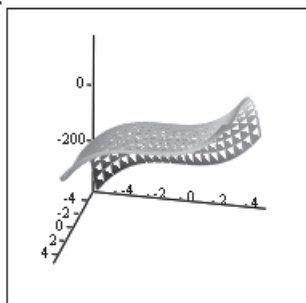


Рис. 3.20. Нахождение экстремума функции двух переменных

Нахождение экстремумов функции нескольких переменных аналогично исследованию функции одной переменной. На примере функции двух переменных рассмотрим влияние ограничений на поиск экстремумов (рис. 3.20 и 3.21). Функции `minimize` и `maximize` способны вычислить, соответственно, минимум и максимум как с ограничениями, так и без них. Результаты решения в значительной степени зависят от выбранных начальных приближений и далеко не всегда соответствуют истине.

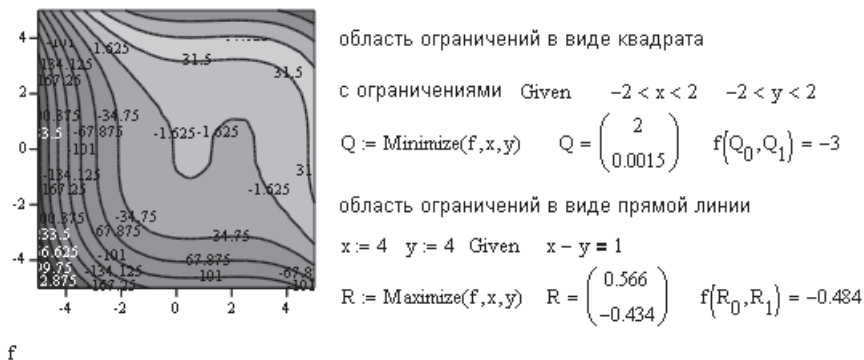


Рис. 3.21. Нахождение экстремума функции двух переменных с ограничениями

СОВЕТ

При анализе конкретного уравнения рекомендуется внимательно изучить поверхностный график функции и график линий уровня, на котором хорошо видны области нахождения экстремумов.

Все три способа определения экстремумов функции имеют свои плюсы и минусы. Творческий подход к их выбору почти всегда позволяет правильно найти экстремумы функции.

3.8. Рекурсивные вычисления

Mathcad допускает внутри тела функции обращение функции к самой себе, позволяя создавать рекурсивные функции, использующие значения, известные на момент вычислений. Примером такой функции является вычисление факториала (рис. 3.22).

На этом же рисунке приведен пример рекурсивного вычисления корня квадратного числа, где последовательность приближений записана в виде вектора. Номер попытки — индекс элемента вектора. В рекурсивном вычислении сначала вычисляется первый элемент массива, затем — последовательные элементы, использующие значения предыдущих элементов.

Для отметки точного вычисления корня на график нанесена метка \sqrt{A} . Для нанесения метки двойным щелчком мыши на графике откройте окно форматирова-

ния графика и на вкладке X-Y Axes установите флажок Show Markers (Показывать маркеры). В одном из появившихся мест ввода впишите \sqrt{A} , а второе оставьте пустым.

Вычисление факториала

```
fact(n) := if(n = 0,1,n*fact(n - 1))
fact(1) = 1      fact(10) = 3628800
```

Вычисление квадратного корня числа A

```
A := 17348      sqrt(A) = 131.711811      i := 0..5
```

```
sqrt_0 := 40    sqrt_{i+1} := (sqrt_i + A/sqrt_i) * 0.5
```

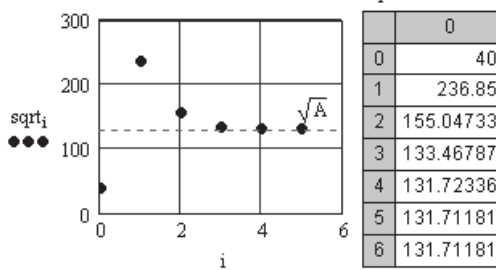


Рис. 3.22. Рекурсивное вычисление корня квадратного числа A

итерационные вычисления модель развития эпидемии i := 0..15 дней

$$\begin{pmatrix} In_0 \\ V_0 \\ L_0 \\ Zd_0 \end{pmatrix} := \begin{pmatrix} 50 \\ 22000 \\ 0 \\ 0 \end{pmatrix} \quad \begin{pmatrix} In_{i+1} \\ V_{i+1} \\ L_{i+1} \\ Zd_{i+1} \end{pmatrix} := \begin{pmatrix} 0.0001 \cdot V_i \cdot In_i \\ V_i - 0.0001 \cdot V_i \cdot In_i \\ L_i + 0.55 \cdot In_i \\ Zd_i + 0.45 \cdot In_i \end{pmatrix}$$

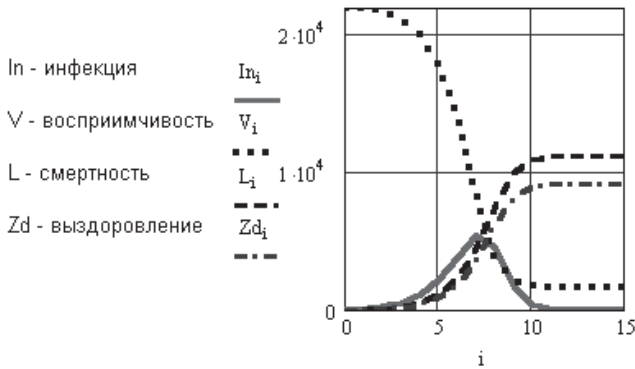


Рис. 3.23. Рекурсивные вычисления в модели распространения инфекции

Можно выполнить многократные рекурсивные вычисления нескольких векторных переменных одновременно. Такая разновидность рекурсивных вычислений используется для решения систем уравнений в методе конечных разностей.

На рис. 3.23 описана модель развития инфекции среди животных. Четыре уравнения связывают друг с другом четыре переменные: число восприимчивых, инфицированных, умерших и выздоровевших животных.

ПРИМЕЧАНИЕ

Mathcad вычисляет все выражения справа перед присваиванием значений величинам слева. Ничто справа не может зависеть ни от чего слева.

ПРИМЕЧАНИЕ

В Mathcad 12 запрещено определять функцию пользователя посредством рекуррентных выражений, например, $f(x) := f(x) + 1$. При попытке вычисления $f(x)$ вместо ее нового (рекуррентного) присваивания, как происходило в предыдущих версиях, будет организован бесконечный цикл. Для организации рекуррентных вычислений используйте новое имя функции, например $f_1(x) := f(x) + 1$, либо именной оператор (namespace operator).

В Mathcad 13 и 14 переопределение функций разрешено.

Глава 4

Работа с векторами и матрицами

Преимущество работы с Mathcad особенно ощутимо при работе с массивами (векторами и матрицами). Как правило, эта работа состоит в многократном повторении однотипных расчетов со всеми элементами матрицы.

Одиночное число будем называть скаляром, столбец чисел — вектором, а прямоугольную таблицу чисел — матрицей. Фактически вектор — это матрица, состоящая из одного столбца чисел.

4.1. Создание матрицы

Есть три способа создания массива чисел:

1. Заполнение шаблона матрицы, содержащего пустые места ввода чисел, что подходит для ввода небольших массивов (не более 100 элементов).
2. Использование дискретной переменной. Этот метод подходит, когда есть явная формула для вычисления элементов массива.
3. Считывание данных из файлов. Этот метод рассматривался в главе 2 (раздел 2.13).

Рассмотрим подробно первый, наиболее простой метод создания матрицы:

- Установите курсор в то место, где надо создать матрицу.
- Щелкните на кнопке математической панели **Vector and Matrix Toolbar** (Векторы и матрицы), а в появившейся панели инструментов щелкните на кнопке **Matrix or Vector** (Матрица или вектор), как показано на рис. 4.1. Можно в главном меню Mathcad выбрать команду **Insert ▶ Matrix** (Вставка ▶ Матрица) или нажать комбинацию клавиш **Ctrl+m**.
- В появившемся диалоговом окне впишите число строк (**Rows**) и число столбцов (**Columns**), а затем щелкните на кнопке **OK**. На месте курсора появится шаблон матрицы.
- В каждое место ввода впишите число, буквенную константу или функцию. Переход от одного места ввода к другому осуществляется клавишами со стрелками. Можно также щелкнуть мышью на нужном месте ввода, но это не так удобно, как перемещение с помощью клавиш.

ПРИМЕЧАНИЕ

Диалоговое окно **Insert Matrix** в Mathcad 13–15 позволяет создавать матрицу из одного числа размерностью 1×1 .

Примеры набора матриц показаны на рис. 4.2.

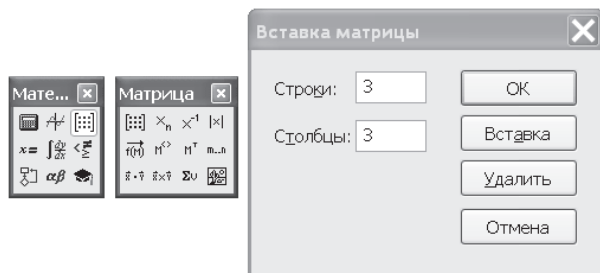


Рис. 4.1. Панели для работы с матрицами

ORIGIN := 1 нумерация элементов в матрицах начинается с 1 (по умолчанию ORIGIN=0) выделение столбца массива

$$A := \begin{pmatrix} 1 & 2 \\ 2 & 3 \\ 3 & 4 \\ 4 & 5 \\ 5 & 1 \end{pmatrix} \quad B := \begin{pmatrix} 3 & 4 & 5 \\ 4 & 5 & 1 \\ 5 & 1 & 2 \\ 1 & 2 & 3 \\ 2 & 3 & 4 \end{pmatrix} \quad V := \begin{pmatrix} 12 \\ 11 \\ 10 \\ 9 \\ 8 \end{pmatrix}$$

элемент массива $A_{1,2} = 2$ $V_3 = 10$ $B^{(2)} = \begin{pmatrix} 4 \\ 5 \\ 1 \\ 2 \\ 3 \end{pmatrix}$

Самостоятельно наберите вектор 3 x 1 и матрицу 3 x 2 из любых чисел и с ними проделайте все последующие операции

Рис. 4.2. Примеры набора массивов и вывода их элементов

Можно изменять размер матрицы, вставляя и удаляя строки и столбцы (рис. 4.3):

- ❑ Щелкните мышью на крайнем элементе матрицы. Синий уголок курсора должен находиться справа от выделенного элемента.
- ❑ Выберите значок матрицы в математическом меню или нажмите комбинацию клавиш **Ctrl+m**. В появившемся окне впишите число строк и столбцов, которые надо вставить или удалить.
- ❑ Щелкните на кнопке **Insert** (Вставить) или **Delete** (Удалить), затем на **ОК**.

Если вы хотите вставить или удалить одну строку, то надо вписать количество строк — 1, количество столбцов — 0 и выбрать пункт **Insert** или **Delete**.

ПРИМЕЧАНИЕ

Mathcad вставляет строки или столбцы вправо и вниз от выделенного элемента матрицы, удаляет строки и столбцы вправо и вниз, включая выделенный элемент.

Чтобы удалить всю матрицу, выделите ее черным цветом и нажмите клавишу **Delete** (для безвозвратного удаления) или выберите **Cut** на стандартной панели Mathcad (чтобы вырезать в буфер). Можно также выбрать команду **Cut** в контекстном меню, появляющемся при щелчке на выделенной матрице правой кнопкой мыши.

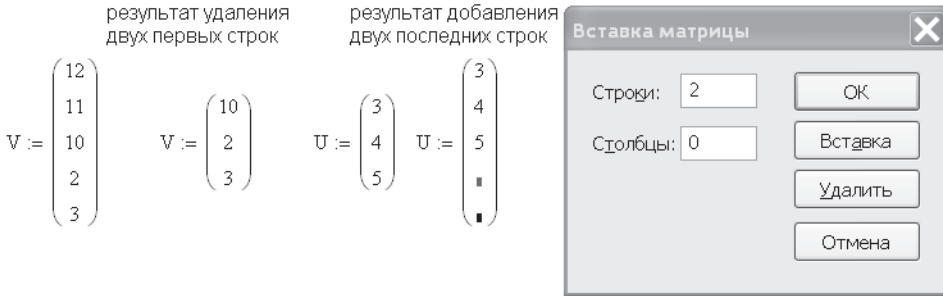


Рис. 4.3. Изменение размеров матрицы

СОВЕТ

При работе с Mathcad почаще нажимайте правую кнопку мыши, проверяя, что можно сделать с помощью контекстного меню. Как правило, в нем появляются самые необходимые в данный момент пункты.

4.2. Нумерация элементов матрицы

Элементы матрицы определяются двумя нижними индексами, элементы вектора — одним.

Для ввода нижнего индекса можно щелкнуть в математической панели на кнопке X_n Subscript (Нижний индекс) (см. рис. 4.1), но лучше использовать клавишу [(открывающая квадратная скобка), так как во время работы с матрицами ввести нижний индекс приходится очень часто.

Чтобы из матрицы выделить вектор (один из столбцов матрицы), используется верхний индекс. Эта операция применяется не очень часто, поэтому для ее выполнения лучше использовать математическую панель инструментов:

- Введите имя матрицы и выделите его синим уголком курсора.
- В математическом меню щелкните на кнопке Matrix and Vectors Toolbar (Панель векторов и матриц) и щелкните на кнопке $M^{<}$ Matrix Column (Столбец матрицы).
- В появившемся месте ввода впишите номер столбца.

4.3. Встроенная переменная ORIGIN

Начало нумерации элементов в векторах и матрицах определяется встроенной переменной ORIGIN.

По умолчанию $ORIGIN = 0$, то есть первый элемент вектора, первая строка и первый столбец матрицы имеют индекс нуль.

Чтобы изменить нумерацию индексов в первой строке документа, наберите (прописными буквами) $ORIGIN:=1$. Можно переопределить встроенную переменную ORIGIN другим способом. Выберите в главном меню команду Tools ▶ Worksheet Options (Инструменты ▶ Параметры документа), в открывшемся окне

перейдите на вкладку **Built-In Variables** (Встроенные переменные) и в поле **Array Origin (ORIGIN)** введите индекс первого элемента массива (рис. 4.4). Переменной **ORIGIN** можно присваивать разные значения, в том числе отрицательные.

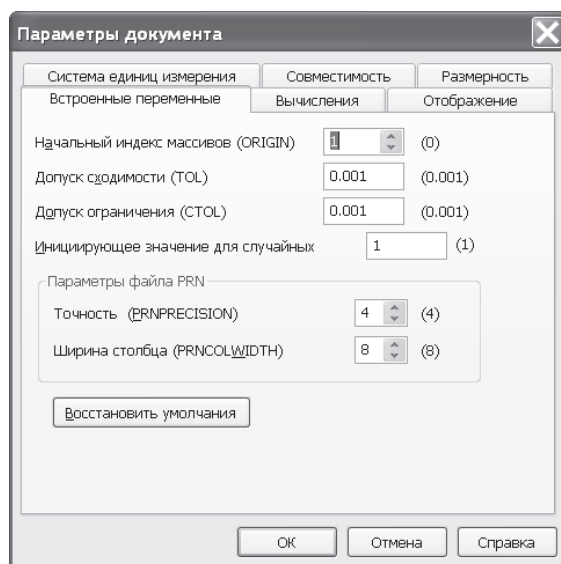


Рис. 4.4. Встроенные переменные Mathcad

В Mathcad есть встроенные функции, в алгоритм которых по умолчанию включены величины с нулевым индексом, например, **mean** (вычисление среднего значения) или **fft** (разложение в ряд Фурье). Если **ORIGIN:=1** введено с клавиатуры, неизбежна ошибка в расчетах. Измените значение встроенной переменной **ORIGIN** в главном меню Mathcad, и ошибки в расчетах не будет.

ПРИМЕЧАНИЕ

Исключение составляет функция **Matrix**. При любом значении **ORIGIN** она возвращает матрицу, нумерация элементов в которой начинается с нуля.

ВНИМАНИЕ

Во всех примерах этой главы установлено значение **ORIGIN = 1**.

4.4. Определение параметров матрицы

В Mathcad есть встроенные функции для определения параметров матрицы (рис. 4.5):

- ❑ **rows(M)** — число строк в массиве или векторе;
- ❑ **cols(M)** — число столбцов в массиве;
- ❑ **last(M)** — индекс последнего элемента в векторе;
- ❑ **max(M)** и **min(M)** — максимальное и минимальное значения элементов в массиве;

- сумма элементов вектора вычисляется нажатием кнопки $\sum V$ на панели Matrix;
- $\text{tr}(M)$ — сумма диагональных элементов квадратной матрицы, называемая следом матрицы, где M — имя матрицы.

Функции для работы с матрицами

$$A := \begin{pmatrix} 1 & 2 \\ 2 & 3 \\ 3 & 4 \\ 4 & 5 \\ 5 & 1 \end{pmatrix} \quad B := \begin{pmatrix} 3 & 4 & 5 \\ 4 & 5 & 1 \\ 5 & 1 & 2 \\ 1 & 2 & 3 \\ 2 & 3 & 4 \end{pmatrix} \quad V := \begin{pmatrix} 12 \\ 11 \\ 10 \\ 9 \\ 8 \end{pmatrix}$$

число строк в массиве $\text{rows}(A) = 5$ $\text{rows}(B) = 5$
число столбцов в массиве $\text{cols}(A) = 2$ $\text{cols}(B) = 3$
индекс последнего элемента в векторе $\text{last}(V) = 5$
max и min элементы в массиве

$$\begin{aligned} \max(A) &= 5 & \min(A) &= 1 \\ \max(V) &= 12 & \min(V) &= 8 \end{aligned}$$

Рис. 4.5. Функции определения параметров массивов

4.5. Образование новых матриц из уже существующих

Рассмотрим встроенные функции для образования новых матриц из уже существующих (рис. 4.6):

- $\text{augment}(A, B)$ объединяет матрицы A и B бок о бок. Матрицы A и B должны иметь одинаковое число строк.

$$A := \begin{pmatrix} 1 & 2 \\ 2 & 3 \\ 3 & 4 \\ 4 & 5 \\ 5 & 1 \end{pmatrix} \quad B := \begin{pmatrix} 3 & 4 & 5 \\ 4 & 5 & 1 \\ 5 & 1 & 2 \\ 1 & 2 & 3 \\ 2 & 3 & 4 \end{pmatrix} \quad C := \text{augment}(A, B) \quad C1 := \text{submatrix}(C, 1, 2, 1, 3)$$

$$C2 := \text{submatrix}(C, 3, 5, 3, 5) \quad \text{поменяйте номера строк и столбцов}$$

$$C = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 4 & 5 & 1 \\ 3 & 4 & 5 & 1 & 2 \\ 4 & 5 & 1 & 2 & 3 \\ 5 & 1 & 2 & 3 & 4 \end{pmatrix} \quad C1 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \end{pmatrix} \quad C2 = \begin{pmatrix} 5 & 1 & 2 \\ 1 & 2 & 3 \\ 2 & 3 & 4 \end{pmatrix}$$

$$D := \text{stack}(C1, C2) \quad D = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 5 & 1 & 2 \\ 1 & 2 & 3 \\ 2 & 3 & 4 \end{pmatrix}$$

Рис. 4.6. Функции объединения и вырезания матриц

- ❑ `stack(A, B)` объединяет матрицы друг над другом. Матрицы A и B должны иметь одинаковое число столбцов.
- ❑ `submatrix(A, irows, jrows, icols, jcols)` создает матрицу, вырезанную из матрицы A . Новая матрица содержит элементы матрицы A , вырезаемые от ряда $irows$ до ряда $jrows$, от столбца $icols$ до столбца $jcols$.

Для ввода указанных функций выполните следующие действия:

- ❑ Наберите имя новой матрицы и оператор присваивания.
- ❑ На стандартной панели выберите кнопку $f(x)$.
- ❑ В открывшемся списке функций найдите нужную и щелкните мышью на кнопке Insert (Вставить). Имя функции можно набирать и с клавиатуры, главное — не сделать в нем ошибки.
- ❑ В места ввода впишите нужные имена массивов.

4.6. Ввод больших массивов

В Mathcad есть ограничения на размер вводимых массивов. При использовании шаблона матрицы можно создать массив, состоящий не более чем из 100 элементов. Если надо ввести массив больших размеров, используются встроенные функции `augment` и `stack`.

Можно ввести массив практически неограниченных размеров с использованием дискретной переменной по какой-либо расчетной формуле либо считать его из внешнего файла функцией `READPRN` (рис. 4.7).

создание массива с помощью дискретной переменной

$n := 4$ $m := 5$ $i := 1..n$ $j := 1..m$ n и m могут быть любыми числами

$$x_i := 0.1 \cdot i \quad y_j := 0.2 \cdot j \quad f_{i,j} := \sin(x_i \cdot y_j) \quad f = \begin{pmatrix} 0.02 & 0.04 & 0.06 & 0.08 & 0.1 \\ 0.04 & 0.08 & 0.12 & 0.159 & 0.199 \\ 0.06 & 0.12 & 0.179 & 0.238 & 0.296 \\ 0.08 & 0.159 & 0.238 & 0.315 & 0.389 \end{pmatrix}$$

поменяйте вид функции и значения n и m

считывание массива из файла с именем "Miu" с помощью функции `Readprn`

$$F := \text{READPRN}(\text{"Miu"}) \quad F^T =$$

	1	2	3	4	5	6	7	8
1	1	2	3	4	5	6	7	8
2	26	4	6	6	30	8	10	10
3	24	26	26	28	28	30	30	32
4	2	2	4	26	6	6	8	30

Рис. 4.7. Способы создания больших массивов

ПРИМЕЧАНИЕ

О дополнительных возможностях преобразования функции в матрицу читайте в разделе 12.5.

4.7. Сортировка векторов и матриц

В Mathcad имеется несколько встроенных функций для сортировки элементов массива в порядке возрастания или убывания:

- $\text{sort}(v)$ — сортировка элементов вектора в порядке возрастания;
- $\text{reverse}(v)$ — перестановка элементов вектора в обратном порядке;
- $\text{csort}(M, i)$ — перестановка строк матрицы M в порядке возрастания элементов i -го столбца;
- $\text{rsort}(M, i)$ — перестановка столбцов матрицы M в порядке возрастания элементов i -й строки.

ПРИМЕЧАНИЕ

Если элементы вектора или матрицы — комплексные числа, то сортировка выполняется по действительной части, а мнимая часть игнорируется.

Примеры сортировки элементов массивов приведены на рис. 4.8.

Функции сортировки

$$\begin{array}{l}
 V := \begin{pmatrix} 12 \\ 5 \\ 10 \\ 9 \\ 8 \end{pmatrix} \quad \text{sort}(V) = \begin{pmatrix} 5 \\ 8 \\ 9 \\ 10 \\ 12 \end{pmatrix} \quad \text{reverse}(\text{sort}(V)) = \begin{pmatrix} 12 \\ 10 \\ 9 \\ 8 \\ 5 \end{pmatrix} \\
 B := \begin{pmatrix} 3 & 4 & 5 \\ 4 & 5 & 1 \\ 5 & 1 & 2 \\ 1 & 2 & 3 \\ 2 & 3 & 4 \end{pmatrix} \quad \text{csort}(B, 2) = \begin{pmatrix} 5 & 1 & 2 \\ 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 5 \\ 4 & 5 & 1 \end{pmatrix} \quad \text{rsort}(B, 5) = \begin{pmatrix} 3 & 4 & 5 \\ 4 & 5 & 1 \\ 5 & 1 & 2 \\ 1 & 2 & 3 \\ 2 & 3 & 4 \end{pmatrix}
 \end{array}$$

Рис. 4.8. Примеры сортировки элементов массивов

4.8. Матричные операторы

Рассмотрим лишь наиболее часто употребляемые операторы.

4.8.1. Транспонирование матрицы

Транспонированием называется операция, в результате которой столбцы исходной матрицы становятся строками, а строки — столбцами.

Для реализации этой и последующих операций выполните следующие действия:

- Наберите имя матрицы.
- Щелкните на кнопке со значком матрицы на математической панели.
- На панели Matrix (Матрицы) щелкните мышью на значке соответствующей операции, в данном случае M^T .

При вводе больших векторов из соображений экономии места удобно вводить их в виде строки с последующим транспонированием (рис. 4.9).



Рис. 4.9. Транспонирование матрицы и вычисление ее определителя

Все матричные операторы и матричные функции работают только с векторами (в виде столбца), но не со строками, поэтому строки вначале приходится транспонировать в столбец, а после выполнения нужной операции вновь транспонировать в строку.

ПРИМЕЧАНИЕ

В Mathcad 12 при преобразовании строки в столбец имена строки и столбца должны быть разными.

4.8.2. Вычисление определителя матрицы

Для нахождения определителя заданной матрицы на панели **Matrix** выберите значок $|X|$, имеющий тройное значение:

- вычисление определителя матрицы, если X — матрица;
- модуль числа, если X — вещественное или комплексное число (см. рис. 4.9);
- длина вектора, если X — вектор.

В Mathcad вычисление определителя матрицы и вывод модуля числа (его положительного значения) производится одним и тем же оператором $|M|$. В Mathcad 11 действие этого оператора определялось здравым смыслом: для квадратной матрицы вычислялся *определитель*, для числа или вектора чисел, а также как результат действия дискретной переменной выводилась *абсолютная величина чисел* или *длина вектора*.

В Mathcad 13–15 требуется в контекстном меню выбрать необходимое действие (рис. 4.10). Неприятный момент состоит в том, что этот выбор не сохраняется при сохранении файла. При открытии файла с таким оператором расчет останавливается на этом операторе, указывая на ошибку. Требуется вмешательство рас-

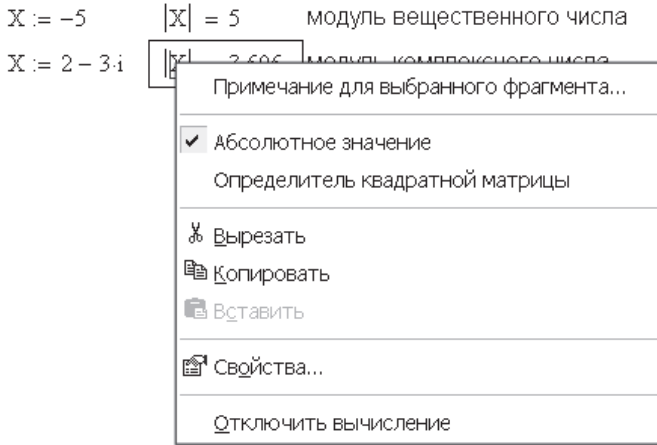


Рис. 4.10. Вычисление модуля числа

четчика, чтобы вручную ввести нужное действие. При сохранении файла в формате xmsd подобной ошибки нет.

4.8.3. Нахождение матрицы, обратной заданной

Для нахождения матрицы, обратной заданной, выберите значок X^{-1} на панели Matrix.

Произведение прямой матрицы на обратную есть единичная матрица (рис. 4.11). Иногда единичная матрица необходима для решения матричных уравнений. Для создания единичной матрицы в Mathcad есть встроенная функция $\text{identity}(n)$, где n — размер квадратной матрицы.

Обращение матрицы

$$C^{-1} = \begin{pmatrix} -0.187 & 0.013 & 0.013 & 0.013 & 0.213 \\ 0.013 & 0.013 & 0.013 & 0.213 & -0.187 \\ 0.013 & 0.013 & 0.213 & -0.187 & 0.013 \\ 0.013 & 0.213 & -0.187 & 0.013 & 0.013 \\ 0.213 & -0.187 & 0.013 & 0.013 & 0.013 \end{pmatrix}$$

Проверка $C \cdot C^{-1} =$

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

Единичная матрица $E := \text{identity}(5)$

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

Рис. 4.11. Вычисление матрицы, обратной заданной

4.8.4. Сложение, вычитание и умножение матриц

Сложение и вычитание матриц выполняется с помощью традиционных операторов сложения и вычитания «плюс» и «минус» (рис. 4.12).

Сложение, вычитание, умножение матриц

$$\begin{aligned}
 B &= \begin{pmatrix} 3 & 4 & 5 \\ 4 & 5 & 1 \\ 5 & 1 & 2 \\ 1 & 2 & 3 \\ 2 & 3 & 4 \end{pmatrix} & D &= \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 5 & 1 & 2 \\ 1 & 2 & 3 \\ 2 & 3 & 4 \end{pmatrix} & B + D &= \begin{pmatrix} 4 & 6 & 8 \\ 6 & 8 & 5 \\ 10 & 2 & 4 \\ 2 & 4 & 6 \\ 4 & 6 & 8 \end{pmatrix} \\
 B \cdot D^T &= \begin{pmatrix} 26 & 38 & 29 & 26 & 38 \\ 17 & 27 & 27 & 17 & 27 \\ 13 & 21 & 30 & 13 & 21 \\ 14 & 20 & 13 & 14 & 20 \\ 20 & 29 & 21 & 20 & 29 \end{pmatrix} & B - D &= \begin{pmatrix} 2 & 2 & 2 \\ 2 & 2 & -3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}
 \end{aligned}$$

$B \cdot D = \blacksquare$ не удовлетворяет правилу перемножения матриц
 Размерности массива не совпадают.

Рис. 4.12. Арифметические действия с матрицами

Для умножения матриц используется традиционный знак умножения (клавиша со звездочкой). Знак умножения в Mathcad-документе по умолчанию обозначается точкой.

Вид значка в документе можно поменять. Для этого щелкните правой кнопкой мыши на выражении, в котором есть знак умножения. В открывшемся контекстном меню найдите пункт View Multiplication as (Вид знака умножения). Щелкните мышью на одном из предлагаемых семи пунктов — значок изменится.

Точно так же можно изменить вид знака присвоения значения, но делать этого не рекомендуется, чтобы не перепутать знак присвоения значения и знак логического равенства.

По правилу перемножения матриц, число столбцов первой матрицы должно быть равно числу строк второй матрицы, $A(m \times n) \cdot B(n \times k) = C(m \times k)$ (в скобках указаны размеры перемножаемых матриц).

ПРИМЕЧАНИЕ

Полезно запомнить, что произведение строки на столбец дает число, а произведение столбца на строку дает полностью заполненную квадратную матрицу (рис. 4.13).

Скалярное произведение векторов дает число, равное сумме попарных произведений элементов этих векторов, $UV = \sum_{i=1}^n U_i V_i$, где n — число элементов каждого вектора, или $UV = |U||V| \cdot \cos(\theta)$, где θ — угол между векторами.

$$\mathbf{V} = \begin{pmatrix} 12 \\ 11 \\ 10 \\ 9 \\ 8 \end{pmatrix} \quad \mathbf{V} \cdot \mathbf{V}^T = \begin{pmatrix} 144 & 132 & 120 & 108 & 96 \\ 132 & 121 & 110 & 99 & 88 \\ 120 & 110 & 100 & 90 & 80 \\ 108 & 99 & 90 & 81 & 72 \\ 96 & 88 & 80 & 72 & 64 \end{pmatrix} \quad \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \cdot (1 \ 2 \ 3) = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & 9 \end{pmatrix}$$

столбец x строка = матрица

$$\mathbf{V}^T \cdot \mathbf{V} = (510) \quad (1 \ 2 \ 3) \cdot \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} = (14)$$

строка x столбец = число

Рис. 4.13. Умножение векторов

Для вычисления скалярного произведения векторов используется обычный знак умножения (клавиша *) или значок $\vec{X} \cdot \vec{Y}$ на панели Matrix (рис. 4.14).

$\mathbf{V} := \begin{pmatrix} 10 \\ 2 \\ 3 \end{pmatrix}$	$\mathbf{U} := \begin{pmatrix} 3 \\ 4 \\ 5 \end{pmatrix}$	скалярное произведение $\mathbf{V} \cdot \mathbf{U} = 53$	векторное произведение $\mathbf{V} \times \mathbf{U} = \begin{pmatrix} -2 \\ -41 \\ 34 \end{pmatrix}$	попарное произведение $\overrightarrow{(\mathbf{V} \cdot \mathbf{U})} = \begin{pmatrix} 30 \\ 8 \\ 15 \end{pmatrix}$
$\mathbf{V1} := (2 \ 5 \ 1 \ 7)^T \quad \mathbf{U1} := (-2 \ 4 \ 8 \ 3)^T \quad \mathbf{V1} \cdot \mathbf{U1} = 45$		$\mathbf{V1} \times \mathbf{U1} = \begin{pmatrix} \blacksquare \\ \blacksquare \\ \blacksquare \end{pmatrix}$		
Значение должно быть вектором с тремя элементами.				

Рис. 4.14. Умножение векторов (продолжение)

Векторное произведение двух векторов, \vec{X} и \vec{Y} , с углом между ними θ равно вектору с модулем $|\mathbf{U}| |\mathbf{V}| \cdot \sin(\theta)$, направленному перпендикулярно плоскости, определяемой векторами \vec{U} и \vec{V} . Оператор векторного произведения можно ввести только с панели Matrix при помощи значка $\vec{X} \times \vec{Y}$.

Векторное произведение можно вычислить только для векторов с тремя элементами (см. рис. 4.14).

4.8.5. Оператор векторизации

В Mathcad существует несколько необычный оператор, называемый оператором векторизации и предназначенный для работы с массивами. Он позволяет выполнить однотипную операцию со всеми элементами массива (рис. 4.15), например, вычислить $\overrightarrow{\cos(\alpha)}$, где α — матрица углов. Для этого

- выделите клавишей пробела все выражение;

если аргумент вектор, оператор векторизации не нужен

$$U = \begin{pmatrix} 3 \\ 4 \\ 5 \end{pmatrix} \quad \cos(U) = \begin{pmatrix} -0.99 \\ -0.654 \\ 0.284 \end{pmatrix} \quad U^3 = \begin{pmatrix} 27 \\ 64 \\ 125 \end{pmatrix} \quad e^U = \begin{pmatrix} 20.086 \\ 54.598 \\ 148.413 \end{pmatrix}$$

если аргумент матрица, оператор векторизации нужен

$$A = \begin{pmatrix} 1 & 2 \\ 2 & 3 \\ 3 & 4 \\ 4 & 5 \\ 5 & 1 \end{pmatrix} \quad \xrightarrow{\cos(A)} \begin{pmatrix} 0.54 & -0.416 \\ -0.416 & -0.99 \\ -0.99 & -0.654 \\ -0.654 & 0.284 \\ 0.284 & 0.54 \end{pmatrix}$$

$$\xrightarrow{A^3} \begin{pmatrix} 1 & 8 \\ 8 & 27 \\ 27 & 64 \\ 64 & 125 \\ 125 & 1 \end{pmatrix} \quad \xrightarrow{e^A} \begin{pmatrix} 2.718 & 7.389 \\ 7.389 & 20.086 \\ 20.086 & 54.598 \\ 54.598 & 148.413 \\ 148.413 & 2.718 \end{pmatrix}$$

Рис. 4.15. Использование оператора векторизации

- на панели матриц щелкните на кнопке $\overrightarrow{f(M)}$. Над выделенным выражением появится знак векторизации;
- нажмите клавишу =.

Mathcad допускает использование вектора в качестве аргумента функции, и в этом случае знак векторизации не требуется. Например, если V – вектор, $\cos(V) = \text{ответ}$ – без знака векторизации.

4.8.6. Последовательное и параллельное вычисление элементов массива

В Mathcad 13–15 матричные операции, такие как вычисление определителя матрицы, перемножение матриц, обращение матрицы, вычисление собственных чисел матриц и их собственных векторов, выполняются более быстрым алгоритмом благодаря одновременному (параллельному) вычислению нескольких элементов массива.

В Mathcad 11 значения массива чисел вычисляются последовательно, а в Mathcad 13–15 – одновременно, что может дать различные ответы (рис. 4.16).

Показанное на рис. 4.17, *слева* последовательное вычисление значений векторов работает в Mathcad 11. В Mathcad 13 для подобных вычислений необходимо использовать программный оператор (рис. 4.17, *справа*).

4.9. Собственные числа и собственные векторы матриц

В технических расчетах весьма актуальна задача определения собственных чисел и собственных векторов квадратных матриц. В Mathcad имеется несколько встроенных функций для решения стандартной и нестандартной задач.

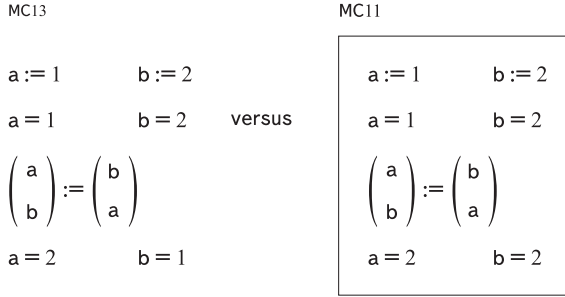


Рис. 4.16. Параллельное в Mathcad 13 и последовательное в Mathcad 11 вычисление элементов массива

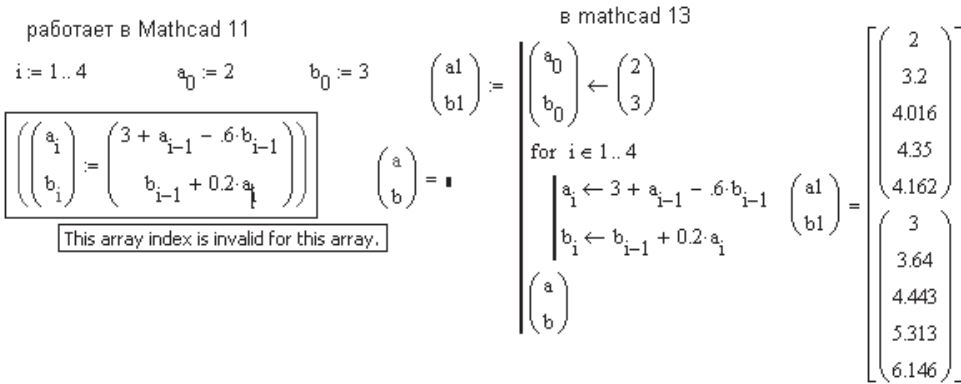


Рис. 4.17. Рекурсия в Mathcad 11 и 13

Рассмотрим стандартную задачу на собственные числа $A \cdot L = \lambda \cdot L$ или $(A - \lambda \cdot E) \cdot L = 0$, где E — единичная матрица, A — квадратная матрица, λ — вектор собственных чисел, L — спектр собственных векторов (рис. 4.18).

Для решения стандартной задачи используются следующие функции:

- ❑ `eigenvals(A)` вычисляет вектор собственных чисел матрицы A , превращающий ее в диагональную матрицу;
- ❑ `eigenvecs(A)` вычисляет спектр собственных векторов матрицы A . Результат расчета — матрица, каждый столбец которой $A^{<i>A</i>}$ представляет собой нормированный собственный вектор для одного из собственных чисел λ_i .

Пример стандартной задачи: по заданному тензору напряжений найти 3 главных напряжения (собственные числа) и соответствующие им направляющие косинусы для нормалей к трем главным площадкам (собственные векторы).

Нестандартная задача на собственные числа имеет вид $A \cdot L = \lambda \cdot B \cdot L$ или $(A - \lambda \cdot B) \cdot L = 0$, где A и B — квадратные матрицы (неединичные), λ — вектор собственных чисел, L — спектр собственных векторов (рис. 4.19).

Стандартная задача на собственные числа

$$A := \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 4 & 5 & 1 \\ 3 & 4 & 5 & 1 & 2 \\ 4 & 5 & 1 & 2 & 3 \\ 5 & 1 & 2 & 3 & 4 \end{pmatrix} \quad \begin{array}{l} A \cdot L = \lambda \cdot L \\ \text{или } (A - \lambda \cdot E) \cdot L = 0 \\ \text{где } E \text{ -- единичная матрица} \end{array}$$

собственные числа

$$\lambda := \text{eigenvals}(A)$$

$$\lambda = \begin{pmatrix} -4.25 \\ -2.63 \\ 2.63 \\ 4.25 \\ 15 \end{pmatrix}$$

собственные векторы

$$L := \text{eigenvecs}(A)$$

$$L = \begin{pmatrix} -0.56 & -0.62 & -0.1 & 0.29 & 0.45 \\ -0.45 & 0.45 & 0.45 & -0.45 & 0.45 \\ 0.29 & -0.1 & -0.62 & -0.56 & 0.45 \\ 0.62 & -0.29 & 0.56 & 0.1 & 0.45 \\ 0.1 & 0.56 & -0.29 & 0.62 & 0.45 \end{pmatrix}$$

Создание диагональной матрицы из собственных чисел

$$A_{\text{глав}} := \text{diag}(\lambda) \quad A_{\text{глав}} = \begin{pmatrix} -4.25 & 0 & 0 & 0 & 0 \\ 0 & -2.63 & 0 & 0 & 0 \\ 0 & 0 & 2.63 & 0 & 0 \\ 0 & 0 & 0 & 4.25 & 0 \\ 0 & 0 & 0 & 0 & 15 \end{pmatrix}$$

Рис. 4.18. Определение собственных чисел квадратной матрицы

Нестандартная задача на собственные числа

$$A \cdot L = \lambda \cdot B \cdot L \quad \text{или} \quad (A - \lambda \cdot B) \cdot L = 0$$

где A и B симметричные матрицы

$$A := \begin{pmatrix} 2 & 3 & 4 \\ 3 & -2 & -5 \\ 4 & -5 & 1 \end{pmatrix} \quad B := \begin{pmatrix} 3 & 1 & 4 \\ 1 & 5 & 2 \\ 4 & 2 & 6 \end{pmatrix}$$

собственные числа

$$\text{genvals}(A, B) = \begin{pmatrix} -9.614 \\ 0.752 \\ 2.612 \end{pmatrix}$$

собственные векторы

$$\text{genvecs}(A, B) = \begin{pmatrix} 0.814 & -0.926 & 0.685 \\ -0.044 & 0.06 & 0.421 \\ -0.578 & -0.373 & -0.595 \end{pmatrix}$$

Рис. 4.19. Обобщенные собственные числа матрицы

Пример нестандартной задачи: определение собственных частот колебаний упругих систем. В этой задаче собственное число $\lambda = \omega^2$ — квадрат собственной частоты колебаний, собственный вектор $L^{<i>$ — форма колебания для собственной частоты ω_i (см. главу 4, примеры 26 и 27 электронной книги).

Для решения нестандартной задачи используются следующие функции:

- $\text{genvals}(A, B)$ вычисляет вектор обобщенных собственных чисел λ ;
- $\text{genvecs}(A, B)$ вычисляет спектр собственных векторов. Результат расчета — матрица, каждый столбец которой $L^{<i>$ соответствует собственному числу λ_i .

ПРИМЕЧАНИЕ

Функции `Genvals` и `Genvecs` принимают к расчету и комплексные матрицы.

Результаты расчета собственных чисел и собственных векторов выводятся в отсортированном виде.

Вектор собственных чисел можно преобразовать в диагональную матрицу с помощью функции `diag(λ)`. Результат ее действия — диагональная матрица, на главной диагонали которой расположены элементы вектора λ .

В старых версиях Mathcad функция `diag` принимала вектор чисел, а возвращала квадратную матрицу с элементами вектора на диагонали. В Mathcad 13 функция `diag` выполняет уже два действия:

- 1) если принимает вектор, то возвращает диагональную матрицу с элементами вектора на главной диагонали;
- 2) если принимает диагональную матрицу, то возвращает вектор чисел, стоящих на главной диагонали (рис. 4.20).

$$i := 0..2 \quad v_1 := 2 \cdot i$$

$$Md := \text{diag}(v) \quad Md = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 4 \end{pmatrix} \quad \text{diag}(Md) = \begin{pmatrix} 0 \\ 2 \\ 4 \end{pmatrix}$$

Рис. 4.20. Два действия, выполняемые функцией `diag`

4.10. Норма квадратной матрицы

Норма квадратной матрицы характеризует порядок величины элементов матрицы.

В Mathcad есть 4 функции для оценки нормы (рис. 4.21):

- `norm1(A)` — норма в пространстве L_1 ;
- `norm2(A)` — норма в пространстве L_2 ;
- `norme(A)` — евклидова норма;
- `normi(A)` — max-норма, или ∞ -норма.

$$C := \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 4 & 5 & 1 \\ 3 & 4 & 5 & 1 & 2 \\ 4 & 5 & 1 & 2 & 3 \\ 5 & 1 & 2 & 3 & 4 \end{pmatrix}$$

<code>norm1(C) = 15</code>	<code>norm2(C) = 15</code>
<code>norme(C) = 16.583</code>	<code>normi(C) = 15</code>

$$CC := 100 \cdot C$$

<code>norm1(CC) = 1500</code>	<code>norm2(CC) = 1500</code>
<code>norme(CC) = 1658</code>	<code>normi(CC) = 1500</code>

Рис. 4.21. Функции вычисления нормы матрицы

Евклидова норма эквивалентна длине многомерного вектора $\|x\| = \sqrt{\sum A_i^2}$.

Как видно из рис. 4.21, величина нормы мало зависит от способа ее вычисления. При увеличении всех элементов матрицы в 100 раз все нормы также увеличиваются в 100 раз.

4.11. Число обусловленности квадратной матрицы

Число обусловленности является мерой чувствительности системы линейных алгебраических уравнений $A \cdot X = B$, определяемой матрицей коэффициентов A , к погрешностям вектора B правых частей уравнений. Чем больше число обусловленности, тем более неустойчив процесс решения системы. В примере, приведенном на рис. 4.22, матрица A хорошо обусловленная, а матрица AA , в которой вторая строка почти пропорциональна первой, плохо обусловленная.

$$\begin{array}{l}
 A := \begin{pmatrix} 2 & 5 \\ 2 & 6 \end{pmatrix} \quad B := \begin{pmatrix} 10 \\ 2 \end{pmatrix} \quad X := A^{-1} \cdot B \quad X = \begin{pmatrix} 25 \\ -8 \end{pmatrix} \\
 \text{cond1}(A) = 44 \quad \text{cond2}(A) = 34.471 \\
 \text{conde}(A) = 34.5 \quad \text{condi}(A) = 44 \quad \text{Матрица } A - \text{хорошо обусловленная} \\
 \\
 AA := \begin{pmatrix} 2 & 5 \\ 2 & 5.001 \end{pmatrix} \quad BB := \begin{pmatrix} 10 \\ 2 \end{pmatrix} \quad XX := AA^{-1} \cdot BB \quad XX = \begin{pmatrix} 20005 \\ -8000 \end{pmatrix} \\
 \text{cond1}(AA) = 35009 \quad \text{cond2}(AA) = 29005 \\
 \text{conde}(AA) = 29005 \quad \text{condi}(AA) = 35009 \quad \text{Матрица } AA - \text{плохо обусловленная}
 \end{array}$$

Рис. 4.22. Функции вычисления обусловленности квадратной матрицы

4.12. Матричные разложения

Большинство матричных методов решения систем линейных алгебраических уравнений основано на представлении матрицы коэффициентов уравнения в виде произведения матриц специального вида, полученных путем разложения исходной матрицы.

$$\begin{array}{l}
 A := \begin{pmatrix} 13 & 7 & 4 \\ 7 & 9 & -3 \\ 4 & -3 & 9 \end{pmatrix} \quad L := \text{cholesky}(A) \quad L = \begin{pmatrix} 3.606 & 0 & 0 \\ 1.941 & 2.287 & 0 \\ 1.109 & -2.253 & 1.64 \end{pmatrix} \\
 \text{Проверка } L \cdot L^T = \begin{pmatrix} 13 & 7 & 4 \\ 7 & 9 & -3 \\ 4 & -3 & 9 \end{pmatrix} \\
 \\
 AA := \begin{pmatrix} 20 & 3 & 4 \\ 3 & 2 & -5 \\ 4 & -5 & 9 \end{pmatrix} \quad \boxed{L := \text{cholesky}(AA)} \\
 \boxed{\text{Матрица должна быть положительно определенной.}}
 \end{array}$$

Рис. 4.23. Разложение матрицы по методу Холесского

В Mathcad имеется несколько функций, реализующих алгоритмы наиболее популярных матричных разложений.

4.12.1. Разложение Холесского

Разложением матрицы по методу Холесского является представление ее в виде $A = L \cdot L^T$, где L — нижняя треугольная матрица (то есть матрица, в которой сверху от диагонали находятся нули). Выполняется это разложение функцией $\text{cholesky}(A)$, где A — симметричная, положительно-определенная матрица (рис. 4.23).

4.12.2. QR-разложение

QR-разложение представляет исходную матрицу в виде $A = Q \cdot R$, где Q — ортонормированная матрица, то есть состоящая из ортогональных друг другу векторов единичной длины, а R — верхняя треугольная матрица (нули под главной диагональю).

Выполняется это разложение функцией $\text{qr}(A)$. Результатом действия функции является матрица L , полученная из двух матриц, Q и R , составленных бок о бок. Чтобы выделить каждую матрицу отдельно, используем функцию submatrix (рис. 4.24).

$$\begin{aligned}
 AA := \begin{pmatrix} 20 & 3 & 4 \\ 3 & 2 & -5 \\ 4 & -5 & 9 \end{pmatrix} \quad L1 := \text{qr}(AA) \quad L1 = \begin{pmatrix} 0.97 & -0.145 & 0.194 & 20.616 & 2.231 & 4.899 \\ 0.146 & -0.292 & -0.945 & 0 & -5.746 & 9.385 \\ 0.194 & 0.945 & -0.262 & 0 & 0 & 3.149 \end{pmatrix} \\
 Q := \text{submatrix}\left(L1, 1, \text{rows}(L1), 1, \frac{\text{cols}(L1)}{2}\right) \quad Q = \begin{pmatrix} 0.97 & -0.145 & 0.194 \\ 0.146 & -0.292 & -0.945 \\ 0.194 & 0.945 & -0.262 \end{pmatrix} \\
 R := \text{submatrix}\left(L1, 1, \text{rows}(L1), \frac{\text{cols}(L1)}{2} + 1, \text{cols}(L1)\right) \quad R = \begin{pmatrix} 20.616 & 2.231 & 4.899 \\ 0 & -5.746 & 9.385 \\ 0 & 0 & 3.149 \end{pmatrix}
 \end{aligned}$$

Рис. 4.24. QR-разложение квадратной матрицы

4.12.3. LU-разложение

LU-разложением матрицы A , или треугольным разложением, является матричное разложение типа $PA = LU$, где U и L — верхняя и нижняя треугольные матрицы. P , A , L , U — квадратные матрицы одного размера.

LU-разложение квадратной матрицы A выполняется функцией $\text{lu}(A)$.

Фактически треугольное разложение матрицы выполняется при решении системы линейных алгебраических уравнений методом Гаусса.

Функция $\text{lu}(A)$, как и функция $\text{qr}(A)$, выводит составную матрицу, содержащую матрицы P , L , U , составленные бок о бок (рис. 4.25).

LU - разложение

$$A := \begin{pmatrix} 20 & 3 & -4 \\ -2 & 2 & -2 \\ 4 & -5 & 9 \end{pmatrix} \quad L1 := lu(A) \quad L1 = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 & -2 & 2 & -2 \\ 1 & 0 & 0 & -10 & 1 & 0 & 0 & 23 & -24 \\ 0 & 0 & 1 & -2 & -0.043 & 1 & 0 & 0 & 3.957 \end{pmatrix}$$

$$P := \text{submatrix}\left(L1, 1, \text{rows}(L1), 1, \frac{\text{cols}(L1)}{3}\right) \quad P = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$L := \text{submatrix}\left(L1, 1, \text{rows}(L1), \frac{\text{cols}(L1)}{3} + 1, \frac{\text{cols}(L1) \cdot 2}{3}\right) \quad L = \begin{pmatrix} 1 & 0 & 0 \\ -10 & 1 & 0 \\ -2 & -0.043 & 1 \end{pmatrix}$$

$$U := \text{submatrix}\left(L1, 1, \text{rows}(L1), \frac{\text{cols}(L1) \cdot 2}{3} + 1, \text{cols}(L1)\right) \quad U = \begin{pmatrix} -2 & 2 & -2 \\ 0 & 23 & -24 \\ 0 & 0 & 3.957 \end{pmatrix}$$

$$\text{Проверка} \quad P \cdot A = \begin{pmatrix} -2 & 2 & -2 \\ 20 & 3 & -4 \\ 4 & -5 & 9 \end{pmatrix} \quad L \cdot U = \begin{pmatrix} -2 & 2 & -2 \\ 20 & 3 & -4 \\ 4 & -5 & 9 \end{pmatrix}$$

Рис. 4.25. LU-разложение квадратной матрицы**4.12.4. Сингулярное разложение**

Сингулярным разложением матрицы A размером $M \times N$, где $M \geq N$, является разложение вида $A = B \times s \times C^T$, где B и C – ортогональные матрицы размером $M \times N$ и $N \times M$ соответственно, s – диагональная матрица с сингулярными числами матрицы A на главной диагонали, $\text{svds}(A)$ – вектор, состоящий из сингулярных чисел, $\text{svd}(A)$ – сингулярное разложение матрицы A .

Примеры поиска сингулярных чисел невырожденной матрицы A и сингулярной матрицы A_1 приведены на рис. 4.26.

Начиная с Mathcad 13 сингулярное разложение выполняется одной функцией svd2 , заменяющей две функции, svd и svds . Новая функция svd2 для сингулярного разложения матриц работает и с комплексными числами. Кроме того, она сделана более удобной в использовании.

Функция svd2 возвращает составной массив, состоящий из трех вложенных массивов, входящих в выражение $A = B \times s \times C^T$. На рис. 4.27 показано использование функции svd2 .

4.13. Вывод векторов и матриц

В Mathcad существует два способа отображения массивов: в виде матрицы и в виде таблицы (рис. 4.28).

Сингулярное разложение

$$A := \begin{pmatrix} 20 & 3 & 4 \\ 3 & 2 & -5 \\ 4 & -5 & 9 \\ 7 & -3 & -2 \end{pmatrix} \quad AA := \text{svd}(A) \quad AA = \begin{pmatrix} -0.92 & 0.142 & -0.347 \\ -0.084 & 0.516 & 0.167 \\ -0.258 & -0.832 & 0.203 \\ -0.281 & 0.147 & 0.9 \\ -0.974 & 0.185 & 0.135 \\ -0.036 & 0.459 & -0.888 \\ -0.226 & -0.869 & -0.441 \end{pmatrix}$$

Вырежем матрицы B и C из матрицы AA

$$B := \text{submatrix}(AA, 1, \text{ceil}\left(\frac{\text{rows}(AA)}{2}\right), 1, \text{cols}(AA)) \quad B = \begin{pmatrix} -0.92 & 0.142 & -0.347 \\ -0.084 & 0.516 & 0.167 \\ -0.258 & -0.832 & 0.203 \\ -0.281 & 0.147 & 0.9 \end{pmatrix}$$

$$C := \text{submatrix}(AA, \text{ceil}\left(\frac{\text{rows}(AA)}{2}\right) + 1, \text{rows}(AA), 1, \text{cols}(AA))$$

$$s := \text{diag}(\text{svds}(A)) \quad s = \begin{pmatrix} 22.251 & 0 & 0 \\ 0 & 11.272 & 0 \\ 0 & 0 & 4.984 \end{pmatrix} \quad C = \begin{pmatrix} -0.974 & 0.185 & 0.135 \\ -0.036 & 0.459 & -0.888 \\ -0.226 & -0.869 & -0.441 \end{pmatrix}$$

Проверка $B \cdot s \cdot C^T = \begin{pmatrix} 20 & 3 & 4 \\ 3 & 2 & -5 \\ 4 & -5 & 9 \\ 7 & -3 & -2 \end{pmatrix}$

Рис. 4.26. Сингулярное разложение матрицы

В Mathcad 13

$$A := \begin{pmatrix} 2.3 & 24 \\ 6.7 & 42 \\ 2.8 & 7 \end{pmatrix} \quad \text{svd2}(A) = \begin{pmatrix} \{2,1\} \\ \{3,2\} \\ \{2,2\} \end{pmatrix} \quad \text{svd2}(A) = \begin{pmatrix} \begin{pmatrix} 49.419 \\ 2.192 \end{pmatrix} \\ \begin{pmatrix} 0.487 & -0.58 \\ 0.861 & 0.192 \\ 0.148 & 0.792 \end{pmatrix} \\ \begin{pmatrix} 0.148 & 0.989 \\ 0.989 & -0.148 \end{pmatrix} \end{pmatrix}$$

Свернутый и развернутый массив вывода

$$\begin{pmatrix} s \\ B \\ C \end{pmatrix} := \text{svd2}(A) \quad s = \begin{pmatrix} 49.419 \\ 2.192 \end{pmatrix} \quad B = \begin{pmatrix} 0.487 & -0.58 \\ 0.861 & 0.192 \\ 0.148 & 0.792 \end{pmatrix} \quad C = \begin{pmatrix} 0.148 & 0.989 \\ 0.989 & -0.148 \end{pmatrix}$$

Проверка $B \cdot \text{diag}(s) \cdot C^T = \begin{pmatrix} 2.3 & 24 \\ 6.7 & 42 \\ 2.8 & 7 \end{pmatrix} \quad A = \begin{pmatrix} 2.3 & 24 \\ 6.7 & 42 \\ 2.8 & 7 \end{pmatrix}$

Рис. 4.27. Применение функции svd2

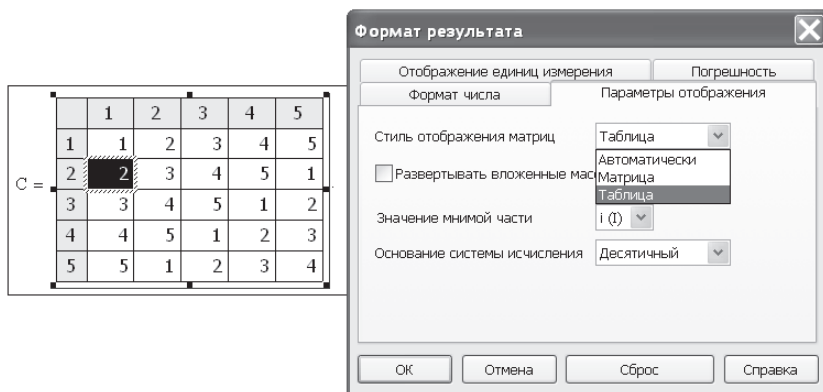


Рис. 4.28. Изменение стиля отображения массива

Для изменения стиля отображения массива выберите в главном меню **Format** ► **Result** (Формат ► Результаты), перейдите на вкладку **Display Options** (Параметры отображения). В поле **Matrix display style** (Стиль отображения матриц) выберите **Matrix** (Матрица), **Table** (Таблица) или **Automatic** (Автоматический выбор).

Как правило, Mathcad выбирает **Matrix** для маленьких массивов и **Table** для больших.

Более простой путь форматирования выведенного массива — дважды щелкнуть левой кнопкой мыши в поле массива, который надо отформатировать. При этом появляется диалоговое окно **Result Format** (Формат результата), открытое на вкладке **Number Format** (Формат чисел). Остается только перейти на нужную вкладку (**Display Options**).

Если число элементов в строке или столбце больше 10, то в стиле **Table** для экономии места в документе Mathcad выводит только 10 чисел, а сбоку и сверху — полосы прокрутки для просмотра остальных элементов массива.

Стиль отображения **Table** допускает различное выравнивание выведенного массива относительно его имени. Для установки выравнивания надо, поместив курсор в область массива, щелкнуть правой кнопкой мыши и в открывшемся контекстном меню выбрать **Alignment** (Выравнивание), где на выбор предлагается 5 мест расположения названия массива: **Top** (Верх), **Center** (Центр), **Bottom** (Низ), **Above** (Выше), **Below** (Ниже) (рис. 4.29). Пункт **Properties** (Свойства) этого же контекстного меню позволяет установить или убрать нумерацию строк и столбцов.

В окне **Display Options** (Параметры отображения) есть пункт **Expand nested arrays** (Разворачивать вложенные массивы). Установите там флажок, если хотите увидеть все массивы, вложенные в составной массив.

4.14. Составные массивы

Для удобства вычислений иногда формируют массив из различных по смыслу и по размерности элементов. Такой массив называется составным. Необходи-

мость в составных массивах возникает при составлении больших программ. В программе одно место вывода, а вывести надо несколько различных величин (чисел, векторов, матриц). Примеры составных массивов приведены в главе 9 и в примерах 6, 20, 30 и 31 электронной книги.

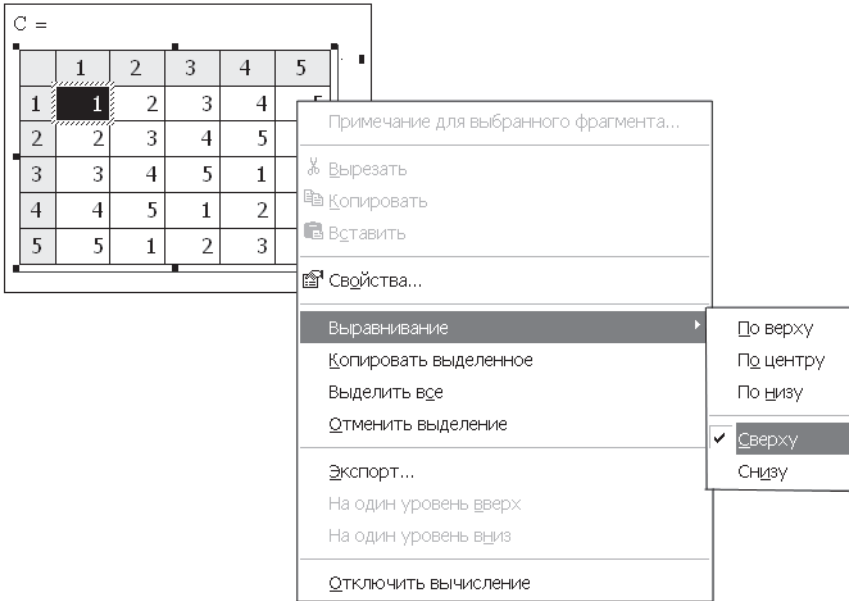


Рис. 4.29. Контекстное меню для матрицы в стиле Table

4.15. Работа с комплексными числами

Mathcad с одинаковым успехом вычисляет выражения с действительными и с комплексными числами.

Комплексные числа можно записывать в любой принятой в математике форме:

- обычной (в виде суммы действительной и мнимой частей числа) — $c = a + bi$;
- в тригонометрической форме — $c = A \cdot (\cos(\varphi) + i \cdot \sin(\varphi))$;
- в экспоненциальной форме — $c = Ae^{i\varphi}$.

Для ввода мнимой единицы надо набрать с клавиатуры $1i$ или $1j$ и ввести число. При выведении курсора из области числа символ 1 исчезает, остается комплексное число в привычном виде (рис. 4.30).

В качестве примера рассмотрим расчет токов и напряжений в цепи переменного тока по законам Кирхгофа. В этом случае ЭДС E в цепи, напряжения U , тока J , сопротивления Z — комплексные числа. ЭДС E задана в экспоненциальной форме, но при нажатии клавиши $=$ выводится в обычном виде. После ввода констант в комплексной форме все формулы записываются так же, как и для действительных чисел. При расчете вектора напряжений на различных участках цепи

по закону Ома использован оператор векторизации для расчета попарных произведений $U_i = J_i Z_i$ (рис. 4.31).

Ввод мнимой единицы 1i или 1j или $i := \sqrt{-1}$
 $C = a + b \cdot i = A \cdot \sin(\phi) + i \cdot \cos(\phi) = A \cdot e^{i \cdot \phi}$
 например
 $C := 2 - 3 \cdot i$ $C = 2 - 3i$
 $D := 2 + 3 \cdot j$ $D = 2 + 3i$
 $CC := 3 \cdot \left(\sin\left(\frac{\pi}{6}\right) + i \cdot \cos\left(\frac{\pi}{6}\right) \right)$ $CC = 1.5 + 2.598i$
 $DD := 3 \cdot e^{-2 \cdot i}$ $DD = -1.248 - 2.728i$

Переменный ток.
 $Em := 179$ $\phi := 20$ $E := \frac{Em}{\sqrt{2}} \cdot e^{j \cdot \phi}$ $E = 51.652 + 115.553i$

Расчет токов в цепи по уравнениям Кирхгофа ORIGIN := 1

$$:= \begin{pmatrix} j \cdot 10 \\ 30 \\ 40 \\ -i \cdot 50 \\ i \cdot 60 \end{pmatrix} \quad A := \begin{pmatrix} 1 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 1 \\ z_1 & z_2 & 0 & 0 & 0 \\ 0 & z_2 & z_3 & z_4 & 0 \\ 0 & 0 & 0 & -z_4 & z_5 \end{pmatrix} \quad B := \begin{pmatrix} 0 \\ 0 \\ E \\ E \\ 0 \end{pmatrix}$$

Рис. 4.30. Ввод комплексных чисел

$$I := A^{-1} \cdot B \quad I = \begin{pmatrix} 2.826 + 2.996i \\ 2.72 + 2.91i \\ -0.106 - 0.086i \\ -0.634 - 0.515i \\ 0.528 + 0.429i \end{pmatrix}$$

Расчет напряжений по закону Ома $U := I \cdot z$ $U = -4.035 + 175.502i$
 без оператора векторизации выведено скалярное произведение векторов

$$U := \overrightarrow{(I \cdot z)} \quad U = \begin{pmatrix} -29.956 + 28.259i \\ 81.608 + 87.295i \\ -4.225 - 3.431i \\ -25.731 + 31.69i \\ -25.731 + 31.69i \end{pmatrix} \quad \bar{U} = \begin{pmatrix} -29.956 - 28.259i \\ 81.608 - 87.295i \\ -4.225 + 3.431i \\ -25.731 - 31.69i \\ -25.731 - 31.69i \end{pmatrix}$$

это не аргумент функции--без оператора векторизации не обойтись вывод комплексно-сопряженных чисел

Рис. 4.31. Расчеты с векторами комплексных чисел

В Mathcad есть оператор преобразования комплексного числа в комплексно-сопряженное. Он не представлен на математической панели, а вызывается с помощью символа кавычки ("") (см. рис. 4.31).

Модуль комплексного числа $A = \sqrt{\operatorname{Re}^2 + \operatorname{Im}^2}$ вычисляется оператором $||$, вводимым с панели Calculator (Калькулятор) или Matrix (Матрица). Тот же оператор позволяет вычислить модуль вектора или определитель матрицы.

Аргумент комплексного числа вычисляется встроенной функцией $\arg(A)$, которая выдает угол поворота вектора в системе координат $\operatorname{Re}-\operatorname{Im}$ (рис. 4.32).

модуль комплексного числа. Необходим оператор векторизации

$$\vec{|I|} = \begin{pmatrix} 4.118 \\ 3.983 \\ 0.136 \\ 0.816 \\ 0.688 \end{pmatrix} \quad \vec{|U|} = \begin{pmatrix} 41.182 \\ 119.5 \\ 5.443 \\ 40.82 \\ 40.82 \end{pmatrix} \quad |I_1| = 4.118$$

аргумент комплексного числа

$$\arg(I_1) = 0.815 \quad \arg(I) = \begin{pmatrix} 0.815 \\ 0.819 \\ -2.46 \\ -2.46 \\ 0.682 \end{pmatrix}$$

$$\phi := \arg(I) \quad \phi = \begin{pmatrix} 0.815 \\ 0.819 \\ -2.46 \\ -2.46 \\ 0.682 \end{pmatrix} \quad \phi = \begin{pmatrix} 46.67 \\ 46.928 \\ -140.925 \\ -140.925 \\ 39.075 \end{pmatrix} \text{ deg}$$

угол в радианах угол в градусах

Рис. 4.32. Вычисление модуля и аргумента комплексных чисел

Глава 5

Символьные вычисления

Основой символьных вычислений в Mathcad являются формулы и правила их преобразования, заимствованные из системы символьной математики Maple 5, предназначенной для математиков высшей квалификации. Из более чем 300 символьных функций Maple 5 в Mathcad включено около 20, но и этого достаточно для инженерно-технических работников, для которых и предназначен Mathcad.

Большинство пользователей Mathcad в своей работе используют в основном численные расчеты и графики для анализа результатов расчетов. Лишь изредка возникает необходимость в символьных вычислениях. Некоторые встроенные функции Mathcad требуют предварительных символьных вычислений для своего успешного применения:

- ❑ При нахождении корней полинома функцией `polyroot` символьным путем желательно найти коэффициенты полинома для ввода их в качестве параметров функции `polyroot`.
- ❑ При использовании функции общей регрессии `genfit` надо создать вектор, содержащий в символьном виде саму аппроксимирующую функцию и ее первую и вторую производные.
- ❑ При решении дифференциальных уравнений необходимо задание высшей производной в явном виде. Иногда это можно сделать, использовав символьное решение уравнения относительно производной высшего порядка.

5.1. Оптимизация численных вычислений

В Mathcad есть возможность оптимизации решений. Если выбрать пункт меню `Tools` ▶ `Optimize` ▶ `Worksheet` (или `Equation`) (`Инструменты` ▶ `Оптимизировать` ▶ `Документ` (или `Выражение`)), то перед вычислением каждого выражения символьный процессор Mathcad будет пытаться упростить все выражения (или одно выражение) для ускорения и уточнения вычислений.

В Mathcad есть два самостоятельных процессора: численный и символьный, которые обычно не имеют связи друг с другом. При включении оптимизации они работают совместно.

Можно включить оптимизацию для вычисления одного конкретного выражения более просто. Для этого щелкните правой кнопкой мыши на интересующем вас выражении и в открывшемся контекстном меню выберите пункт `Optimize` (`Оптимизировать`). Около выражения, для которого Mathcad будет искать символьное решение, появится красная звездочка (рис. 5.1).

ВНИМАНИЕ

Оптимизация возможна только для выражений, содержащих оператор присваивания (`:=`).

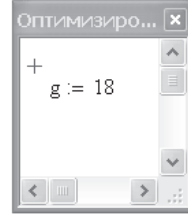
Оптимизация расчетов

$$g := \int_0^1 \int_0^2 \int_0^3 x^2 \cdot y^3 \cdot z \, dx \, dy \, dz *$$

$$g = 18$$

нет оптимизации

$$\int_0^1 \int_0^2 \int_0^3 x^2 \cdot y^3 \cdot z \, dx \, dy \, dz = 18$$

**Рис. 5.1.** Оптимизация вычислений

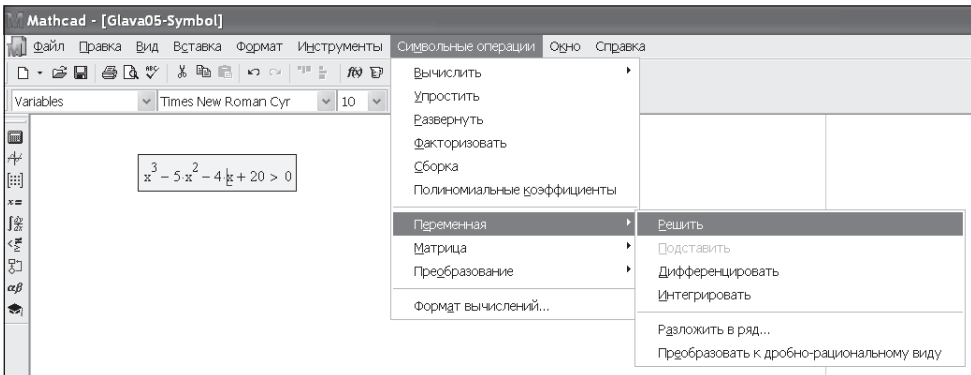
В некоторых случаях, например, при вычислении интегралов по поверхности или объему (двойной или тройной интеграл), удается найти символьное решение интеграла (см. рис. 5.1), что резко ускоряет расчет, ведь вычисление интегралов — самая медленная операция в Mathcad. В Mathcad есть два способа символьных вычислений:

1. С помощью меню Symbolics (Символьные вычисления) из главного меню Mathcad.
2. С помощью панели Symbolic Keyword Toolbar (Символьная панель инструментов), вызываемой соответствующим значком математической панели.

5.2. Использование меню символьных вычислений Symbolics

Этот способ предоставляет меньше возможностей, чем символьная панель инструментов. Однако он прост и тем удобен. В ряде случаев лучше обращаться к меню Symbolics, чем к символьной панели инструментов.

В главном меню Mathcad выберите пункт Symbolics, откроется подменю, содержащее ряд команд (рис. 5.2). Для выполнения любой из этих команд необходимо сначала выделить объект вычислений:

**Рис. 5.2.** Вид меню Symbolics

- ❑ Щелкните мышью на выражении, если надо произвести символьную операцию над всем выражением.
- ❑ Если надо произвести действия над частью выражения, выделите эту часть черным цветом, то есть подведите указатель мыши к началу нужной части выражения, нажмите левую кнопку мыши и переместите курсор до конца этой части при нажатой левой кнопке мыши.
- ❑ Щелкните мышью на переменной, относительно которой надо произвести символьные действия.

ПРИМЕЧАНИЕ

Выделение черным цветом используется в Mathcad, когда какие-либо действия нужно выполнить над частью выражения: вырезать в буфер, скопировать, удалить, изменить размер или стиль шрифта и т. д.

Выполняя символьные операции, следует помнить, что не всякое выражение поддается символьным преобразованиям, что поставленная задача часто не имеет символьного решения (в отличие от численного).

Рассмотрение символьных операций начнем с последнего пункта меню **Symbolics** ▶ **Evaluation Style** (Стиль вычислений). Этот пункт позволяет выбрать стиль выводимого символьного решения: горизонтально — рядом с исходным выражением, вертикально — под ним, с комментариями и без них (рис. 5.3). На рис. 5.4–5.9 выбрано горизонтальное расположение ответов с комментариями.

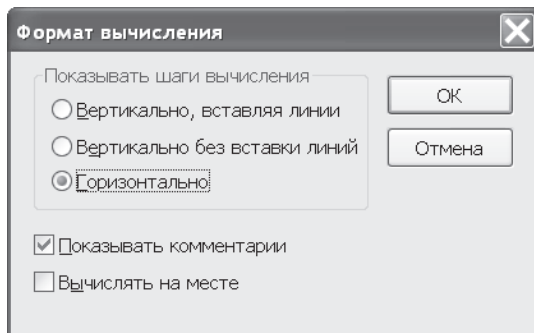


Рис. 5.3. Меню выбора стиля вывода символьного результата

ВНИМАНИЕ

Перед выполнением символьных операций убедитесь, что включен (помечен флажком) пункт **Automatic Calculation** (Автоматический режим вычислений) в меню **Math** (Математика).

5.2.1. Символьная оценка выражения

Для выполнения символьной оценки:

- ❑ выделите выражение или его часть;
- ❑ в главном меню выберите команду **Symbolics** ▶ **Evaluate** (Символьные вычисления ▶ Вычислить).

Этот пункт предлагает на выбор три варианта вычисления выражения: Symbolically (Символьное), Floating Point (С плавающей точкой), Complex (В комплексной форме).

Символьный процессор по-разному обрабатывает числа с десятичной точкой и без нее. Сформулируем общие правила:

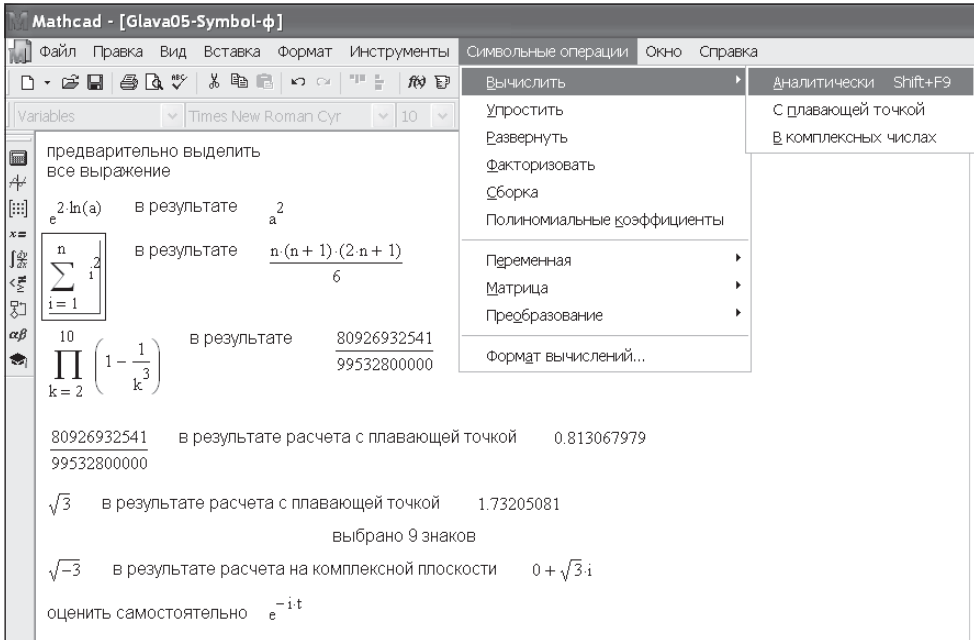


Рис. 5.4. Примеры символьной оценки выражений

- ❑ Если символьный процессор получает числа, содержащие десятичную точку, то любые возвращаемые численные результаты будут десятичными приближениями точного решения.
- ❑ Если символьный процессор получает числа без десятичных точек, то любые возвращаемые численные результаты будут выражены без десятичных точек, когда это возможно.
- ❑ Если символьная операция дает ответ в виде десятичной дроби, то этот ответ всегда отображается с 20 значащими цифрами. На это отображение не действуют ни локальный, ни глобальный числовой формат.

Примеры использования команды Evaluate приведены на рис. 5.4.

5.2.2. Упрощение выражений

Команда Simplify (Упростить) служит для основных алгебраических и тригонометрических упрощений выбранного выражения. Она выполняет арифметические преобразования, сокращает общие множители, использует основные тождества для тригонометрических и обратных функций, упрощает степени. Можно

упрощать как все выражение целиком, так и его части. Если преобразование всего выражения не дает желаемого результата, попробуйте выделить и упростить часть выражения. Примеры использования команды `Simplify` приведены на рис. 5.5.

Simplify - упростить

$$\sum_{i=0}^{\infty} a^i \quad \text{упрощение до} \quad \begin{cases} \frac{1}{a-1} & \text{if } a \neq 1 \wedge |a| < 1 \\ \infty & \text{if } 1 \leq a \end{cases}$$

$$\sum_{k=0}^3 \left(\frac{3!}{k! \cdot 3 - k!} \cdot x^k \cdot 2^{3-k} \right) \quad \text{упрощение до} \quad \sum_{k=0}^3 \left(\frac{3}{k!} \cdot x^k \cdot 2^{3-k} \right)$$

$$\frac{x^2 - 3x - 4}{x - 4} + 2x - 5 \quad \text{упрощение до} \quad x + 1 + 2x - 5$$

$$\sin(2x) + \cos(3x) \quad \text{упрощение до} \quad \sin(2x) + \cos(3x)$$

$$\frac{3}{19.0} + \frac{47}{93} \quad \text{упрощение до} \quad 0.66327108092812676853$$

Рис. 5.5. Примеры упрощения выражений

5.2.3. Раскрытие выражений

Операция символьного разложения по смыслу противоположна операции упрощения. В ходе разложения раскрываются все суммы и произведения, а сложные тригонометрические выражения разлагаются с помощью тригонометрических тождеств. Если выражение — дробь, числитель будет разложен и выражение будет представлено как сумма дробей. Примеры использования команды `Expand` (Раскрытие) приведены на рис. 5.6.

Expand - разложить

$$(x+4)^4 \quad \text{расширение до} \quad x^4 + 16x^3 + 96x^2 + 256x + 256$$

$$\tan(3x) + \cos(2x) \quad \text{расширение до} \quad \cos(x)^2 - \sin(x)^2 - \frac{3 \cdot \tan(x) - \tan(x)^3}{3 \cdot \tan(x)^2 - 1}$$

$$(x+1) \cdot (x-1) \quad \text{расширение до} \quad x^2 - 1$$

$$\cos(5x) \quad \text{расширение до} \quad \cos(x)^5 - 10 \cdot \cos(x)^3 \cdot \sin(x)^2 + 5 \cdot \cos(x) \cdot \sin(x)^4$$

разложить самостоятельно

$$\tan(3x) \quad (x+3) \cdot (2x+4)$$

Рис. 5.6. Примеры разворачивания выражений

5.2.4. Разложение на множители

Команда Factor (Разложить на множители) позволяет разложить полиномы на произведение более простых полиномов, а целые числа — на простые сомножители. Команда будет объединять сумму дробей в одну дробь и упрощать многоэтажную дробь с несколькими дробными чертами. Примеры использования команды Factor приведены на рис. 5.7.

ВНИМАНИЕ

Mathcad разлагает на множители только то, что выделено.

разложить самостоятельно

$$\tan(3 \cdot x) \quad (x + 3) \cdot (2 \cdot x + 4)$$

Factor - разложить на множители

19125 в результате факторизации $3^2 \cdot 5^3 \cdot 17$

$a \cdot b + a \cdot c$ в результате факторизации $a \cdot (b + c)$

$x^3 + y^3$ в результате факторизации $(x + y) \cdot (x^2 - x \cdot y + y^2)$

$\frac{2 \cdot x^3 - 3 \cdot x + 1}{x^3 + 2 \cdot x^2 - 9 \cdot x - 18}$ в результате факторизации $\frac{(x - 1) \cdot (2 \cdot x^2 + 2 \cdot x - 1)}{(x - 3) \cdot (x + 3) \cdot (x + 2)}$

разложить самостоятельно

$$2 \cdot a^2 \cdot y^5 + a \cdot b \cdot y^5 - a \cdot b \cdot y^3 - 2 \cdot a^2 \cdot y^3 \quad \frac{\frac{3}{x+1} - \frac{1}{x+3}}{\frac{3}{1-x} - \frac{1}{3-x}}$$

Рис. 5.7. Примеры разложения выражений на множители

5.2.5. Приведение подобных слагаемых

Команда Collect (Привести подобные слагаемые) объединяет члены, содержащие одинаковые степени выделенной курсором переменной. Примеры использования команды Collect приведены на рис. 5.8.

Collect - приведение подобных слагаемых

$$x + 2 \cdot x - 3 \cdot y + 4 \cdot y + x \cdot y - 2 \cdot x^2 \cdot y - 3 \cdot x \cdot y^2$$

по переменной x в результате приведения подобных членов

$$-2 \cdot y \cdot x^2 + (-3 \cdot y^2 + y + 3) \cdot x + y$$

по переменной y в результате приведения подобных членов

$$-3 \cdot x \cdot y^2 + (-2 \cdot x^2 + x + 1) \cdot y + 3 \cdot x$$

самостоятельно приведите

подобные слагаемые

по x и по y

$$x^2 - a \cdot y \cdot x^2 + 2 \cdot y^2 \cdot x - x$$

Рис. 5.8. Примеры приведения подобных слагаемых

5.2.6. Коэффициенты полинома

Многие выражения или части выражения могут быть представлены в виде полиномов от выделенной переменной. Коэффициенты полинома могут быть найдены символьным процессором. При этом коэффициенты могут быть сложными функциями других переменных. Примеры использования команды `Polynomial Coefficients` (Коэффициенты полинома) приведены на рис. 5.9.

Нахождение коэффициентов полинома `Polynomial Coefficients`

$3 \cdot a^3 \cdot x^4 \cdot y + 2 \cdot x^2 \cdot (a + x \cdot y)^3 - a \cdot x \cdot y^2$ имеются коэффициенты

найденные коэффициенты полинома, разложенного по степеням x

имеются коэффициенты

$$\begin{pmatrix} 0 \\ -a \cdot y^2 \\ 2 \cdot a^3 \\ 6 \cdot a^2 \cdot y \\ 3 \cdot a^3 \cdot y + 6 \cdot a \cdot y^2 \\ 2 \cdot y^3 \end{pmatrix} \begin{pmatrix} \text{по степеням } y \\ 2 \cdot x^2 \cdot a^3 \\ 3 \cdot a^3 \cdot x^4 + 6 \cdot x^3 \cdot a^2 \\ 6 \cdot x^4 \cdot a - a \cdot x \\ 2 \cdot x^5 \end{pmatrix} \begin{pmatrix} \text{по степеням } a \\ 2 \cdot x^5 \cdot y^3 \\ 6 \cdot x^4 \cdot y^2 - x \cdot y^2 \\ 6 \cdot x^3 \cdot y \\ 3 \cdot x^4 \cdot y + 2 \cdot x^2 \end{pmatrix}$$

самостоятельно разложить по степеням a, b, c

$$(a^3 - b - c^2)^3 \cdot (b \cdot c - 3 \cdot a)^2$$

Рис. 5.9. Примеры нахождения коэффициентов полинома

5.2.7. Вычисление пределов

Для вычисления пределов в Mathcad выполните следующие действия:

- На математической панели выберите кнопку со знаком интеграла, откроется панель `Calculus` (Исчисление), на которой внизу есть три оператора вычисления пределов. Выберите один из них.
- Введите выражение в поле ввода справа от `lim`.
- В поле ввода под словом `lim` введите имя переменной, по которой надо вычислить предел, и ее предельное значение.
- Выделите уголком или черным цветом все выражение целиком.
- В главном меню Mathcad выберите `Symbolics` ▶ `Evaluate` ▶ `Symbolically` (Символьные вычисления ▶ Вычислить ▶ Символьно). Mathcad возвращает значение предела, если оно существует. Примеры вычисления пределов приведены на рис. 5.10.

Вычислить предел выражения можно только символично.

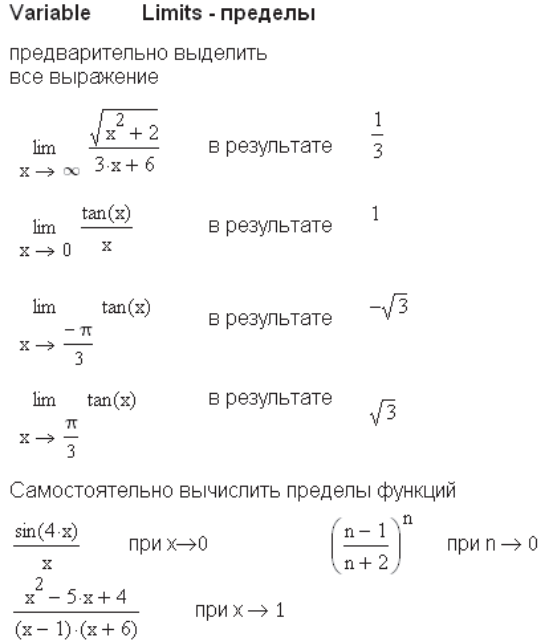


Рис. 5.10. Примеры вычисления пределов функций

5.2.8. Замена переменной

Команда Substitute (Подстановка) заменяет выделенным и скопированным в буфер выражением заданную переменную. Чтобы использовать эту команду, нужно:

- выделить выражение, которое будет заменять переменную;
- скопировать его в буфер обмена;
- выделить переменную, которую надо заменить;
- в главном меню выбрать Symbolics ▶ Variable ▶ Substitute (Символьные вычисления ▶ Переменная ▶ Подстановка), как показано на рис. 5.11.

Mathcad подставит выражение из буфера обмена вместо выделенной переменной.

Примеры подстановки переменной показаны на рис. 5.12.

5.2.9. Вычисление производной или интеграла

С помощью меню Symbolics производную или интеграл можно вычислить двумя способами:

1. В нужном выражении, записанном без знаков производной или интеграла, выделите переменную, по которой надо вычислить производную или взять интеграл. В главном меню выберите команду Symbolics ▶ Variable ▶ Differentiate (Символьные вычисления ▶ Переменная ▶ Вычислить производную) или

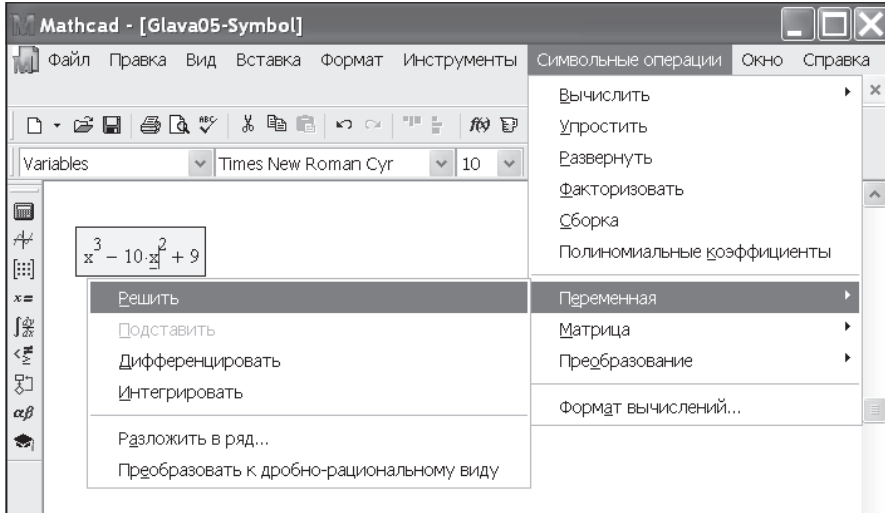


Рис. 5.11. Меню Symbolics ▶ Variable

Variable Substitute - подставить в выражение

подставить $f(\sin(x))$ вместо y

$f(\sin(x))$ скопировать в буфер

$\cos(y) + \sqrt{1+y^2}$ в результате подстановки $\cos(f(\sin(x))) + \sqrt{f(\sin(x))^2 + 1}$

Вместо x подставить вектор $\begin{pmatrix} a \\ b \end{pmatrix}$

$\begin{pmatrix} a \\ b \end{pmatrix}$ скопировать в буфер

в результате подстановки

$$\begin{pmatrix} \cos(x) & \sin(x) \\ -\sin(x) & \cos(x) \end{pmatrix}$$

$$\begin{bmatrix} \cos\left(\begin{pmatrix} a \\ b \end{pmatrix}\right) & \sin\left(\begin{pmatrix} a \\ b \end{pmatrix}\right) \\ -\sin\left(\begin{pmatrix} a \\ b \end{pmatrix}\right) & \cos\left(\begin{pmatrix} a \\ b \end{pmatrix}\right) \end{bmatrix}$$

Рис. 5.12. Примеры замены переменных

Symbolics ▶ Variable ▶ Integrate (Символьные вычисления ▶ Переменная ▶ Вычислить интеграл).

2. На панели Calculus (Вычисления) выберите знак интеграла или производной и запишите с его помощью нужное выражение. В главном меню выберите команду Symbolics ▶ Evaluate ▶ Symbolically (Символьные вычисления ▶ Вычислить ▶ Символьно).

Примеры вычисления производной и интеграла приведены на рис. 5.13.

Variable**Differentiate Integrate - дифференцировать интегрировать**

$x^2 \cdot e^x$	в результате дифференцирования	$2 \cdot x \cdot \exp(x) + x^2 \cdot \exp(x)$
$x^2 \cdot e^x$	в результате дифференцирования	$x^2 \cdot \exp(x) - 2 \cdot x \cdot \exp(x) + 2 \cdot \exp(x)$
$\frac{d^2}{dz^2}(z \cdot \operatorname{atan}(z))$	в результате дифференцирования	$\frac{2}{1+z^2} - 2 \cdot \frac{z^2}{(1+z^2)^2}$
$\int_0^{\infty} e^{-x^2} dx$	в результате дифференцирования	$\frac{1}{2} \cdot \pi^{\frac{1}{2}}$
Самостоятельно вычислить производную и интеграл от следующих выражений		
$\frac{x+a}{x^2+b}$	$x^2 - 2 \cdot x - 10$	$\frac{d}{dx} \sqrt{a^2 - x^2}$
$\tan(\sqrt{x})$	$\frac{\sin(x)}{x+2}$	$\frac{\ln(x)^2}{x}$

Рис. 5.13. Примеры символьного дифференцирования и интегрирования выражений

5.2.10. Символьное решение уравнений и неравенств

Решить уравнение символьно труднее, чем численно. Может оказаться, что решения в символьном виде не существует. Чтобы решить уравнение символьно, следует:

- напечатать уравнение, используя жирный знак равенства (знак логического равенства, а не знак присвоения значения). Если в выражении отсутствует знак равенства, Mathcad полагает, что выражение равно нулю;
- выделить переменную, относительно которой надо решить уравнение;
- в главном меню выбрать команду **Symbolics** ▶ **Variable** ▶ **Solve** (Символьные вычисления ▶ Переменная ▶ Решить).

Если решений несколько, Mathcad выводит вектор решений.

Примеры символьного решения уравнений приведены на рис. 5.14.

5.2.11. Разложение в ряд Тейлора

Чтобы разложить выражение в ряд Тейлора с остаточным членом в форме Пиано, выполните следующие действия:

- выделите переменную в выражении, которое надо разложить в ряд;
- в главном меню выберите команду **Symbolics** ▶ **Variable** ▶ **Expand to Series** (Символьные вычисления ▶ Переменная ▶ Разложить в ряд);
- в появившемся диалоговом окне укажите порядок остаточного члена.

Variable	Solve - решить уравнение или неравенство
$x^3 - 10x^2 + 9$	имеются решения
$x^3 - 5x^2 - 4x + 20 > 0$	
имеются решения	$\left[\begin{array}{l} (-2 < x) \cdot (x < 2) \\ 5 < x \end{array} \right]$
Самостоятельно решить уравнение и неравенство	
$x^2 + x + 3 = 0$	$x^2 + 3x - 7 < 0$

$$\left(\begin{array}{l} 1 \\ \frac{9}{2} + \frac{3\sqrt{13} + 9}{2} \\ \frac{9}{2} - \frac{3\sqrt{13}}{2} \end{array} \right)$$

Рис. 5.14. Символьное решение уравнений и неравенств

В полученном ответе остаточный член обозначен $O(x^6)$, то есть отброшены члены ряда, содержащие x^6 и выше. Прежде чем использовать разложение для дальнейших вычислений, остаточный член надо удалить.

Ряд Тейлора для функции обычно сходится только в малой окрестности выбранной точки. В качестве точки разложения Mathcad принимает нуль. Для разложения функции в точке, отличной от нуля, необходима замена переменной. Например, для разложения в ряд функции $\ln(x)$ в окрестности точки $x = 1$ надо разложить в ряд функцию $\ln(x + 1)$, а затем переменную x заменить разностью $x - 1$. Этот и другие примеры разложения в ряд приведены на рис. 5.15.

5.2.12. Разложение на простейшие дроби

Сложное выражение можно преобразовать в сумму простейших дробей:

- Выделите переменную в заданном выражении.
- В главном меню выберите команду **Symbolics** ▶ **Variable** ▶ **Convert to Partial Fraction** (Символьные вычисления ▶ Переменная ▶ Разложить на простейшие дроби).

Символьный процессор будет пытаться разложить знаменатель выражения на множители. Mathcad не станет разлагать выражение, которое содержит десятичные точки. Примеры разложения выражения на простейшие дроби приведены на рис. 5.16.

5.2.13. Символьные операции с матрицами

Mathcad позволяет выполнить с матрицами ряд символьных операций. Для этого

- создайте матрицу с цифровыми или буквенными элементами;
- выделите ее любым способом;
- в главном меню выберите пункт **Symbolics** ▶ **Matrix** (Символьные вычисления ▶ Матрица).

$\cos(x)$ преобразование в серию $1 - \frac{x^2}{2} + \frac{x^4}{24}$
 $1 - \frac{1}{2} \cdot x^2 + \frac{1}{24} \cdot x^4 + O(x^6)$
 разбиение на 2 строки
 длинного ответа
 $\cos(x)$ преобразование в серию

$$\left(\begin{array}{l} 1 - \frac{1}{2} \cdot x^2 + \frac{1}{24} \cdot x^4 - \frac{1}{720} \cdot x^6 + \frac{1}{40320} \cdot x^8 - \frac{1}{3628800} \cdot x^{10} \dots \\ + \frac{1}{479001600} \cdot x^{12} - \frac{1}{87178291200} \cdot x^{14} + O(x^{16}) \end{array} \right)$$

 $\frac{2 \cdot x^2 - 3 \cdot x + 1}{x^3 + 2 \cdot x^2 - 9 \cdot x + 18}$ преобразование в серию $\frac{1}{18} - \frac{5}{36} \cdot x + \frac{23}{648} \cdot x^2 + \frac{13}{432} \cdot x^3 + O(x^4)$
 Разложить в ряд функцию $\ln(x)$ выбрано 4 члена ряда
 $\ln(x+1)$ преобразование в серию $1 \cdot x - \frac{1}{2} \cdot x^2 + \frac{1}{3} \cdot x^3 + O(x^4)$
 замена переменной
 $x - 1$
 в результате подстановки
 $x - 1 - \frac{1}{2} \cdot (x-1)^2 + \frac{1}{3} \cdot (x-1)^3 + O[(x-1)^4]$
 Самостоятельно разложить в ряд
 $\sin(x) \frac{\tan(x)}{e^{x^2}}$
 $x^2 - 6x + 3$

Рис. 5.15. Примеры разложения функций в ряд Тейлора

Разложить на элементарные дроби Convert to Partial fractions

$\frac{2 \cdot x^2 - 3 \cdot x + 1}{x^3 + 2 \cdot x^2 - 9 \cdot x - 18}$ расширение элементарных дробей до
 $\frac{1}{3 \cdot (x-3)} + \frac{14}{3 \cdot (x+3)} - \frac{3}{x+2}$
 $\left(\prod_{n=1}^3 \frac{1}{x+n} \right) \cdot x^4$ расширение элементарных дробей до
 $x - 6 + \frac{1}{2 \cdot (x+1)} - \frac{16}{x+2} + \frac{81}{2 \cdot (x+3)}$

Самостоятельно разложить на элементарные дроби

$\frac{x^3 - 2 \cdot x^2 + 3 \cdot x - 6}{(x-2)^3} \sum_{a=1}^4 \frac{a \cdot x + x - 2}{(a+x)^2}$

Рис. 5.16. Примеры разложения выражений на простейшие дроби

Откроется подменю из трех пунктов: Transpose (Транспонировать), Invert (Найти обратную) и Determinant (Вычислить определитель). Выбрав один из этих пунктов, можно найти результат вычисления. На рис. 5.17 указанные операции выполнены на примере матрицы направляющих косинусов для плоской задачи.

Попутно эти примеры доказывают свойство ортогональности матрицы направляющих косинусов, то есть то, что обратная матрица равна транспонированной.

Чтобы выполнить арифметические действия над несколькими (чаще всего двумя) матрицами, надо выделить целиком все матричное выражение и в главном меню выбрать команду **Symbolics** ▶ **Evaluate** ▶ **Symbolically** (Символьные вычисления ▶ Вычислить ▶ Символьно). На рис. 5.17 показано символьное решение системы линейных алгебраических уравнений, записанных в матричном виде.

Символьные операции с матрицами

$$\begin{pmatrix} \cos(x) & \sin(x) & 0 \\ -\sin(x) & \cos(x) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{в результате} \quad \begin{pmatrix} \cos(x) & -\sin(x) & 0 \\ \sin(x) & \cos(x) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{транспонирования матрицы}$$

$$\begin{pmatrix} \cos(x) & \sin(x) & 0 \\ -\sin(x) & \cos(x) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{в результате} \quad \frac{1}{\cos(x)^2 + \sin(x)^2} \begin{pmatrix} \cos(x) & -\sin(x) & 0 \\ \sin(x) & \cos(x) & 0 \\ 0 & 0 & \cos(x)^2 + \sin(x)^2 \end{pmatrix} \quad \text{обращения матрицы}$$

$$\begin{pmatrix} \cos(x) & \sin(x) & 0 \\ -\sin(x) & \cos(x) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{имеется определитель} \quad \cos(x)^2 + \sin(x)^2$$

Решение СЛАУ

$$\begin{aligned} a_1 \cdot x + b_1 \cdot y &= c_1 \\ a_2 \cdot x + b_2 \cdot y &= c_2 \end{aligned} \quad \begin{pmatrix} x \\ y \end{pmatrix} := \begin{pmatrix} a_1 & b_1 \\ a_2 & b_2 \end{pmatrix}^{-1} \cdot \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}$$

Вычислить аналитически

в результате

$$\begin{pmatrix} x \\ y \end{pmatrix} := \begin{pmatrix} \frac{b_2}{a_1 \cdot b_2 - b_1 \cdot a_2} \cdot c_1 - \frac{b_1}{a_1 \cdot b_2 - b_1 \cdot a_2} \cdot c_2 \\ \frac{-a_2}{a_1 \cdot b_2 - b_1 \cdot a_2} \cdot c_1 + \frac{a_1}{a_1 \cdot b_2 - b_1 \cdot a_2} \cdot c_2 \end{pmatrix}$$

Рис. 5.17. Примеры символьных операций с матрицами

5.2.14. Вычисление интегральных преобразований

Mathcad позволяет вычислить символьно три интегральных преобразования: Фурье, Лапласа и z -преобразование.

Прямое преобразование Фурье возвращает функцию от ω , определяемую формулой

$$\int_{-\infty}^{+\infty} f(t) e^{-i\omega t} dt,$$

где $f(t)$ — выражение, которое надо преобразовать.

Обратное преобразование Фурье возвращает функцию от t , задаваемую выражением

$$\frac{1}{2\pi} \int_{-\infty}^{+\infty} F(\omega) e^{i\omega t} d\omega,$$

где $F(\omega)$ — выражение, которое надо преобразовать.

Прямое преобразование Лапласа возвращает функцию от s , определяемую формулой

$$\int_0^{+\infty} f(t) e^{-st} dt,$$

где $f(t)$ — выражение, которое надо преобразовать.

Обратное преобразование Лапласа возвращает функцию от t , задаваемую выражением

$$\frac{1}{2\pi i} \int_{\sigma-i\infty}^{\sigma+i\infty} F(s) e^{st} ds,$$

где $F(s)$ — выражение, которое надо преобразовать, и все его особенности находятся слева от линии $\text{Re}(s) = \sigma$.

Прямое z -преобразование возвращает функцию от s , определяемую формулой

$$\sum_{n=0}^{+\infty} f(n) z^{-n},$$

где $f(n)$ — выражение, которое нужно преобразовать.

Обратное z -преобразование возвращает функцию от n , задаваемую контурным интегралом вокруг начала координат

$$\frac{1}{2\pi} \oint_C F(z) z^{n-1} dz,$$

где $F(z)$ — выражение, которое нужно преобразовать; C — контур, включающий в себя все особенности подынтегрального выражения.

Для выполнения любого из шести указанных преобразований выполните следующее:

- введите нужное выражение;
- выделите переменную курсором;
- в главном меню выберите Symbolics ▶ Transform ▶ Fourier (Символьные операции ▶ Преобразование ▶ Фурье) или укажите другую нужную команду: Inverse Fourier, Laplace, Inverse Laplace, Z, Inverse Z (Обратное Фурье, Лапласа, Обратное Лапласа...).

Примеры указанных преобразований приведены на рис. 5.18. В выводимых ответах Mathcad использует специальные функции (рис. 5.19). Так, в преобразованиях Фурье используется функция Dirac (Дельта-функция Дирака). Об этой и других специальных функциях можно прочитать в Mathcad Help (Справка Mathcad).

Интегральные преобразования

$\cos(x)^2$	имеется преобразование Фурье	$\frac{1}{2} \cdot \pi \cdot \Delta(\omega - 2) + \pi \cdot \Delta(\omega) + \frac{1}{2} \cdot \pi \cdot \Delta(\omega + 2)$
	имеется обратное преобразование Фурье	$\frac{1}{4} \cdot \exp(2 \cdot i \cdot t) + \frac{1}{2} + \frac{1}{4} \cdot \exp(-2 \cdot i \cdot t)$
$\cos(x)^2$	имеется преобразование Лапласа	$\frac{s^2 + 2}{s^3 + 4 \cdot s}$
	имеется обратное преобразование Лапласа	$\frac{1}{2} \cos(2 \cdot t) + \frac{1}{2}$
$(x + a)^2$	имеется Z-преобразование	$\frac{z \cdot (a^2 \cdot z^2 - 2 \cdot a^2 \cdot z + a^2 + 2 \cdot a \cdot z - 2 \cdot a + z + 1)}{(z - 1)^3}$
	имеется обратное Z-преобразование	$n^2 + 2 \cdot a \cdot n + a^2$

Рис. 5.18. Примеры интегральных преобразований функций

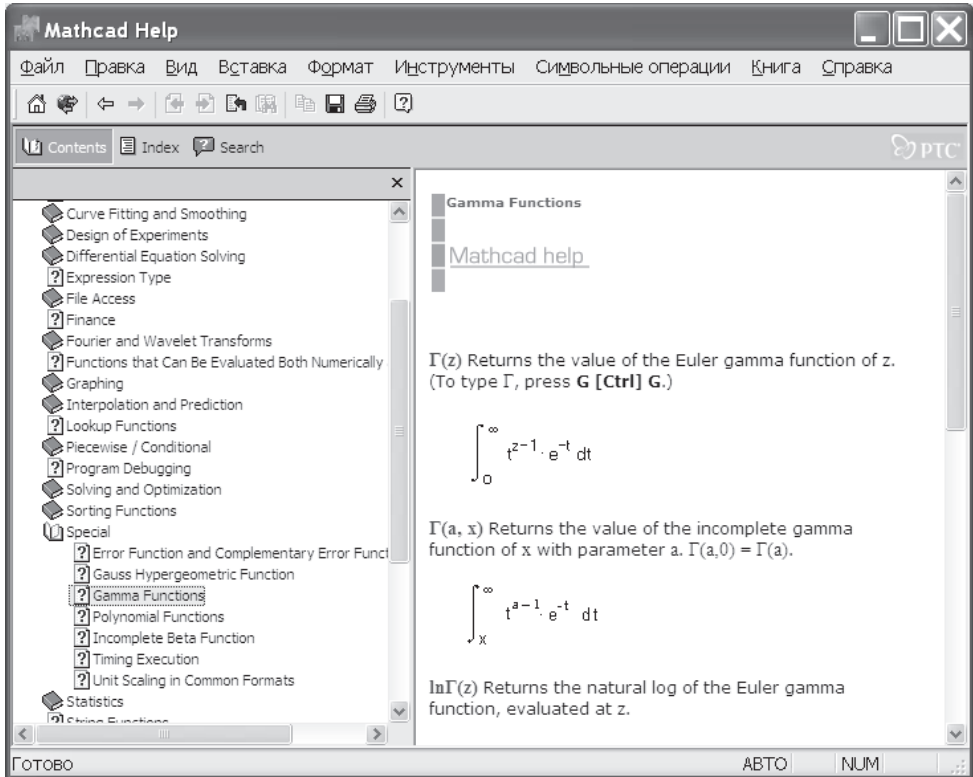


Рис. 5.19. Поиск специальных функций в Mathcad Help

5.3. Использование символьной панели инструментов

Символьная панель инструментов вызывается из математической панели щелчком на значке Symbolic Keyword Toolbar (значок с изображением шапочки академика). Содержание панели показано на рис. 5.20.

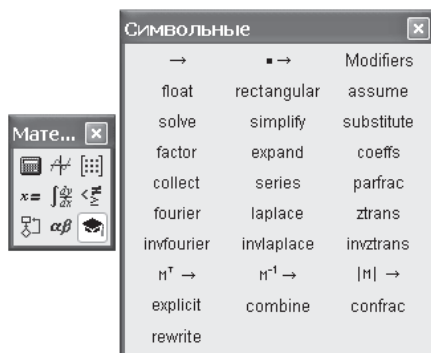


Рис. 5.20. Символьная панель инструментов

В Mathcad существует символьный знак равенства \rightarrow — он вводится с клавиатуры сочетанием клавиш $\text{Ctrl}+$ (точка) или с панелей Symbolic Keyword Toolbar (Символьная панель инструментов) и Evaluation Toolbar (Панель вычислений). В отличие от обычного знака равенства, символьный знак равенства возвращает не число, а выражение.

По умолчанию символьный знак равенства упрощает выражение слева от знака аналогично команде Evaluate Symbolically (Вычислить символьно) из меню Symbolics. Для всестороннего контроля выполнения символьных вычислений нужно использовать символьную панель инструментов.

ПРИМЕЧАНИЕ

Как уже говорилось, при использовании меню Symbolics Mathcad игнорирует предыдущие определения всех величин, входящих в рассматриваемое выражение, и **не учитывает функции пользователя**.

При использовании символьного знака равенства и панели инструментов Symbolic Mathcad учитывает все предыдущие определения и при пересчете всей программы пересчитывает и результат символьных вычислений **и работает с функциями пользователя**.

Чтобы отменить предыдущее присвоение значения переменной, например, x , введите выражение $x:=x$.

На символьной панели инструментов (см. рис. 5.20) есть два символьных знака равенства: с одним местом ввода и с двумя местами ввода. Одно место ввода предназначено для ввода выражения или функции (встроенной или пользовательской), второе — для ввода ключевого слова, означающего выполняемое действие.

Символьный знак равенства с двумя местами ввода позволяет создавать программы символьных вычислений, включающих в себя несколько последовательных действий.

5.3.1. Символьные операции без использования ключевых слов

Одноместный символьный знак равенства используется для символьного вычисления интегралов, производных, сумм и произведений (рис. 5.21) и эквивалентен выполнению команды Simplify (Упростить).

Очистка использованного ранее значения переменной $x := x$

$$\sum_{n=0}^{\infty} \frac{x^n}{2^n \cdot n!} \rightarrow e^{\frac{x}{2}}$$

Символьное интегрирование и дифференцирование

$$\frac{d^2}{dz^2} \tan(z) \rightarrow 2 \cdot \tan(z) \cdot (\tan(z)^2 + 1)$$

$$\frac{d^4}{dz^4} \tan(z) \rightarrow 16 \cdot \tan(z) \cdot (\tan(z)^2 + 1)^2 + 8 \cdot \tan(z)^3 \cdot (\tan(z)^2 + 1)$$

$$\int \tan(z) dz \rightarrow -\ln(\cos(z))$$

$$\int_a^b \int_b^c x \cdot y dx dy \rightarrow \frac{(a^2 - b^2) \cdot (b^2 - c^2)}{4}$$

Рис. 5.21. Символьные вычисления с одноместным знаком равенства

При выполнении большинства символьных операций удобно использовать функции пользователя (рис. 5.22). Это облегчает понимание смысла выполняемых действий и делает документ менее громоздким.

5.3.2. Символьные операции с использованием ключевых слов

На символьной панели инструментов находится 26 кнопок с названиями символьных операций. Почти все эти операции есть и в меню Symbolics. При нажатии одной из кнопок на экране появляется шаблон символьной операции с указанием ключевого слова (двухместный символьный знак равенства). В шаблоне надо заполнить места ввода, а если остаются или появляются лишние места ввода, их надо удалить (выделить место ввода курсором и нажать клавишу Delete).

ПРИМЕЧАНИЕ

Символьную операцию можно ввести и с клавиатуры. Для этого вначале надо нажать Ctrl+Shift+. (точка). Появится шаблон с двумя местами ввода. В левое место ввода ввести математическое выражение, а в правое — ввести ключевое слово для данной операции.

использование функций пользователя

$$f(x) := x^2 - 6x + 2x - 12 \quad \frac{d}{dx} f(x) \rightarrow 2x - 4$$

$$\int_a^b f(x) dx \rightarrow 2 \cdot a^2 - \frac{a^3}{3} + 12 \cdot a + \frac{b^3}{3} - 2 \cdot b^2 - 12 \cdot b$$

Решение уравнений и неравенств

$$f(xa) = 0 \text{ solve, } xa \rightarrow \begin{pmatrix} -2 \\ 6 \end{pmatrix}$$

$$f(xa) > 0 \text{ solve, } xa \rightarrow xa < -2 \vee 6 < xa$$

$$xa^3 - 5 \cdot xa^2 - 4 \cdot xa + 20 > 0 \text{ solve, } xa \rightarrow 5 < xa \vee -2 < xa < 2$$

Рис. 5.22. Использование функций пользователя в символьных вычислениях

Команды, их шаблоны и действия команд приведены в табл. 5.1. В ней жирные точки обозначают места ввода выражений (слева) и ключевых слов (справа).

Таблица. 5.1. Действия, выполняемые с помощью ключевых слов

Команда	Шаблон	Действие команды
float	• float, • →	Вычисления с плавающей точкой
rectangular	• rectangular →	Возвращает комплексное число
expand	• expand, • →	Раскрытие степеней и произведений, сумм нескольких переменных
solve	• solve, • →	Решение уравнений и систем уравнений для указанной после запятой переменной
simplify	• simplify →	Упрощение выражений
substitute	• substitute, • = • →	Вычисление выражений с подстановкой определенных значений или с заменой переменной
collect	• collect, • →	Упрощение выражений представлением их суммой простых полиномов
series	• series, • = • →	Разложение выражения в степенной ряд. Число членов вводится в правое поле ввода
assume	• assume, • →	Вычисления с присвоением определенных значений переменных (действительное, мнимое, целое, >0, <0)
parfrac	• parfrac →	Разложение выражения на простые дроби
coeffs	• coeffs, • →	Определение вектора коэффициентов полинома
factor	• factor, • →	Разложение выражения на произведения
fourier	• fourier, • →	Прямое преобразование Фурье

продолжение ↗

Таблица. 5.1 (продолжение)

Команда	Шаблон	Действие команды
laplace	• laplace, • →	Прямое преобразование Лапласа
ztrans	• ztrans, • →	Прямое z-преобразование
invfourier	• invfourier, • →	Обратное преобразование Фурье
invlaplace	• invlaplace, • →	Обратное преобразование Лапласа
invztrans	• invztrans, • →	Обратное z-преобразование
$M^T \rightarrow$	• $^T \rightarrow$	Транспонирование матрицы
$M^{-1} \rightarrow$	• $^{-1} \rightarrow$	Обращение матрицы
$ M \rightarrow$	$ \cdot \rightarrow$	Вычисление определителя матрицы
explicit	• explicit →	Подстановка числовых значений и размерностей в формулу
combine	<i>Употребляются в сочетании с другими словами</i>	Упрощает выражение с exp или ln
confrac		Разложение непрерывной дроби на фракции
rewrite		Переписывает выражение с указанными элементарными функциями
Modifiers		Вывод панели Modifier

Подробно о предназначении и использовании каждого ключевого слова можно узнать, выбрав Mathcad Help ► Symbolics and Keywords ► Using keywords for symbolic operations (Mathcad Помощь ► Символы и ключевые слова ► Использование ключевых слов для символьных операций), а также посмотрев в начале главы разделы 5.2.1–5.2.9. Примеры использования ключевых слов с односторонним символьным знаком равенства приведены на рис. 5.23.

Некоторые команды, имеющие в своих шаблонах места ввода, стоящие после запятой, могут иногда использоваться и без указания там аргумента. Так применяются команды expand, factor, collect, solve, coeff, если в выражении есть только один аргумент (рис. 5.23 и 5.24).

Далее приведем некоторые пояснения к отдельным ключевым словам.

При использовании команды substitute (Заменить) необходимо задать вариант подстановки. Если нужно заменить не одну переменную, а несколько, то второй и последующие варианты подстановки вводятся через запятую и набираются с клавиатуры (см. рис. 5.24).

Ключевое слово assume позволяет ввести ограничения на значения переменных, что бывает необходимо, если, например, функция имеет разрыв в некоторой точке (см. рис. 5.23). Ключевое слово assume — должно появляться после ключевого слова, обозначающего действие, на которое вы хотите ввести ограничения.

Ключевое слово coeff выводит коэффициенты полинома, начиная с наименьшей степени. Если полином содержит более одной переменной, необходимо после ключевого слова coeff поставить запятую и ввести имя переменной, относительно которой нужно вывести коэффициенты. В Mathcad 14 появилась возможность,

$x := x$ $y := y$ очистка переменных x и y

assume assume должно появляться после ключевого слова обозначающего действие на которое вы хотите ввести ограничения

$$\begin{array}{l}
 (x^3 - 1)(x^2 - 2) = 0 \text{ solve} \rightarrow \begin{pmatrix} 1 \\ \sqrt{2} \\ -\sqrt{2} \\ -\frac{1}{2} + \frac{\sqrt{3}i}{2} \\ -\frac{1}{2} - \frac{\sqrt{3}i}{2} \end{pmatrix} \quad (x^3 - 1)(x^2 - 2) = 0 \left| \begin{array}{l} \text{solve} \\ \text{assume, } x = \text{real} \end{array} \right. \rightarrow \begin{pmatrix} 1 \\ \sqrt{2} \\ -\sqrt{2} \end{pmatrix} \\
 (x^3 - 1)(x^2 - 2) = 0 \text{ solve} \rightarrow \begin{pmatrix} 1 \\ \sqrt{2} \\ -\sqrt{2} \\ -\frac{1}{2} + \frac{\sqrt{3}i}{2} \\ -\frac{1}{2} - \frac{\sqrt{3}i}{2} \end{pmatrix} \quad (x^3 - 1)(x^2 - 2) = 0 \left| \begin{array}{l} \text{solve} \\ \text{assume, } x = \text{RealRange}(0, 2) \end{array} \right. \rightarrow \begin{pmatrix} 1 \\ \sqrt{2} \end{pmatrix} \\
 (x^3 - 1)(x^2 - 2) = 0 \text{ solve} \rightarrow \begin{pmatrix} 1 \\ \sqrt{2} \\ -\sqrt{2} \\ -\frac{1}{2} + \frac{\sqrt{3}i}{2} \\ -\frac{1}{2} - \frac{\sqrt{3}i}{2} \end{pmatrix} \quad (x^3 - 1)(x^2 - 2) = 0 \left| \begin{array}{l} \text{solve} \\ \text{assume, } x = \text{integer} \end{array} \right. \rightarrow 1
 \end{array}$$

coeff работает только в Mathcad 14

$$x^4 + 3x^2 + 7x - 3 \text{ coeffs} \rightarrow \begin{pmatrix} -3 \\ 7 \\ 3 \\ 0 \\ 1 \end{pmatrix}$$

$$x^3 \cdot y + x^2 \cdot y^{-2} - 3x \cdot y^{-1} - 12y + 5 \text{ coeffs, } y, \text{ degree} \rightarrow \begin{pmatrix} x^2 & -2 \\ -3x & -1 \\ 5 & 0 \\ x^3 - 12 & 1 \end{pmatrix}$$

Рис. 5.23. Применение ключевых слов в символических вычислениях

$$\begin{array}{l}
 \text{expand} \quad \int_0^{\infty} e^{-z \cdot t} dt \rightarrow \frac{1}{z} - \frac{\lim_{t \rightarrow \infty} e^{-t \cdot z}}{z} \quad (x + y)^3 \text{ expand} \rightarrow x^3 + 3x^2 \cdot y + 3x \cdot y^2 + y^3 \\
 \cos(2 \cdot x) \text{ expand} \rightarrow \cos(x)^2 - \sin(x)^2 \\
 \text{factor} \quad \int_0^{\infty} e^{-z \cdot t} dt \text{ ass1}_{100} \text{ factor} \rightarrow 2^1 \cdot 2 \cdot 2^2 \quad x^3 - y^3 \text{ factor} \rightarrow (x - y) \cdot (x^2 + x \cdot y + y^2) \\
 \text{float} \quad \pi \text{ float} \rightarrow 3.1415926535897932385 \\
 \pi \text{ float, } 7 \rightarrow 3.141593 \quad 7 - \text{указано число знаков после запятой} \\
 \text{работает только в Mathcad 14 и 15} \\
 \text{series} \quad e^x \text{ series} \rightarrow 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} + \frac{x^5}{120} \quad e^x \text{ series, } 3 \rightarrow 1 + x + \frac{x^2}{2} \quad \text{здесь } 3 - \text{число членов ряда} \\
 e^x \text{ series, } x = 2 \rightarrow e^2 + e^2 \cdot (x - 2) + \frac{e^2 \cdot (x - 2)^2}{2} + \frac{e^2 \cdot (x - 2)^3}{6} + \frac{e^2 \cdot (x - 2)^4}{24} + \frac{e^2 \cdot (x - 2)^5}{120} \\
 \text{simplify} \quad (x + 3)^3 + 2x - 6 \text{ simplify} \rightarrow (x + 1) \cdot (x^2 + 8x + 21) \\
 \sin(x)^2 + \cos(x)^2 \text{ simplify} \rightarrow 1 \\
 \text{substitute} \quad 5x^2 + 3x - 6 \text{ substitute, } x = b - 2 \rightarrow 5 \cdot b^2 - 17 \cdot b + 8 \\
 a \cdot x^3 - b \cdot x + c \text{ substitute, } x = \frac{2}{z} \rightarrow \frac{c \cdot z^3 - 2 \cdot b \cdot z^2 + 8 \cdot a}{z^3}
 \end{array}$$

Рис. 5.24. Применение ключевых слов в символических вычислениях

введя два слова, `coeff`, `degree`, вывести два столбца: значения коэффициентов и показатель степени данного члена полинома (см. рис. 5.23).

При разложении в ряд Тейлора (команда `series`) необходимо указать координаты точки, в окрестности которой производится разложение, и число членов ряда. Если не указать это число, то Mathcad по умолчанию выводит результат с шестью членами ряда Тейлора (см. рис. 5.24).

5.3.3. Символьное решение уравнений

Для символьного решения уравнений и неравенств на символьной панели инструментов нужно выбрать команду `solve` (Решить). При этом на экране появляется символьный знак равенства с местом ввода, где надо указать переменную, относительно которой решается уравнение или неравенство (см. рис. 5.22, 5.25).

The image shows a collection of mathematical expressions demonstrating the use of the `solve` command in Mathcad. The expressions include:

- `solve sin(x) = cos(x) solve → π/4`
- `solve sin(x) = cos(x) solve, fully → π/4 + π·n if n ∈ ℤ`
- `solve sin(4·z) expand → 4·cos(z)3·sin(z) - 4·cos(z)·sin(z)3 if k ∈ ℤ`
- `solve (x+1)·(x-1) expand → x2 - 1`
- `solve (a-x)·(a2 - a·x + x2) collect, x → 2·a·x2 - x3 + -2·a2·x + a3`
- `solve a3 + 3·a2·b + 3·a·b2 + b3 factor → (a+b)3`
- `solve (r+s)/(s) + (r-s)/(r-s) simplify → r2/s2`
- `solve (a·z+1)/(z-b) - e-a = 0 solve, z → -b·e-a+1 / (a - e-a)`
- `solve (a·z+1)/(z-b) - e-a solve, b → z - ea·(a·z+1)`
- `solve ei·n·θ complex → cos(θ·n) + sin(θ·n)·i`
- `solve f(a) := (3a-7)·x = 1 solve, x, fully → 1/(3a-7) if a ≠ 7/3`
- `solve f(a) := (3a-7)·x = 1 solve, x, fully → undefined if a = 7/3`

Рис. 5.25. Символьное решение уравнения с использованием ключевого слова `solve`

Для символьного решения уравнений можно использовать также функцию `root` и вычислительный блок `Given–find` (рис. 5.26). При этом для вывода результата используется одноместный символьный знак равенства `→` (клавиши `Ctrl+`).

В Mathcad 13–15 функция `root` для символьного решения уравнения требует указать диапазон поиска корней (см. рис. 5.26).

5.3.4. Использование нескольких команд одновременно

Использование нескольких команд одновременно — довольно сложный процесс, так как надо знать, может ли быть преобразован промежуточный символьный результат. Чтобы убедиться в этом, его надо увидеть, то есть вывести каждый результат отдельно. Одновременное действие нескольких команд показано на рис. 5.27.

Символьные вычисления начинают с ввода исходного выражения, подлежащего преобразованию:

- Для выполнения других символьных операций после ввода выражения нажмите на символьной панели инструментов кнопку с названием необходимой операции. В появившемся на экране шаблоне заполните нужные места ввода и удалите ненужные.
- Для одновременного выполнения нескольких символьных операций выберите на панели двухместный знак символьного равенства. В появившемся шаблоне заполните нужные места ввода и удалите ненужные.
- Для вывода результатов символьных вычислений выведите курсор за пределы выражения и нажмите левую кнопку мыши.

ВНИМАНИЕ

При использовании двухместного символьного знака равенства после ввода каждого ключевого слова с названием символьной операции автоматически появляется свободное место для ввода следующей символьной операции. После ввода всех символьных операций надо удалить лишние места ввода и автоматически появляющиеся лишние скобки.

5.3.5. Символьные операции с матрицами

Одноместный символьный знак равенства позволяет выполнять различные операции с массивами. Кроме трех функций (транспонирования, обращения и вычисления определителя матрицы), находящихся на символьной панели инструментов, можно выполнять операции сложения, вычитания и перемножения матриц (рис. 5.28), а также использовать встроенные функции для работы с матрицами, например, для нахождения собственных чисел (*eigenvals*) и собственных векторов (*eigenvecs*) (рис. 5.29).

$$\begin{aligned}
 A &:= \begin{pmatrix} a & b \\ c & d \end{pmatrix} & B &:= \begin{pmatrix} k \\ m \end{pmatrix} & C &:= \begin{pmatrix} p \\ q \end{pmatrix} \\
 A^{-1} &\rightarrow \begin{bmatrix} \frac{d}{(a \cdot d - b \cdot c)} & \frac{-b}{(a \cdot d - b \cdot c)} \\ \frac{-c}{(a \cdot d - b \cdot c)} & \frac{a}{(a \cdot d - b \cdot c)} \end{bmatrix} \\
 A \cdot B &\rightarrow \begin{pmatrix} a \cdot k + b \cdot m \\ c \cdot k + d \cdot m \end{pmatrix} & B \cdot C^T &\rightarrow \begin{pmatrix} k \cdot p & k \cdot q \\ m \cdot p & m \cdot q \end{pmatrix} \\
 B^T \cdot C &\rightarrow k \cdot p + m \cdot q & B \cdot C &\rightarrow k \cdot p + m \cdot q
 \end{aligned}$$

Решение матричного уравнения $A \cdot X = C$

$$X := A^{-1} \cdot C \quad X \rightarrow \begin{bmatrix} \frac{d}{(a \cdot d - b \cdot c)} \cdot p - \frac{b}{(a \cdot d - b \cdot c)} \cdot q \\ \frac{-c}{(a \cdot d - b \cdot c)} \cdot p + \frac{a}{(a \cdot d - b \cdot c)} \cdot q \end{bmatrix}$$

Рис. 5.28. Символьные операции с массивами

$$\begin{aligned} \text{eigenvals}(A) &\rightarrow \left[\begin{array}{c} 9 + \frac{1}{2} \cdot d + \frac{1}{2} \cdot \left(324 - 36 \cdot d + d^2 + 4 \cdot b \cdot c \right)^{\frac{1}{2}} \\ 9 + \frac{1}{2} \cdot d - \frac{1}{2} \cdot \left(324 - 36 \cdot d + d^2 + 4 \cdot b \cdot c \right)^{\frac{1}{2}} \end{array} \right] \\ \text{eigenvecs}(A) &\rightarrow \blacksquare \quad \text{отключено, большое выражение} \end{aligned}$$

Рис. 5.29. Символьное определение собственных чисел и собственных векторов матрицы

Возможность получения символьного решения почти любого матричного выражения является большим достоинством Mathcad.

5.4. Новые возможности символьных вычислений в Mathcad 13–15

В Mathcad 14 применен совершенно новый символьный процессор MuPad. В результате теперь выполняется ряд новых операций и расширены возможности существовавших ранее.

5.4.1. Новые ключевые слова

На символьной панели появились новые ключевые слова:

- **Combine** (Комбинировать) позволяет упростить выражения в случае экспоненциальной или логарифмической функции. Обратите внимание на то, что ключевое слово **Expand** (Разложить) отменяет действие слова **Combine**. Наберите запятую после ключевого слова **Combine**, а за ней — модификатор, определяющий элементарную функцию, например **exp** или **ln** (рис. 5.30).

$$\begin{aligned} x^m \cdot x^n \text{ combine} &\rightarrow x^{m+n} \\ e^x \cdot e^{2y} \text{ combine, exp} &\rightarrow e^{x+2 \cdot y} \\ \ln(x) + \ln(2) + 3 \ln\left(\frac{3}{2}\right) \text{ combine, ln} &\rightarrow \ln\left(\frac{27 \cdot x}{4}\right) \\ x^m \cdot x^n \text{ combine} &\rightarrow x^{m+n} \\ x^{(m+n)} \text{ expand} &\rightarrow x^m \cdot x^n \end{aligned}$$

Рис. 5.30. Использование ключевого слова **Combine**

$$\begin{aligned} \sin(x) \text{ rewrite, exp} &\rightarrow -\frac{e^{x \cdot i}}{2} + \frac{e^{-x \cdot i}}{2} \\ \left(\frac{1}{2}\right) \cdot i \cdot e^{(-1) \cdot x} - e^{i \cdot x} \left[i \cdot \left(\frac{1}{2}\right)\right] \text{ rewrite, sincos} &\rightarrow \sin(x) \end{aligned}$$

Рис. 5.31. Использование ключевого слова **Rewrite**

- ❑ Rewrite (Переписать) позволяет переписать выражения с использованием некоторых элементарных функций, например `exp`, `sincos` (рис. 5.31).
- ❑ `Confrac` — разложение непрерывной дроби на фракции. По умолчанию Mathcad возвращает такое количество фракций, чтобы точность вычислений составляла 10 значащих цифр. Изменить точность вычислений можно, введя через запятую число выводимых фракций. Чтобы увидеть полные фракции, введите через запятую `Confrac, fraction` (рис. 5.32).
- ❑ `Explicit` (Явная подстановка) означает подстановку числовых значений и размерностей в формулу. Примеры использования `explicit` приведены на рис. 5.33.

$$\begin{aligned}
 \frac{2+\sqrt{3}}{7} \text{ confrac} &\rightarrow \begin{pmatrix} 0 \\ 1 \\ 1 \\ 7 \\ 24 \\ 8 \\ 24 \end{pmatrix} & \frac{2+\sqrt{3}}{7} \text{ confrac,4} &\rightarrow \begin{pmatrix} 0 \\ 1 \\ 1 \\ 7 \\ 24 \end{pmatrix} \\
 \frac{2+\sqrt{3}}{7} \text{ confrac, fraction} &\rightarrow \frac{1}{1 + \frac{1}{1 + \frac{1}{7 + \frac{1}{24 + \frac{1}{8 + \frac{1}{24}}}}}}
 \end{aligned}$$

$$\cos(\alpha) \text{ confrac} \rightarrow \begin{pmatrix} 1 & \alpha^2 \\ -2 & \alpha^2 \\ -6 & 0 \end{pmatrix} \quad \cos(\alpha) \text{ confrac, fraction} \rightarrow 1 + \frac{\alpha^2}{-2 + \frac{\alpha^2}{-6}}$$

Рис. 5.32. Использование ключевого слова `Confrac`

Записав математическое выражение, надо ввести ключевое слово `explicit`, выбрав его на математической панели. Далее через запятую нужно ввести имена параметров, чьи численные значения надо подставить, и знак символьного вычисления `→`. Чтобы получить окончательный результат вычислений, следует после символьного результата нажать клавишу `=` (равно) и, если нужно, ввести в поле ввода производную размерность.

ПРИМЕЧАНИЕ

В Mathcad 13 появилась возможность использовать в одной строке два знака равенства: символьный `→` и численный `=`. Попытка третий раз ввести знак присвоения значения результата не имеет (см. рис. 5.33). Приходится вновь записывать (копировать) имя выражения, чтобы ввести знак численного равенства.

$$M := 200 \quad b := 2 \quad h := 5$$

$$\sigma_{\text{изг}} := \frac{M}{\frac{b \cdot h^2}{6}} \text{ explicit, } M, b, h \rightarrow \frac{200}{\frac{2 \cdot 5^2}{6}} \quad \sigma_{\text{изг}} = 24 \quad \text{УДОБНО}$$

$$M_{\text{изг}} := \frac{M + F_0 \cdot L_0 + F_1 \cdot L_1}{\frac{b \cdot h^2}{6}} \rightarrow \frac{3 \cdot F_0 \cdot L_0}{25} + \frac{3 \cdot F_1 \cdot L_1}{25} + 24 \quad \begin{array}{l} \text{исходные данные} \\ \text{в векторной форме} \\ \text{НЕУДОБНО} \end{array}$$

$$M_{\text{изг}} := \frac{M + F_0 \cdot L_0 + F_1 \cdot L_1}{\frac{b \cdot h^2}{6}} \text{ explicit, } M, b, h, F, L \rightarrow \frac{200 + F_0 \cdot L_0 + F_1 \cdot L_1}{\frac{2 \cdot 5^2}{6}}$$

$$M1 := \frac{M}{W(b, h)} \rightarrow \frac{200}{W(2, 5)} \quad M2 := \frac{M}{W(b, h)} \text{ explicit, } M, b, h \rightarrow \frac{200}{W(2, 5)}$$

Рис. 5.33. Ключевое слово explicit в символьных вычислениях

В случае если символьные вычисления производятся с элементами матриц, результат получается несимпатичным, что называется, «не смотрится».

В сложных выражениях, содержащих внутри себя имена других функций, Mathcad так и переписывает имена этих функций, поставив в скобках численные значения их аргументов.

5.4.2. Новые слова-модификаторы

В Mathcad 14 введено несколько новых модификаторов, позволяющих изменить смысл ключевых слов:

- All применяет записанное далее условие ко всем переменным в выражении.
- Degree используется при вычислении коэффициентов полинома. Выводит вторую колонку чисел — степень того члена полинома, для которого выведен коэффициент.
- Domain используется с ключевым словом factor или parfrac, чтобы определить область чисел, в которой должно быть выполнено разложение на множители или на простые дроби.
- Exp используется со словом Combine или Rewrite, чтобы записать выражение с использованием экспоненты (рис. 5.34).
- Ln используется со словом Combine, чтобы представить выражение в виде логарифмов.
- Sincos используется со словом Rewrite, чтобы записать результат вычислений в терминах sin и cos.

$$\begin{aligned}
 & \sqrt{x^2} + \sqrt{y^2} + \sqrt{z^2} \text{ assume, ALL} > 0 \rightarrow x + y + z \\
 & p := 5 \quad q := 3 \quad r := -1 \\
 & \frac{p^2 + 3q}{5 + r} \text{ explicit, ALL} \rightarrow \frac{5^2 + 3 \cdot 3}{5 + -1} \\
 & 3z^3 + 5z^2 - 4 \text{ coeffs, degree} \rightarrow \begin{pmatrix} -4 & 0 \\ 0 & 1 \\ 5 & 2 \\ 3 & 3 \end{pmatrix} \\
 & x^3 - 2 \left| \begin{array}{l} \text{factor, domain = complex} \\ \text{float, 3} \end{array} \right. \rightarrow (x - 1.26) \cdot (x + 0.63 - 1.09i) \cdot (x + 0.63 + 1.09i) \\
 & a \cdot z = 1 \text{ solve, z, fully} \rightarrow \begin{cases} \frac{1}{a} & \text{if } a \neq 0 \\ \text{undefined} & \text{if } a = 0 \end{cases} \quad a \cdot z = 1 \text{ solve, z} \rightarrow \frac{1}{a} \\
 & \cos(z) = 0 \text{ solve, fully} \rightarrow \begin{cases} \frac{\pi}{2} + \pi \cdot _n & \text{if } _n \in \mathbb{Z} \\ \text{undefined} & \text{otherwise} \end{cases} \\
 & \cos(z) = 0 \text{ solve, fully} \rightarrow \begin{cases} \frac{\pi}{2} + \pi \cdot _n & \text{if } _n \in \mathbb{Z} \\ \text{undefined} & \text{otherwise} \end{cases} \quad \text{substitute, \%n = k} \rightarrow \begin{cases} \frac{\pi}{2} + \pi \cdot _n & \text{if } _n \in \mathbb{Z} \\ \text{undefined} & \text{otherwise} \end{cases}
 \end{aligned}$$

Рис. 5.34. Использование модификаторов All, degree, domain

- Fully возвращает полное подробное решение уравнения. Для уравнений, имеющих периодическое решение, использование слова fully после слова solve возвращает результат, содержащий произвольную постоянную %n, которую можно заменить, использовав подстановку со словом substitute или using. Mathcad использует символ %, чтобы избежать конфликта с другими переменными в данном Mathcad-документе (рис. 5.35).
- Using используется для замены произвольной постоянной %n при решении уравнения, имеющего периодическое решение.

$$\begin{aligned}
 & a \cdot z = 1 \text{ solve, z, fully} \rightarrow \begin{cases} \frac{1}{a} & \text{if } a \neq 0 \\ \text{undefined} & \text{if } a = 0 \end{cases} \quad a \cdot z = 1 \text{ solve, z} \rightarrow \frac{1}{a} \\
 & \cos(z) = 0 \text{ solve, fully} \rightarrow \begin{cases} \frac{\pi}{2} + \pi \cdot _n & \text{if } _n \in \mathbb{Z} \\ \text{undefined} & \text{otherwise} \end{cases} \\
 & \cos(z) = 0 \text{ solve, fully} \rightarrow \begin{cases} \frac{\pi}{2} + \pi \cdot _n & \text{if } _n \in \mathbb{Z} \\ \text{undefined} & \text{otherwise} \end{cases} \quad \text{substitute, \%n = k} \rightarrow \begin{cases} \frac{\pi}{2} + \pi \cdot _n & \text{if } _n \in \mathbb{Z} \\ \text{undefined} & \text{otherwise} \end{cases}
 \end{aligned}$$

Рис. 5.35. Использование модификаторов fully и using

- Raw фактически отменяет действие ранее записанного ключевого слова, возвращая исходное выражение.

5.4.3. Символьная векторизация выражений

В Mathcad 14 допускается символьное вычисление с использованием оператора векторизации (рис. 5.36). В более ранних версиях Mathcad этот оператор использовался лишь в процессе численных вычислений.

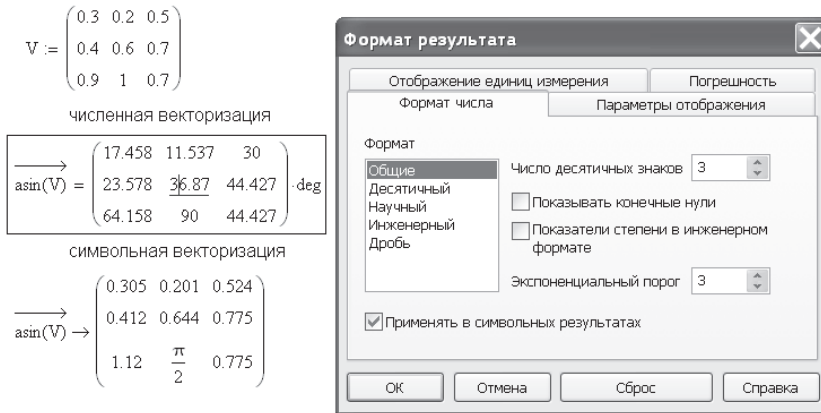


Рис. 5.36. Символьная векторизация выражений

5.4.4. Новые функции символьных вычислений

В Mathcad 14 появились три новые функции, используемые при символьных вычислениях.

- Numer, Denom возвращают числитель или знаменатель дроби (рис. 5.37).

$$\text{numer}\left(\frac{a + b + d}{x^2 + y^2}\right) \rightarrow a + b + d$$

$$\text{denom}\left(\frac{a + b + c}{x^2 + y^2}\right) \rightarrow x^2 + y^2$$

$$\text{IsPrime}(8) \rightarrow 0 \qquad \text{IsPrime}(7) \rightarrow 1$$

$$\text{IsPrime}(2.5) \rightarrow \text{Дробь}$$

$$\text{Integer expected}$$

Рис. 5.37. Использование новых функций Numer, Denom и IsPrime

- `IsPrime` возвращает 1, если целое число простое, и 0 — в противном случае. Попытка оценить дробное число или матрицу приводит к ошибке.

При использовании символьного знака равенства `Mathcad` учитывает все предыдущие присвоения значений константам и переменным и подставляет их в результат символьных вычислений. Если значение не задано, `Mathcad` окрашивает данный символ красным цветом, но символьное выражение вычисляет правильно. Поэтому лучше использовать величины, которым не присвоены численные значения. Если это нежелательно, надо использовать меню `Symbolics`, которое игнорирует все ранее присвоенные численные значения.

5.5. Встроенные функции и переменные

Символьный и численный процессоры используют несовпадающие списки встроенных функций. Эти списки частично пересекаются. Часть встроенных функций, имеющих общепринятое математическое значение, подобно `sin` или `log`, участвует в символьных преобразованиях. Другие функции, подобно `interp` или `rnd`, не имеющие общепризнанного математического значения, не преобразуются символьным процессором. В то же время символьный процессор использует ряд функций, не вычисляемых численным процессором. Список таких функций можно найти, обратившись за помощью к `Mathcad`.

Список встроенных функций, используемых в символьных вычислениях, включает в себя тригонометрические и гиперболические функции и обратные им, логарифмические и показательные функции, функции `Re` и `Im` вещественной и мнимой частей комплексного числа, функцию ошибок `erf`, гамма-функцию `Г`, функцию вычисления остатка от деления `mod`, ступенчатую функцию Хэвисайда `Ф`, функции `max` и `min`, матричные функции `identity` и `eigenvals`. В целом эти функции трактуются одинаково символьным и численным процессорами.

Встроенные константы, имеющие общепринятое математическое значение, трактуются одинаково символьным и численным процессорами. Символьный процессор распознает π , e и ∞ . Они сохраняют свой смысл в символьных вычислениях.

Встроенные константы, не имеющие общепринятого математического значения, не распознаются символьным процессором. Например, `TOL` или `ORIGIN` будут обрабатываться подобно любой неопределенной переменной.

ВНИМАНИЕ

Функции и переменные, задаваемые пользователем самостоятельно, распознаются символьным процессором при использовании символьного знака равенства и не распознаются при использовании меню `Symbolics`.

5.6. Особенности символьной обработки

При работе с символьным процессором необходимо помнить следующее:

- Многие вычисления могут быть выполнены только численно.
- Еще больше вычислений возвращают такие длинные ответы, что их неудобно использовать, поэтому удобнее выполнять эти вычисления численно.

Для компьютера символьные вычисления намного труднее, чем соответствующие численные операции. Так, большинство уравнений не могут быть решены символьно. Большинство интегралов, порой выглядящих просто, не имеют символьного решения, выраженного через элементарные функции.

Символьные вычисления могут производить настолько длинные ответы, что они не помещаются в рабочем окне. Если ответ состоит из слагаемых, его можно разделить «плюсами» на несколько строк (рис. 5.38). Для этого нужно:

- ❑ выделить весь ответ. Вновь установить указатель курсора на черном выражении и, нажав правую кнопку мыши, переместить ответ на свободное место. Появится копия ответа;
- ❑ уголком курсора выделить всю часть выражения перед знаком «плюс», после которого надо сделать перенос;
- ❑ нажать комбинацию клавиш Ctrl+Enter;
- ❑ удалить лишний знак «плюс»;
- ❑ повторить перенос строки столько раз, сколько строк вы хотите получить.

Иногда символьный ответ получается настолько длинным, что его не удастся удобно разместить в рабочем документе. На рис. 5.38 при решении уравнения

Перенос длинной строки

$$e^{x+y} \text{ series, } x=0, y=0,4 \rightarrow 1 + x + y + \frac{1}{2} \cdot x^2 + y \cdot x + \frac{1}{2} \cdot y^2 + \frac{1}{6} \cdot x^3 + \frac{1}{2} \cdot y \cdot x^2 + \frac{1}{2} \cdot y^2 \cdot x + \frac{1}{6} \cdot y^3$$

Скопировать ответ. Перед знаком плюс нажать Ctrl+Enter
Удалить лишнее место ввода

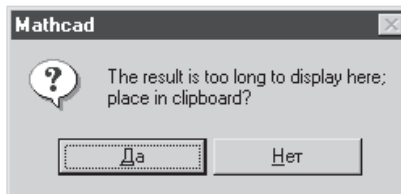
$$\left(1 + x + y + \frac{1}{2} \cdot x^2 + y \cdot x + \frac{1}{2} \cdot y^2 \right) \dots$$

$$+ \frac{1}{6} \cdot x^3 + \frac{1}{2} \cdot y \cdot x^2 + \frac{1}{2} \cdot y^2 \cdot x + \frac{1}{6} \cdot y^3$$

$$\left(\frac{1}{x} + \frac{1}{y} \right)^3 = a + 3 \cdot b \text{ solve, } x \rightarrow \text{Здесь нет ответа}$$

используем меню
Symbolic - Variable - Solve

$$\left(\frac{1}{k} + \frac{1}{y} \right)^3 = a + 3 \cdot b$$



MATRIX([[(1/2*y/(-1+a*y^3+3*b*y^3))*(4*a+12*b+4*a^2*y^3+24*a*y^3*b+36*b^2*y^3-4*((a+3*b)^2)^(1/2)+4*((a+3*b)^2)^(1/2)*a*y^3+12*((a+3*b)^2)^(1/2)*b*y^3)^(1/3)+2*y^2*(a+3*b)/(-1+a*y^3+3*b*y^3)]/(4*a+12*b+4*a^2*y^3+24*a*y^3*b+36*b^2*y^3-4*((a+3*b)^2)^(1/2)+4*((a+3*b)^2)^(1/2)*a*y^3+12*((a+3*b)^2)^(1/2)*b*y^3)-^(1/3)+1/(-1+a*y^3+3*b*y^3)]*y], это лишь часть ответа

Рис. 5.38. Обработка длинных ответов

с использованием символьного знака равенства процессор сообщил о длинном ответе, но не предложил его вывести. При использовании меню **Symbolics** ▶ **Variable** ▶ **Solve** (Символьные вычисления ▶ Переменная ▶ Решить) Mathcad спрашивает, не следует ли ответ поместить в буфер обмена. Если вы нажали кнопку **OK**, Mathcad поместит строку в буфер обмена. Выбрав команду **Paste** (Вставить), вы увидите на экране ответ на языке, похожем на Fortran. О том, как разобраться в таком ответе, рассказано в Resource Center (Центре документации) Mathcad.

Подводя итоги изучения символьных вычислений, следует отметить, что в Mathcad символьные вычисления имеют вспомогательное значение:

- ❑ В инженерных расчетах к символьным операциям приходится прибегать довольно редко.
- ❑ Часто вообще не существует символьного результата вычислений.
- ❑ Ряд символьных операций проще выполнить вручную, чем обращаться к Mathcad.
- ❑ В разделах справки (**Help**) и в Центре документации сведений о символьных операциях очень мало, мало также примеров символьных вычислений.
- ❑ Выполнение некоторых, даже очевидных, символьных операций вызывает непонятное затруднение у символьного процессора Mathcad.
- ❑ Mathcad — это инструмент расчетчика. Инструмент должен быть удобным. Если выполнение некоторых операций сложно, неудобно и сравнительно быстрое получение результатов расчета невозможно, то вряд ли есть смысл в этих случаях обращаться к Mathcad.

Профессиональные математики и ученые, которым часто приходится прибегать к символьным вычислениям, вряд ли обратятся за этим к Mathcad. Для них существуют более мощные пакеты, например Maple и Mathematica.

Глава 6

Решение дифференциальных уравнений

Дифференциальное уравнение называется обыкновенным (ОДУ), если в него входят производные только по одной переменной. В противном случае говорят об уравнении в частных производных. Mathcad предоставляет большие возможности для решения ОДУ и очень ограниченные — для решения уравнений в частных производных.

Поскольку решение дифференциальных уравнений состоит в интегрировании, чтобы обеспечить однозначность решения, необходимо задавать дополнительные условия для определения постоянных интегрирования.

Mathcad решает ОДУ двух типов:

- задачи Коши — ОДУ с начальными условиями, в которых задаются значения функции и ее производных в начальной точке интервала интегрирования;
- краевые задачи — ОДУ с граничными условиями, где задаются значения функции и ее производных в начале и в конце интервала интегрирования.

ВНИМАНИЕ

Mathcad в состоянии решить только ОДУ, которые можно записать в стандартном виде, то есть решить алгебраически относительно производной высшего порядка и записать в виде $y'(x) = f(x)$.

Для численного интегрирования одного ОДУ (равно как и системы ОДУ) можно использовать вычислительный блок Given–Odesolve (рис. 6.1) или применить встроенные функции, унаследованные от ранних версий Mathcad. Рассмотрим оба варианта решения ОДУ.

6.1. Вычислительный блок Given–Odesolve

Простота и удобство использования блока Given–Odesolve позволили описать его применение уже в главе 2. Напомним кратко, как использовать его для решения одного дифференциального уравнения.

Применение функции Odesolve требует записи вычислительного блока, состоящего из трех частей:

- ключевого слова Given (Дано);
- дифференциального уравнения и начальных или граничных условий к нему;
- функции Odesolve(x, xk, n) (решение ОДУ), где x — имя переменной, относительно которой решается уравнение; xk — конец интервала интегрирования (начало интервала интегрирования указано ранее, в начальных условиях); n — необязательный внутренний параметр, определяющий число интервалов, используемых для интерполяции функции решения уравнения.

Примеры использования функции Odesolve приведены в главе 2 на рис. 2.19–2.22 и далее на рис. 6.1–6.5.

6.1.1. Решение ОДУ с начальными и граничными условиями

Число условий в вычислительном блоке Given–Odesolve должно быть равно порядку ОДУ. Если все условия задаются в точке начала интервала интегрирования, такие условия называются начальными. Пример решения ОДУ с начальными условиями приведен на рис. 6.1.

Если часть условий задана в одной точке, а часть — в другой точке, то это граничные условия. Независимо от порядка ОДУ Mathcad принимает граничные условия только в двух точках. Пример решения ОДУ с граничными условиями приведен на рис. 6.2.

Одна точка обязательно должна быть началом интервала интегрирования, другая точка — любая. Однако решение оказывается достаточно точным, только если обе точки являются границами интервала интегрирования.

$$y'''(x) - 2^{-x} \cdot y'(x) = x \cdot \sin(x)$$

Given $\frac{d^3}{dx^3}y(x) - 2^{-x} \cdot \frac{d}{dx}y(x) = x \cdot \sin(x)$ исходное уравнение

$y(0) = 1$ $y'(0) = 1$ $y''(0) = -0.5$ начальные условия

$y := \text{Odesolve}(x, 2)$ $x := 0, 0.1..2$

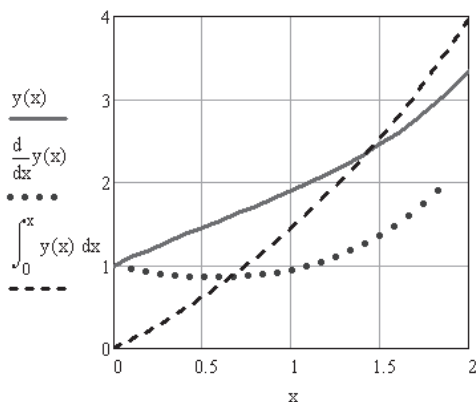


Рис. 6.1. Использование функции Odesolve

6.1.2. Объединение ОДУ с помощью операторов условия

Если вид дифференциального уравнения изменяется с изменением аргумента, например, уравнение содержит ступенчатую разрывную функцию, то такое уравнение можно записать с использованием условия, то есть записать несколько уравнений в виде одного уравнения (рис. 6.3).

Mathcad позволяет записать условие несколькими способами:

□ с использованием булевых операторов:

$$ay''(x) + by(x)x^2(x < 0) + y(x)(1-x)(x \geq 0) = c;$$

$$x(t)^2 \cdot \frac{d^2}{dt^2} x(t) + \left(\frac{d}{dt} x(t) \right)^3 = t \quad \begin{array}{l} \text{поменяйте местами} \\ \text{Выражения} \end{array}$$

Given $x(t)^2 \cdot x''(t) + x'(t)^3 = t$

$x(0) = 3 \quad x(6) = 2 \quad x := \text{Odesolve}(t, 9) \quad t := 0, 0.2.. 9$

граничные условия можно задавать лишь в двух точках
в начале и в конце интервала интегрирования

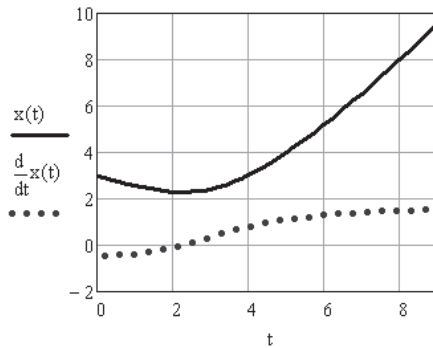


Рис. 6.2. Решение уравнения с граничными условиями

поменяйте местами
Выражения

$$y''(x) + y'(x) \cdot \text{if}[x < 1, x^2, (3 - x^{0.2})] = \text{if}(x < 2, \sin(x), 5) \quad \text{с функцией if}$$

Given $y''(x) + x^2 \cdot (x < 1) + y'(x) \cdot (3 - x^{0.2}) \cdot (x \geq 1) = \sin(x) \cdot (x < 2) + 5 \cdot (x \geq 2)$ с булевыми операторами

$y(0) = 1 \quad y(2) = -1 \quad y := \text{Odesolve}(x, 3) \quad x := 0, 0.1.. 3$

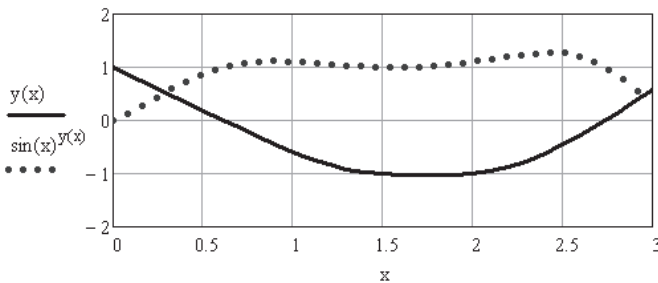


Рис. 6.3. Объединение двух дифференциальных уравнений в одно

□ с использованием функции if:

$$ay''(x) + \text{if}[x < 0, by(x)x^2, y(x)(1 - x)] = c.$$

В данном случае невозможно записать условие с использованием условного оператора программирования if, так как вертикальной черте (обязательному признаку программирования) должен предшествовать оператор присваивания :=, тогда как в нашем случае использован знак + или знак логического равенства =.

6.1.3. Решение ОДУ, заданных неявно

В Mathcad все встроенные функции для решения дифференциальных уравнений требуют записи производной высшего порядка в явном виде. Однако, как показано на рис. 6.2, присутствие коэффициента (постоянного или переменного) перед производной высшего порядка не помеха при решении ОДУ. Если уравнение есть сложная функция от высшей производной, предварительно надо решить это уравнение алгебраически относительно высшей производной. На рис. 6.4 исходное уравнение решено символично относительно $y'(x)$. Для выполнения этой операции надо выделить переменную, затем в главном меню выбрать команду **Symbolics** ▶ **Variable** ▶ **Solve** (Символьные вычисления ▶ Переменная ▶ Решить). Найденный результат подставлен в дифференциальное уравнение, которое решено с помощью функции **Odesolve**.

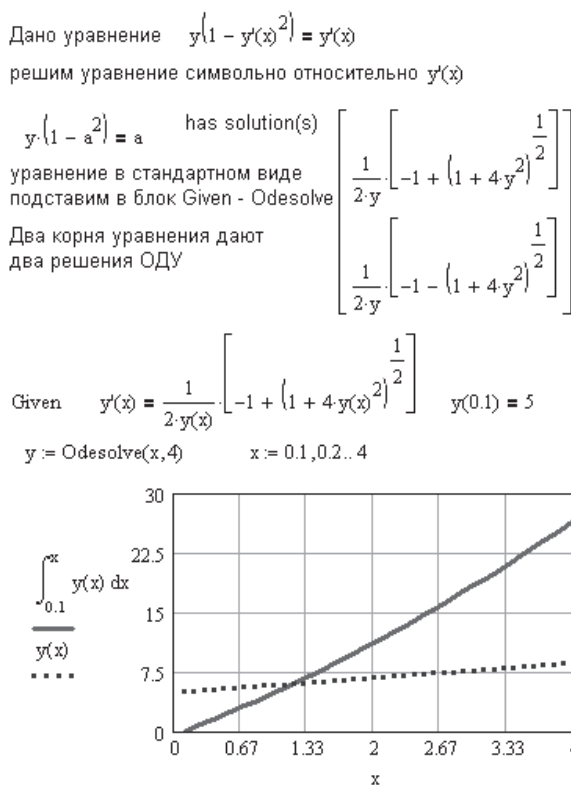


Рис. 6.4. Приведение дифференциального уравнения к стандартному виду и его решение

6.1.4. Решение систем ОДУ

Функция **Odesolve** может решать и системы дифференциальных уравнений. При этом несколько изменяется обращение к функции.

Обращение к функции для решения одного уравнения выглядит так:

$\text{Odesolve}(x, xk, n)$.

Для решения системы дифференциальных уравнений —
 $\text{Odesolve}(\text{вектор имен неизвестных}, x, xk, n)$.

На рис. 6.5 приведено решение системы из трех дифференциальных уравнений первого порядка. Соответственно, вектор имен неизвестных содержит имена трех неизвестных, x , y и z . Граничные условия заданы в точке $t = -1$. Конечная точка интервала интегрирования $t = 3$. Следовательно, интервал интегрирования — от -1 до 3 . Конечное значение всегда должно быть больше начального. Если график строится для значений, выходящих за пределы интервала интегрирования, Mathcad экстраполирует значения функции за пределы интервала интегрирования, получая чаще всего неправдоподобные результаты.

ВНИМАНИЕ

Если в Mathcad 11 функция Odesolve решает системы уравнений только с начальными параметрами, то начиная с Mathcad 12 — и с начальными, и с граничными условиями.

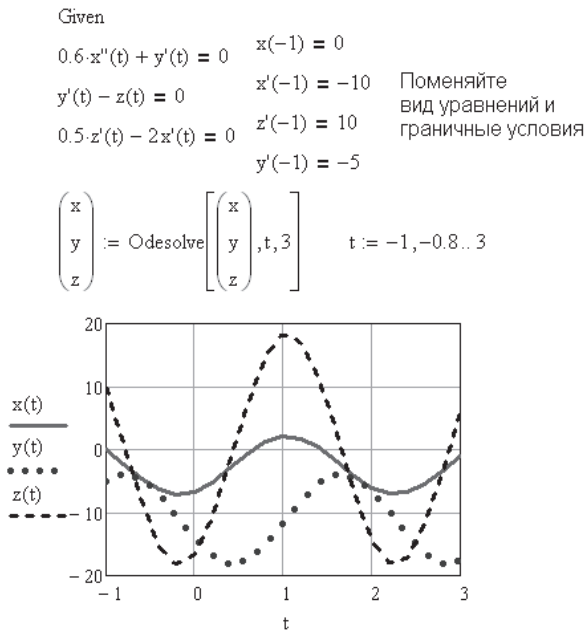


Рис. 6.5. Решение системы ОДУ функцией Odesolve

Функция Odesolve может принимать в качестве ограничений алгебраические уравнения (рис. 6.6 и 6.7), но только в задачах с начальными условиями.

6.1.5. Работа блока Given–Odesolve

Рассмотрим механизм работы блока Given–Odesolve . Фактически блок решения ОДУ Odesolve включает в себя все лучшее, что было создано для решения ОДУ в предыдущих версиях Mathcad. Блок Odesolve преобразует заданные дифференциальные уравнения к стандартному виду, используемому функциями rkfixed и

Rkadapt, применение которых описано далее. С помощью указанных функций он решает систему ОДУ (или одно ОДУ) с начальными условиями. В результате расчета получают массив значений функции и ее производные.

Given

$x'''(t) + y(t) = 0$	дифференциальные уравнения	алгебраическое уравнение
$y''(t) - z(t) = 0$		должно удовлетворять граничным условиям
$0.3 \cdot z(t) - x(t) = 0$	алгебраическое уравнение	

$x(-1) = 0$ $x'(-1) = -10$ $x''(-1) = 1$ требуется 6 начальных условий

$y(-1) = -10$ $y'(-1) = 0$ $z(-1) = 0$

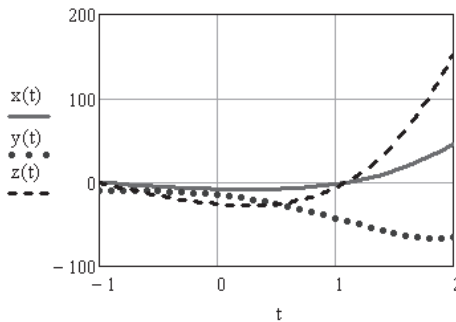
$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} := \text{Odesolve} \left[\begin{pmatrix} x \\ y \\ z \end{pmatrix}, t, 2 \right] \quad t := -1, -0.9..2$$


Рис. 6.6. Решение системы ОДУ с алгебраическими ограничениями функцией Odesolve

Далее используются функции интерполяции lspline и interp, преобразующие массив решений системы ОДУ в функцию, которую затем можно дифференцировать или интегрировать. Кстати, необязательный параметр steps в обращении к функции Odesolve как раз и задает число точек интерполяции. Чем больше steps, тем выше точность интерполяции, но тем больше время решения задачи.

Для решения системы жестких ОДУ используется функция Radau. Ее описание приводится далее в данной главе (раздел 6.3.1).

Если граничные условия заданы в двух точках, в начале решения функция Odesolve преобразует задачу с граничными условиями к задаче с начальными условиями, используя для этого функции sbval и bvalfit, описанные далее. Затем решается задача с начальными условиями с помощью функций rkfixed, Rkadapt или Radau в зависимости от выбранного пользователем метода решения — Fixed, Adaptive или Stiff.

Таким образом, функция Odesolve объединила все лучшие функции решения ОДУ, существующие в данный момент, в единый блок.

При решении системы ОДУ с помощью функции Odesolve есть возможность выбрать метод решения. Для этого надо установить курсор на слове Odesolve и нажать правую кнопку мыши. В открывшемся контекстном меню выбрать Fixed (Решение с фиксированным шагом функцией rkfixed), Adaptive (Решение с переменным шагом функцией Rkadapt) или Stiff (Решение жесткой системы ОДУ функцией Radau), как показано на рис. 6.7.

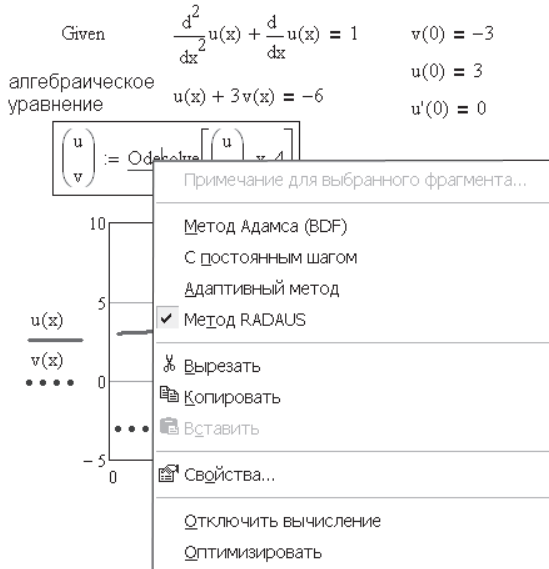


Рис. 6.7. Выбор метода решения дифференциальных уравнений

6.1.6. Новые функции решения ОДУ в Mathcad 14

В Mathcad 14 добавлены три новых функции решения ОДУ:

- Adams, которая базируется на методе Адамса и предназначена для решения нежестких систем.

Обращение к этой функции: $\text{Adams}(y, x_1, x_2, m, D, [\text{tol}])$.

- BDF, которая использует формулы обратного дифференцирования и предназначена для решения жестких систем.

Обращение к этой функции: $\text{BDF}(y, x_1, x_2, m, D, [J], [\text{tol}])$.

- AdamsBDF — гибридная функция, которая сама определяет, является система жесткой или нежесткой, и вызывает функции Adams или BDF соответственно.

Обращение к этой функции: $\text{AdamsBDF}(y, x_1, x_2, m, D, [J], [\text{tol}])$.

Аргументы всех трех функций те же, что у функции rkfixed (раздел 6.2): y — вектор начальных условий размерности n . Здесь n — порядок дифференциального уравнения или число уравнений в системе (если решается система дифференциальных уравнений); x_1 и x_2 — начало и конец интервала интегрирования, на котором ищется решение дифференциального уравнения. Начальные условия, заданные в векторе y , — это значения решения в точке x_1 ; m — число точек (не считая

начальной точки), в которых ищется приближенное решение. При помощи этого аргумента определяется число строк $(1 + m)$ в матрице, возвращаемой указанными функциями; [tol] — необязательный параметр, определяющий точность расчета каждой переменной. Вводится как число или вектор чисел. Можно использовать параметр [tol], чтобы изменить используемую по умолчанию точность 10^{-5} ; [J] — матрица Якоби (матрица частных производных системы ОДУ по всем переменным). Необязательный параметр. Используется только в функциях BDF или AdamsBDF. Ввод его может повысить точность решения.

Эти три новые функции включены в алгоритм функции Odesolve. В контекстном меню, появляющемся при наведении стрелки мыши на слово «Odesolve», предлагается выбор способа решения ОДУ (см. рис. 6.7):

- Adams (Метод Адамса);
- Fixed (С постоянным шагом);
- Adaptive (Адаптивный метод);
- Radau (Метод Radaus).

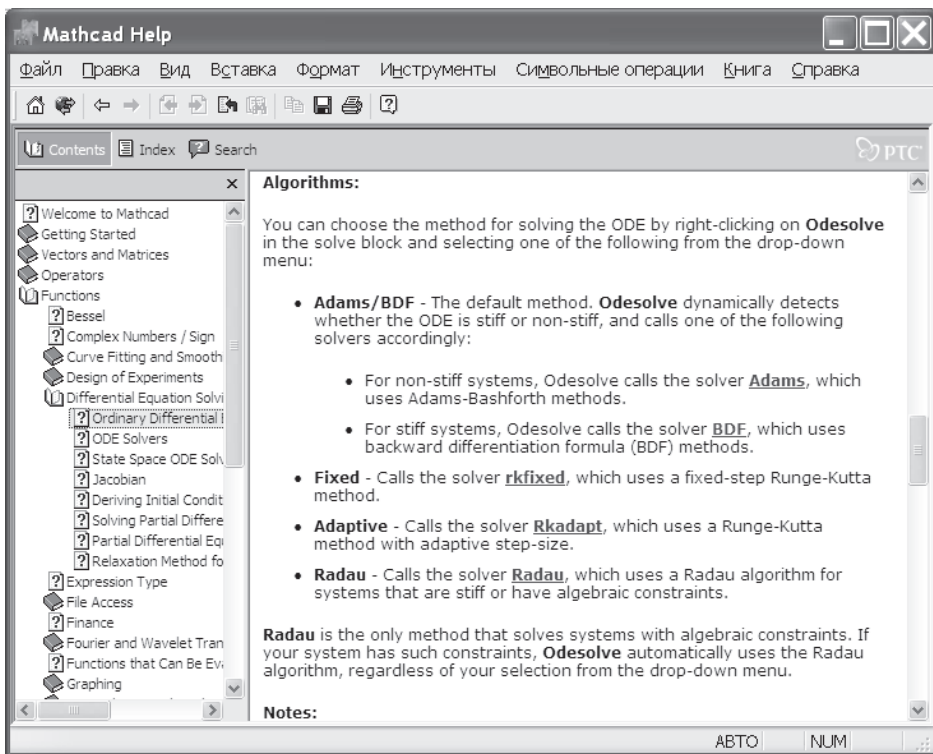


Рис. 6.8. Пояснения к выбору метода решения ОДУ в разделе Помощь Mathcad

Подробно о методах решения ОДУ функцией Odesolve читайте в разделе Помощь Mathcad (рис. 6.8).

Обратите внимание на то, что если Mathcad-документ был создан в более ранней версии, то в Mathcad 14 он будет открываться с использованием пункта Adams.

6.1.7. Решение ОДУ с переменными параметрами

В Mathcad 12 и 13 функция Odesolve может решать дифференциальные уравнения с переменными параметрами. Появилась возможность записать функцию пользователя с использованием встроенной функции Odesolve (рис. 6.9). Число параметров функции пользователя может быть любым. Для вывода результатов расчета эти параметры можно задать числом или в виде дискретной переменной. Результаты расчета можно вывести в виде таблицы, графика или семейства кривых.

```
Given  a·x''(t) + c·x'(t) = tb  x(t0) = 5  x(tk) = 0  D(a,b,c) := Odesolve(t,tk)
      b := 1  c := 1  F(a,t) := D(a,b,c,t)
      a := 0..3  b := 0..3
```

Семейства кривых при переменных значениях параметров

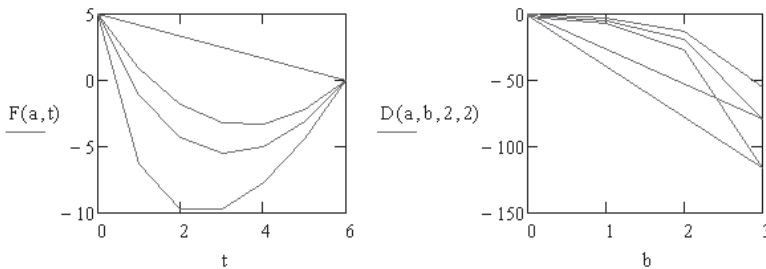
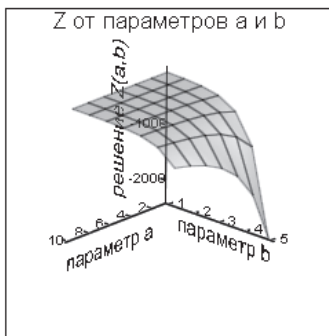


Рис. 6.9. Решение ОДУ с переменными параметрами

Вывод решения при двух переменных и двух постоянных параметрах (поменяйте параметры)

```
F(a,b) := D(a,b,2,4)
```



Отключен длительный расчет

Чтобы ускорить расчет на вкладке Quick Plot Data задайте малое число точек (# of Grids) например, по 6 точек

F

Рис. 6.10. Вывод решения ОДУ с двумя переменными параметрами

При двух переменных параметрах можно вывести график поверхности, но в этом случае возрастет время расчета (рис. 6.10).

В Mathcad 13 появилась возможность использовать функцию `Odesolve` в программных модулях. Примеры использования этой функции в программах приведены в главе 9.

6.2. Использование функций `rkfixed`, `Rkadapt`, `Bulstoer`

Рассмотрим три функции, предназначенные для решения дифференциальных уравнений и их систем: `rkfixed`, `Rkadapt` и `Bulstoer`.

Обращение к этим трем функциям одинаково:

$$\text{rkfixed}(y, x_1, x_2, m, D),$$

где y — вектор начальных условий размерности n . Здесь n — порядок дифференциального уравнения или число уравнений в системе (если решается система дифференциальных уравнений); x_1 и x_2 — начало и конец интервала интегрирования, на котором ищется решение дифференциального уравнения. Начальные условия, заданные в векторе y , — это значения решения в точке x_1 ; m — число точек (не считая начальной точки), в которых ищется приближенное решение. При помощи этого аргумента определяется число строк ($1 + m$) в матрице, возвращаемой указанными функциями.

Функции `rkfixed` и `Rkadapt` реализуют метод Рунге—Кутты четвертого порядка. Этот метод не является самым быстродействующим, но почти всегда приводит к искомому результату.

ВНИМАНИЕ

Все три функции решают ОДУ и их системы только с начальными, но не с граничными условиями, в отличие от функции `Odesolve`, которая решает дифференциальное уравнение любого порядка как с начальными, так и с граничными условиями.

Функция `rkfixed` ищет решение с постоянным шагом. Функция `Rkadapt` проверяет, как быстро изменяется приближенное решение, и, соответственно, адаптирует шаг решения. Адаптивный контроль величины шага дает возможность функции `Rkadapt` вычислять значения приближенного решения на более мелкой сетке в тех местах, где оно меняется быстро, и на более крупной — там, где оно меняется медленно. Это позволяет и повысить точность, и сократить время решения задачи.

Хотя функция `Rkadapt` при решении дифференциального уравнения использует во внутренних расчетах переменный шаг, возвращает она приближенное решение на равномерной сетке (в равноотстоящих точках).

Когда известно, что решение является гладкой функцией, лучше применить функцию `Bulstoer`, которая использует метод Булирша—Штера вместо метода

Рунге—Кутта, как в функциях rkfixed и Rkadapt. Найденное функцией Bulstoer решение будет немного более точным, чем при использовании других функций. Для решения дифференциального уравнения или системы уравнений необходимо задать:

- начальные условия;
- число точек, в которых надо найти решение;
- само дифференциальное уравнение, записанное в некотором специальном виде, который будет описан далее.

6.2.1. Решение дифференциального уравнения первого порядка

При интегрировании дифференциального уравнения первого порядка появляется одна произвольная постоянная, поэтому надо задать одно начальное условие y_0 , если ORIGIN = 0, или y_1 , если ORIGIN = 1 (Mathcad ожидает начальные условия в виде вектора).

Далее нужно ввести функцию двух переменных $D(x, y)$, которая содержит вектор первых производных от неизвестных функций. Для ОДУ первого порядка функция $D(x, y)$ содержит одну первую производную $y'(x)$.

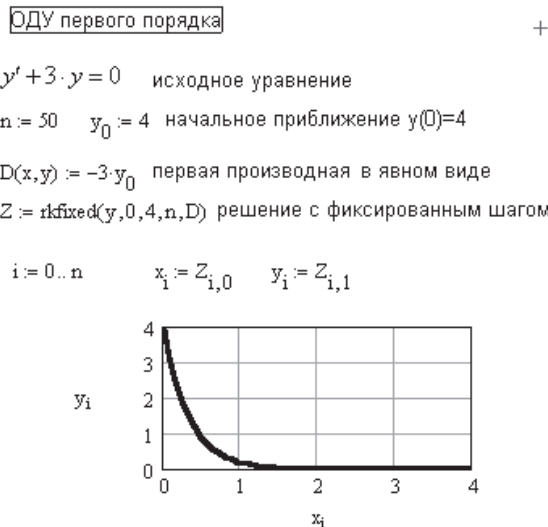


Рис. 6.11. Решение дифференциального уравнения первого порядка

В примерах на рис. 6.11 и 6.12 было просто решить уравнение относительно первой производной и определить функцию $D(x, y)$. В случае нелинейного дифференциального уравнения необходимо решить это уравнение алгебраически относительно высшей производной (в данном случае первой производной). Это можно сделать в символьном виде и подставить решение в функцию $D(x, y)$. Далее вызывается функция rkfixed (или Rkadapt, или Bulstoer), которая возвращает

решение уравнения. В результате решения получается матрица, содержащая два столбца:

- первый столбец содержит координаты x точек, в которых найдено решение;
 - второй столбец содержит значения найденного решения в указанных точках.
- Найденное решение можно вывести в виде таблиц или графиков.

```

y' + y^2 · x = 0  Нелинейное уравнение
y_0 := 1  начальное приближение y(0)=1
n := 50  D(x,y) := -(y_0)^2 · x
Z := Rkadapt(y,0,3,n,D)  решение с переменным шагом
x := Z<0>  y := Z<1>  i := 0..n

```

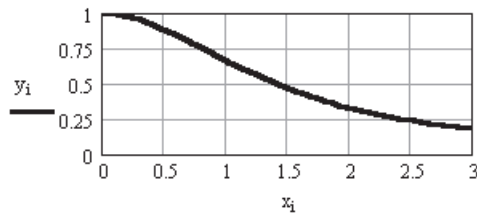


Рис. 6.12. Решение дифференциального уравнения с переменным шагом

ВНИМАНИЕ

Функции `rkfixed`, `Rkadapt`, `Bulstoer` возвращают матрицы, а не функции, поэтому интегрирование и дифференцирование найденного решения стандартными методами Mathcad невозможно. Необходимо использовать функции интерполяции для перехода от матрицы к функции.

6.2.2. Дифференциальные уравнения более высокого порядка

Для решения уравнений высокого порядка используются функции `rkfixed`, `Rkadapt`, `Bulstoer`. Для решения дифференциального уравнения порядка n нужно:

- задать вектор начальных условий, который состоит из n элементов, определяющих начальные условия для искомой функции и ее производных $y, y', y'', y''', \dots, y^{(n-1)}$;
- ввести функцию $D(x, y)$, которая теперь является вектором, содержащим n элементов:

$$D(x, y) = \begin{bmatrix} y'(x) \\ y''(x) \\ \dots \\ \dots \\ y^{(n)}(x) \end{bmatrix};$$

□ ввести выражение, содержащее функцию rkfixed, Rkadapt или Bulstoer.

В результате решения получается матрица, содержащая $n+1$ столбец: первый — для значений аргумента x , оставшиеся столбцы содержат искомую функцию y и ее производные $y(x), y'(x), y''(x), \dots, y^{(n-1)}(x)$.

Пример решения ОДУ четвертого порядка показан на рис. 6.13 и 6.14.

Исходное уравнение

$$\frac{d^4}{dx^4}y(x) - 2 \cdot x \cdot \frac{d^2}{dx^2}y(x) + \sin(x) \cdot y(x) = 0$$

$x1 := 10$ $x2 := 11$ $n := 20$ первые производные
в явном виде

граничные условия

$$y := \begin{pmatrix} 0 \\ 1 \\ 2 \\ 3 \end{pmatrix} \begin{matrix} y(0) = 0 \\ y'(0) = 1 \\ y''(0) = 2 \\ y'''(0) = 3 \end{matrix} \quad D(x, y) := \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ 2 \cdot x \cdot y_2 - \sin(x) \cdot y_0 \end{pmatrix} \begin{matrix} y_1 = y' \\ y_2 = y'' \\ y_3 = y''' \\ y_4 \end{matrix}$$

$Z := \text{Rkadapt}(y, x1, x2, n, D)$ $i := 0..n$

где y --вектор начальных условий
 $L1$ --начало интервала интегрирования
 $L2$ --конец интервала интегрирования
 n --число точек для вычислений
 D --вектор первых производных в явном виде

Поменяйте Rkadapt на rkfixed или Bulstoer

Рис. 6.13. Решение дифференциального уравнения четвертого порядка

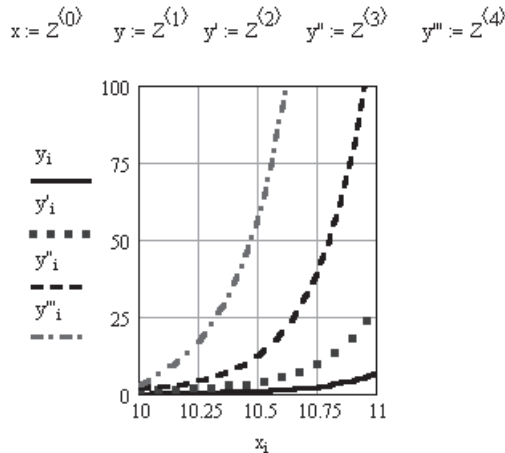


Рис. 6.14. Результаты решения уравнения, приведенного на рис 6.13

6.2.3. Системы дифференциальных уравнений

Методика решения систем дифференциальных уравнений такая же, как и методика решения одного дифференциального уравнения высокого порядка.

Любое дифференциальное уравнение порядка n

$$y^{(n)} = f(x, y^{(n-1)}, y^{(n-2)}, \dots, y', y)$$

путем замены

$$\begin{aligned} y_1(x) &= y(x); \\ y_2(x) &= y'(x); \\ &\dots \\ y_n(x) &= y^{(n-1)}(x) \end{aligned}$$

может быть представлено в виде системы дифференциальных уравнений:

$$\begin{aligned} y_1'(x) &= y_2(x); \\ y_2'(x) &= y_3(x); \\ &\dots \\ y_n'(x) &= f(x, y^{(n-1)}, y^{(n-2)}, \dots, y', y). \end{aligned}$$

Заметим, что в приведенных уравнениях содержатся производные только первого порядка. Преобразовав подобным образом каждое из уравнений, входящих в исходную систему, получим систему уравнений с большим количеством неизвестных функций, но с производными только первого порядка.

Исходная система уравнений

$$\frac{dx}{dt} = y + t \quad \frac{dy}{dt} = x + e^t \quad \text{при } x(0)=1 \text{ и } y(0)=0$$

Решение системы уравнений

$$\begin{aligned} x &:= \begin{pmatrix} 1 \\ 0 \end{pmatrix} & D(t, x) &:= \begin{pmatrix} x_1 + t \\ x_n + e^t \end{pmatrix} & n &:= 10 \\ t &:= F^{(0)} & x &:= F^{(1)} & y &:= F^{(2)} \end{aligned} \quad \begin{aligned} & F := \text{rkfixed}(x, 0, 1, n, D) \end{aligned}$$

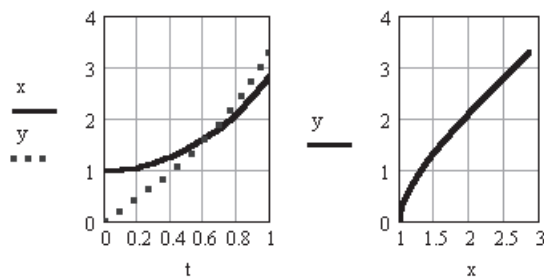


Рис. 6.15. Решение системы ОДУ первого порядка

В результате решения получается матрица, содержащая $n+1$ столбец: первый — для значений аргумента x в точках, где найдены решения, оставшиеся столбцы содержат искомые функции и их производные $y(x), y'(x), y''(x), \dots, y^{(n-1)}(x)$.

Примеры решения систем ОДУ первого и второго порядка показаны на рис. 6.15 и 6.16.

Фактически так и решалось дифференциальное уравнение высокого порядка (см. рис. 6.13), где функция $D(x, y)$ представляла собой систему уравнений первого порядка.

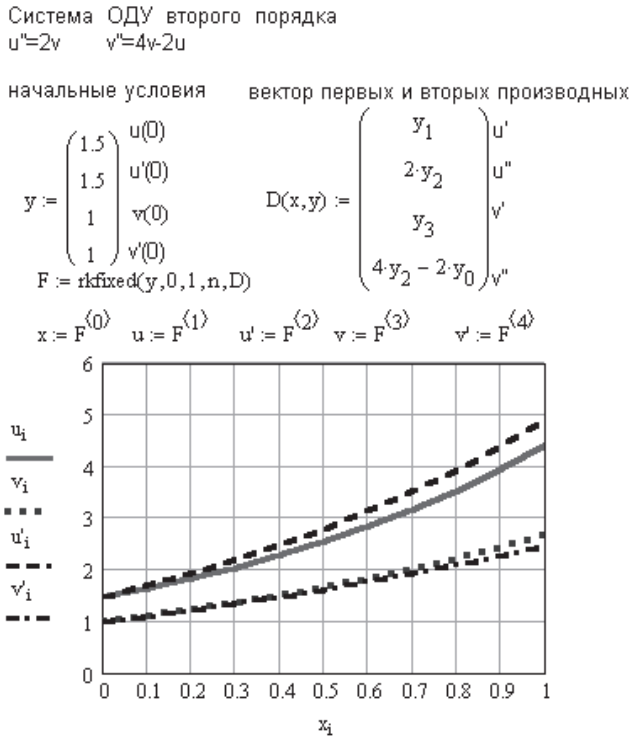


Рис. 6.16. Решение системы ОДУ второго порядка

Примеры для самостоятельного решения дифференциальных уравнений и их систем приведены на рис. 6.17.

6.3. Жесткие системы дифференциальных уравнений

Система дифференциальных уравнений, записанная в матричном виде, $Y = A \times X$, считается жесткой, если матрица коэффициентов почти вырожденная. В этом случае решение, возвращаемое функцией `rkfixed`, может быть неустойчивым. При решении жесткой системы необходимо использовать одну из трех функций, специально разработанных для решения жестких систем дифференциальных уравнений. К ранее существовавшим функциям `Stiffb` и `Stiffrr`, реализующим

Решить самостоятельно с помощью функций `rkfixed` или `Rkadapt`, с начальными условиями

$$y' = \frac{y}{\ln(y) \cdot \cos(x)} \quad y(0) = 0$$

$$y''' \cdot (x - 1) - y'' = 0 \quad y(2) = 2 \quad y'(2) = 1 \quad y''(2) = 1$$

$$y''' - 2 \cdot y'' + y' = 0 \quad \text{варьируйте граничные условия}$$

Системы ОДУ

$$\frac{d}{dt}x_1 = x_1 - x_2$$

$$\frac{d}{dt}x = \frac{x}{3 \cdot x + 2 \cdot y}$$

$$\frac{d}{dt}x_2 = x_1$$

$$\frac{d}{dt}y = \frac{y}{2 \cdot x + 3 \cdot y}$$

$$\frac{d}{dt}x_3 = x_1 - x_2$$

Рис. 6.17. Примеры для самостоятельной работы с ОДУ и их системами

методы Булирша—Штера или Розенброка соответственно, в Mathcad 2001i добавилась функция `Radau`, реализующая метод RADAU5. Рассмотрим вначале более ранние функции `Stiffb` и `Stiffr`.

Вид матрицы, возвращаемой этими функциями, совпадает с возвращаемым функцией `rkfixed`. Однако функции `Stiffr` и `Stiffb` требуют дополнительно задания якобиана системы уравнений.

Обращение к этим функциям:

$$\text{Stiffb}(y, x_1, x_2, n, D, J);$$

$$\text{Stiffr}(y, x_1, x_2, n, D, J).$$

Здесь y — вектор начальных условий размерности m , где m — порядок ОДУ или число уравнений в системе ОДУ; x_1 и x_2 — начало и конец интервала интегрирования, на котором ищется решение системы ОДУ; начальные условия, заданные вектором y , — это значение решения в точке x_1 ; n — число точек (не считая начальной), в которых ищется решение; $D(x, y)$ — m -мерный вектор, который содержит первые производные неизвестных функций; $J(x, y)$ — функция, которая возвращает матрицу размером $m \times (m + 1)$. Первый столбец ее содержит производные $F(t, y)$ по y . Остальные столбцы представляют собой матрицу Якоби системы ОДУ.

Пример решения жесткой системы ОДУ приведен на рис. 6.18.

Следует заметить, что увеличение числа шагов в 1000 раз позволяет решить приведенную систему с помощью функции `rkfixed` (рис. 6.19), но это оттого, что заданная система не слишком жесткая. Если увеличить разницу в коэффициентах уравнений, то функция `rkfixed` потребует гигантского числа шагов и в принципе не сможет решить систему.

Жесткие ОДУ

$$y_0 := \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad F(t, y) := \begin{pmatrix} -0.1 \cdot y_0 + 100 \cdot y_1 \cdot y_2 \\ 0.1 \cdot y_0 - 100 \cdot y_1 \cdot y_2 - 1000 \cdot y_1 \\ 1000 \cdot y_1 \end{pmatrix}$$

$$J(t, y) := \begin{pmatrix} 0 & -0.1 & 100 \cdot y_2 & 100 \cdot y_1 \\ 0 & 0.1 & -100 \cdot y_2 - 1000 & -100 \cdot y_1 \\ 0 & 0 & 1000 & 0 \end{pmatrix}$$

D := Stiffb(y0, 0, 50, 50, F, J)

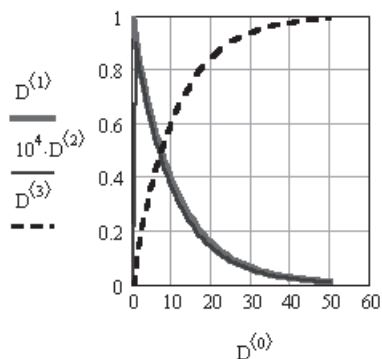


Рис. 6.18. Пример решения жесткой системы ОДУ

$$y_0 := \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad F(t, y) := \begin{pmatrix} -0.1 \cdot y_0 + 100 \cdot y_1 \cdot y_2 \\ 0.1 \cdot y_0 - 100 \cdot y_1 \cdot y_2 - 1000 \cdot y_1 \\ 1000 \cdot y_1 \end{pmatrix}$$

D := rkfixed(y0, 0, 50, 20000, F)

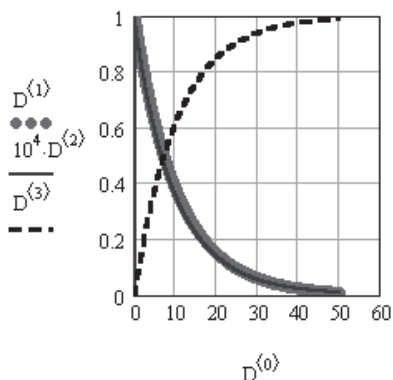


Рис. 6.19. Решение жесткой системы ОДУ функцией rkfixed

Новая функция Radau имеет такой же список аргументов, как и функции Rkadapt и Bulstoer, а именно: $\text{Radau}(y, x_1, x_2, D)$.

Функция Radau предназначена для решения систем жестких ОДУ, как и функции Stiffr и Stiffb. Преимуществом функции Radau перед Stiffr и Stiffb является то, что она не требует указания якобиана в качестве параметра функции.

Сравнение результатов расчета жесткой системы ОДУ с помощью функций Radau и Stiffr приведено на рис. 6.20. Численные результаты, естественно, совпадают, но использование функции Radau проще.

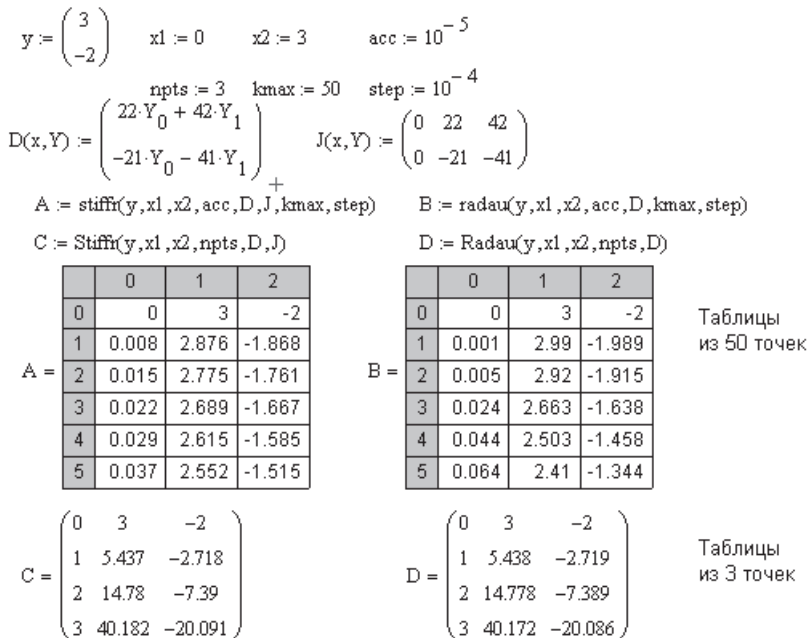


Рис. 6.20. Решение жесткой системы уравнений функциями Radau и Stiffr

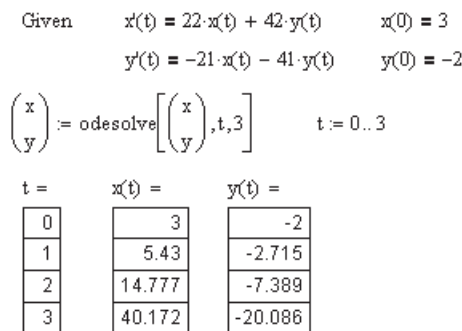


Рис. 6.21. Решение жесткой системы уравнений функцией Odesolve

На рис. 6.21 показано решение той же системы ОДУ с использованием функции `Odesolve` (в контекстном меню выбран способ решения `Stiff`).

6.3.1. Функция `Radau` в `Mathcad 14`

В `Mathcad 14` в обращении к функции `Radau` появилось три новых необязательных аргумента:

- J – матрица Якоби;
- M – матрица связи переменных в форме $M \left(\frac{dy}{dt} \right) = f(t, y)$;
- `[tol]` – число или вектор чисел, указывающих точность расчета по каждой переменной.

Обращение к функции `Radau`:

$$\text{Radau}(y, x_1, x_2, m, D, [J], [M], [tol]).$$

В приведенном на рис. 6.22 примере решается система из двух дифференциальных и одного алгебраического уравнения.

$$\begin{aligned} \frac{d}{dt}x &= -0.4 \cdot x + 10^4 \cdot y \cdot z \\ \frac{d}{dt}y &= 0.04 \cdot x - 10^4 \cdot y \cdot z - 3 \cdot 10^7 \cdot y^2 \\ x + y + z - 1 &= 0 \end{aligned} \quad \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \end{pmatrix}$$

Для справки

$$M \cdot \left(\frac{d}{dt}y \right) = f(t, y) \quad M \cdot \left(\frac{d}{dt}y \right) = \begin{pmatrix} \frac{d}{dt}y_0 \\ \frac{d}{dt}y_1 \\ 0 \end{pmatrix} \quad \begin{matrix} \text{Матрица связи} \\ M := \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{matrix}$$

Матрица Якоби

$$f(t, y) := \begin{bmatrix} -0.4 \cdot y_0 + 10^4 \cdot y_1 \cdot y_2 \\ 0.04 \cdot y_0 - 10^4 \cdot y_1 \cdot y_2 - 3 \cdot 10^7 \cdot (y_1)^2 \\ y_0 + y_1 + y_2 - 1 \end{bmatrix} \quad J(t, y) := \begin{pmatrix} -0.4 & 10^4 \cdot y_2 & 10^4 \cdot y_1 \\ 0.04 & -10^4 \cdot y_2 - 6 \cdot 10^7 \cdot y_1 & -10^4 \cdot y_1 \\ 1 & 1 & 1 \end{pmatrix}$$

Рис. 6.22. Подготовка системы ОДУ к решению

В матрице связи M ноль внизу на главной диагонали означает, что правая часть третьего уравнения равна нулю.

Для повышения точности расчетов в обращении к функции `Radau` включена матрица Якоби. Однако для данного примера якобиан можно удалить. Результаты расчета приведены на рис. 6.23.

Начальные условия Диапазон интегрирования

$$y_0 := 1 \quad y_1 := 0 \quad y_2 := 0 \quad t_H := 0 \quad t_K := 4 \cdot 10^4$$

$$Z := \text{Radau}(y, t_H, t_K, 1000, f, M)$$

$$t := Z^{(0)} \quad x := Z^{(1)} \quad y := Z^{(2)} \quad z := Z^{(3)}$$

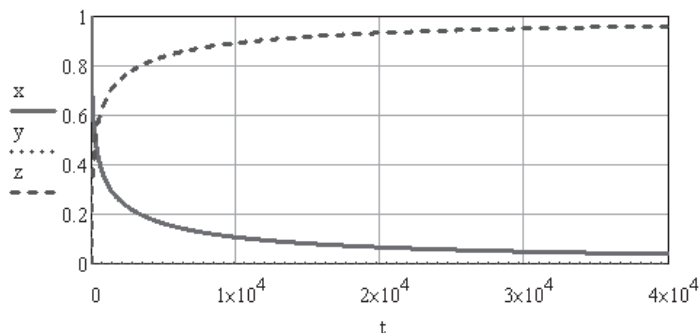


Рис. 6.23. Результат решения системы уравнений с помощью функции Radau

6.4. Нахождение решения только в конечной точке

Функции для численного решения дифференциальных уравнений, описанные ранее, искали решение $y(x)$ в равноотстоящих точках на отрезке от x_1 до x_2 . Однако часто возникает задача, когда необходимо найти решение только в конечной точке интервала $y(x_2)$. Функции, описанные ранее, будут давать такое решение, однако при этом делать много ненужной работы. Если нужно только значение функции в конце интервала $y(x_2)$, можно использовать функции, описанные в дальнейшем. Каждая из них соответствует одной из функций, перечисленных ранее, и обладает аналогичными свойствами. Имена таких функций начинаются со строчной буквы.

Обращение к этим функциям:

`bulstoer(y, x1, x2, acc, D, kmax, save);`

`rkadapt(y, x1, x2, acc, D, kmax, save);`

`radau(y, x1, x2, acc, D, kmax, save);`

`stiffb(y, x1, x2, acc, D, J, kmax, save);`

`stiffrr(y, x1, x2, acc, D, J, kmax, save),`

где y — вектор начальных условий размерности n (n — порядок ОДУ или число уравнений в системе ОДУ); x_1 и x_2 — граничные точки интервала, на котором ищется решение системы ОДУ; начальные условия, заданные вектором y , — это значение решения в точке x_1 ; `acc` — параметр, контролирующий качество решения.

Малое значение *acc* вынуждает алгоритм брать меньшие шаги вдоль траектории, что увеличивает точность решения, значения *acc*, близкие к 0,001, приводят к хорошей точности решения; $D(x, y)$ – n -мерный вектор, который содержит первые производные неизвестных функций; $J(x, y)$ – функция, которая возвращает матрицу размером $n \times (n + 1)$, первый столбец ее содержит производные $F(t, y)$ по y , остальные столбцы представляют собой матрицу Якоби системы ОДУ; k_{\max} – максимальное число промежуточных точек, в которых ищется решение, ограничивает число строк матрицы, возвращаемой этими функциями; *save* – минимально допустимый интервал между точками, в которых ищется приближенное решение. Он определяет минимальное различие между любыми двумя числами в первом столбце матрицы, возвращаемой функцией.

Следует отметить, что экономия времени при использовании таких функций невелика, и поэтому применение их вряд ли целесообразно.

Примеры использования таких функций приведены на рис. 6.20 и 6.24.

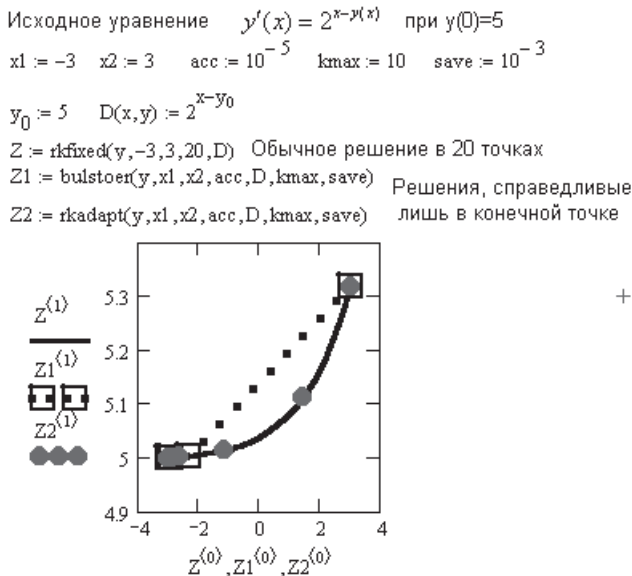


Рис. 6.24. Нахождение решения ОДУ только в конечной точке

При использовании этих функций точный результат получается только в конечной точке. Вывод промежуточных точек бесполезен (см. рис. 6.24).

6.5. Двухточечные краевые задачи

При решении дифференциального уравнения первого порядка задается одно граничное условие, и оно обязательно должно быть задано в начале интервала интегрирования. При решении одного дифференциального уравнения выше первого порядка число граничных условий должно быть равно порядку уравнения. Эти условия могут быть заданы в одной точке, в которой обязательно должно

быть начало интервала интегрирования, или в двух точках, одна из которых обязательно должна быть началом интервала интегрирования. Вторая точка может быть любой в Mathcad 13 и 14. В старых версиях — обязательно правее начала интервала.

Для решения дифференциальных уравнений любого порядка с граничными условиями лучше всего использовать функцию `Odesolve`. Однако можно использовать и все остальные рассмотренные функции, например `rkfixed`, `Rkadapt`, `Bulstoer` и др. Можно, но сложно. Гораздо сложнее, чем `Odesolve`.

Допустим следующее:

- ❑ дана система дифференциальных уравнений или одно дифференциальное уравнение порядка n ;
- ❑ в начальной точке x_1 интервала, на котором ищется решение, задана часть граничных условий;
- ❑ внутри или в конце интервала в точке x_2 заданы недостающие граничные условия;
- ❑ общее количество граничных условий равно n .

В этом случае необходимо использовать функцию `sbval`, чтобы найти недостающие граничные условия в точке x_1 . После этого можно будет решать обычную задачу с начальными условиями (задачу Коши). Для решения задачи Коши можно использовать любую из описанных ранее в этой главе функций.

ВНИМАНИЕ

Функция `sbval` не решает дифференциальное уравнение — она только вычисляет недостающие начальные условия.

Для того чтобы найденные функцией `sbval` недостающие начальные условия были согласованы с граничными условиями во второй точке интервала, используется алгоритм стрельбы. Суть метода стрельбы заключается в пробном задании недостающих начальных условий и решении затем задачи Коши известными методами.

Обращение к функции `sbval` выглядит следующим образом:

$$\text{sbval}(z, x_1, x_2, L, \text{load}, \text{score}),$$

где z — вектор начальных приближений для недостающих начальных условий в точке x_1 ; x_1 и x_2 — точки интервала, в которых заданы граничные условия; $D(x, y)$ — функция, содержащая в виде вектора из n элементов первые производные неизвестных функций; $\text{load}(x_1, y)$ — векторная функция, содержащая значения начальных условий в точке x_1 . Вектор состоит из n элементов, которые соответствуют вектору начальных условий в точке x_1 . Некоторые из этих условий будут константами, определяемыми начальными условиями. Другие элементы неизвестные и в дальнейшем будут определены функцией `sbval`; $\text{score}(x_2, y)$ — векторная функция, возвращающая вектор, количество элементов которого равно порядку уравнения n . Каждый элемент вектора содержит разность между начальным условием, заданным в точке x_2 , и значением искомого решения уравнения

в этой точке. Вектор `score` показывает, насколько значения найденного решения в точке x_2 близки к значениям, заданным в точке x_2 . Значение 0 для любого элемента указывает на полное совпадение начального условия и того значения, которое возвращается функцией `sbval`.

В качестве примера решения двухточечной краевой задачи на рис. 6.25 показано решение дифференциального уравнения упругой линии балки с двумя консолями (определение прогиба балки). И что же? Mathcad не справился с поставленной задачей. Все встроенные функции для решения дифференциальных уравнений, включая и функцию `Odesolve`, требуют обязательно задания одного из двух граничных условий на левом конце балки. Поэтому прогиб балки на левой консоли определить не удастся (рис. 6.26). Mathcad определяет прогиб балки во всех точках, начиная с левой опоры балки (точка A), $x = a$.

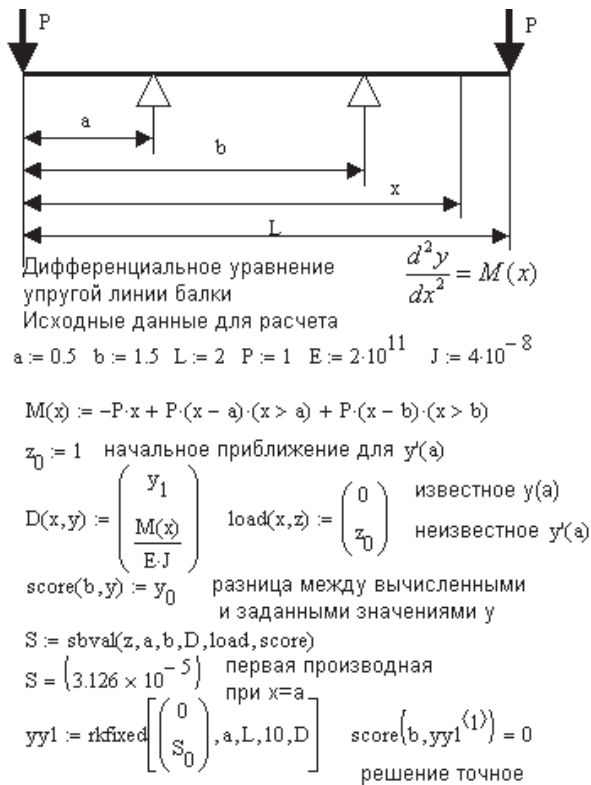


Рис. 6.25. Расчет прогиба балки функциями `sbval` и `rkfixed`

На балке три участка, поэтому уравнение изгибающего момента $M(x)$ записано с использованием булевых операторов (возможна запись $M(x)$ с использованием функции условия `if` или условного оператора `if` с панели программирования).

Для сравнения на рис. 6.26 показано решение той же задачи с помощью функции `Odesolve`.

Для сравнения то же решение с помощью функции `odesolve`

Given $E \cdot J \cdot y''(x) = M(x)$ $y(a) = 0$ $y(b) = 0$

$yy := \text{odesolve}(x, L)$ $x := a, a + \frac{L}{10} .. L$

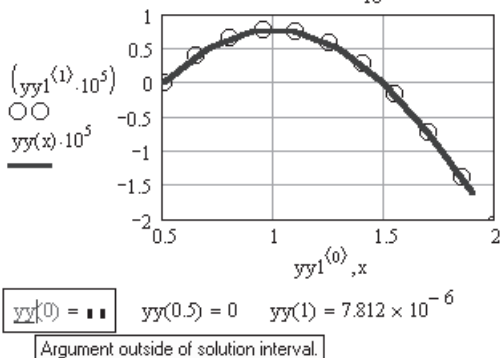


Рис. 6.26. Определение прогиба балки (см. рис. 6.25) функцией `Odesolve`

В некоторых случаях бывает, что кроме граничных условий в двух точках задано условие и в некоторой третьей точке внутри интервала интегрирования. В этом случае необходимо использовать функцию `bvalfit`, которая, как и `sbval`, вычисляет недостающие начальные условия для последующего решения задачи Коши.

Обращение к функции `bvalfit`:

`bvalfit(z1, z2, x1, x2, xf, D, load1, load2, score)`,

где z_1 — вектор, содержащий начальное приближение для начальных условий, не заданных в точке x_1 ; z_2 — то же самое для граничных условий, не заданных в точке x_2 ; x_1 и x_2 — точки интервала, где заданы граничные условия; x_f — некоторая точка между x_1 и x_2 . В этой точке решения, начинающиеся в точках x_1 и x_2 , должны быть равны; $D(x, y)$ — функция, содержащая в виде вектора первые производные неизвестных функций; `load1(x1, z1)` — векторная функция, возвращающая значения начальных условий в точке x_1 . Вектор состоит из n элементов, которые соответствуют вектору начальных условий в точке x_1 . Некоторые из этих условий будут константами, определяемыми начальными условиями. Другие элементы неизвестны и в дальнейшем будут определены функцией `bvalfit`; `load2(x2, z2)` — функция, аналогичная `load1`, но для значений в точке x_2 ; `score(xf, y)` — векторная функция, возвращающая вектор из n элементов. Он определяет, насколько значения решений, начинающихся из точек x_1 и x_2 , соответствуют друг другу в точке x_f . Если необходимо, чтобы решения совпали, определите `score(xf, y) = y`.

Этот метод нужно использовать, когда производные имеют разрыв где-либо на интервале поиска решения.

Пример использования функции `bvalfit` приведен на рис. 6.27, результаты расчета — на рис. 6.28. Для дифференциального уравнения второго порядка с разрывной правой частью d в точке $x_f = 0$ с помощью функции `bvalfit` найдены недостающие граничные условия в точках x_1 и x_2 . Далее с помощью функции `rkfixed` решено уравнение для $x < 0$.

Решение этого уравнения в точке x_f использовано в качестве начальных условий для решения этого же уравнения при $x > 0$.

```

Дифференциальное уравнение с разрывной правой частью
краевые условия
y'' = | y | при x < 0
      | -y | при x > 0   y(1) = 2   y(-1) = 1

D(x,y) := [ y_1
            y_0*(x < 0) - y_0*(x >= 0) ]   xf := 0 точка стыковки двух
                                                кусков решения

z1_0 := 1   стартовые значения y'(1)
           для поиска величин
z2_0 := 1   y'(-1)

load1(x1,z1) := ( 1 ) y(-1)
                ( z1_0 ) стартовое значение y'(-1)

load2(x2,z2) := ( 2 ) y(1)
                ( z2_0 ) стартовое значение y'(1)

score(xf,y) := y   условие, приводящее к стыковке
                решений в точке x=xf

S := bvalfit(z1,z2,-1,1,0,D,load1,load2,score)
S = (0.092 -0.678)   содержит y(-1) и y(1)
    
```

Рис. 6.27. Краевая задача с граничным условием в промежуточной точке

Решаем 2 дифференциальных уравнения с известными начальными условиями

```

Y1 := rkfixed( ( 1 )
              ( S0,0 ), -1,0,10,D)   Y2 := rkfixed( ( Y110,1 )
              ( Y110,2 ), 0,1,10,D)
    
```

Для сравнения то же решение с помощью функции *odesolve*

```

y''(x) = if(x < 0, y(x), -y(x))   Поменяйте местами
Given   y''(x) = y(x) * (x < 0) - y(x) * (x >= 0)   эти 2 уравнения
y(-1) = 1   y(1) = 2   y := Odesolve(x,1)   x := -1,-0.9..1
Решение уравнения с использованием функций
odesolve                               bvalfit и rkfixed
    
```

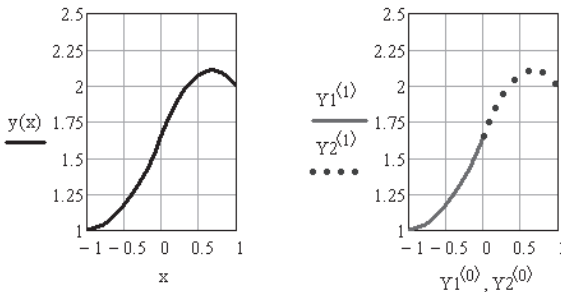


Рис. 6.28. Решение уравнения (см. рис. 6.27) функцией *Odesolve*

Теперь посмотрите, как это же уравнение решает функция `Odesolve` с использованием булевых операторов для записи правой части уравнения или функции условия `if` (см. рис 6.28). Условный оператор программирования `if` в данном случае применить нельзя, так как перед вертикальной чертой, указывающей на ввод программы, должен стоять оператор присвоения значения `:=`, а не логический знак равенства `=`.

На рис. 6.28 также выведены графики решения дифференциального уравнения с помощью функции `Odesolve` и функций `bvalfit` и `rkfixed`. Оба графика полностью совпадают. Функция `Odesolve` возвратила функцию, а функция `rkfixed` — массив.

6.6. Решение дифференциальных уравнений в частных производных

Дифференциальные уравнения в частных производных требуют нахождения функции не одной, а нескольких переменных. `Mathcad` имеет очень ограниченные возможности для решения таких уравнений, ведь для решения каждого вида уравнений в частных производных требуется свой метод решения.

Уравнения в частных производных можно разделить на три типа:

- ❑ параболические, содержащие первую производную по одной переменной и вторую — по другой, причем все производные входят в уравнение с одинаковым знаком;
- ❑ гиперболические, содержащие первую производную по одной переменной и вторую — по другой, входящие в уравнение с разными знаками;
- ❑ эллиптические, содержащие только вторые производные, причем одного знака.

6.6.1. Функции решения параболических и гиперболических уравнений

`Mathcad` включает в себя две функции для решения параболических и гиперболических уравнений, `pdesolve` и `numol`.

Функция `pdesolve` используется в составе вычислительного блока `Given–pdesolve` для решения гиперболических или параболических уравнений (или систем уравнений) в частных производных, имеющих в качестве аргументов, как правило, время t и пространственную координату x .

Обращение к этой функции:

$$\text{pdesolve}(u, x, x\text{range}, t, t\text{range}, [x\text{pts}], [t\text{pts}])$$

возвращает скалярную (для одного уравнения) или векторную (для системы уравнений) функцию, являющуюся решением уравнения (или системы уравнений). Здесь u — явно заданный вектор имен функций (без указания имен аргументов), подлежащих вычислению. Эти функции, а также граничные условия должны быть определены внутри вычислительного блока `Given–pdesolve`; x — пространственная координата; $x\text{range}$ — вектор значений аргумента x для граничных условий. Он должен состоять из двух чисел, представляющих две границы

расчетного интервала; t — время (имя второго аргумента неизвестной функции); $trange$ — вектор значений аргумента t для граничных условий, состоящий из двух чисел, представляющих две границы расчетного интервала; $xpts$ — количество пространственных точек дискретизации (может не указываться); $tpts$ — количество временных слоев (также может не указываться).

Пример использования функции `pdesolve` приведен на рис. 6.29. Запись вычислительного блока с функцией `pdesolve` аналогична записи блока с функцией `Odesolve`. Результаты расчета показаны на рис. 6.30.

Решение одномерного волнового уравнения

$$\frac{d^2}{dt^2}w(x,t) = a^2 \cdot \frac{d^2}{dx^2}w(x,t)$$

Здесь w - перемещение, v - скорость перемещения

где $\frac{\partial}{\partial t}w(x,t) = v(x,t)$ тогда $\frac{\partial}{\partial t}v(x,t) = a^2 \cdot \frac{d^2}{dx^2}w(x,t)$

Представим первое уравнение как систему двух уравнений первого порядка

Given $v_t(x,t) = a^2 \cdot w_{xx}(x,t)$ $w_t(x,t) = v(x,t)$

граничные условия

$$w(x,0) = \sin\left(\frac{\pi \cdot x}{L}\right) \quad v(x,0) = 0$$

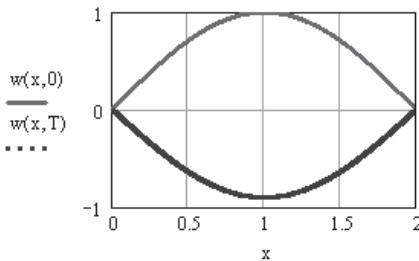
$$w(0,t) = 0 \quad w(L,t) = 0$$

$$\begin{pmatrix} w \\ v \end{pmatrix} := \text{Pdesolve} \left[\begin{pmatrix} w \\ v \end{pmatrix}, x, \begin{pmatrix} 0 \\ L \end{pmatrix}, t, \begin{pmatrix} 0 \\ T \end{pmatrix} \right]$$

$a \equiv 1$ Глобальное присвоение параметров
 $L \equiv 2$ Глобальное присвоение параметров
 $T \equiv 2 \cdot \pi$ Глобальное присвоение параметров

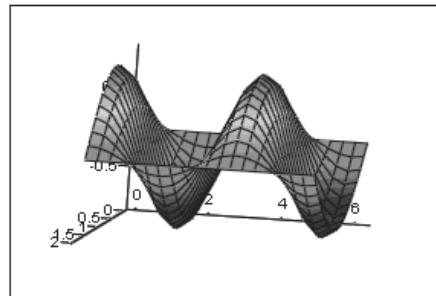
Рис. 6.29. Пример использования функции `pdesolve`

Единичное решение волнового уравнения



Сетка решений волнового уравнения на временном и пространственном интервалах

`M := CreateMesh(w,0,L,0,T)`



M

Рис. 6.30. Результаты решения волнового уравнения

Обратите внимание на то, что уравнения внутри вычислительного блока должны записываться с аргументами. Для идентификации частных производных следует использовать нижний индекс, например, $u_{xx}(x, t)$ — вторая производная функции u по пространственной координате x .

Недостатком функции `pdesolve` (как и функции `Odesolve`) является невозможность ее использования в составе выражения — программы для многократного решения дифференциального уравнения. При необходимости многократного решения обыкновенных дифференциальных уравнений в состав программного модуля можно включать старые функции `Rkadapt` и `Bulstoer`

При необходимости многократного решения дифференциальных уравнений в частных производных в состав программного модуля можно включать функцию `numol`, которая, как и `pdesolve`, появилась в Mathcad 11.

Функция `numol` предназначена для решения тех же уравнений, что и функция `pdesolve`.

Обращение к этой функции:

`numol(xrange, xpts, trange, tpts, Npde, Nae, rhs, init, bc)`

возвращает матрицу решения дифференциального уравнения в частных производных в каждой точке по пространственной (по строкам) и временной (по столбцам) координате. Если решается не одно уравнение, а система уравнений, то результатом решения является составная матрица, образованная путем слияния (слева направо) со значениями каждой искомой сеточной функции. Здесь `xrange` — вектор значений аргумента x для граничных условий. Он должен состоять из двух чисел, представляющих две границы расчетного интервала; `trange` — вектор значений аргумента t для граничных условий, состоящий из двух чисел, представляющих две границы расчетного интервала; `xpts` — количество пространственных точек дискретизации (может не указываться); `tpts` — количество временных слоев (также может не указываться); `Npde` — количество дифференциальных уравнений в частных производных в системе; `Nae` — количество дополнительных алгебраических уравнений, входящих в систему; `rhs` — вектор правых частей уравнений; `init` — векторная функция, определяющая начальные условия для каждой неизвестной функции; `bc` — функциональная матрица граничных условий.

Вектор граничных условий может иметь значения трех типов:

- ❑ `rhs` содержит вторые пространственные производные: граничные условия (или Дирихле «D», или Неймана «N») требуются по одному с каждой стороны интервала интегрирования;
- ❑ `rhs` содержит первые пространственные производные: граничные условия Дирихле на левой или правой границе интервала, на другой стороне NA;
- ❑ нет пространственных производных — граничные условия не требуются.

Функциональная матрица `bc` содержит три столбца, имеющие следующий вид:

- ❑ `(init_left(t) init_right(t) «D»)` — для граничных условий Дирихле;
- ❑ `(init_left(t) init_right(t) «N»)` — для граничных условий Неймана.

Пользоваться функцией `numol` намного сложнее, чем функцией `pdesolve` (рисунки 6.31 и 6.32).

исходное уравнение то же, что и при решении функцией **pdsolve** $\frac{d^2}{dt^2} w(x,t) = a^2 \cdot \frac{d^2}{dx^2} w(x,t)$

заменено системой уравнений $\frac{\partial}{\partial t} w(x,t) = v(x,t)$ $\frac{\partial}{\partial t} v(x,t) = a^2 \cdot \frac{d^2}{dx^2} w(x,t)$

$Npde := 2$ Дифференциальных уравнений в частных производных 2 $Nae := 0$ алгебраических уравнений нет

вектор правых частей уравнений $rhs(x,t,u,u_x,u_{xx}) := \begin{pmatrix} u_1 \\ a^2 \cdot u_{xx} \end{pmatrix}$
 здесь, $u_1 = v$ и $u_0 = w$.

Вектор начальных условий $init(x) := \begin{pmatrix} \sin\left(\frac{\pi \cdot x}{L}\right) \\ 0 \end{pmatrix}$

Граничные условия в начале и в конце выбранных интервалов x и t $bc_func(t) := \begin{pmatrix} init(0)_0 & init(L)_0 & "D" \\ "NA" & "NA" & "D" \end{pmatrix}$

Решение уравнения:

$nx := 30$ $nt := 20$

В этом примере 20 точек по времени на каждую функцию. Итого получаем матрицу из 40 колонок.

$sol := \text{numol}\left[\begin{pmatrix} 0 \\ L \end{pmatrix}, nx, \begin{pmatrix} 0 \\ T \end{pmatrix}, nt, Npde, Nae, rhs, init, bc_func\right]$

Рис. 6.31. Решение волнового уравнения с помощью функции numol

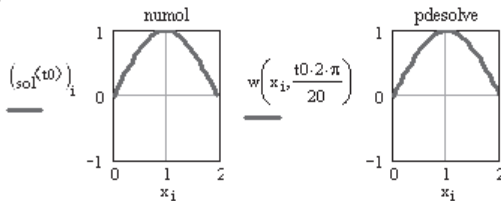
$\text{rows}(sol) = 30$ $\text{cols}(sol) = 40$

Матрица sol содержит два решения: первое для $u_0=w$ и второе для $u_1=v$ (посмотрите их. Для этого на трехмерном графике замените SOL на sol)

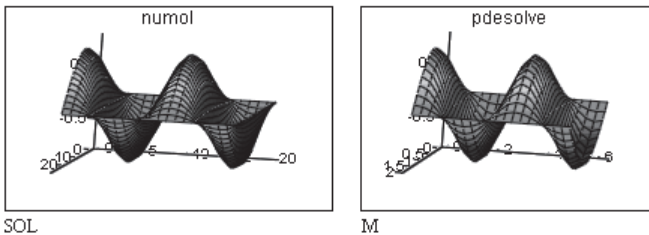
Исключим второе решение u_1 $SOL := \text{submatrix}(sol, 0, nx - 1, 0, nt)$

$i := 0..30$ $L = 2$ $x_i := \frac{i \cdot L}{\%n}$ $t0 := 0$

Сравним решение уравнения функциями numol и pdsolve при $t0=0$ - поменяйте значение t0.



Сравним сетку решений в пространстве и времени



SOL

M

Рис. 6.32. Сравнение решения волнового уравнения функциями numol и pdsolve

ПРИМЕЧАНИЕ

Граничное условие называется условием Дирихле, если задано значение функции, или Неймана, если задана первая производная функции.

Процесс изменения решения во времени удобно наблюдать с использованием анимации результатов расчета (рис. 6.33). Подробно об анимации графиков рассказано в главе 10.



Рис. 6.33. Анимация решения волнового уравнения

6.6.2. Решение эллиптических уравнений (Лапласа и Пуассона)

С помощью двух встроенных функций, `multigrid` и `relax`, можно решить простейшие случаи уравнения Пуассона. Это уравнение очень часто используется в технике, например, для описания полей напряжений и деформаций в плоской задаче теории упругости, в задачах теплопроводности, гидроаэродинамики, электростатики.

В Mathcad для численного решения уравнения Пуассона используется метод конечных разностей.

Уравнение Пуассона имеет вид

$$\frac{\partial^2 F(x, y)}{\partial x^2} + \frac{\partial^2 F(x, y)}{\partial y^2} = -\rho(x, y).$$

Если правая часть уравнения равна нулю, такое уравнение называется уравнением Лапласа

$$\frac{\partial^2 F(x, y)}{\partial x^2} + \frac{\partial^2 F(x, y)}{\partial y^2} = 0.$$

На квадратной области уравнение Пуассона представляется в виде

$$au_{i+1, j} + bu_{i-1, j} + cu_{i, j+1} + du_{i, j-1} + eu_{i, j} = p_{i, j}.$$

Численное решение ищется в Mathcad только на квадратной области, состоящей из $(n + 1) \times (n + 1)$ точек. Поэтому граничные условия должны быть определены пользователем для всех четырех сторон квадрата. Самый простой (и наиболее

часто используемый) вариант — это нулевые граничные условия (уравнение Лапласа). В таком случае можно использовать функцию `multigrid`.

Обращение к функции:

$$\text{multigrid}(M, N\text{cycle}),$$

где M — квадратная матрица размером $1 \times 2n$, которая содержит значения правой части уравнения Пуассона в соответствующей точке квадратной области; $N\text{cycle}$ — число циклов на каждом уровне итерации функции `multigrid`. Значение $N\text{cycle} = 2$ обычно дает хорошую аппроксимацию решения.

Пример использования функции `multigrid` приведен на рис. 6.34.

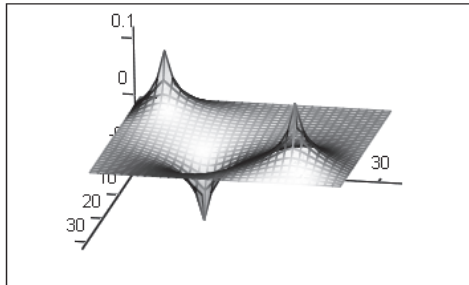
`M := 0` обнуление предыдущих значений M

`n := 32` `Mn,n := 0` Обнуление матрицы правых частей уравнения

`i := 1..n` `j := 1..n` `Mi,j := -2` Включите выражения в рамке

`M4,4 := -200` `M20,12 := 200` `M25,25 := -200` Три точечных источника

`F := multigrid(M, 2)` Значения правой части уравнения Пуассона постоянны во всех точках



F

Рис. 6.34. Решение уравнения Лапласа с помощью функции `multigrid`

Если граничные условия по сторонам квадрата ненулевые, необходимо использовать функцию `relax`.

Обращение к функции:

$$\text{relax}(a, b, c, d, e, f, u, r\text{jac}),$$

где a, b, c, d, e — квадратные матрицы одинакового размера, содержащие коэффициенты аппроксимирующего уравнения; f — квадратная матрица, содержащая значения правой части уравнения в каждой точке области, где ищется решение; u — квадратная матрица, содержащая граничные значения решения на границах квадратной области и начальное приближение для решения внутри области; $r\text{jac}$ — спектральный радиус итераций Якоби. Это число между 0 и 1, которое управляет сходимостью процесса релаксации.

Использование этой функции требует глубокого знания метода конечных разностей для составления указанных матриц. Пример использования функции `relax` приведен на рис. 6.35 и 6.36.

```

Определим размеры сетки
n := 32 i := 0..n j := 0..n
ai,j := 1 b := a c := a d := a e := -4·a
Определим положение и интенсивность источника тепла
Пусть во всех узлах интенсивность постоянная
Si,j := 0.05
Граничные условия
на верхней границе f0,j := 0
на нижней границе fn,j := 2·cos(4·π·j/n)
по бокам fi,0 := 2·i/n fi,n := 2·i/n
Спектральный радиус Якоби r := 1 - 2·π/n r = 0.804
Решение уравнения Пуассона F := relax(a,b,c,d,e,S,f,r)

```

Рис. 6.35. Решение уравнения Пуассона с помощью функции `relax`

В части IV книги приводится пример решения уравнения Пуассона при кручении стержней произвольного поперечного сечения с помощью значительно более мощного метода конечных элементов (пример 54 электронной книги). Этот пример может быть полезен и при других приложениях уравнения Пуассона.

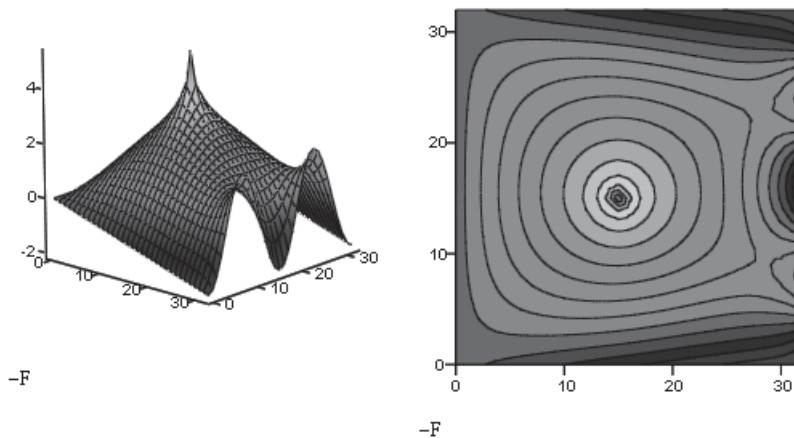


Рис. 6.36. Результаты решения уравнения Пуассона с помощью функции `relax`

Завершая рассмотрение возможностей решения дифференциальных уравнений, отметим, что созданная универсальная функция `Odesolve`, включившая в свой механизм решения ОДУ все созданные ранее функции плюс функции интерполяции, одна может справиться с любой из рассмотренных в этой главе задач. Простота использования этой функции в сочетании с огромными возможностями восхищает. Тем не менее Mathcad пока не в силах решать некоторые дифференциальные уравнения.

Глава 7

Обработка экспериментальных данных

При обработке экспериментальных данных, как правило, возникает задача аппроксимации результатов эксперимента аналитической зависимостью $y = f(x)$, которую можно использовать в последующих расчетах.

Существует три возможности аппроксимации опытных данных.

1. Аппроксимирующая функция $f(x)$ должна проходить через все опытные точки. Такой способ аппроксимации называется интерполяцией.
2. Выбрать аппроксимирующую функцию таким образом, чтобы она сглаживала, усредняла опытные данные. Такой способ аппроксимации называется регрессией, или сглаживанием.
3. Подобрать аппроксимирующую функцию, отбросив систематическую погрешность, так называемые шумы, наложившиеся на экспериментальные данные. Такой способ называется сглаживанием с фильтрацией данных.

В Mathcad имеется много встроенных функций для обработки массивов данных. Рассмотрим последовательно все три способа обработки данных.

7.1. Интерполяция

Встроенные функции Mathcad позволяют при интерполяции проводить через экспериментальные точки кривые разной степени гладкости.

7.1.1. Линейная интерполяция

При линейной интерполяции аппроксимирующая функция соединяет опытные точки отрезками прямых линий. Для линейной интерполяции используется встроенная функция `linterp`.

Обращение к функции:

$$\text{linterp}(x, y, t),$$

где x — вектор опытных значений аргумента; y — вектор опытных значений функции; t — значение аргумента, при котором вычисляется интерполирующее значение функции.

Иногда надо знать одно интерполирующее значение функции, но чаще определяется ряд значений функции, тогда t — вектор значений, а результат расчета — массив интерполирующих значений. Еще чаще t — просто переменная (аргумент интерполирующей функции), тогда результат расчета — функция, которую можно далее интегрировать, дифференцировать и т. д. Примеры линейной интерполяции показаны на рис. 7.1. Исходный массив данных Y на этом рисунке создан генератором случайных чисел `rnd`, о котором рассказано в следующей главе. Координаты X в этом примере взяты равноотстоящими друг от друга, но это не обязательно.

массив экспериментальных данных (X, Y)

$$X := (0 \ 1 \ 3 \ 4 \ 6 \ 7 \ 10 \ 11 \ 13 \ 14 \ 15)^T$$

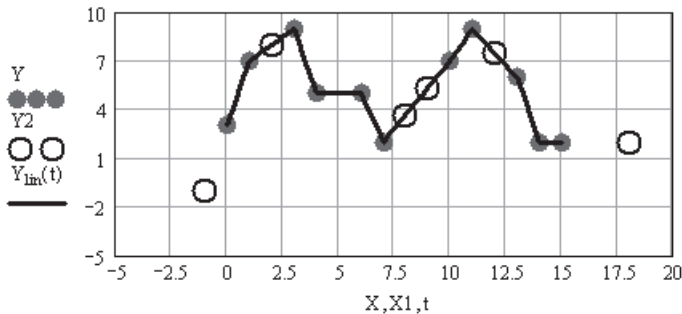
$$Y := (3 \ 7 \ 9 \ 5 \ 5 \ 2 \ 7 \ 9 \ 6 \ 2 \ 2)^T$$

$Y1 := \text{linterp}(X, Y, 5)$ $Y1 = 5$ интерполирующее значение
в одной точке $X=5$

$X1 := (-1 \ 2 \ 8 \ 9 \ 12 \ 18)^T$ $Y2 := \text{linterp}(X, Y, X1)$

$Y2$ - массив интерполирующих значений массива Y в точках,
заданных вектором $X1$

$t := 0, 0.2.. \max(X)$ $Y_{\text{lin}}(t) := \text{linterp}(X, Y, t)$ непрерывная интерполирующая
функция



Крайние точки на графике - экстраполяция за пределы массива X

Рис. 7.1. Линейная интерполяция экспериментальных данных

ВНИМАНИЕ

В этой главе для всех встроенных функций в массиве опытных данных элементы вектора x должны быть расположены обязательно в порядке возрастания.

7.1.2. Кубическая сплайн-интерполяция

В большинстве случаев желательно соединять экспериментальные точки не ломаной линией, а гладкой кривой. Для этих целей используется сплайн-интерполяция. Кубическая сплайн-интерполяция позволяет провести кривую через набор точек таким образом, что первые и вторые производные кривой непрерывны в каждой точке. Эта кривая образуется путем создания ряда кубических полиномов, проходящих через наборы из трех соседних точек. Кубические полиномы затем состыкуются друг с другом так, чтобы образовать единую кривую.

Для кубической сплайн-интерполяции используется встроенная функция `interp`.

Обращение к функции:

$$\text{interp}(vs, X, Y, t),$$

где vs — вектор вторых производных, созданный одной из трех функций, `lspline(X, Y)`, `pspline(X, Y)`, `cspline(X, Y)`; X — вектор опытных значений аргумента, расположен-

ных в порядке возрастания; Y – вектор опытных значений функции; t – значение аргумента, при котором вычисляется интерполирующее значение функции.

Сплайн-интерполяция организована в Mathcad несколько сложнее, чем линейная. Перед применением функции `interp` необходимо определить первый из ее аргументов – вектор ts . Делается это при помощи одной из трех встроенных функций тех же аргументов (x, y) :

- ❑ `lspline(X, Y)` создает вектор коэффициентов кривой, которая приближается к прямой линии в граничных точках;
- ❑ `pspline(X, Y)` создает вектор коэффициентов кривой, которая приближается к квадратичной параболе в граничных точках;
- ❑ `cspline(X, Y)` создает вектор коэффициентов кривой, которая приближается к кубической параболе в граничных точках.

Выбор конкретной функции сплайн-коэффициентов влияет на интерполяцию вблизи конечных точек интервала. Использование двух функций для интерполяции можно совместить, записав, например, `interp(cspline(X, Y), X, Y, t)`. Примеры кубической сплайн-интерполяции показаны на рис. 7.2.

$$Y_{\text{lspl}}(t) := \text{interp}(\text{lspline}(X, Y), X, Y, t)$$

$$Y_{\text{pspl}}(t) := \text{interp}(\text{pspline}(X, Y), X, Y, t)$$

$$Y_{\text{cspl}}(t) := \text{interp}(\text{cspline}(X, Y), X, Y, t)$$

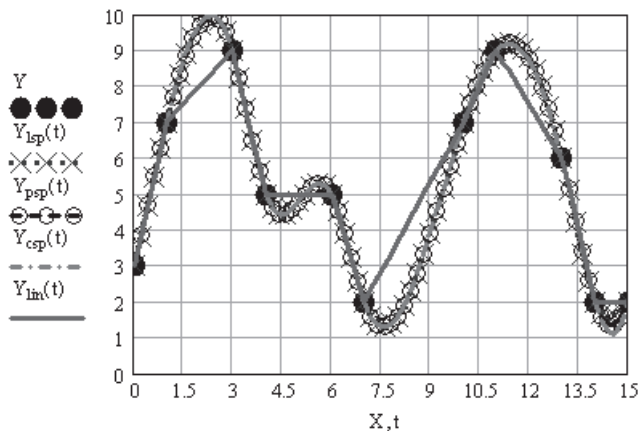


Рис. 7.2. Кубическая сплайн-интерполяция массива Y , приведенного на рис. 7.1

7.1.3. В-сплайн-интерполяция

В-сплайн-интерполяция отличается от обычной сплайн-интерполяции тем, что сшивка отдельных отрезков (сплайнов) производится не в точках X_i , а в точках, координаты которых задаются пользователем. Применяется интерполяция В-сплайнами так же, как и обычная сплайн-интерполяция, но для определения сплайн-коэффициентов применяется функция `bspline`.

Обращение к функции:

$$\text{bspline}(X, Y, XX, n),$$

где X — вектор опытных значений аргумента, расположенных в порядке возрастания; Y — вектор опытных значений функции; XX — вектор значений аргумента, при которых вычисляются интерполирующие значения функции; n — порядок полиномов сплайн-интерполяции (1, 2 или 3).

Размерность вектора XX должна быть на $(n - 1)$ меньше, чем размерность вектора X . Первый элемент вектора XX должен быть меньше первого элемента вектора X или равен ему. Последний элемент вектора XX должен быть больше последнего элемента X или равен ему.

Наложенные на вектор XX ограничения делают крайне неудобным использование В-сплайн-интерполяции, так как пользователь лишен возможности свободного выбора точек сшивки полиномов. К тому же по непонятным причинам функция bspline отказывается интерполировать некоторые наборы данных. В центре документации Mathcad примеры использования этой функции отсутствуют. Пример использования функции bspline приведен на рис. 7.3.

```

nn := 2      поменяйте порядок полиномов nn=1,2 или 3

X := (0 1 2 3 4 5 6)T      массивы
                          исходных
                          данных
Y := (4 2 3 4 3 5 6)T
XX := (-1 1 3 4 5 7)T      аргументы точек соединения
                          полиномов интерполяции

xmin := min(XX)  xmax := max(XX)  x := xmin + xmin +  $\frac{x_{\max} - x_{\min}}{30} \cdot x_{\max}$ 

Ycsp(t) := interp(cspine(X, Y), X, Y, t)  кубическая сплайн-интерполяция

YY(t) := interp(bspline(X, Y, XX, nn), X, Y, t)  В - сплайн-интерполяция

```

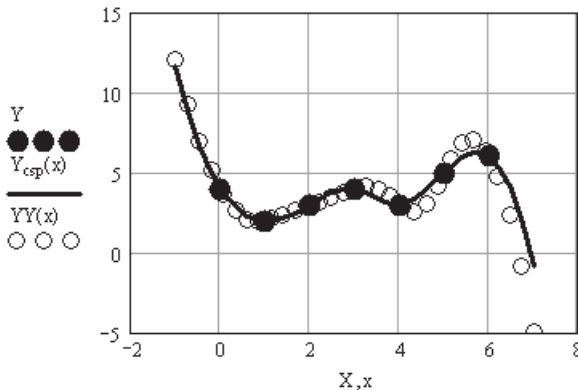


Рис. 7.3. В-сплайн-интерполяция массива Y , приведенного на рис. 7.1

ПРИМЕЧАНИЕ

Если два соседних значения аргумента близки, все функции интерполяции, кроме `linterp`, дают всплеск на графике в этом месте (рис. 7.4). Чтобы убрать всплеск, раздвиньте близлежащие точки. В этих случаях желательно использовать функции регрессии вместо функций интерполяции.

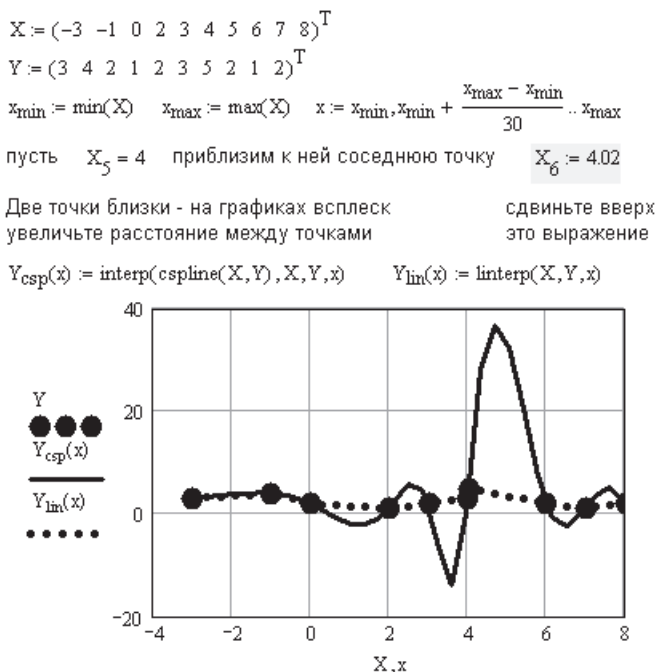


Рис. 7.4. Влияние близких точек на результат интерполяции. Массив Y приведен на рис. 7.1

7.1.4. Линейное предсказание

Кроме интерполяции в промежуточных точках иногда нужно оценить значения функции за пределами области заданных точек. Все рассмотренные ранее функции интерполяции выполняют экстраполяцию данных за пределами интервала заданных точек с помощью соответствующей зависимости, основанной на анализе расположения нескольких исходных точек на границе интервала данных. Но в Mathcad имеется и специальная функция предсказания `predict`.

Обращение к функции:

$$\text{predict}(Y, m, n),$$

где Y — вектор заданных значений функции, обязательно взятых через равные интервалы аргумента; m — число последовательных значений Y , на основании которых функция `predict` возвращает n предсказанных значений Y .

Значений аргумента для данных не требуется, поскольку по определению функция действует на данные, следующие друг за другом с одинаковым шагом.

Обратите внимание на то, что результат функции `predict` надо вставлять «в хвост» исходных данных.

Функция `predict` использует линейный алгоритм предсказания, который является полезным, когда экстраполируется функция гладкая и осциллирующая, хотя необязательно периодическая. Эта функция использует последние m исходных значений данных, чтобы вычислить коэффициенты предсказания. Как только это сделано, она использует последние m точек, чтобы предсказать координаты $(m + 1)$ -й точки, фактически создавая скользящее окно шириной m точек.

Как видно из рис. 7.5, функция предсказания может быть полезной при экстраполяции данных на небольшие расстояния. Вдали от исходных данных результат чаще всего оказывается неудовлетворительным. Если исходных данных мало, то результат также окажется бесполезным.

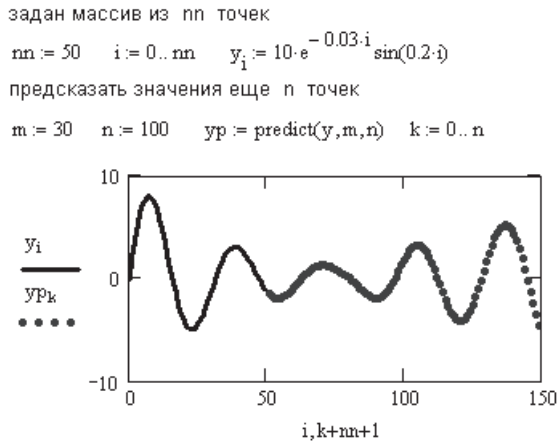


Рис. 7.5. Функция предсказания в Mathcad

7.1.5. Интерполяция функции двух переменных

Mathcad выполняет интерполяцию кубическими сплайнами функции двух переменных таким же образом, как и при одномерной интерполяции, рассмотренной ранее. Mathcad проводит через сетку узлов поверхность, составленную из кубических полиномов от x и y , так, что первые и вторые частные производные являются непрерывными в каждом узле сетки.

Первый шаг в двухмерной сплайн-интерполяции (как и в одномерной) — определение узлов, через которые поверхность должна пройти:

- Создайте матрицу XU размерностью $n \times 2$, элементы которой определяют координаты x и y диагонали прямоугольной сетки. Таким образом, число узлов по осям координат x и y обязательно должно быть одинаковым, что резко ограничивает возможности двухмерной интерполяции или заставляет прибегать к специальным приемам, чтобы привести неквадратную область к квадратной.

- Введите матрицу Z размерностью $n \times n$, каждый элемент которой — заданное значение функции в соответствующем узле.
- Вычислите вектор вторых производных $\text{cspline}(XY, Z)$ функции Z в узлах.
- Чтобы найти интерполируемое значение функции Z в произвольной точке, скажем, в точке с координатами x_0 и y_0 , используйте функцию `interp`:

$$Z_0 := \text{interp}\left(\text{cspline}(XY, Z), XY, Z, \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}\right).$$

На практике приходится вычислять интерполируемые значения функции сразу во всех узлах сетки. Для этого записываем Z как функцию от x и y и с помощью функции `CreateMech` создаем массив точек для графика поверхности:

$$ZZ(x, y) := \text{interp}\left(\text{cspline}(XY, Z), XY, Z, \begin{bmatrix} x \\ y \end{bmatrix}\right);$$

$$Z_1 := \text{CreateMech}(ZZ, x_1, x_2, y_1, y_2, nx, ny).$$

В обращении к функции `CreateMech` обязательным является только параметр ZZ , остальные можно опустить. Здесь x_1, x_2, y_1, y_2 — нижний и верхний пределы аргументов x и y (по умолчанию от -5 до 5); nx и ny — число точек сетки по переменным x и y (по умолчанию 20). Примеры двумерной интерполяции приведены на рис. 7.6 и 7.7. О применении функции `CreateMech` написано в главе 15 (раздел 15.3.3).

Массив значений функции Z задан на квадратной области в точках с координатами X и Y

$$X := (0 \ 1 \ 2 \ 3 \ 4 \ 5)^T \qquad Y := (0 \ 1 \ 2 \ 3 \ 4 \ 5)^T$$

Координаты точек перестраиваем в двумерный массив XY и вводим в параметры функции `interp`

$$XY^{(0)} := X \qquad XY^{(1)} := Y$$

$$XY = \begin{pmatrix} 0 & 0 \\ 1 & 1 \\ 2 & 2 \\ 3 & 3 \\ 4 & 4 \\ 5 & 5 \end{pmatrix} \qquad Z := \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.345 & 0.559 & 0.559 & 0.345 & 0 \\ 0 & 0.559 & 0.905 & 0.905 & 0.559 & 0 \\ 0 & 0.559 & 0.905 & 0.905 & 0.559 & 0 \\ 0 & 0.345 & 0.559 & 0.559 & 0.345 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$ZZ(x, y) := \text{interp}\left[\text{cspline}(XY, Z), XY, Z, \begin{pmatrix} x \\ y \end{pmatrix}\right]$$

Рис. 7.6. Двухмерная интерполяция массива данных

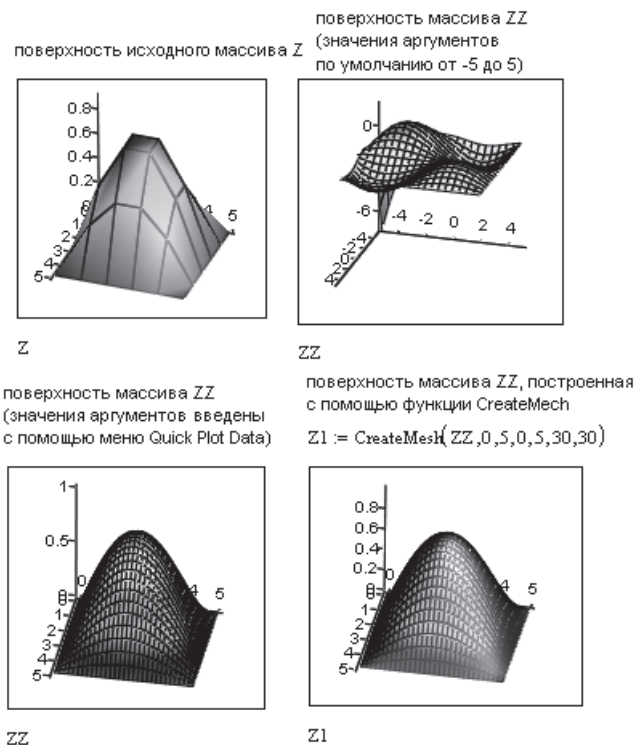


Рис. 7.7. Графики поверхности исходных точек Z и поверхности интерполирующей функции

При построении графика поверхности можно обойтись без явного использования функции `CreateMesh`. На математической панели выбрать значок графика `Graph Toolbar` ▶ `Surface Plot` (Панель графиков ▶ График поверхности). В поле ввода ввести имя функции двух переменных (без параметров). Появится график поверхности функции. По умолчанию берутся пределы изменения переменных от -5 до $+5$. Чтобы их изменить, сделайте двойной щелчок в поле графика и в открывшемся окне форматирования графика выберите вкладку `Quick Plot Data` (Данные быстрого графика), на которой впишите необходимые пределы изменения переменных (`range1` и `range2`) (рис. 7.8). При таком построении графика внутренний алгоритм `Mathcad` самостоятельно использует функцию `CreateMesh`.

Для двумерной интерполяции векторы значений координат по осям x и y должны иметь одинаковый размер, то есть интерполяция выполняется только на квадратной поверхности. Однако это требование можно обойти.

Перед вычислением значений функции на неквадратной плоскости нужно обнулить массив значений функции на квадратной поверхности, большей, чем заданная неквадратная. После этого вычислить значения функции на заданной поверхности (рис. 7.9) и построить график поверхности (рис. 7.10). В результате поверхность появится лишь над частью квадратной плоскости.

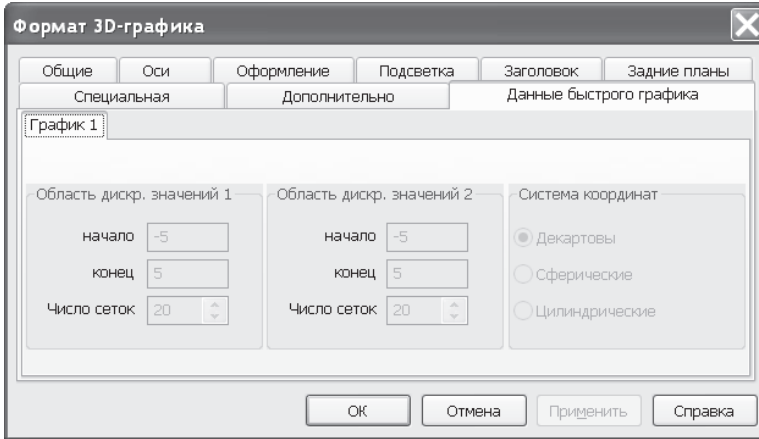


Рис. 7.8. Установка пределов значений координат

```

X := (0 1 2 3 4 5)T   Y := (0 1 2 3 4 5)T   массивы данных
n := 10   i := 0..n   j := 0..n   Zi,j := 0   Обнуление функции на
                                         квадратной поверхности
XYi,0 := i   XYj,1 := j   XYT =  $\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{pmatrix}$ 
i1 := 0..3   j1 := 0..4   индексы элементов массива данных Z1
Z1 :=  $\begin{pmatrix} 0.345 & 0.559 & 0.559 & 0.345 & 0.345 \\ 0.559 & 0.905 & 0.505 & 0.559 & 0.559 \\ 0.559 & 0.305 & 0.905 & 0.559 & 0.559 \\ 0.345 & 0.559 & 0.559 & 0.345 & 0.345 \end{pmatrix}$    Неквадратный массив Z1
                                         можно поместить в любое
                                         место квадратной области Z
                                         (поменяйте индексы массива Z)
                                         Zi1+5,j1+3 := Z1i1,j1
    
```

Рис. 7.9. Двухмерная интерполяция на неквадратной области

На рис. 7.10 поверхность Z_1 (справа) — это график массива, созданного функцией CreateMech, после двухмерной интерполяции массива Z (см. рис. 7.9). Поверхность ZZ построена непосредственно после интерполяции с использованием вкладки Quick Plot Data. Работая с этим рисунком, поменяйте размеры неквадратной области.

7.2. Функции регрессии

Mathcad включает ряд функций для вычисления регрессии. Эти функции создают кривые или поверхности определенного типа с минимальным отклонением от имеющегося набора данных. Функции различаются прежде всего типом кривой или поверхности, которую они используют, чтобы аппроксимировать данные.

В отличие от функций интерполяции, эти функции не требуют, чтобы аппроксимирующая кривая или поверхность проходила через точки данных.

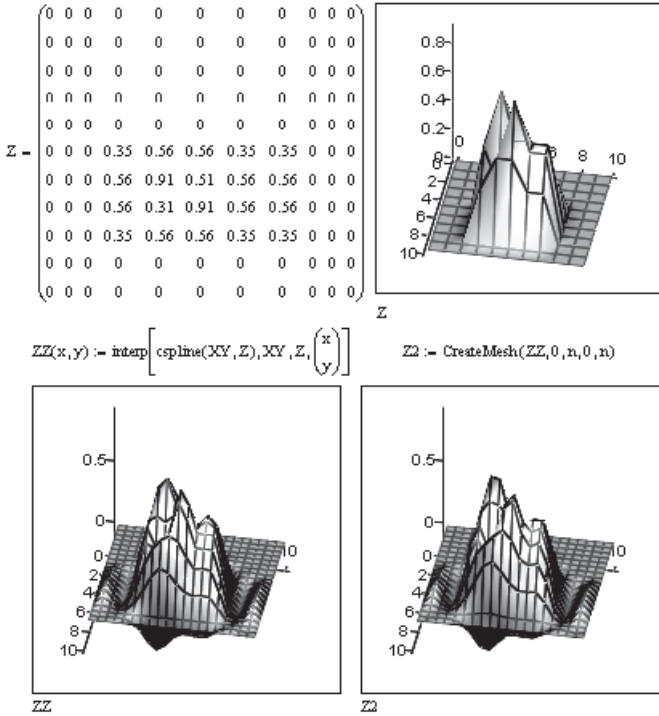


Рис. 7.10. Результаты интерполяции на неквадратной поверхности

7.2.1. Одномерная регрессия

Mathcad допускает регрессию одномерных и многомерных массивов. Вначале рассмотрим регрессию одномерных массивов.

Ряд функций регрессии возвращают коэффициенты аналитической функции и позволяют записать аппроксимирующую функцию в явном виде. Эти функции описаны в главе 2 (раздел 2.11.2), и здесь не рассматриваются.

Ряд функций регрессии не позволяют получить аппроксимирующую функцию в явном виде. Они возвращают лишь численные результаты в виде таблиц или графиков. Две из них — regress и loess — также рассмотрены в разделе 2.11.2. Эти две функции не выводят коэффициенты и аппроксимируют массив данных одним степенным полиномом (regress) или отрезками нескольких полиномов (loess).

В Mathcad регрессия с использованием одного полинома реализуется комбинацией встроенных функций регрессии regress и интерполяции interp.

Обращение к указанным функциям:

$$s := \text{regress}(X, Y, n); YY(t) := \text{interp}(s, X, Y, t)$$

или

$$YY(t) := \text{interp}(\text{regress}(X, Y, n), X, Y, t),$$

где X – вектор значений аргумента, элементы которого расположены в порядке возрастания; Y – вектор значений функции того же размера; s – вектор коэффициентов для построения аппроксимирующего полинома, создаваемый функцией `regress`; t – значение аргумента, при котором вычисляется интерполирующая функция; n – степень аппроксимирующего полинома.

Степень аппроксимирующего полинома может быть любой.

Регрессия с использованием нескольких отрезков полинома также реализуется комбинацией встроенных функций регрессии и интерполяции

$$\text{interp}(s, x, y, t) \text{ и } \text{loess}(x, y, \text{span}),$$

где $s := \text{loess}(X, Y, \text{span})$ – вектор коэффициентов для построения аппроксимирующего полинома второй степени, требуемый функцией `interp`; $\text{span} > 0$ – параметр, определяющий размер отрезков полиномов.

Параметр `span` задает степень сглаженности данных. Аргумент `span` определяет размер отрезков полиномов, из которых «сшивается» аппроксимирующая кривая. На практике $0,2 < \text{span} < 2$. При $\text{span} = 2$ результат аппроксимации тот же, что и при аппроксимации одной параболой. При $\text{span} = 0,2$ аппроксимирующая кривая почти точно описывает любой набор данных.

Примеры использования функций `regress` и `loess` приведены в главе 2 на рис. 2.26 и на рис. 7.11.

```

Y := (3 0 -5 -4 -1 2 5 6 4 1)ᵀ массив значений функции
X := (-2 0 1 3 5 6 9 11 12 15)ᵀ массив значений аргумента
                               в возрастающем порядке

поменяйте параметры 1 < N < 20 и 0.2 < L < 2
N := 4 степень полинома
L := 0.5 характеристика длины отрезков полиномов
Y_regress(x) := interp(regress(X, Y, N), X, Y, x)
Y_loess(x) := interp(loess(X, Y, L), X, Y, x)
x_min := min(X) x_max := max(X) x := x_min, x_min + (x_max - x_min) / 30 .. x_max
    
```

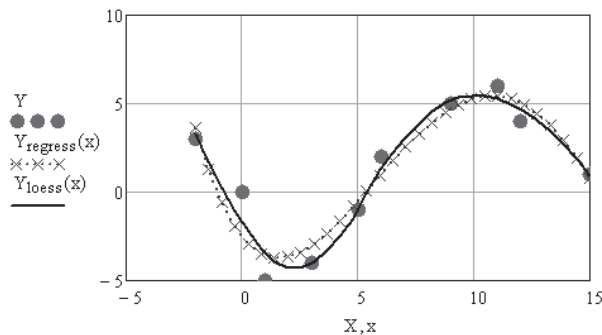


Рис. 7.11. Функции регрессии, не дающие аналитического выражения

7.2.2. Двухмерная регрессия

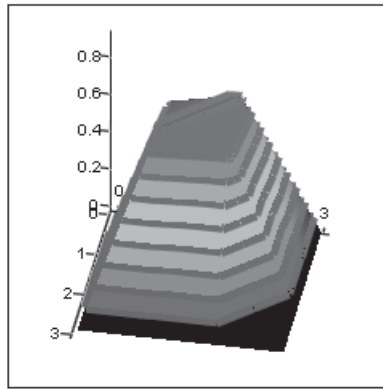
Функции `regress` и `loess` допускают и многомерную регрессию. Пример двухмерной регрессии приведен на рис. 7.12 и 7.13. В этом случае массив XU координат точек x становится двухмерным массивом, состоящим из двух столбцов: первый столбец — это координаты по оси x , второй — координаты по оси y .

$$Z_A := \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0.84 & 0.91 & 0 \\ 0 & 0.91 & 0.84 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad \text{Исходный массив точек}$$

координаты точек

$$x_i := \begin{pmatrix} 0 \\ 1 \\ 2 \\ 3 \end{pmatrix} \quad y_j := \begin{pmatrix} 0 \\ 10 \\ 20 \\ 30 \end{pmatrix}$$

x и y даны для построения массива XU



Z_A

Рис. 7.12. Исходные массивы и их перестройка для двухмерной регрессии

ВНИМАНИЕ

При двухмерной интерполяции массив координат точек XU состоит из координат точек диагонали квадратной области. Например, для массива точек 4×4 вводятся координаты лишь 4 точек. Координаты остальных точек определяются автоматически. При двухмерной регрессии массив координат точек XU состоит из координат всех точек прямоугольной области (не обязательно квадратной), записанных по столбцам. Например, для массива точек 4×5 вводятся координаты всех 20 точек.

Массив значений функции Z во всех точках прямоугольной области вводится по столбцам. Число строк вектора Z равно числу строк массива XU .

Двухмерная интерполяция производится как функцией `regress(XU, Z, N)`, так и функцией `loess(XU, Z, span)`. Здесь N — порядок аппроксимирующего полинома, `span` определяет размер отрезков полиномов, как и при одномерной регрессии.

Mathcad допускает многомерную регрессию. Можно увеличивать число независимых переменных, просто добавляя столбцы в массив XU , а затем добавляя соответствующее число строк к вектору Z , которые передаются функции `interp`.

Функция regress может иметь любое число независимых переменных, но она будет работать слишком медленно и требовать много памяти. Функция loess допускает максимум 4 независимые переменные.

Перестройка двумерных массивов функции и ее аргументов в одномерные (по столбцам) массив координат x и y

$$XY := \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 3 \\ 0 & 10 & 20 & 30 & 35 & 0 & 10 & 20 & 30 & 35 & 0 & 10 & 20 & 30 & 35 & 0 & 10 & 20 & 30 & 35 \end{pmatrix}^T$$

$$Z := (0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0.84 \ 0.91 \ 0.61 \ 0 \ 0 \ 0.91 \ 0.84 \ 0.77 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0)^T$$

$$Z_{regress}(x, y) := \text{interp} \left[\text{regress}(XY, Z, 4), XY, Z, \begin{pmatrix} x \\ y \end{pmatrix} \right]$$

$$Z_{loess}(x, y) := \text{interp} \left[\text{loess}(XY, Z, 0.6), XY, Z, \begin{pmatrix} x \\ y \end{pmatrix} \right]$$

значения аргументов $x=0.3$ и $y=0.35$ установлены при форматировании графика в меню QuickPlot Data

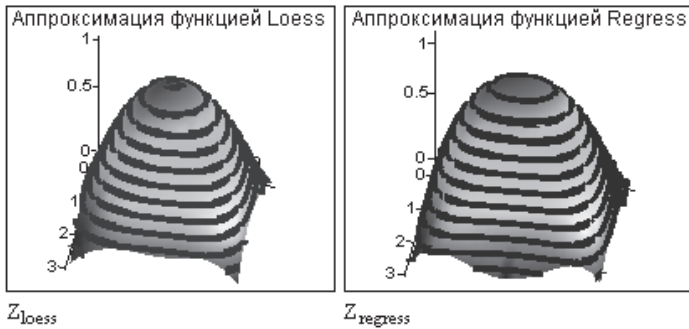


Рис. 7.13. Результаты двумерной интерполяции массива, приведенного на рис. 7.10

Для функции regress число значений данных m должно удовлетворять соотношению

$$m > \binom{n+k-1}{k} \frac{n+k}{n},$$

где n — число независимых переменных (число столбцов XY); k — желаемая степень полинома; m — число значений данных (число строк в массиве XY). Например, если имеется 5 независимых переменных и ищется приближение полиномом 4-й степени, потребуется более 126 наблюдений.

7.2.3. Обобщенная регрессия

Бывают случаи, когда закон изменения функции известен и требуется подобрать лишь ее коэффициенты, однако сама аппроксимирующая функция такова, что использовать функции специальной регрессии невозможно. В этом случае Mathcad предлагает использовать две функции регрессии общего вида — linfit и genfit.

Если предполагаемая зависимость представляет собой линейную комбинацию некоторых функций

$$y = a_0 f_0(x) + a_1 f_1(x) + \dots + a_n f_n(x)$$

и надо лишь подобрать коэффициенты этих функций, то используется функция `linfit`, например:

$$f(x) = a_1 \sin(2x) + a_2 \cos(3x),$$

`linfit(x, y, F)` возвращает вектор, содержащий набор коэффициентов при функциях $f_i(x)$, $y(x)$ – вектор заданных значений функции при значениях аргумента, содержащихся в векторе x , F – вектор функций, входящих в линейную комбинацию $y(x)$.

Для произвольной функции с неизвестными коэффициентами надо использовать функцию `genfit`, например:

$$f(x) = 2\sin(a_1 x) + 3\cos(a_2 x),$$

`genfit(x, y, C, F)` возвращает вектор, содержащий n параметров, наилучшим образом аппроксимирующих данные из векторов x и y заданной функцией $f(x)$. Здесь $F(x)$ – функция, которая возвращает $(n + 1)$ -мерный вектор, содержащий функцию $f(x)$ и ее частные производные относительно неизвестных параметров; C – n -мерный вектор начальных значений для n неизвестных параметров.

Примеры применения функций `linfit` и `genfit` приведены на рис. 7.14–7.19.

```

Y := (0.6 2 2.8 3.5 4.2 4.6 4 3.2 2 4)T    массив значений функции
X := (0 1 2 3 5 6 9 11 12 15)T          массив значений аргумента
                                       в возрастающем порядке
F(x) :=  $\begin{pmatrix} x^2 \\ \cos(x) \\ x \end{pmatrix}$  набор функций, входящих в линейную комбинацию
                                       поменяйте функции в векторе F(x)
S := linfit(X, Y, F)  S =  $\begin{pmatrix} -0.059 \\ -0.363 \\ 1.024 \end{pmatrix}$  найденные коэффициенты
                                       при функциях
f(x) := F(x)·S    линейная комбинация функций
ff(x) := -0.059·x2 - 0.365·sin(x) + 1.024·x
эта формула - всего лишь иллюстрация к f(x).
В документе она не нужна.
xmin := min(X)    xmax := max(X)    x := xmin, xmin +  $\frac{x_{max} - x_{min}}{20}$  .. xmax
    
```

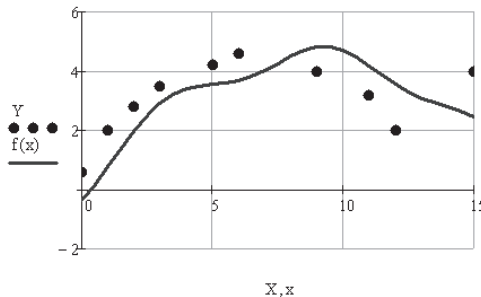


Рис. 7.14. Регрессия данных функцией `linfit`

```

массив значений аргумента в возрастающем порядке
X := (0 1 2 3 4 5 6 7 8 9 10)T
массив значений функции
Y := (0 0.7 0.9 0.3 0 -0.2 -0.4 -0.4 -0.4 -0.3 -0.2)T
f(x, a, b, c) :=  $\frac{a \cdot x^3}{x^4 + b} \cdot \cos(c \cdot x)$  предлагаемая функция
поменяйте функцию

символьное вычисление производных
 $\frac{d}{da} f(x, a, b, c) \rightarrow \frac{x^3 \cdot \cos(c \cdot x)}{x^4 + b}$ 
 $\frac{d}{dxc} f(x, a, b, c) \rightarrow -\frac{a \cdot x^3 \cdot \sin(c \cdot x)}{x^4 + b}$ 
 $\frac{d}{db} f(x, a, b, c) \rightarrow -\frac{a \cdot x^3 \cdot \cos(c \cdot x)}{(x^4 + b)^2}$ 

k :=  $\begin{pmatrix} 2 \\ 2 \\ 0.4 \end{pmatrix}$  a начальные приближения
c
    
```

Рис. 7.15. Нелинейная регрессия данных функцией genfit в Mathcad 11 и более старых версиях

```


$$FF(x, k) := \begin{pmatrix} k_0 \cdot \frac{x^3}{x^4 + k_1} \cdot \cos(k_2 \cdot x) \\ \frac{x^3}{x^4 + k_1} \cdot \cos(k_2 \cdot x) \\ k_0 \cdot \frac{x^3}{(x^4 + k_1)^2} \cdot \cos(k_2 \cdot x) \\ k_0 \cdot \frac{x^4}{x^4 + k_1} \cdot \sin(k_2 \cdot x) \end{pmatrix}$$

предлагаемая функция
ее производная по a
ее производная по b
ее производная по c

K := genfit(X, Y, k, FF) K =  $\begin{pmatrix} 2.56 \\ 1.877 \\ 0.414 \end{pmatrix}$  вектор найденных параметров функции

F(x) := FF(x, K)0 искомая функция с найденными параметрами

f(x) :=  $\frac{2.49 \cdot x^3}{x^4 + 1.855} \cdot \cos(0.406 \cdot x)$  эта формула - всего лишь иллюстрация к F(x). В документе она не нужна
    
```

Рис. 7.16. Нелинейная регрессия данных функцией genfit в Mathcad 11 (окончание)

До появления Mathcad 13 использование функции `genfit` было весьма сложным в связи с необходимостью вычисления частных производных искомой функции по всем неизвестным параметрам. Символьное вычисление таких производных и их использование показано на рис. 7.15 и 7.16.

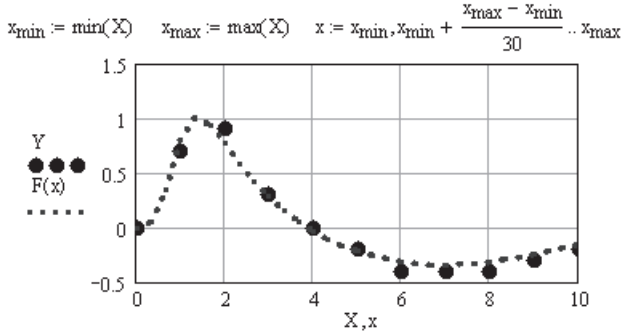


Рис. 7.17. Результаты нелинейной регрессии данных функцией `genfit`

В Mathcad 13 использование функции `genfit` значительно упростилось. Функция `genfit` теперь допускает задание параметров в явном виде (рис. 7.18). Однако частные производные от аппроксимирующей функции по ее параметрам в этом случае надо определять самому пользователю.

```
X := (0.5 1 1.5 2 2.5 3 3.5 4 4.5)T
Y := (16 12 20 23 19 24 46 69 120)T
f(x,A,γ) := A·eγ·x аппроксимирующая функция
k :=  $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$  начальные приближения параметров A и γ
F(x,A,γ) :=  $\begin{pmatrix} f(x,A,\gamma) \\ e^{\gamma \cdot x} \\ A \cdot x \cdot e^{\gamma \cdot x} \end{pmatrix}$  функция и ее производные по A и γ
K := genfit(X,Y,k,F)
FF(x) := f(x,K0,K1)
```

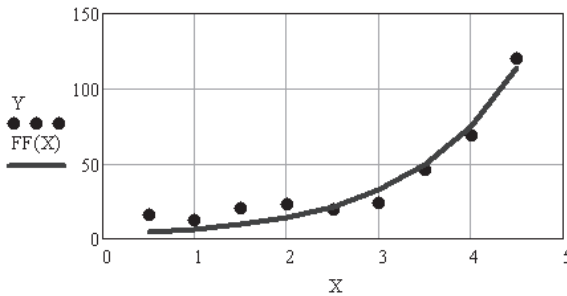


Рис. 7.18. Ввод явных параметров в функцию `genfit` в Mathcad 13

В Mathcad 13 функция `genfit` не требует обязательного вычисления производных по всем расчетным параметрам. Если в обращение к функции вписано имя исходной функции, то необходимые производные вычисляются численно внутренним алгоритмом Mathcad. Правда, все параметры, которые надо определить, должны быть элементами вектора. Для этого вектора и задаются начальные приближения (рис. 7.19).

В контекстном меню функции `genfit` предлагается 2 метода `Levenberg–Maquardt` (Левенберга–Маккарта) и `Optimized Levenberg–Maquardt` (Оптимизированный метод Левенберга–Маккарта). Если используемый по умолчанию метод `Levenberg–Maquardt` не дает решения, то выберите в контекстном меню метод `Optimized Levenberg–Maquardt`. Так пришлось поступить в примере на рис. 7.19. При таком вводе несколько уменьшается точность расчета.

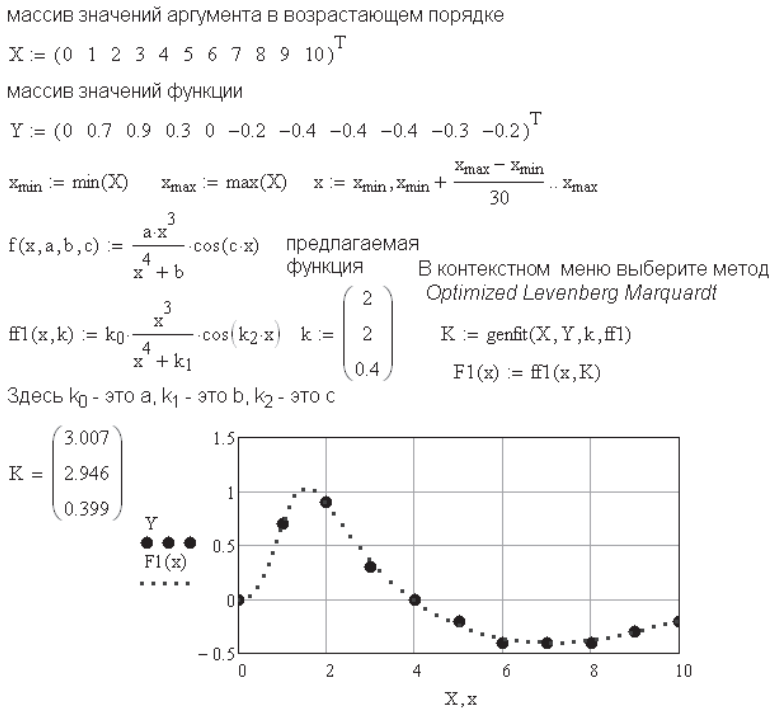


Рис. 7.19. Применение функции `genfit` без вычисления производных в Mathcad 13

7.3. Функции сглаживания

В процессе сглаживания из исходного набора данных получается новый набор данных, более гладкий, чем исходный.

СОВЕТ

Убедитесь, что каждый элемент в векторе исходных данных определен. Mathcad присваивает значение 0 любым элементам, значения которых не определены.

Mathcad предлагает три встроенные функции сглаживания (рис. 7.20):

- $\text{medsmooth}(y, n)$ возвращает m -мерный вектор, созданный сглаживанием с помощью скользящей медианы. Здесь y — m -мерный вектор вещественных чисел; n — ширина окна, по которому происходит сглаживание, n должно быть нечетным числом, меньшим или равным числу элементов вектора y .
- $\text{ksmooth}(x, y, b)$ возвращает n -мерный вектор. Здесь x и y — n -мерные векторы, вектор y создан сглаживанием при помощи гауссова ядра данных из y ; b — параметр, управляющий окном сглаживания. Он должен быть в несколько раз больше величины интервала между точками x . Чем больше b , тем больше окно, используемое при сглаживании.
- $\text{supsmooth}(x, y)$ возвращает n -мерный вектор, созданный локальным использованием симметричной линейной процедуры сглаживания методом наименьших квадратов по правилу k ближайших соседей, в которой k выбирается адаптивно. Здесь x и y — n -мерные векторы исходных данных. Элементы x должны быть расположены в порядке возрастания.

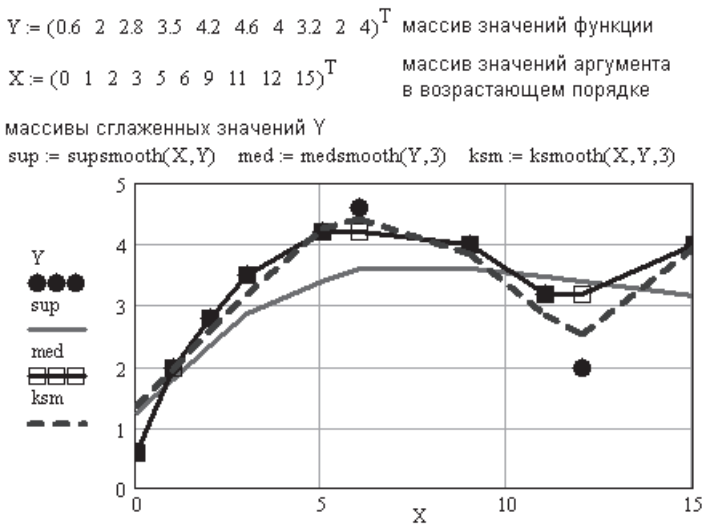


Рис. 7.20. Сглаживание данных с созданием массива значений

Функция medsmooth наиболее устойчивая из всех трех встроенных функций, так как в меньшей степени подвержена влиянию зашумленных данных.

Функция ksmooth использует гауссово ядро, чтобы вычислить локально взвешенные средние значения исходного вектора y . Такое сглаживание наиболее полезно, когда интервалы между точками данных примерно одинаковой величины.

Функция supsmooth полезна, если длина интервалов существенно изменяется. Эта функция адаптивно выбирает различную ширину полосы сглаживания для различных частей вектора данных.

В отличие от регрессии и интерполяции, функции сглаживания возвращают массив, а не функцию. Для получения функции после сглаживания необходимо использовать интерполяцию или регрессию (рис. 7.21).

Функции сглаживания возвращают массив, а не функцию.

Чтобы получить функцию, добавьте интерполяцию

$Y_{\text{interp}}(x) := \text{interp}(\text{cspline}(X, Y), X, Y, x)$ интерполяция без сглаживания

$Y_{\text{interp.sup}}(x) := \text{interp}(\text{cspline}(X, \text{sup}), X, \text{sup}, x)$ интерполяция

$Y_{\text{interp.med}}(x) := \text{interp}(\text{cspline}(X, \text{med}), X, \text{med}, x)$ со сглаживанием

$Y_{\text{interp.ksm}}(x) := \text{interp}(\text{cspline}(X, \text{ksm}), X, \text{ksm}, x)$ $x := 0, 0.1 \dots \max(X)$

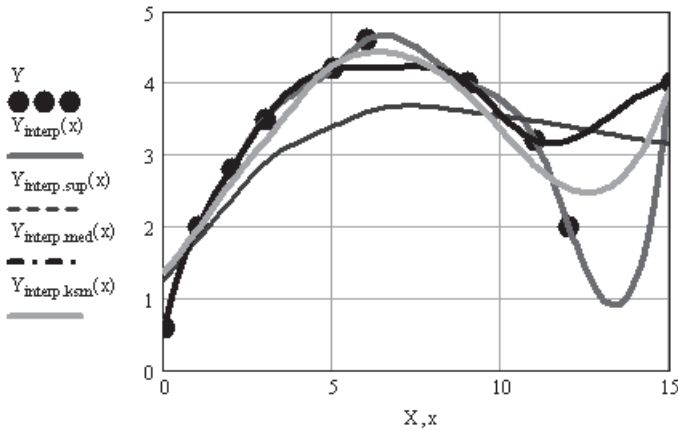


Рис. 7.21. Сглаживание данных с созданием функции

7.4. Дискретные преобразования

Mathcad содержит функции для быстрого дискретного преобразования Фурье (БПФ) и его обращения, а также для одномерного волнового преобразования и его обращения. Все эти функции имеют векторные аргументы.

7.4.1. Преобразование Фурье

В Mathcad входят два типа функций для дискретного преобразования Фурье: `fft` и `ifft`, `cfft` и `icfft`. Эти функции дискретны: они берут в качестве аргументов и возвращают векторы и матрицы. Они не могут быть использованы с другими функциями.

Используйте функции `fft` и `ifft`, если выполнены следующие два условия:

- аргументы вещественные;
- вектор данных имеет 2^m элементов.

Используйте функции `cfft` и `icfft` во всех прочих случаях.

Соблюдать первое условие необходимо, потому что функции `fft` и `ifft` используют тот факт, что для вещественных данных вторая половина преобразования Фурье является комплексно-сопряженной с первой. Mathcad отбрасывает вторую половину вектора результата, что сохраняет время и память при вычислениях.

Функции `cfft` и `icfft` не используют симметрию в преобразовании и могут использоваться для вещественных и комплексных чисел.

Второе условие требуется, потому что пара функций, `fft` и `ifft`, использует высокоэффективный алгоритм быстрого преобразования Фурье. Для этого вектор аргумента, используемого функцией `fft`, должен иметь 2^m элементов. Алгоритм функций `cfft` и `icfft` допускает в качестве аргументов векторы и матрицы произвольного размера. Для двухмерного преобразования Фурье используются только эти функции.

Существуют различные формулировки определения преобразования Фурье, использующие различные нормировочные коэффициенты и соглашения о знаке перед мнимой единицей в показателе экспоненты прямого и обратного преобразований. Функции `fft`, `ifft`, `cfft`, `icfft` используют нормировочный коэффициент $1/\sqrt{n}$ и положительный показатель степени в прямом преобразовании. Функции `FFT`, `IFFT`, `CFFT`, `ICFFT` используют нормировочный коэффициент $1/n$ и отрицательный показатель степени в прямом преобразовании.

ВНИМАНИЕ

Аргумент x не входит в параметры встроенных функций, так как по умолчанию x берется через равные интервалы в возрастающем порядке.

Перечислим функции дискретного преобразования Фурье:

- `fft(y)` возвращает коэффициенты ряда Фурье вектора y , содержащего 2^m элементов. Результат — комплекснозначный вектор размерности $1+2^{m-1}$. Элементы вектора коэффициентов, возвращаемого `fft`, вычисляются по формуле

$$c_j = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} y_k e^{2\pi i(j/n)k}.$$

В этой формуле n — число элементов вектора y ; i — мнимая единица. Если частота измерения исходного сигнала f_s , то частота, соответствующая c_k , $f_k = \frac{k}{n} f_s$.

- `ifft(c)` возвращает обратное дискретное преобразование Фурье. Результат — вещественный. Вектор c должен иметь $1+2^m$ элементов, где m — целое. Функция возвращает вектор d , чьи элементы вычисляются по формуле

$$d_j = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} \omega_k e^{-2\pi i(j/n)k},$$

где ω — вектор, комплексно-сопряженный y . Это та же самая формула, что и для `fft`, за исключением знака «минус» в показателе степени. Функции `fft` и `ifft` — точные обращения. Для всех вещественных y справедливо `ifft(fft(y)) = y`.

- `cfft(y)` возвращает коэффициенты ряда Фурье вектора или массива, содержащих как вещественные, так и комплексные числа. Число элементов вектора y может быть любым. Возвращаемый массив имеет тот же размер, что и массив, используемый как аргумент.
- `icfft(c)` возвращает обратное преобразование Фурье вектора или матрицы данных.

Функции `cfft` и `icfft` — обратные друг к другу. То есть `icfft(cfft(y)) = y`.

Рассмотренные преобразования Фурье не являются единственно возможными. Широко распространен следующий вид формул преобразования Фурье:

$$c_j = \frac{1}{n} \sum_{k=0}^{n-1} y_k e^{-2\pi i(j/n)k} \text{ и } d_j = \sum_{k=0}^{n-1} w_k e^{2\pi i(j/n)k}.$$

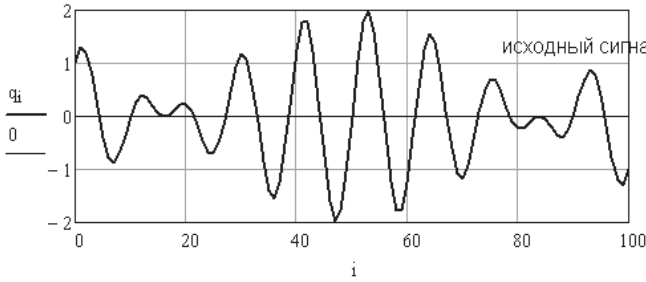
Эти формулы используются функциями FFT и IFFT.

Функции FFT, IFFT, CFFT и ICFFT используются аналогично функциям fft, ifft, cfft, icfft.

В рассмотренном на рис. 7.22 примере экспериментальная функция создана путем наложения на произвольную аналитическую функцию (sin + cos) случайных чисел, созданных функцией rnd(n), возвращающей случайное число, лежащее между 0 и n. Функция cfft возвращает коэффициенты ряда Фурье для созданного массива чисел. Из набора найденных коэффициентов можно выбрать ряд коэффициентов, удовлетворяющих заданному условию $k_1 \leq [f_i] \leq k_2$.

Зададим сигнал `n := 100 i := 0..n`

`qi := sin(i/100 * 16 * pi) + cos(i/100 * 19 * pi)` Поменяйте аргументы sin и cos и число точек n



Добавим помехи `si := qi + rnd(2) - 1`

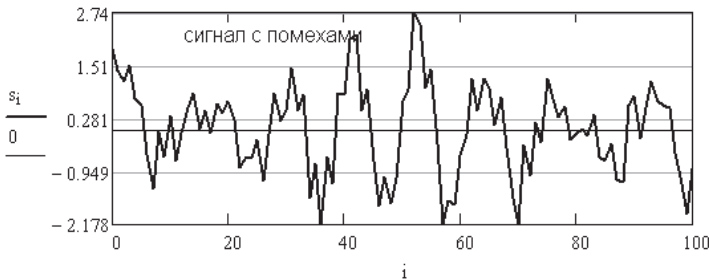


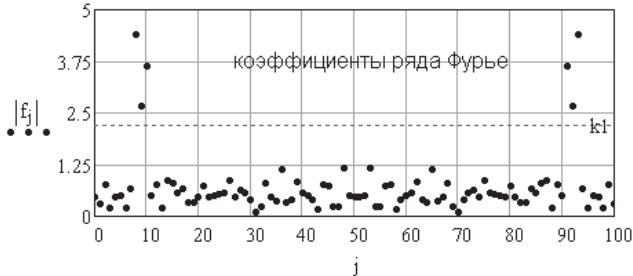
Рис. 7.22. Формирование массива зашумленных данных для последующей очистки

Далее с помощью функции icfft находим функцию, представляющую собой сумму синусоид с коэффициентами, выбранными согласно заданному условию. Меняя условие (величины k_1 и k_2), можно включать в создаваемую функцию только необходимые гармоники (рис. 7.23). Поработайте с интересным примером.

прямое преобразование $f := \text{cfft}(s)$

построить ряд Фурье, значения коэффициентов которого лежат между $k1$ и $k2$

$k1 := 2.2$ $k2 := 7$ поменяйте $k1$ и $k2$ $j := 0..n$
 Поменяйте cfft на CFFT и значения $k1$



выбор коэффициентов

обратное преобразование

$g := \text{if}(k1 \leq |f_j| \leq k2, f_j, 0)$ $h := \text{icfft}(g)$

Поменяйте cfft и icfft на CFFT и ICFFT

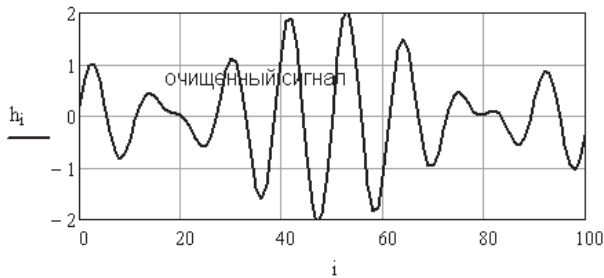


Рис. 7.23. Фильтрация данных с использованием разложения в ряд Фурье

Поменяйте саму функцию, условие, число точек. Функция cfft позволяет работать с массивом из любого числа точек.

Разложение функции в ряд Фурье использовано в примере 4 электронной книги для определения параметров движения кулачкового механизма.

7.4.2. Двухмерное преобразование Фурье

В Mathcad имеется возможность применения встроенных функций комплексного преобразования Фурье не только к одномерным, но и к двумерным массивам.

На рис. 7.24 приведена ступенчатая функция $Y(x, y)$ и показана ее поверхность. Обратите внимание на запись условия при создании ступенчатой функции. Условие $C(x, y)$ записано в виде произведения булевых операторов. $C(x, y) = 1$, если условие выполняется, и $C(x, y) = 0$, если условие не выполняется. По функции $Y(x, y)$ сформирован массив данных YU . Работая с примером, поменяйте исходные данные.

С помощью генератора случайных чисел rnd на массив YU наложены помехи.

Сформируем массив значений сигнала YY
на области значений аргументов α, β

```
A := 3    амплитуда сигнала    n := 30
i := 0..n    j := 0..n     $\alpha_i := 7 \cdot \frac{i}{30}$      $\beta_j := 7 \cdot \frac{j}{30}$  область аргументов
a := 2    b := 4    c := 3    d := 5 границы области сигнала
C(x,y) := (a ≤ x ≤ b) · (c ≤ y ≤ d) область сигнала
 $Y(x,y) := \begin{cases} A & \text{if } C(x,y) \\ 0 & \text{otherwise} \end{cases}$      $YY_{i,j} := Y(\alpha_i, \beta_j)$  Исходный сигнал
```

Добавить помехи $YR_{i,j} := YY_{i,j} + \text{rnd}(1) - 0.5$

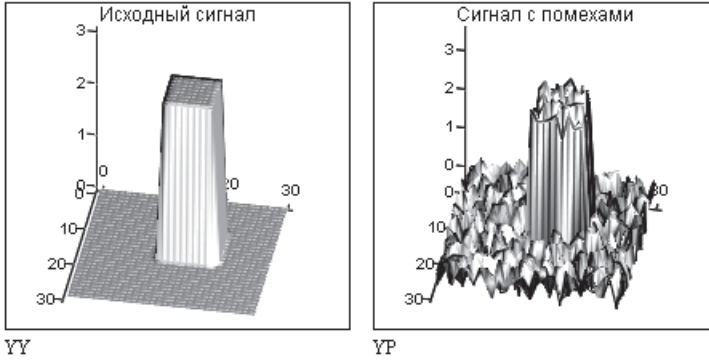


Рис. 7.24. Наложение помех на двумерный массив

На рис. 7.25 показано дискретное преобразование Фурье функции YR с помехами. Слева показан массив коэффициентов, справа — результат обратного преобразования Фурье.

Выбираем ряд значений коэффициентов ряда Фурье между $k1$ и $k2$

```
K := cfft(YR)    k1 := 1    k2 := 15    поменяйте k1 и k2    Поменяйте cfft на CFFT
 $g_{i,j} := \text{if}(|k1 \leq |K_{i,j}| \leq k2, K_{i,j}, 0)$     YA := icfft(g)    и icfft на ICFFT
```

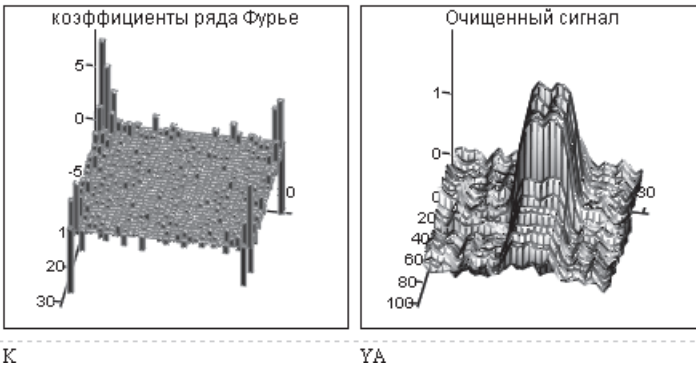


Рис. 7.25. Двухмерное преобразование Фурье массива данных с помехами (см. рис. 7.24)

Для обратного преобразования использованы только гармоники, у которых коэффициенты лежат внутри выбранного диапазона значений g .

7.4.3. Волновое преобразование

Волновое (вейвлетное) преобразование в ряде случаев оказывается более эффективным, чем преобразование Фурье. Преобразование Фурье раскладывает заданную функцию на ряд гармоник (бесконечно осциллирующих синусоид). При волновом преобразовании базовые функции локализованы в некоторой ограниченной области значений аргумента, а вдали от нее — малы или равны нулю.

Mathcad имеет одну встроенную функцию для волновых преобразований, использующую функцию Даубечи:

- $wave(y)$ — вектор прямого волнового преобразования;
- $iwave(v)$ — вектор обратного волнового преобразования.

Здесь y — вектор данных, взятых через равные промежутки значений аргумента; v — вектор данных вейвлет-спектра (коэффициентов волнового преобразования).

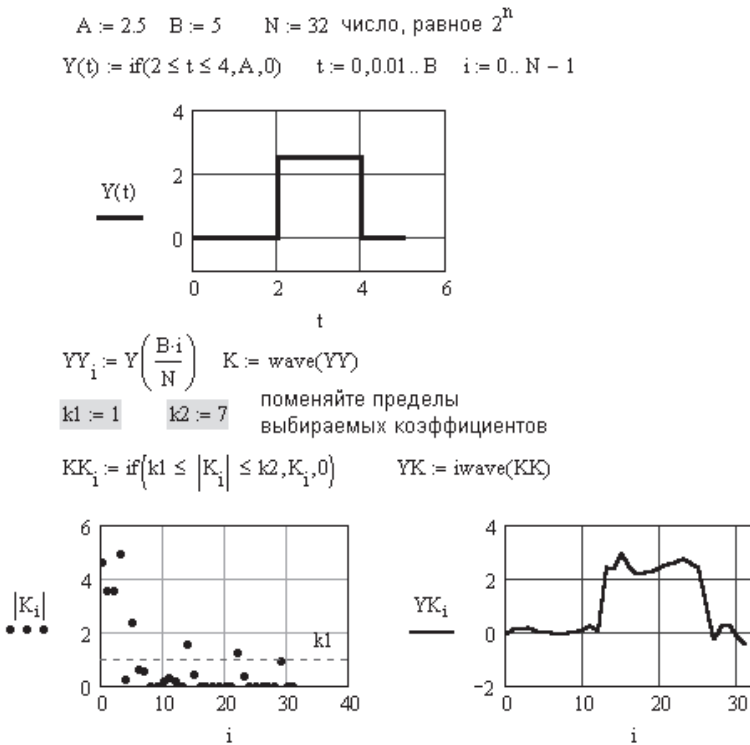


Рис. 7.26. Пример волнового преобразования данных

Вектор y должен содержать 2^n элементов (n — целое число). Результатом функции `wave` является вектор, содержащий несколько коэффициентов двухпараметрического вейвлет-спектра. Пример волнового преобразования приведен на рис. 7.26. Выполняемые действия, включая фильтрацию коэффициентов, аналогичны действиям по преобразованию Фурье, показанному на рис. 7.22–7.25.

ПРИМЕЧАНИЕ

В Mathcad 11 и более поздние версии включена электронная книга, посвященная волновым преобразованиям. Она содержит дополнительные функции обработки сигналов и много примеров волновых преобразований. Правда, воспользоваться сразу этими функциями нельзя — необходимо приобрести полный пакет на сайте фирмы Mathsoft.

Глава 8

Математическая статистика

Mathcad имеет большой набор функций математической статистики, позволяющих вычислять характеристики выборки данных: средние величины, дисперсию, коэффициенты корреляции и другие коэффициенты, вычислять плотности вероятности, функции вероятности, квантили вероятности для 17 различных видов распределения случайных величин. Кроме того, в нем есть генераторы случайных чисел, соответствующие всем 17 видам распределения, позволяющие эффективно выполнять моделирование методом Монте-Карло.

8.1. Характеристики выборки данных и связи двух массивов

Mathcad содержит 16 различных функций для оценки параметров выборки данных. Перечислим основные из них:

- $\text{mean}(A)$ возвращает среднее значение элементов массива A ;
- $\text{hmean}(x)$ возвращает среднее гармоническое значение элементов массива A ;
- $\text{gmean}(A)$ возвращает среднее геометрическое значение элементов массива A ;
- $\text{var}(A)$ возвращает дисперсию элементов массива A ;
- $\text{Var}(A)$ возвращает несмещенную дисперсию элементов массива A ;
- $\text{stdev}(A)$ возвращает среднее квадратическое отклонение элементов массива A ;
- $\text{Stdev}(A)$ возвращает несмещенное среднее квадратическое отклонение элементов массива A ;
- $\text{median}(A)$ возвращает медиану массива A , которая делит гистограмму плотности вероятностей на две равные части;
- $\text{mode}(A)$ возвращает моду массива A (наиболее часто встречающееся значение выборки данных);
- $\text{skew}(A)$ возвращает асимметрию массива A (степень асимметричности гистограммы плотности вероятности относительно оси, проходящей через ее центр тяжести);
- $\text{kurt}(x)$ возвращает эксцесс массива A (степень сглаженности плотности вероятности в окрестности главного максимума);
- $\text{stderr}(A, B)$ возвращает стандартную ошибку при линейной регрессии массивов A и B ;
- $\text{cvar}(A, B)$ возвращает ковариацию элементов двух массивов, A и B ;
- $\text{corr}(A, B)$ возвращает коэффициент корреляции двух массивов, A и B ;
- $\text{hist}(\text{int}, y)$ — функция построения гистограммы массива A ;
- $\text{histogram}(n, y)$ — функция построения гистограммы массива A .

Формулы для расчета указанных характеристик приведены на рис. 8.1 и 8.2.

$$\begin{aligned} \text{ORIGIN} &:= 1 \\ \text{Векторы наблюдений } K &:= (25 \ 33 \ 33 \ 30 \ 34 \ 37) \quad K := K^T \quad n := \text{rows}(K) \\ \text{среднее арифметическое } \text{mean}(K) &= 32 \quad \frac{1}{n} \cdot \sum_{i=1}^n K_i = 32 \\ m &:= \text{mean}(K) \\ \text{среднее геометрическое } \text{gmean}(K) &= 31.766 \quad \sqrt[n]{\prod_{i=1}^n K_i} = 31.766 \\ \text{среднее гармоническое } \text{hmean}(K) &= 31.516 \quad \left(\frac{1}{n} \cdot \sum_{i=1}^n \frac{1}{K_i} \right)^{-1} = 31.516 \end{aligned}$$

Рис. 8.1. Оценка параметров выборки данных

$$\begin{aligned} \text{дисперсия} \\ \text{смещенная оценка } \text{var}(K) &= 14 \quad \frac{1}{n} \cdot \sum_{i=1}^n (K_i - m)^2 = 14 \\ \text{несмещенная оценка } \text{Var}(K) &= 16.8 \quad \frac{1}{n-1} \cdot \sum_{i=1}^n (K_i - m)^2 = 16.8 \\ \text{среднее квадратическое отклонение} \\ \text{смещенная оценка } \text{stdev}(K) &= 3.742 \quad \sqrt{\text{var}(K)} = 3.742 \\ \text{несмещенная оценка } \text{Stdev}(K) &= 4.099 \quad \sqrt{\text{Var}(K)} = 4.099 \\ \text{медиана } \text{median}(K) &= 33 \\ \text{мода } \text{mode}(K) &= 33 \\ \text{Эксцесс } \text{kurt}(K) &= 1.339 \\ \text{асимметрия } \text{skew}(K) &= -0.941 \end{aligned}$$

Рис. 8.2. Оценка параметров выборки данных (продолжение)

Все приведенные статистические функции могут использоваться для работы как с векторами, так и с матрицами. При этом статистические характеристики рассчитываются для совокупности значений всех элементов матрицы, без разделения ее на строки и столбцы. Так, для матрицы размерностью $m \times n$ объем выборки равен $m \times n$.

$$\begin{aligned} \text{Массивы наблюдений} \\ K &:= (25 \ 33 \ 33 \ 30 \ 34 \ 37) \quad K := K^T \quad M := \begin{pmatrix} 16 & 22 \\ 31 & 9 \\ 24 & 17 \end{pmatrix} \\ A &:= (10 \ 15 \ 13 \ 20 \ 22 \ 27) \\ \text{mean}(K, A, M) &= 23.222 \quad \text{mean}(M, A, K) = 23.222 \\ \text{median}(K, A, M) &= 23 \quad \text{stdev}(A, M, K) = 8.417 \end{aligned}$$

Рис. 8.3. Оценка параметров нескольких массивов данных

В качестве аргументов функций можно указать любое количество векторов, матриц и скаляров. Пример вычисления статистических характеристик для нескольких массивов приведен на рис. 8.3. Порядок указания массивов не имеет значения.

В Mathcad имеется три функции для оценки связи двух векторов или матриц:

- `stderr(A, B)` возвращает стандартную ошибку при линейной регрессии массивов A и B ;
- `cvar(A, B)` возвращает ковариацию элементов массивов A и B ;
- `corr(A, B)` возвращает коэффициент корреляции массивов A и B .

Коэффициент корреляции и ковариация различаются лишь нормировкой.

Ковариация определяется по формуле

$$C = \frac{1}{n} \sum_{i=1}^n (A_i - A_{\text{cp}})(B_i - B_{\text{cp}}),$$

коэффициент корреляции — по формуле

$$R = \frac{C}{\sigma_A \sigma_B},$$

где σ_A и σ_B — средние квадратические отклонения массивов A и B .

Чаще на практике используется коэффициент корреляции, дающий относительную, а не абсолютную (как ковариация) оценку связи двух массивов. Чем ближе к единице коэффициент корреляции, тем теснее связь. Пример расчета параметров связи массивов приведен на рис. 8.4.

Векторы наблюдений	$K := (25 \ 33 \ 33 \ 30 \ 34 \ 37)$	$K := K^T$
	$A := (10 \ 15 \ 13 \ 20 \ 22 \ 27)$	$A := A^T$
Оценка связи двух массивов		
ковариация	$cvar(K, A) = 16.167$	
коэффициент корреляции	$corr(K, A) = 0.751$	
стандартная ошибка	$stderr(K, A) = 4.659$	

Рис. 8.4. Оценка связи векторов A и K

8.2. Функции распределения вероятностей

В Mathcad имеются встроенные функции для оценки 17 видов распределения случайных величин. Эти функции рассчитывают плотность вероятности, функцию распределения, квантиль вероятности, генерируют вектор случайных чисел, распределенных по любому из 17 видов распределения.

Функции различаются написанием первой буквы, а другая часть имени функции указывает выбранный вид распределения (в списке функций эта часть обозначена звездочкой):

- $d^*(x, \text{par})$ — плотность вероятности;
- $p^*(x, \text{par})$ — функция распределения;
- $q^*(P, \text{par})$ — квантиль распределения;
- $r^*(M, \text{par})$ — вектор из M независимых чисел, распределенных по выбранному закону.

Здесь x — значение случайной величины; P — значение вероятности; par — список параметров распределения.

Чтобы получить функции, относящиеся, например, к нормальному распределению, надо вместо звездочки поставить `norm`. Список параметров для нормального распределения содержит две величины: μ — математическое ожидание и σ — среднее квадратическое отклонение. Указанные функции приобретают следующий вид: `dnorm(x,μ,σ)`, `pnorm(x,μ,σ)`, `qnorm(P,μ,σ)`, `rnorm(M,μ,σ)`.

Перечислим все типы распределения, использованные в Mathcad, вместе с их параметрами, обозначив звездочкой первую букву встроенных функций:

- $*\beta(x, s_1, s_2)$ — бета-распределение (x — значение случайной величины $0 < x < 1$, $s_1, s_2 > 0$ — параметры);
- $*\text{binom}(k, n, p)$ — биномиальное распределение (n — целый параметр, $0 < k < n$, $0 < p < 1$ — вероятность успеха единичного испытания);
- $*\text{cauchy}(x, l, s)$ — распределение Коши (l — параметр разложения, $s > 0$ — параметр масштаба);
- $*\text{chisq}(x, d)$ — χ^2 («хи-квадрат»)-распределение ($d > 0$ — число степеней свободы);
- $*\text{exp}(x, r)$ — экспоненциальное распределение ($r > 0$ — показатель экспоненты);
- $*\text{F}(x, d_1, d_2)$ — распределение Фишера ($d_1, d_2 > 0$ — числа степеней свободы);
- $*\gamma(k, p)$ — гамма-распределение ($s > 0$ — параметр формы);
- $*\text{geom}(k, p)$ — геометрическое распределение ($0 \leq p \leq 1$ — параметр, равный вероятности успеха единичного испытания, $k > 0$ — целое число);
- $*\text{hypergeom}(k, a, b, n)$ — гипергеометрическое распределение (a, b, n — целые числа);
- $*\text{lnorm}(x, \mu, \sigma)$ — логарифмически нормальное распределение (μ — натуральный логарифм математического ожидания, $\sigma > 0$ — натуральный логарифм среднего квадратического отклонения);
- $*\text{logis}(x, l, s)$ — логистическое распределение (l — математическое ожидание, $s > 0$ — параметр масштаба);
- $*\text{nbinom}(k, n, p)$ — отрицательное биномиальное распределение ($n > 0$ — целое число, $0 < p \leq 1$);
- $*\text{norm}(x, \mu, \sigma)$ — нормальное распределение (μ — математическое ожидание, $\sigma > 0$ — среднее квадратическое отклонение);
- $*\text{pois}(k, \lambda)$ — распределение Пуассона ($\lambda > 0$ — параметр);
- $*\text{t}(x, d)$ — распределение Стьюдента ($d > 0$ — число степеней свободы);

- * $\text{unif}(x, a, b)$ — равномерное распределение ($a < b$ — границы интервала);
- * $\text{weibull}(x, s)$ — распределение Вейбулла ($s > 0$ — параметр).

Примеры определения плотности вероятности, функции распределения и квантилей распределения приведены на рис. 8.5 и 8.6.

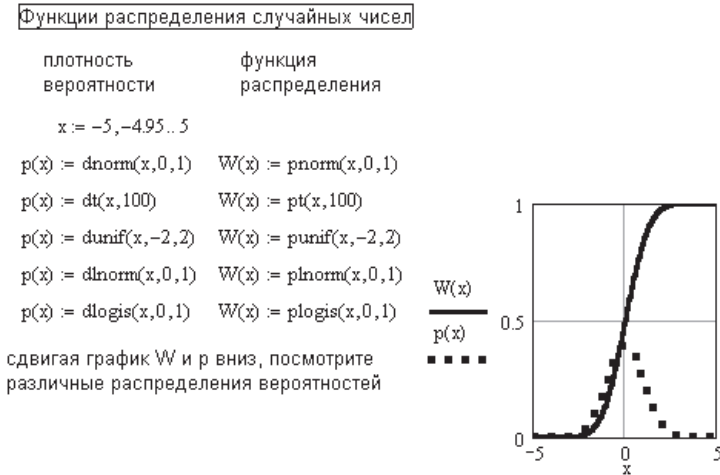


Рис. 8.5. Плотность вероятности p и функция распределения вероятностей W

Обращение функций распределения (квантиль вероятности)

$$\begin{aligned}
 q(p) &:= \text{qnorm}(p, 0, 1) & q(p) &:= \text{qpois}(p, 1) \\
 q(p) &:= \text{qt}(p, \text{qt}(p, 10)) & q(p) &:= \text{qlnorm}(p, 0, 1) \\
 p &:= 0, 0.05 \dots 1 & & \text{Сдвиньте вниз одно из выражений } q(p)
 \end{aligned}$$

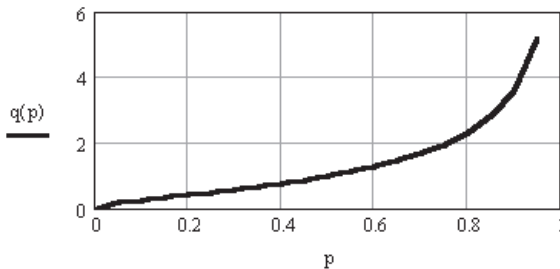


Рис. 8.6. График определения квантиля вероятности q

8.3. Генераторы случайных чисел

Для каждого закона распределения в Mathcad имеется функция — генератор случайных чисел, что позволяет выполнять моделирование методами Монте-Карло. Простейшая из них, функция $\text{rnd}(N)$, возвращает одно случайное число, лежащее в интервале от 0 до N . Все другие функции — генераторы случайных чисел создают вектор чисел, распределенных по выбранному закону. Например, $\text{runif}(M, a, b)$ создает вектор из M чисел, равномерно распределенных в интервале от a до b . Примеры работы генератора случайных чисел приведены на рис. 8.7. и 8.8.

создать одно случайное число, равномерно распределенное между 0 и x
 (сотрите знак равенства и введите его вновь-число поменяется) $rnd(x)$
 $rnd(2) = 2.537 \times 10^{-3}$ одно число $i := 1..4$ $V_i := rnd(10)$ вектор
 $V := morm(5, 10, 2)$ $V := rchisq(5, 10)$ $V := rexp(5, 0.3)$ $V := rgeom(5, 0.2)$

Опустите вниз одно из выражений V
 $V := sort(V)$ $V := reverse(V)$ сортировка вектора по росту и обратно
 $(V) = \begin{pmatrix} 8 \\ 7 \\ 3 \\ 0 \\ 0 \end{pmatrix}$ среднее арифметическое $mean(V) = 3.6$
 дисперсия $var(V) = 11.44$
 среднее квадратическое отклонение $stdev(V) = 3.382$
 медиана $median(V) = 3$

Рис. 8.7. Работа генератора случайных чисел

Формирование массива случайных чисел
 Для всех функций, кроме rnd $N := 4$ $a := 1$ $b := 4$ $k := 1..N$
 $\varepsilon^{(k)} := runif(3, a, b)$
 поменяйте генератор чисел $\varepsilon = \begin{pmatrix} 1.79 & 3.17 & 3.04 & 1.36 \\ 2.84 & 3.45 & 1.85 & 2.96 \\ 1.08 & 3.75 & 1.98 & 3.77 \end{pmatrix}$
 Для функции rnd $n := 3$ $m := 4$ $i := 1..n$ $j := 1..m$
 $\Delta_{i,j} := rnd(5)$ $\Delta = \begin{pmatrix} 0.213 & 1.877 & 2.44 & 3.016 \\ 0.142 & 2.236 & 2.86 & 0.38 \\ 3.063 & 3.484 & 4.398 & 3.256 \end{pmatrix}$

Рис. 8.8. Работа генератора случайных чисел (продолжение)

8.4. Построение гистограмм

Гистограммой называется график, аппроксимирующий по случайным данным плотность их распределения. При построении гистограммы область значений случайной величины разбивается на заданное число сегментов, а затем подсчитывается число попаданий в каждый сегмент.

В Mathcad встроено две функции построения гистограмм:

- ❑ **histogram(n , data).** Здесь n — число интервалов, на которые разбивается весь диапазон исходных данных **data**. Эта функция возвращает 2 столбца. В первом содержатся средние точки каждого из n интервалов, во втором — частоты попадания случайных чисел вектора **data** в каждый из n интервалов. Пример построения гистограммы приведен на рис. 8.9. Массивы исходных данных V из 1000 чисел на этом рисунке создаются генераторами случайных чисел, соответствующих выбранному закону распределения. Число интервалов nn выбрано 21.
- ❑ **hist(int, data).** Здесь **int** имеет двоякий смысл: либо это вектор середин интервалов (можно задать интервалы разной ширины), либо число интервалов. **data** — вектор случайных чисел. Пример использования функции **hist** приведен на рис. 8.10.

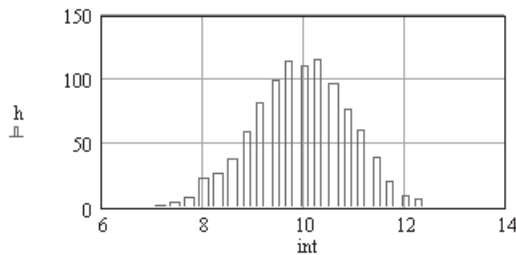
нормальное распределение	$V := \text{norm}(1000,10,1)$
распределение Стьюдента	$V1 := \text{rt}(1000,100)$
равномерное распределение	$V2 := \text{runif}(1000,-2,2)$
логарифмически нормальное	$V3 := \text{rlnorm}(1000,0,1)$
логистическое распределение	$V4 := \text{rllogis}(1000,0,1)$

построение гистограммы

$nn := 21$ $hh := \text{histogram}(nn, V)$

$int := hh^{(1)}$ вектор средних точек интервалов

$h := hh^{(2)}$ частота попадания чисел в интервал
(в функции `histogram` поменяйте имя массива данных V)

**Рис. 8.9.** Применение функции `histogram`

$V := \text{rpois}(1000,10)$ или $V := \text{norm}(1000,10,1)$

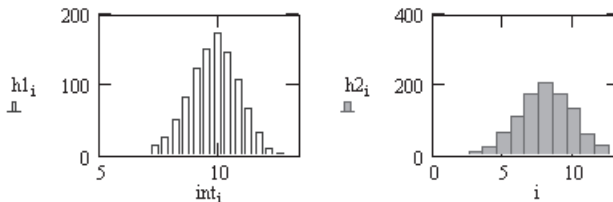
$int := 0$ сброс предыдущих значений int $n := 15$ $i := 1..n$

$max := \text{max}(V)$ $min := \text{min}(V)$ $int_i := \frac{max - min}{n} \cdot i + min$

$max = 13.429$ $min = 6.883$

$h1 := \text{hist}(int, V)$ int - вектор середин интервалов

$h2 := \text{hist}(n, V)$ n - число интервалов

**Рис. 8.10.** Применение функции `hist`

Для закрашивания гистограммы надо дважды щелкнуть мышью на графике и в открывшемся окне форматирования графика (рис. 8.11) выбрать `Traces` ▶ `Type` ▶ `Solidbar` (Трассировка ▶ Тип ▶ Сплошные столбики) или `Traces` ▶ `Type` ▶ `Bar` (Трассировка ▶ Тип ▶ Выбрать тип столбиков).

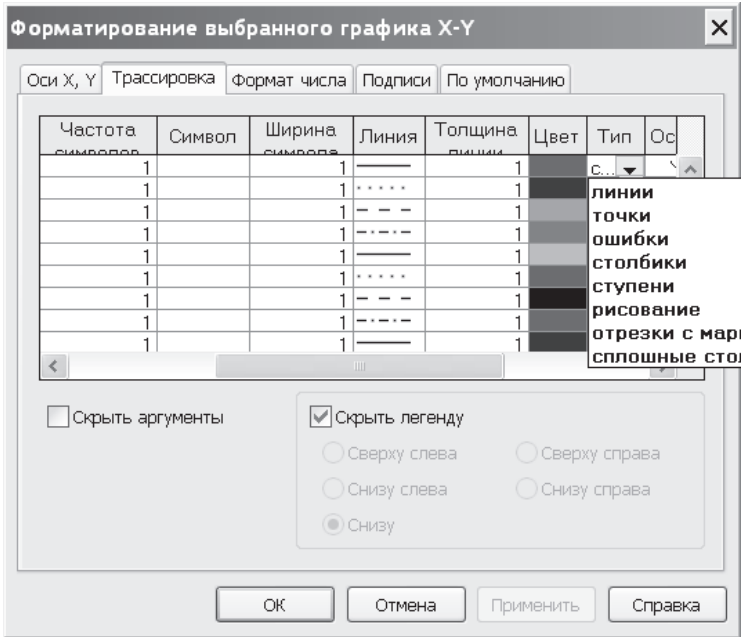


Рис. 8.11. Выбор внешнего вида графиков

8.5. Расчет доверительного интервала

В качестве примера статистических расчетов рассмотрим расчет доверительного интервала математического ожидания для малой выборки K из 6 чисел (рис. 8.12). Для расчета использовано распределение Стьюдента. Квантиль распределения определяет ширину доверительного интервала. Он показывает, сколько значений среднего квадратического отклонения надо отложить влево и вправо от среднего арифметического значения. Доверительный интервал определен для вероятности попадания числа внутрь интервала 99 %.

В технике существует понятие «расчет на три девятки». Смысл его в том, что при расчете с вероятностью неразрушения 99,9 % надо отложить от средней долговечности в сторону ее уменьшения «три сигма» (три средних квадратических отклонения), потому что квантиль нормального распределения примерно равен 3 при вероятности 99,9 %.

Расчет доверительного интервала

$$K := (25 \ 33 \ 33 \ 30 \ 34 \ 36) \quad K := K^T \quad K_{cp} := \text{mean}(K) \quad K_{cp} = 31.833$$

по Стьюденту квантиль $t := \text{qt}(0.99, n) \quad t = 2.602$

отклонение от среднего арифметического $di := t \cdot \text{stdev}(K) \quad di = 9.191$

Доверительный интервал $K_{cp} - di < KK < K_{cp} + di$

$$31.833 - 9.191 < KK < 31.833 + 9.191$$

Рис. 8.12. Оценка доверительного интервала для малой выборки данных

Часть III

Дополнительные возможности Mathcad

В этой части книги рассматриваются дополнительные возможности Mathcad: программирование, анимация, учет размерностей, подготовка документов.

Глава 9

Программирование

Раздел «Программирование» занимает особое место в Mathcad. При начальном обучении этот раздел совершенно не нужен. Огромные возможности Mathcad позволяют решать подавляющее число задач без использования программирования, да к тому же, как правило, несколькими способами. Но есть класс задач, которые невозможно решить без программирования. Это задачи, в которых часть документа из нескольких или многих операторов надо выполнить многократно. В таких случаях документ должен состоять из отдельных подпрограмм, объединенных в единую «головную» программу. И в этом случае программирования можно было бы избежать, если бы создатели Mathcad предусмотрели оператор перехода к метке, который есть (был) во многих языках программирования, в частности в Fortran. Серьезные программисты считают дурным тоном использование перехода к метке, полагая, что в любой программе можно обойтись без меток. Это верно, но зачастую требует больших усилий. Система Mathcad предназначена для простых инженеров, не всегда являющихся специалистами в программировании, и поэтому должна быть простой в использовании.

Использование раздела «Программирование» позволяет записать в Mathcad программы любой сложности. Образцом такой программы является решение плоской упруго-пластической задачи методом конечных элементов, приведенной в примере 55 электронной книги.

Программирование в Mathcad чрезвычайно наглядно и понятно, так как программа представляет собой последовательность формул. Далее программами будем называть не все обычные Mathcad-документы, а только те из них, которые написаны с использованием панели программирования.

Основные операторы программирования расположены на панели Programming Toolbar (Панель программирования), вызываемой соответствующей кнопкой математической панели. Прибегать к помощи панели программирования придется в основном при неоднократном повторении каких-либо частей документа.

9.1. Создание программ

Программа Mathcad есть частный случай выражения Mathcad. Подобно любому выражению, программа возвращает значение, если за ней следует знак равенства. Обычное выражение Mathcad состоит из одной строки. Выражение-программа содержит много строк. Фактически это составное выражение.

Выражение-программа состоит из названия выражения, следующего за ним знака присвоения значения и необходимых выражений в правой части, записанных в столбик и объединенных слева вертикальной чертой.

9.1.1. Ввод строк в программу

Для создания программы надо:

- ввести имя выражения-программы;
- ввести оператор присваивания ($:=$);

- ❑ щелкнуть на кнопке панели программирования Add Line (Добавить строку) столько раз, сколько строк должна содержать программа;
- ❑ в появившиеся места ввода ввести нужные операторы, лишние места ввода удалить.

Чтобы создать недостающие места ввода, надо поставить синий уголок курсора в конец строки, после которой ввести новую строку. Клавишей пробела следует выделить полностью всю строку и нажать кнопку Add Line. При этом возможны два варианта:

- ❑ Если синий уголок курсора находится в начале строки, то после нажатия Add Line место ввода появится выше этой строки.
- ❑ Если синий уголок курсора находится в конце строки, то после нажатия Add Line место ввода появится ниже этой строки (рис. 9.1).

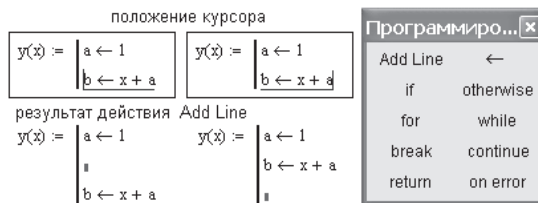


Рис. 9.1. Ввод строки в программу

В некоторых случаях, например, при необходимости вставить строку между двумя вложенными циклами, что при создании программы встречается очень часто, указанный прием не работает. В этом случае необходимо:

- ❑ выделить черным цветом весь внутренний цикл;
- ❑ вырезать его в буфер кнопкой Cut (Вырезать);
- ❑ нажать кнопку Add Line;
- ❑ вставить из буфера вырезанный фрагмент на прежнее место;
- ❑ заполнить появившееся дополнительное место ввода.

При определенном навыке этот прием удобно использовать во всех случаях вставки дополнительных строк.

9.1.2. Локальное присвоение значений

Программа Mathcad — это обособленный модуль внутри Mathcad-документа со своими именами переменных и их значениями, которые не видны из других мест Mathcad-документа.

Присваивание значений переменным и константам в программах производится с помощью программного оператора присваивания \leftarrow , который вводится с панели

программирования нажатием кнопки \leftarrow . При создании программы, когда этот знак приходится использовать часто, полезно пользоваться клавишей $\{$.

Локальные значения переменных нельзя использовать вне программы. Если это все же необходимо, их надо вывести из программы специальным образом (см. раздел 9.4, «Вывод результатов расчета из программы»).

ВНИМАНИЕ

Ни оператор присвоения значения ($:=$), ни оператор вывода ($=$) в пределах программ не применяются.

Из основного документа в программу передаются в виде констант все значения величин, имеющих одинаковые имена в основном документе и в программном модуле. Эти значения сохраняются неизменными при любых действиях с ними внутри программы.

Для того чтобы любые величины стали переменными, им надо присвоить значения внутри программы с помощью программного оператора присваивания. На рис. 9.7 первый оператор в программе $A \leftarrow A$ замените на $AA \leftarrow A$ или удалите его. Результат изменится.

В большой программе, например, при решении упруго-пластической задачи методом конечных элементов (МКЭ) (пример 55, часть IV), где в головную программу ZZ надо передать много переменных величин, всем им в начале программы надо присвоить значения. Обратите внимание на то, что имена передаваемых и принимаемых величин чаще всего совпадают, чтобы программу было легче понять. В указанном примере передается 6 величин, и каждая занимает одну строку в программе, что увеличивает длину программы.

Передать нужные величины в программу можно и другим способом, создав программу-функцию, где передаваемые величины записаны в параметры функции. Так в рассмотренном примере оформлены все подпрограммы, используемые затем в головной программе. При этом программа занимает меньше места в длину, но больше — в ширину.

Для экономии места в документе можно использовать наложение объектов друг на друга. Щелкните правой кнопкой мыши и выберите в контекстном меню Bring to Front (Выдвинуть на передний план) или Send to Back (Убрать на задний план). Документ получается более компактным. Именно так скомпонованы все программы в этой книге.

ПРИМЕЧАНИЕ

Оператор программирования в Mathcad 13 не допускает наличия одинаковых имен в левой и правой частях оператора локального присвоения \leftarrow . Например, массиву нельзя присвоить имя функции.

Любая программа представляет собой сочетание обычных математических выражений с операторами условия и цикла. Разберем последовательно действие этих операторов.

9.2. Условный оператор if

Действие условного оператора состоит из двух частей. Сначала проверяется условие, записанное справа от оператора if. Если оно верно, выполняется выражение, стоящее слева от if, если не верно, происходит переход к следующей строке программы.

Чтобы вставить условный оператор в программу, нужно:

- ❑ в создаваемой программе установить курсор на свободное место ввода, где должен появиться условный оператор;
- ❑ на панели программирования (Programming Toolbar) щелкнуть на кнопке if. В программе появится шаблон оператора с двумя местами ввода;
- ❑ в правое место ввода ввести условие. Пользуйтесь при этом логическими операторами, вводя их с панели Boolean (Булевы операторы);
- ❑ слева от оператора if ввести выражение, которое должно выполняться, если условие верно;
- ❑ если при выполнении условия должно выполняться сразу несколько выражений, надо иметь несколько мест ввода. Установите курсор на место ввода слева от if и нажмите Add Line столько раз, сколько строк надо ввести. Обратите внимание на то, что при этом изменяется вид условного оператора. Столбик мест ввода появляется не слева, а под оператором if (рис. 9.2).

использование оператора программирования if

a := 8 поменяйте a

```
y := if a < 3
      | b ← √a
      | ln(b)
      | 2 if 3 ≤ a ≤ 5
      | otherwise
      | c ← a + 1
      | c2 - a
```

использование булевых операторов

$$y1 := [\ln(\sqrt{a}) \cdot (a < 3) + 2 \cdot (3 \leq a \leq 5) + [(a + 1)^2 - a] \cdot (a > 5)]$$

использование функции if

$$y2 := \text{if}[a < 3, \ln(\sqrt{a}), \text{if}[3 \leq a \leq 5, 2, (a + 1)^2 - a]]$$

y = 73 y1 = 73 y2 = 73

Рис. 9.2. Три способа записи условия в Mathcad

Если условие не верно, происходит переход к следующей строке программы. Она может содержать новое условие или быть обычным выражением.

Часто встречается условие с двумя вариантами действия: если ..., то ..., иначе Для слова «иначе» на панели Programming имеется оператор otherwise (Иначе), который вводится так же, как if.

На практике лучше вводить условие в несколько ином порядке:

- ❑ На свободное место ввода в программе вводят выражение, выполняемое, если условие верно.

- ❑ Выделяют курсором выражение так, чтобы уголок курсора был в конце выражения, и на панели программирования нажимают if.
- ❑ Справа от оператора if вводят условие.
- ❑ На следующем свободном месте ввода (на следующей строке) пишут выражение, выполняемое «иначе», если условие не выполняется.
- ❑ Выделяют курсором выражение так, чтобы уголок курсора был в конце выражения, и на панели программирования нажимают otherwise.

На рис. 9.3 приведен пример использования условного оператора if. В качестве примера выбрано построение эпюр изгибающих моментов и поперечных сил в балке (рис. 9.4). Применение условного оператора особенно удобно при выполнении операции со ступенчатыми функциями. Участков на балке может быть несколько, но функция изгибающих моментов одна.

Использование оператора if для описания ступенчатых функций (построение эпюр Q и M при изгибе балки)

$$\begin{aligned}
 & q := 20 \quad P1 := 20 \quad P2 := 30 \quad M0 := 30 \quad a := 3 \quad b := 2 \\
 & Q(x) := \begin{cases} P1 - q \cdot x & \text{if } x < a \\ (P1 + P2 - q \cdot a) & \text{otherwise} \end{cases} \quad x := 0, 0.1 \dots 5 \\
 & MP(x) := \begin{cases} \left[P1 \cdot x - \left(q \cdot \frac{x^2}{2} \right) \right] & \text{if } 0 \leq x < a \\ \left[P1 \cdot x - q \cdot a \cdot \left(x - \frac{a}{2} \right) + P2 \cdot (x - 2) + M0 \right] & \text{otherwise} \end{cases}
 \end{aligned}$$

Рис. 9.3. Использование условного оператора if

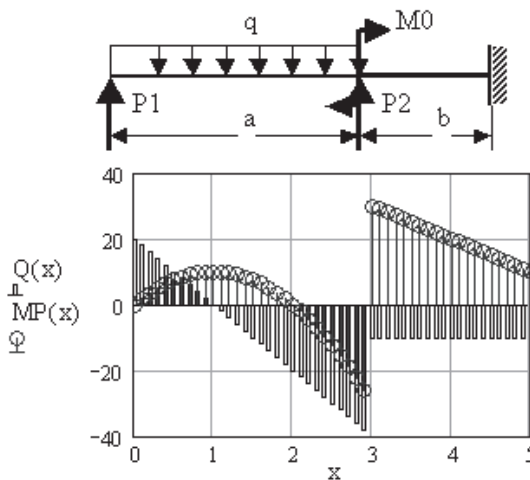


Рис. 9.4. Эпюры поперечной силы и изгибающего момента при изгибе балки (см. рис. 9.3)

Напомним, что в Mathcad есть три способа записи условия:

- с помощью оператора программирования if;
- с помощью булевых операторов;
- с помощью функции if.

Применение всех трех способов записи условия показано на рис. 9.2.

9.3. Операторы цикла

Важнейшим элементом программирования, помимо условного оператора, является оператор цикла. В обычном Mathcad-документе использование дискретной переменной фактически аналогично применению оператора цикла, используемого для вычисления одного выражения. Mathcad вычисляет выражения сверху вниз и переходит к следующему выражению, лишь завершив все вычисления предыдущего выражения, поскольку вернуться к нему уже не сможет. Если же в каждом цикле должно быть вычислено несколько выражений, необходимо составлять программу.

ВНИМАНИЕ

Использование дискретных переменных внутри программ невозможно.

Панель программирования Mathcad содержит два оператора цикла, for и while:

- Если число выполнений цикла заранее известно, используется оператор for.
- Если цикл должен завершиться по выполнении некоторого условия и момент выполнения этого условия неизвестен, используется оператор while.

Рассмотрим каждый из них.

9.3.1. Оператор while

Цикл while выполняется до тех пор, пока остается истинным условие цикла, поэтому нет необходимости знать число вычислений заранее. Важно только, чтобы где-либо внутри цикла или в другом выполняемом участке программы присутствовал оператор, делающий условие цикла ложным. В противном случае цикл будет выполняться бесконечно.

Если выполняемая программа заиклилась, ее можно остановить, нажав клавишу Esc.

Чтобы записать цикл while, выполните следующие действия:

- Установите курсор на свободное место ввода в программе (справа от вертикальной черты).
- На панели программирования нажмите кнопку while. Появится шаблон с двумя местами ввода.
- Справа от слова «while» введите условие выполнения цикла. Обычно это логическое выражение.
- В оставшееся поле ввода (внизу под словом «while») введите выражение, которое вычисляется в цикле.
- Если в цикле надо вычислять несколько выражений, то вначале установите курсор на место ввода и нажмите кнопку Add Line (или клавишу J) столько

раз, сколько строк будет содержать цикл. Затем заполните все места ввода, введя нужные выражения. Удалите лишние места ввода.

Обнаружив заголовок цикла while, Mathcad проверяет условие цикла. Если оно истинно, то Mathcad выполняет тело цикла и снова проверяет условие. Если оно ложно, Mathcad заканчивает выполнение цикла.

Пример использования цикла while приведен на рис. 9.5.

Нахождение первого элемента вектора, большего заданного значения $K := 1.6$

$j := 1..7$ $v_j := \sqrt[3]{j-1}$ $ORIGIN := 1$

$v^T =$	1	2	3	4	5	6	7	
	1	0	1	1.26	1.442	1.587	1.71	1.817

$t :=$ $\begin{cases} j \leftarrow 1 & j - \text{номер элемента} \\ \text{while } |v_j| \leq K \\ j \leftarrow j + 1 \\ \begin{pmatrix} j \\ v_j \end{pmatrix} \end{cases}$ $t = \begin{pmatrix} 6 \\ 1.71 \end{pmatrix}$

Рис. 9.5. Применение оператора цикла while

9.3.2. Оператор for

В цикле for число повторений цикла определяется переменной, задаваемой в начале цикла. Рассмотрим создание такого цикла:

- ❑ Установите курсор на свободное место ввода в программе (справа от вертикальной черты).
- ❑ На панели программирования нажмите кнопку for Loop (Цикл for). Появится шаблон с тремя местами ввода.
- ❑ Справа от слова «for» введите имя переменной цикла. После знака ∈ введите диапазон изменения переменной цикла так же, как это делается с помощью дискретной переменной. Переменной цикла может быть ряд чисел, или вектор, или список скаляров, диапазонов, векторов, разделенных запятой (рис. 9.6).

переменная цикла по типу дискретной переменной

$Z :=$ $\begin{cases} m \leftarrow 1 \\ \text{for } s \in 1,1.2..2 \\ X_m \leftarrow \sqrt{s+1} \\ m \leftarrow m + 1 \\ X \end{cases}$ $Z = \begin{pmatrix} 1.414 \\ 1.483 \\ 1.549 \\ 1.612 \\ 1.673 \\ 1.732 \end{pmatrix}$

переменная цикла - 2 вектора

$A := \begin{pmatrix} 13 \\ 15 \\ 17 \end{pmatrix}$ $B := \begin{pmatrix} 1 \\ 3 \end{pmatrix}$

$Z1 :=$ $\begin{cases} m \leftarrow 1 \\ \text{for } s \in A, B \\ X_m \leftarrow s \\ m \leftarrow m + 1 \\ X \end{cases}$ $Z1 = \begin{pmatrix} 13 \\ 15 \\ 17 \\ 1 \\ 3 \end{pmatrix}$

Рис. 9.6. Применение оператора цикла for

- ❑ В оставшееся поле ввода (внизу, под словом «for») введите выражение, которое вычисляется в цикле.
- ❑ Если в цикле надо вычислять несколько выражений, то вначале установите курсор на место ввода и нажмите кнопку Add Line (или клавишу \downarrow) столько раз, сколько строк будет содержать цикл. Затем заполните все места ввода, введя нужные выражения. Удалите лишние места ввода.

Mathcad легко справляется со сложными математическими проблемами и легко решает сложные уравнения. А вот простые для любого человека логические задачи решаются с большим трудом. На рис. 9.7 – 9.9 показаны примеры использования программирования для решения логических задач выборки нужных значений вектора или матрицы. В этих примерах показаны различные приемы записи условия в операторе if: в виде обычных булевых операторов (рис. 9.9) или в виде логической функции (рис. 9.8). Здесь демонстрируются также приемы формирования массивов выходных данных. При работе с массивами удобно использовать привычную нумерацию элементов массива, начиная с 1, поэтому в примерах на программирование далее используется $ORIGIN = 1$.

```

Вставка массива В в массив А      ORIGIN := 1
A := identity(5)      B := ( 3 2 1 )
                        1 2 3 )
место вставки      ряд r := 3
                   столбец c := 2
C := A ← A
   "Передача массива А в программу"
   for i ∈ 1..rows(B)
     for j ∈ 1..cols(B)
       Ar+i-1,c+j-1 ← Bi,j
   A
A = ( 1 0 0 0 0 )
     0 1 0 0 0
     0 0 1 0 0
     0 0 0 1 0
     0 0 0 0 1 )
C = ( 1 0 0 0 0 )
     0 1 0 0 0
     0 3 2 1 0
     0 1 2 3 0
     0 0 0 0 1 )
    
```

Рис. 9.7. Вставка массива в массив

```

Извлечение элементов вектора D,
удовлетворяющих условию c(x) := 2 ≤ x ≤ 4      ORIGIN := 1
D := mom(8,3,2)      DT = (4.116 2.51 3.179 5.524 1.589 3.004 5.215 4.785)
Извлечение := i ← 1
              for j ∈ 1..rows(D)
                if c{Dj} = 1
                  Ui ← Dj
                  Ai ← j
                  i ← i + 1
              W{1} ← A
              W{2} ← U
              W
логическая функция c(D)
принимает значения
1 - верно, 0 - неверно
c{Dj} = 1 эквивалентно
условию 2 ≤ Dj ≤ 4
E := Извлечение
E = ( 2 2.51 )
     3 3.179 )
A - номер элемента
U - значение элемента,
удовлетворяющего условию
    
```

Рис. 9.8. Извлечение нужных элементов вектора

```

Извлечение элементов массива M,
удовлетворяющих условию 3 < x < 5
ORIGIN := 1

i := 1..4    j := 1..3    Mi,j := rnd(10)  создание массива
                                                случайных чисел

M =  $\begin{pmatrix} 8.76 & 9.559 & 5.393 \\ 4.621 & 8.622 & 7.797 \\ 9.968 & 6.115 & 2.662 \\ 8.401 & 3.759 & 6.772 \end{pmatrix}$ 

Элементы := Z(M)

Элементы =  $\begin{pmatrix} 2 & 4 \\ 1 & 2 \\ 4.621 & 3.759 \end{pmatrix}$    строка
                                                столбец
                                                число

Z(M) :=  $\begin{array}{l} i \leftarrow 1 \\ \text{for } m \in 1.. \text{rows}(M) \\ \quad \text{for } n \in 1.. \text{cols}(M) \\ \quad \quad \text{if } 3 < M_{m,n} < 5 \\ \quad \quad \quad \left. \begin{array}{l} ML_i \leftarrow M_{m,n} \\ L \langle i \rangle \leftarrow \begin{pmatrix} m \\ n \\ ML_i \end{pmatrix} \end{array} \right\} \\ \quad \quad \quad L \\ \quad \quad \quad i \leftarrow i + 1 \end{array}$ 

```

Рис. 9.9. Извлечение нужных элементов матрицы

В примере на рис. 9.7 показан ввод комментария в программу.

Для ввода комментария в программу нужно установить курсор на свободное место ввода в программе и ввести символ " (кавычки). Появится символ введения строковой переменной: две пары кавычек с местом ввода между ними. Введите туда текст комментария.

Mathcad вычисляет выражения слева направо, сверху вниз. Результаты действия программы можно вывести правее и ниже места присвоения значения.

Для экономии места в документе используется наложение одной области на другую, ведь программа занимает прямоугольную область, большая часть которой пустует. Для этого переместите любое выражение в область программы и при выделенном выражении, нажав правую кнопку мыши, в контекстном меню выберите Bring to Front (На передний план). Можно также, выделив программу, выбрать в контекстном меню Send to Back (На задний план). Таким образом, скомпонованы все примеры в части IV книги.

9.3.3. Операторы break, continue, return

Эти операторы используются для управления работой циклов и всей программы в целом:

- continue возвращает расчет к началу цикла;
- break обеспечивает выход из цикла и продолжение работы программы;
- return обеспечивает выход из программы.

Работа этих операторов показана на рис. 9.10. Оператор continue в этом примере создает список нечетных чисел (вектор V), осуществляя возврат в начало цикла, если остаток от деления числа n на 2 (mod(n, 2)) равен 0. Оператор break осуществляет выход из цикла и продолжение работы программы, если n < 0. Оператор

return осуществляет выход из программы, если $n = -10$, с записью введенного комментария (выход из программы).

$f(n) :=$	$s \leftarrow 1$ $a \leftarrow 1$ for $k \in 1..n$ continue if $\text{mod}(k,2) = 0$ return "выход из программы" if $n = -10$ $a \leftarrow a + 1$ $V_a \leftarrow k$ break if $n < 0$ $s \leftarrow s + k$ "Конец цикла" $s \leftarrow s \cdot 1000$ $\begin{pmatrix} V \\ s \end{pmatrix}$	$f(9) = \begin{pmatrix} (6,1) \\ 26000 \end{pmatrix}$ $f(-9) = \begin{pmatrix} (2,1) \\ 1000 \end{pmatrix}$ $f(-9)_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ $f(-10) = \text{"выход из программы"}$	$f(9)_1 = \begin{pmatrix} 0 \\ 1 \\ 3 \\ 5 \\ 7 \\ 9 \end{pmatrix}$
-----------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------

Рис. 9.10. Применение операторов continue, break, return

ВНИМАНИЕ

Для вставки в программу операторов continue, break, return, как и других операторов программирования, можно пользоваться только панелью программирования. Их нельзя набирать с клавиатуры.

9.4. Вывод результатов расчета из программы

Итогом действия программы является результат выполнения последнего оператора программы. Если последним оператором является математическое выражение, то результат его вычисления и будет результатом действия программы. Если надо выводить данные, это делается путем формирования внутри программы массивов и составных массивов.

На рис. 9.5 из программы надо вывести 2 числа: номер первого элемента, большего K , и численное значение этого элемента. Как уже сказано, выражение-программа принимает значение последнего оператора программы.

Чтобы вывести результаты расчета, выполните следующие действия:

- Установите курсор на пустое место ввода последнего оператора.
- Для вывода одного числа введите имя соответствующей переменной (любой, которой присвоено какое-либо значение в программе или в основном документе).

□ Для вывода вектора или матрицы введите их имя, но каждый элемент массива должен быть определен либо в программе, либо в основном документе.

Элементы вектора или матрицы можно ввести их именами (см. рис. 9.5). Для этого, установив курсор на место ввода последнего оператора, комбинацией клавиш **Ctrl+m** надо вызвать шаблон матрицы, ввести число строк и столбцов (в нашем случае 2 строки, 1 столбец), ввести имена элементов, которым присвоены значения.

При разработке серьезных больших программ стоит задача вывода результатов расчета. Одновременно надо вывести и скаляры, и векторы, и матрицы. Но ведь место вывода только одно!

Напомним, что Mathcad присваивает выражению с именем программы значение последнего оператора программы.

Выход из трудного положения дает составной массив, который имеет структуру обычного массива, но его элементы могут быть одновременно числами, векторами или матрицами, совершенно не связанными друг с другом.

Массив Z на рис. 9.11 — это матрица-столбец (вектор), для уменьшения размера программы записанный в транспонированном виде. Он содержит 3 элемента: Z_1 — 4 массива различной размерности, Z_2 — 4 вектора, расположенных бок о бок, Z_3 — скаляр (число итераций). На этом рисунке показано, как, используя верхние и нижние индексы, вывести любое расчетное значение. Поупражняйтесь самостоятельно в выводе расчетных значений. Убедитесь в неограниченных возможностях вывода любых расчетных данных.

$Z := \begin{array}{l} a \leftarrow 1 \\ \text{while } a < 5 \\ \quad \text{for } i \in 1..a \\ \quad \quad V_i \leftarrow a + i \\ \quad \quad \quad \text{for } j \in 1..a \\ \quad \quad \quad \quad M_{i,j} \leftarrow a + i + j \\ \quad \quad MM_a \leftarrow M \\ \quad \quad VV^{(a)} \leftarrow V \\ \quad \quad a \leftarrow a + 1 \\ (MM \ VV \ a)^T \end{array}$	$Z = \begin{pmatrix} (4,1) \\ (4,4) \\ 5 \end{pmatrix}$	<p>Z_1 - 4 массива различной размерности</p> <p>Z_2 - 4 вектора бок о бок</p> <p>Z_3 - число итераций</p>
	$Z_1 = \begin{pmatrix} (1,1) \\ (2,2) \\ (3,3) \\ (4,4) \end{pmatrix}$	$(Z_1)_4 = \begin{pmatrix} 6 & 7 & 8 & 9 \\ 7 & 8 & 9 & 10 \\ 8 & 9 & 10 & 11 \\ 9 & 10 & 11 & 12 \end{pmatrix}$
	$Z_2 = \begin{pmatrix} 2 & 3 & 4 & 5 \\ 0 & 4 & 5 & 6 \\ 0 & 0 & 6 & 7 \\ 0 & 0 & 0 & 8 \end{pmatrix}$	$(Z_2)^{(2)} = \begin{pmatrix} 3 \\ 4 \\ 0 \\ 0 \end{pmatrix}$

константе можно присвоить значение элемента массива

$Z_3 = 5 \quad A := (Z_1)_2 \quad A = \begin{pmatrix} 4 & 5 \\ 5 & 6 \end{pmatrix} \quad B := [(Z_1)_4]_{2,3} \quad B = 9$

Замените матрицу-столбец вывода $(MM \ VV \ a)^T$ на матрицу $\begin{pmatrix} MM & VV \\ a & 1 \end{pmatrix}$

а потом на матрицу-строку $(MM \ VV \ a)$ и самостоятельно выведите результаты расчета

Рис. 9.11. Вывод результатов работы программы с помощью составного массива

Составные массивы не имеют физического и математического смысла, поэтому операции матричного исчисления, за исключением операции транспонирования, к ним применять нельзя.

9.5. Подпрограммы-функции

Любая программа является выражением, состоящим из ряда выражений (объединенных вертикальной чертой), каждое из которых, в свою очередь, может быть программой. Например, если часть программы, состоящая из нескольких операторов, должна выполняться несколько раз внутри программы, ее целесообразно записать в виде подпрограммы-функции и вызывать, меняя внутренние параметры, каждый раз, когда это нужно. Формально подпрограммой, состоящей из одного оператора, является любая функция пользователя, входящая в состав другой программы.

На рис. 9.12 головная программа Z содержит внутри себя две подпрограммы, f1(x) и f2(x), которые многократно вызываются внутри цикла for. В конце программы с помощью составного массива выводятся необходимые результаты расчета.

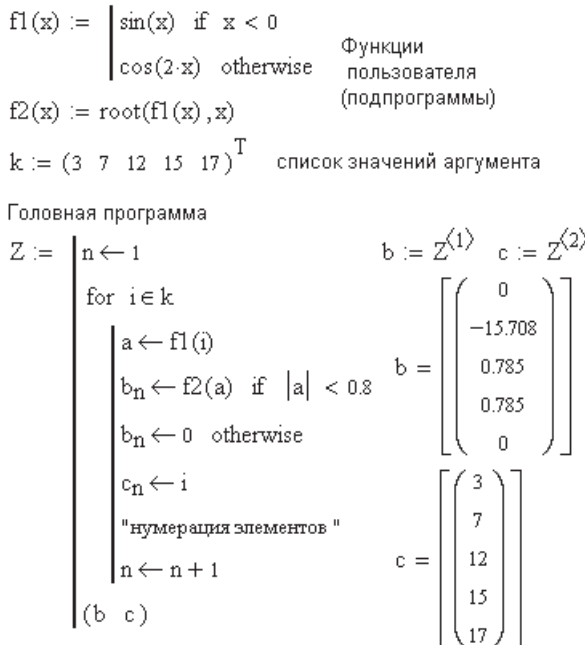


Рис. 9.12. Вложенные подпрограммы

Как правило, подпрограмма-функция располагается где-то выше головной программы, в которой она используется. Результаты расчета можно вывести в любой позиции правее и ниже имени выражения-программы. Примеры сложных

программ с использованием многих подпрограмм находятся в части IV книги (примеры 55 и 56).

9.6. Интегрирование функции методом трапеций

Операция интегрирования самая медленная в Mathcad. Особенно длительным является вычисление интегральной функции с использованием дискретной переменной при построении графика. Смена метода интегрирования, предоставляемая Mathcad при вызове контекстного меню, не приводит к ускорению расчета.

Для ускорения операции интегрирования предлагаю простую программу интегрирования функции методом трапеций, самым простым из всех возможных методов интегрирования. Это программа $\text{Int}(F, L_1, L_2, N)$, показанная на рис. 9.13. Здесь F — имя функции, L_1 и L_2 — верхний и нижний пределы интегрирования, N — число промежуточных точек интегрирования. Сама программа очень проста и не требует комментариев.

Для примера выбрана функция, полученная интерполяцией экспериментального массива данных. Формирование этой функции показано на рис. 9.13, *вверху*. Сравнение результатов интегрирования с помощью функции Int и стандартным оператором интегрирования Mathcad показано на рис. 9.14. Функция Int позволяет в несколько раз сократить время операции. Совпадение графиков показывает, что функция Int обеспечивает достаточную точность расчета даже при малом количестве точек интегрирования.

```

массив экспериментальных данных (X,Y)
X := (0 1 3 4 6 7 10 11 13 14 20)T
Y := (3 7 9 5 5 2 7 9 6 2 2)T

Преобразуем массив в функцию путем интерполяции или регрессии
F(t) := interp(1spline(X, Y), X, Y, t)    приближение к прямой линии
                                           в граничных точках

Функция интегрирования методом трапеций

Int(F, L1, L2, N) :=
  ff ← 0
  ΔL ←  $\frac{L_2 - L_1}{N}$ 
  for i ∈ 0.. N
    f1 ← F(L1)
    x ← L1 + ΔL
    L1 ← x
    f2 ← F(x)
    ff ← ff +  $\frac{f1 + f2}{2} \cdot \Delta L$ 

```

F -- имя интегрируемой функции
(без параметров)

L1 и L2 -- верхний и нижний
пределы интегрирования

N -- число точек
интегрирования

Рис. 9.13. Программа интегрирования функции методом трапеций

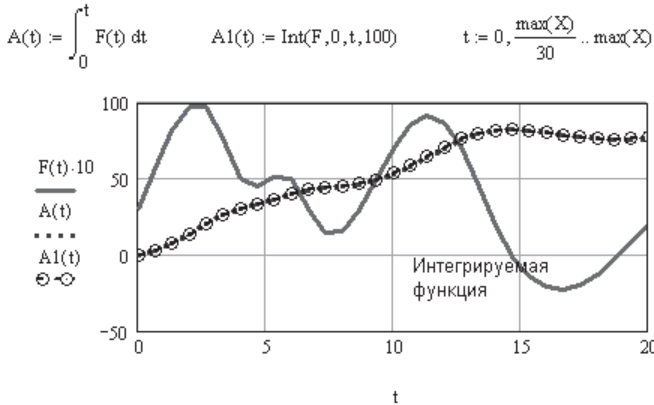


Рис. 9.14. Сравнение результатов интегрирования, выполненного с помощью функции Int и стандартным оператором интегрирования Mathcad

9.7. Решение уравнений и систем уравнений в программе

В главе 3 (раздел 3.2) описано использование функции пользователя для нахождения корней уравнения при переменных значениях параметров b , перечисленных в названии функции пользователя (см. рис. 3.5 и 3.6). То же самое для систем уравнений описано в разделе 3.4 (см. рис. 3.11, 3.12).

В старых версиях Mathcad функцию root можно было использовать в программных модулях, как показано на рис. 9.15. Решаемое уравнение записывалось в виде функции $f(a, b, z)$, равной нулю. Далее внутри двух вложенных циклов for задавались переменные параметры b и c . Корень уравнения для каждого значения a и b , определяемый функцией root, записывался в массив Z. Это было очень удобно для расчетов. К сожалению, в новых версиях Mathcad использование функции root в программных модулях невозможно.

Довольно сложным моментом является решение системы уравнений в программном модуле. Вычислительный блок Given–find, оформленный как функция пользователя, вместе с начальными приближениями для каждой переменной размещается выше программы. Решение системы уравнений в этом случае выводится в виде массива, каждый столбец которого соответствует вектору решения для одной переменной.

ПРИМЕЧАНИЕ

Существенным недостатком использования вычислительного блока со словом «given» (Given–find, Given–minerr, Given–maximize, Given–Odesolve) является то, что начальные приближения задаются вне программного модуля и не могут быть изменены внутри него. В ряде случаев (для ступенчатых функций) это приводит к невозможности решения задачи.

В примере на рис. 9.15 внутри двух вложенных циклов for на каждом шаге вызывается функция пользователя $w(a, b)$, где значения a и b заданы как переменные цикла, хотя их можно задать и массивом значений и вычислять по какой-либо формуле. Результаты вычисления значений x и y записываются в массив K при

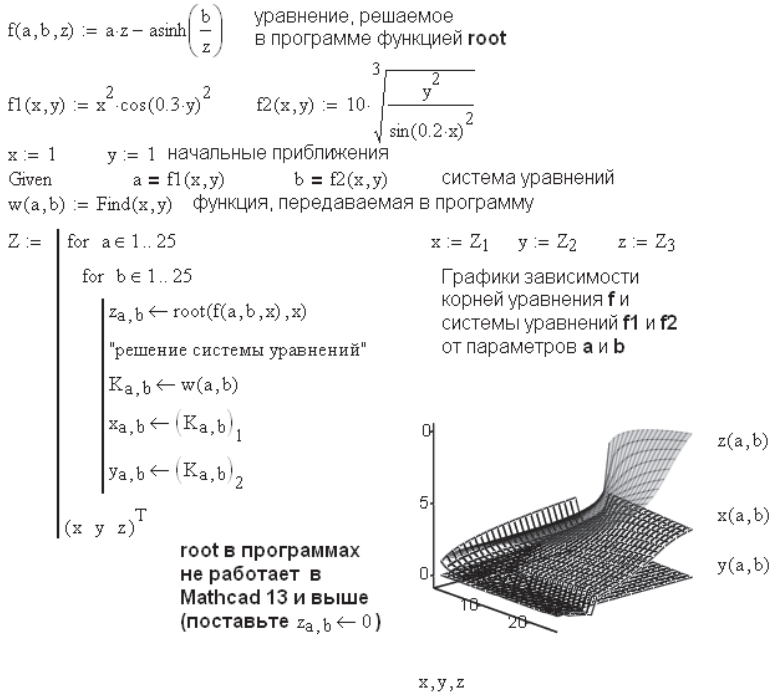


Рис. 9.15. Решение уравнений и систем уравнений в программе

каждом значении переменных a и b . При этом сама функция $w(a, b)$ на каждом шаге вызывается один раз, а не два, чтобы найти значения x и y . Такой прием позволяет вдвое сократить время выполнения программы, которое при большем количестве уравнений может быть значительным, ведь основное время в программе тратится на вычисление функции $w(a, b)$. С помощью составного массива (последний оператор программы) расчетные значения x , y и z выводятся из программы и представлены в виде графика поверхностей $x(a, b)$, $y(a, b)$ и $z(a, b)$.

9.8. Нахождение экстремума функции в программе

Средства нахождения экстремума функции в Mathcad (функции `minerr` и `maximize`) имеют серьезные ограничения. Они не могут определить экстремум функций со скачками и разрывами. Приходится выбирать свои начальные приближение и ограничения для каждого участка ломаной кривой графика функции, а также выбирать из нескольких локальных экстремумов один главный. Это означает, что расчет не проходит автоматически и требует ручного вмешательства пользователя, для того чтобы корректировать начальные условия и ограничения.

Организовать автоматическое прохождение всех участков внутри программного модуля тоже не удастся. При использовании вычислительного блока `Given`–(`find`, `minerr`, `maximize` или `minimize`) внутри программы вызывается только функция, но не меняется начальное приближение. К тому же не срабатывают ограничения на диапазон поиска экстремума. Функции `maximize` и `minimize` в принципе не допус-

кают присутствия переменных параметров. В результате эти функции внутри программного модуля часто дают неверные результаты.

В большинстве практических задач можно использовать простейший перебор значений заданной функции в заданном диапазоне значений аргумента и выбрать из них экстремальное. На рис. 9.16 и 9.17 показан поиск максимального по абсолютной величине значения ступенчатой функции $M(x)$ без учета размерностей. На графике значения функции взяты по модулю. Поиск максимума функции выполняется с помощью программы-функции $F_{\max}(F, L_{\text{н}}, L_{\text{к}}, N)$. Ее параметры: F — имя функции, для которой ищется максимум; $L_{\text{н}}$ — начало интервала поиска; $L_{\text{к}}$ — конец интервала поиска; N — число точек внутри интервала поиска, в которых определяется значение функции.

Без учета размерностей

Поменяйте числовые значения в функции $L := 2 \quad x := 0, \frac{L}{100} \dots L$

$$M(x) := 200 - 50 \cdot x \cdot \left(x > \frac{L}{10}\right) + 400 \cdot \left(x > \frac{2 \cdot L}{10}\right) - 500 \cdot \left(x > 5 \cdot \frac{L}{10}\right) + 400 \cdot x \cdot \left(x > 7 \cdot \frac{L}{10}\right)$$

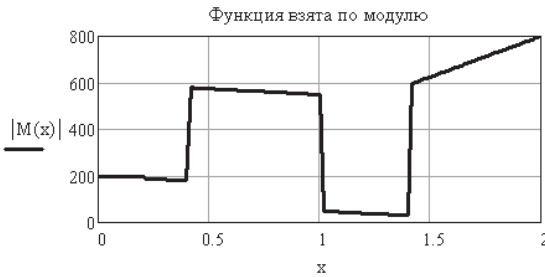


Рис. 9.16. Безразмерная функция, максимум которой найден на рис. 9.17

Поиск максимума функции путем перебора ее значений

Поменяйте параметры поиска

```

F_max(F, L_n, L_k, N) :=
  L ← L_n
  A1 ← |F(L)|
  while L < L_k
    L ← L + (L_k - L_n) / N
    A2 ← |F(L)|
    if A2 > A1
      A1 ← A2
      L1 ← L
  (A1 L1)ᵀ
    
```

$L_{\text{н}} := 0 \quad L_{\text{к}} := L \quad N := 1000$

Вывод результатов поиска

$M_{\text{max}} := F_{\max}(M, L_{\text{н}}, L_{\text{к}}, N)_1$
 $x_{\text{max}} := F_{\max}(M, L_{\text{н}}, L_{\text{к}}, N)_2$
 $M_{\text{max}} = 800$
 $x_{\text{max}} = 2$

Рис. 9.17. Поиск максимума безразмерной функции со скачками (см. рис. 9.16)

Внутри программы вначале определяются значение аргумента L и значение функции $F(L)$ в начале поиска. Затем внутри цикла ($\text{while } L < L_K$) для N значений аргумента определяются значения функции $F(L)$. Каждое значение функции сравнивается с предыдущим и, если оно больше предыдущего, записывается как максимальное A_1 . Как L_1 записывается значение ее аргумента. По окончании расчета (закрытии цикла while) последние значения A_1 и L_1 выводятся из программы с помощью составного массива. Для расчета максимума заданной функции надо в параметры функции вписать истинное имя функции (без параметров) и числовые значения L_H, L_K, N .

С учетом размерностей

```

H := N      кН := 1000·N  м := m  см := 0.01·м  Нм := Н·м  кНм := кН·м
M := (200 400 500)T·Нм  F := (50 400)T·Н  L := 2·м  Поменяйте исходные данные  x := 0,  $\frac{L}{100} \dots L$ 
M(x) := [ M1 - F1·x (x >  $\frac{L}{10}$ ) + M2·(x >  $\frac{2·L}{10}$ ) - M3·(x >  $5·\frac{L}{10}$ ) + F2·x (x >  $7·\frac{L}{10}$ ) ]
    
```



Рис. 9.18. Размерная функция, максимум которой найден на рис. 9.19

```

L_H := 0·м  L_K := L  N := 1000  Поменяйте параметры поиска
    
```

В составном массиве вывода результатов расчета сделать все размерности одинаковыми

```

F_max(F, L_H, L_K, N) := | L ← L_H
                        | A1 ← |F(L)|
                        | while L < L_K
                        |   | L ← L +  $\frac{L_K - L_H}{N}$ 
                        |   | A2 ← |F(L)|
                        |   | if A2 > A1
                        |   |   | A1 ← A2
                        |   |   | L2 ← L
                        |   | (A1 L2·H)T
                        | M_max := F_max(M, L_H, L_K, N)1
                        | Возвращение к исходной размерности
                        |   F_max(M, L_H, L_K, N)2
                        | x_max :=  $\frac{F_{max}(M, L_H, L_K, N)_2}{H}$ 
                        | M_max = 800·Нм
                        | x_max = 2·м
                        | Чтобы избежать сложностей с размерностями в программах, используйте квазиразмерности
    
```

Рис. 9.19. Поиск максимума размерной функции со скачками (см. рис. 9.18)

Эта программа простая, не использует градиентных методов поиска, но работает хорошо и полезна для практических расчетов.

В случае выполнения расчетов с учетом размерности в начале расчета вводятся производные размерности (на русском языке). Далее вводятся исходные данные, умноженные на производную размерность, и заданная функция (в данном случае $M(x)$). Исходные данные и график заданной функции показаны на рис. 9.18. Поиск максимального значения функции в заданном интервале значений аргумента x показан на рис. 9.19.

Программа поиска максимума та же, что и для расчета безразмерной функции, но есть нюансы ее использования. Они описаны в следующем разделе.

9.9. Учет размерностей в программе

Размерности в программном модуле можно учесть, если все выводимые из модуля данные имеют одинаковую размерность.

Производные размерности надо задавать в начале документа. Во внутренних операторах программы могут использоваться величины любой размерности. Из программы выводится значение последнего оператора. Если это составной массив, то все элементы составного массива должны иметь одинаковую размерность. На рис. 9.20 все элементы составного массива имеют размерность длины, поэтому проблем нет. Программе в этом случае безразлично, учитывается размерность или нет.

```

м := 100·см    сек := s    кг := kg    Н := N    производные
                                     размерности
а := 4· $\frac{м}{сек^2}$     Δt := 1·сек    м := 3·кг
Все результаты расчета одной размерности
Z := for i ∈ 1..4
    | t ← Δt·i
    | S1i ← а·t2
    | S2i ← 0.5·а·(t - 1·сек)2
    | S3 ← S1 + S2
    | (S1 S2 S3)T
    Z =  $\begin{pmatrix} \{4,1\} \\ \{4,1\} \\ \{4,1\} \end{pmatrix} м$ 
    Z1 =  $\begin{pmatrix} 4 \\ 16 \\ 36 \\ 64 \end{pmatrix} м$ 
    (Z2)4 = 18 м
    (Z3)4 = 82 м

```

Рис. 9.20. Все выводимые величины одной размерности

На рис. 9.21 из программы выводятся время t , скорость v и путь S , имеющие, естественно, разную размерность. В этом случае надо выровнять размерности, сделать их одинаковыми. Для этого элементы составного массива умножаются на выравнивающую производную размерность, по смыслу каждый раз разную. При выводе результатов расчета выражение надо умножить на размерность, обратную выравнивающей. Теперь ответы выводятся в правильной размерности.

Надо сделать одинаковыми размерности выходного массива.
Затем вернуться к исходной размерности

$$\begin{array}{l}
 Z := \text{for } i \in 1..4 \\
 \quad t \leftarrow \Delta t \cdot i \\
 \quad V_i \leftarrow a \cdot t \\
 \quad S_i \leftarrow V_i \cdot t \\
 \quad T_i \leftarrow t \\
 \left(\begin{array}{l} V \cdot \text{сек} \\ S \\ T \cdot \frac{m}{\text{сек}} \end{array} \right)
 \end{array}
 \quad
 \begin{array}{l}
 Z = \begin{pmatrix} \{4,1\} \\ \{4,1\} \\ \{4,1\} \end{pmatrix} m \\
 V := \frac{Z_2}{\text{сек}} \\
 V = \begin{pmatrix} 4 \\ 16 \\ 36 \\ 64 \end{pmatrix} \frac{m}{s}
 \end{array}
 \quad
 \begin{array}{l}
 S := Z_2 \\
 S = \begin{pmatrix} 4 \\ 16 \\ 36 \\ 64 \end{pmatrix} m \\
 t := Z_3 \cdot \frac{\text{сек}}{m} \\
 t = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix} s
 \end{array}$$

Рис. 9.21. Выводимые величины имеют разную размерность

Аналогичное выравнивание размерностей проведено в программе Fmax на рис. 9.19. Так как в последнем операторе (составном массиве величина A1 имеет размерность момента (Нм), то и второй элемент этого массива, L2, должен иметь размерность момента силы, а не длины (м), поэтому он умножается на размерность Н — ньютон. Для возвращения к правильной размерности результат расчета x_{\max} разделен на размерность Н — ньютон.

С использованием размерностей в программе выполнен пример 5 части IV книги.

9.10. Решение дифференциальных уравнений внутри программы

Возможности решать дифференциальные уравнения в теле программного модуля в Mathcad крайне ограничены. Каждый оператор внутри программного модуля должен возвращать число. Функция Odesolve возвращает функцию и по этой причине не может использоваться в программах старых версий Mathcad. Остается использовать в программах старые функции решения дифференциальных уравнений. На рис. 9.22 показано использование функции rkfixed. В программе можно варьировать лишь два параметра: границы интервала интегрирования и число точек, в которых выводится решение.

ПРИМЕЧАНИЕ

В старых версиях Mathcad нельзя изменять параметры самого дифференциального уравнения, что как раз и необходимо в практических задачах.

В Mathcad 13 уже разрешено изменение параметров функции Odesolve внутри программного модуля (рис. 9.23 и 9.24).

Использование функций, дающих решение в двух точках, возможно. Оно аналогично примеру, приведенному на рис. 9.22, и столь же бесполезно.

В Mathcad 13 появилась возможность использовать функцию Odesolve в программных модулях. На рис. 9.23 приведено решение дифференциального урав-

```

y := 0    обнуление предыдущего массива y
y' + 3 · y = 0    исходное уравнение
y1 := 5    начальное приближение y(0)=5
D(x,y) := -3 · (y1)    первая производная в явном виде

Варируемые параметры:
a := 0    b := 3    границы интервала интегрирования
n := 1    число точек, в которых выводится решение

ZZ := | for i ∈ 1..b
      |   Z ← Rkadapt(y, a, i, n, D)
      |   bbi ← i
      |   Z1 ← Z(2)
      |   ZZ(i) ← Z1
      | (bb ZZ)T
      |
      |   bb := ZZ1    Z := ZZ2
      |
      |   bb =  $\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$     правая граница
      |                                     интервала
      |                                     интегрирования
      |
      |   ZT =  $\begin{pmatrix} 5 & 0.249 \\ 5 & 0.012 \\ 5 & 0.001 \end{pmatrix}$     решение уравнения в
      |                                     начале и в конце
      |                                     интервала
      |                                     интегрирования

```

Рис. 9.22. Решение дифференциального уравнения внутри программы

нения свободных затухающих колебаний с переменными параметрами, оформленное в виде функции пользователя $D(m, h, k)$.

Далее поставлена задача найти значения коэффициентов дифференциального уравнения, при которых решение ОДУ наилучшим образом соответствует экспериментальным данным. Решение ОДУ (функция $x \leftarrow D(m, h, k)$) введено в подпрограмму локально (см. рис. 9.24).

В предыдущих версиях каждый оператор в программном модуле должен был быть числом. В Mathcad 13 такой оператор может быть просто функцией, которой в дальнейшем надо присвоить численное значение. Так, на рис. 9.24 первый оператор в подпрограмме $Err\ x \leftarrow D(m, h, k)$ является функцией и как бы принят к сведению. Второй оператор, $x(tt)$, является уже числом и участвует в расчете.

Вектор отклонения решения ОДУ от экспериментальных данных рассчитывается в подпрограмме Err . Оператор $(x(tt) - \overline{xx})$ благодаря векторизации одновременно вычисляет весь вектор отклонений аналогично оператору цикла.

Для нахождения параметров, обеспечивающих минимальное расхождение расчетных и опытных данных, использована функция $minerr$, в вычислительном блоке которой в итерационном процессе многократно вызывается подпрограмма Err . В процессе выполнения подпрограммы $minerr$ многократно вызывается функция $Odesolve$, отчего время выполнения оператора велико. Сопоставление опытных данных с рассчитанными по оптимальным параметрам приведено на рис. 9.25.

Экспериментальные данные (перемещение xx от времени tt)

$$tt := (0 \ 0.6 \ 1.2 \ 1.8 \ 2.4 \ 3 \ 3.6 \ 4.2 \ 4.8 \ 5.4 \ 6)^T$$

$$xx := (1 \ 0.902 \ 0.655 \ 0.337 \ 0.023 \ -0.228 \ -0.382 \ -0.432 \ -0.419 \ -0.382 \ -0.245)^T$$

Подобрать коэффициенты m , h , k дифференциального уравнения свободных затухающих колебаний системы, наиболее удовлетворяющие опытным данным

$t_0 := 0$ $t_k := 6$ начало и конец интервала интегрирования Поменяйте условия задачи $t := 0$

$A := 1$ $B := 1$ граничные условия

Given $m \cdot \frac{d^2}{dt^2} x(t) + 2 \cdot h \cdot m \cdot \frac{d}{dt} x(t) + k \cdot x(t) = 0$

$x(t_0) = A$ $x'(t_0) = B$

$D(m, h, k) := \text{Odesolve}(t, t_k)$

Рис. 9.23. Функция пользователя с использованием Odesolve

Найдем вектор ошибки Err , используя Mathcad-программу

Функцию x определяем в программе локально

$$Err(m, h, k) := \begin{array}{l} x \leftarrow D(m, h, k) \\ \xrightarrow{\quad} \\ (x'(tt) - xx) \end{array}$$

Для отыскания минимальной ошибки используем функцию $Minerr$

$m := 1$ $h := 0.2$ $k := 1000$ начальные приближения параметров

$$\text{Given} \quad Err(m, h, k) = 0 \quad \begin{pmatrix} m \\ h \\ k \end{pmatrix} := \text{Minerr}(m, h, k)$$

Решение $m = 0.99$ $h = 0.205$ $k = 1017.037$

$y := D(m, h, k)$

Рис. 9.24. Использование функции Odesolve в программе

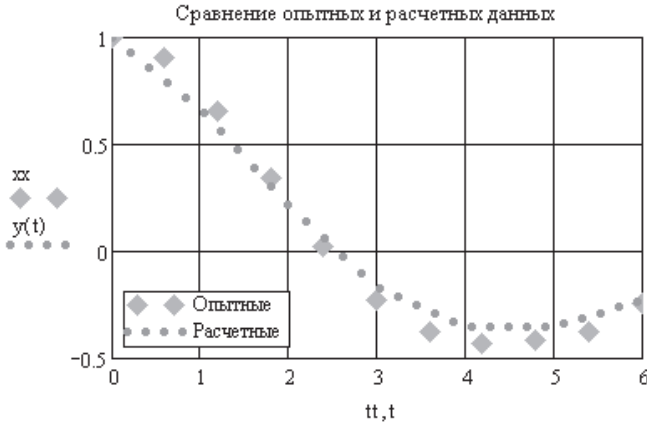


Рис. 9.25. Результаты решения ОДУ (см. рис. 9.23)

Решение ОДУ с переменными параметрами, но без программирования

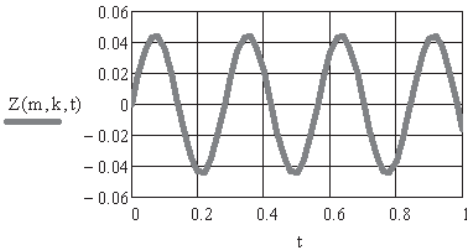
$t_0 := 0$ $t_k := 6$ начало и конец интервала интегрирования

$A := 1$ $B := 1$ граничные условия

Given $m \cdot \frac{d^2}{dt^2} y(t) + k \cdot y(t) = 0$

$y(t_0) = 0$ $y'(t_0) = 1$ $t := 0, 0.02..1$

$Z(m,k) := \text{Odesolve}(t, t_k)$ $m := 1$ $k := 500$



Решение предыдущего ОДУ с переменными параметрами m и k при $t = 1$ с использованием программирования

$$Z1(m) := \begin{cases} \text{for } i \in 1..5 \\ \begin{cases} k \leftarrow 1000 \cdot i \\ m \leftarrow 1 + 0.2 \cdot i \\ x \leftarrow Z(m, k, 1) \\ H_i \leftarrow x \end{cases} \\ H \end{cases} \quad Z1(m) = \begin{pmatrix} -0.019 \\ 0.003 \\ -0.015 \\ -0 \\ -0.005 \end{pmatrix}$$

Рис. 9.26. Решение ОДУ с переменными параметрами

Простой пример создания функции пользователя для решения ОДУ с переменными параметрами приведен на рис. 9.26. Такая функция может использоваться как внутри программного модуля, так и самостоятельно. Обратите внимание на то, что функция пользователя $Z(m, k) = \text{Odesolve}(t, t_k)$ содержит два параметра, m и k , хотя является функцией трех переменных, m, k, t . При выводе результатов (на графике и в программе) необходимо писать эту функцию с тремя параметрами $Z(m, k, t)$.

9.11. Отладка программ

Mathcad — перспективное средство программирования. Создать расчетную программу в Mathcad можно во много раз быстрее и значительно проще, чем на других языках программирования. Кроме того, программа Mathcad — это фактически записанный в виде последовательности формул алгоритм решения задачи. Она абсолютно понятна каждому, знающему суть задачи. В отличие от нее программы на других языках представляют собой очень трудный для понимания длинный список операторов.

Приведенная в примере 55 части IV программа решения упруго-пластической задачи методом конечных элементов — это 3 страницы, на которых в виде формул записаны операторы, подпрограммы и головная программа. Та же программа на языке Fortran представляет собой 15 страниц сложных для понимания операторов.

Складывается впечатление, что создатели Mathcad (фирма Mathsoft) не рассчитывали на применение Mathcad в качестве средства создания серьезных больших программ. Раздел программирования в Mathcad проработан явно слабее остальных разделов.

В Mathcad 13 наконец-то появились две функции, Trace и Debug, предназначенные для отладки программ. Увы, эти средства работают лишь в отлаженных программах и не работают в неотлаженных, где они, собственно, и нужны.

В Mathcad 13 разработчики предусмотрели специальную область окна Mathcad, называемую Trace Window (Окно отладки), а также дополнительную панель инструментов Debug (Отладка) и встроенные функции trace и pause. Функция trace осуществляет вывод наблюдаемых переменных в окно отладки; функция pause приостанавливает вычисления, позволяя анализировать выведенные в окне данные промежуточных вычислений.

Обращение к этим функциям:

```
trace("format string", x, y, z...);
pause("format string", x, y, z...),
```

где *format string* — строка, содержащая текст (на английском языке), предшествующий выводимому числу; $x, y, z...$ — список аргументов, значения которых выводятся на экран.

ВНИМАНИЕ

Для включения этих функций в работу нужно в меню View выбрать Toolbars ► Debug и в появившейся на экране панели включить обе кнопки, Toggle Debugging и Trace Window (рис. 9.27).



Рис. 9.27. Панель Debug

Строка текста, взятая в кавычки, должна заканчиваться знаком = и порядковым номером метки в фигурных скобках, которая отмечает оператор, чей результат действия выводится в окне Trace Window (рис. 9.28).

Нахождение первого элемента вектора,
 большего заданного значения $K := 1.6$ $ORIGIN := 1$

$j := 1..7$ $v_j := \sqrt[3]{j-1}$ j -номер элемента

$v^T =$		1	2	3	4	5	6	7
	1	0	1	1.26	1.442	1.587	1.71	1.817

$t1 :=$ $\left| \begin{array}{l} j \leftarrow 1 \\ \text{while } |v_j| \leq K \\ \quad \left| \begin{array}{l} zz \leftarrow K \\ \text{trace}("K=(0),j=(1),v_j=(2)",K,j,v_j) \\ j \leftarrow j + 1 \end{array} \right. \quad t1 = \begin{pmatrix} 1.6 \\ 6 \\ 1.71 \end{pmatrix} \\ z \leftarrow zz \\ (z \ j \ v_j)^T \end{array} \right.$

```

< >
Окно трассировки - Glava09-ф-Program
K=1.6, j=3, vj=1.25992104989487
K=1.6, j=4, vj=1.44224957030741
K=1.6, j=5, vj=1.5874010519682
j=1, vj=0
j=2, vj=1
j=3, vj=1.25992104989487
j=4, vj=1.44224957030741
j=5, vj=1.5874010519682
The value = vector(1) [0]

```

Рис. 9.28. Результат действия функции trace

ПРИМЕЧАНИЕ

Для вывода результата действия функции `trace` выделить все выражение с этой функцией и нажать клавишу = (равно) или написать имя выражения-программы с функцией `trace` и оценить выражение, нажав клавишу = (равно).

Функция `pause` отказывается принимать конкретные числовые значения либо указанный элемент массива. Примеры, приведенные в файлах помощи `Mathcad`, зависят и останавливаются после нескольких операций. Сложность использования функции `pause` делает ее бесполезной без доработки в последующих версиях `Mathcad`.

Функция `Debug` работает только в работающей программе. Между тем требуется вывести сообщение о том, в каком месте происходит сбой программы, то есть требуется вывод пошаговых значений. В таком виде, в котором она существует, функция `Debug` оказывается бесполезной.

Рекомендации, описанные далее, составлены на основании моего личного опыта программирования.

При разработке программы начните со следующих действий:

- Составьте формульный алгоритм программы.
- Выделите и запишите в виде подпрограмм-функций повторяющиеся несколько раз или просто загромождающие программу части.
- Составьте головную программу. Она должна быть сравнительно краткой и включать в себя обращения к подпрограммам.

Далее начинается отладка программы. Вначале отлаживается каждая подпрограмма.

Просмотрите список величин, используемых в данной подпрограмме.

Константы, не изменяющиеся внутри программы, должны быть заданы в основном `Mathcad`-документе где-либо выше места использования подпрограммы. Можно присвоить им значения и в самой подпрограмме, но нежелательно без нужды увеличивать размер подпрограммы.

ВНИМАНИЕ

Переменным величинам, изменяющимся внутри подпрограммы, должны быть присвоены значения обязательно внутри подпрограммы, или эти величины должны быть записаны в число параметров в названии подпрограммы-функции. В противном случае величины будут иметь значения, присвоенные им в основном документе, и не смогут изменяться в процессе расчета в подпрограмме.

В конце подпрограммы создайте место вывода результатов расчета в виде составного массива, включив в него как можно больше величин промежуточных расчетов, выполняемых в подпрограмме. Посмотрите на результаты расчета. Как правило, программа с первого раза не работает и результатов не дает.

Щелкните мышью внутри подпрограммы. Посмотрите, есть ли сообщение об ошибках. Если сообщение об ошибках есть, постарайтесь исправить их, учтя сообщение.

Чаще всего появляется сообщение «This variable or function is not defined above» («Эта переменная или функция еще не определена»). В этом случае проследите по документу, где этой величине должно быть присвоено значение (внутри программы, в подпрограмме, в основном документе). Если оно должно быть присвоено внутри программы, присвойте правильное значение, если в основном документе — проверьте значение, написав имя переменной и нажав клавишу =. Если же значение должно быть присвоено в подпрограмме, щелкните на ней. Если сообщений нет, нужно проанализировать подпрограмму, как описано далее.

Часто встречается сообщение «Value of subscript or superscript is too big (or too small) for this array» («Значение верхнего или нижнего индекса слишком велико или слишком мало для этого массива»). В этом случае, если массив вводился в основном документе, проверьте его значения. Если массив больших размеров, чем вы задали, значит, надо обнулить массив перед вводом значений. Если массив вводился внутри программы или подпрограммы, проверьте диапазоны переменных циклов, их соответствие значению ORIGIN. Если ORIGIN = 1, переменные цикла не должны иметь значение 0.



Рис. 9.29. Изготовление проверочной программы

Если сообщений об ошибках нет, предстоит разделить программу на части для проверки:

- ❑ Выделите черным цветом проверяемый кусок программы и вырежьте его в буфер, нажав кнопку Cut (Вырезать).
- ❑ Щелкните мышью вне программы и вставьте в это место из буфера вырезанный фрагмент, нажав кнопку Paste (Вставить).
- ❑ Используемый в программировании знак присвоения \leftarrow замените обычным оператором присвоения значения $(:=)$.
- ❑ Задайте численные значения величин, входящих в вырезанное (и вставленное) выражение, и посмотрите результат расчета, нажав клавишу =. Добейтесь правильного ответа. Оператор цикла или условия вырезается по частям (рис. 9.29).
- ❑ В любом месте вне программы напишите имя будущей проверочной программы, оператор присваивания $(:=)$ и, щелкнув на кнопке Add line, создайте несколько мест ввода (про запас).
- ❑ Выделите тело цикла или условия и вырежьте его в буфер.
- ❑ Вставьте вырезанный фрагмент в любое место вне программы.
- ❑ Выделите и вырежьте в буфер заголовок цикла или условия.
- ❑ Вставьте заголовок цикла или условия в нужное место ввода проверочной программы.
- ❑ Выделите перечеркиванием мышью при нажатой левой кнопке вырезанное ранее тело цикла или условия и скопируйте его в буфер.
- ❑ В место ввода в проверочной программе вставьте из буфера тело цикла или условия.
- ❑ Закончите составление проверочной программы, не забыв вставить составной массив вывода результатов расчета. Проверьте результаты расчета.

Если при выполнении программы не появилось сообщений об ошибке, но и численные результаты не выводятся, предстоит полная разборка программы, как описано ранее. Отлаживайте каждый кусок программы отдельно, а затем вставляйте в основную программу. Важно получить хоть какие-нибудь результаты.

Если наконец вы получили результат, но он неверен, добавьте строки в составной массив вывода результатов. Выведите для просмотра большинство расчетных величин. Постепенно возвращайте на место вырезанные куски программы после их отладки.

Конечно, эти способы отладки очень примитивны, но других разработчики Mathcad пока нам не предоставили. Тем не менее повторяю, что составление программ в Mathcad более удобно и просто, чем при использовании других языков программирования, а отладка не труднее, чем, скажем, на языке Fortran. Время составления и отладки программы в Mathcad в несколько раз меньше, чем требуется для выполнения той же работы на языке Fortran.

9.12. Возможности ускорения работы Mathcad

Решение задачи в Mathcad обычно продолжается дольше, чем решение аналогичной задачи, полученное при использовании любого из компилируемых язы-

ков программирования. Это не имеет значения при однократном вычислении выражения. Однако в больших программах при выполнении многократных расчетов (итерационных процедур, расчетов, где переменной является время или температура) время расчета становится недопустимо большим.

Эта задержка связана с тем, что язык программирования в Mathcad является не компилируемым, а интерпретируемым, то есть Mathcad перед выполнением каждого оператора выполняет некие подготовительные операции. В компилируемых же языках программирования подготовительные операции выполняются один раз, до решения задачи, а после редактирования и компиляции программы создается объектный модуль (exe-файл).

Рассмотрим некоторые случаи, когда ускорение решения задачи в Mathcad все-таки возможно.

Больше всего времени в Mathcad занимает операция интегрирования. Если курсор установлен на знаке интеграла, при нажатии правой кнопки мыши открывается контекстное меню, предлагающее четыре способа вычисления интеграла. По умолчанию Mathcad автоматически выбирает адаптивный метод, наиболее точный, но и наиболее длительный. Выберите другой метод интегрирования. Для этого щелкните правой кнопкой мыши на значке интеграла. В открывшемся контекстном меню выберите один из предлагаемых методов интегрирования. Уменьшите стандартную постоянную ТОС из меню Tools ► Worksheet Option. Возможно, скорость вычисления интеграла возрастет. Проверьте это на примере вычисления интеграла Дюамеля, приведенного в примере 36 части IV. В примерах 51 и 52 матрицу масс элемента можно вычислить, как интеграл

$$M_{i,j} = \int_0^L mN(x)_i N(x)_j dx,$$

но это очень долго. Счет идет во много раз быстрее, если введены стандартные матрицы с элементами-формулами.

При интегрировании функции одной переменной расчет многократно ускоряется, если вместо стандартного оператора интегрирования использовать функцию `Int`, описанную в разделе 9.6.

В ряде случаев вместо функции можно использовать матрицы. Если каждый элемент матрицы является функцией от других матриц, то при вычислении каждого элемента матрицы программа вычисляет всю матрицу, а затем берет из нее лишь одно значение. В программах в методе конечных элементов при формировании матрицы жесткости можно записать два вида выражений:

$$\begin{aligned} KE(k) &:= A(k)^T \cdot C \cdot A(k) \cdot F_k \cdot t; \\ KE_k &:= A_k^T \cdot C \cdot A_k \cdot F_k \cdot t. \end{aligned}$$

При большом числе элементов первое из этих выражений «висит», второе — «летит». Скорость вычисления второго выражения в сотни раз больше, чем первого.

При работе с большими матрицами, когда используется обращение матрицы (решение системы уравнений), оператор $\Delta := K^{-1}P$, хорошо работающий в приме-

рах 50–54, требует много времени (по методу Гаусса идет пересчет всех элементов матрицы, включая нули). В методе конечных элементов при работе с ленточными матрицами рекомендуется использовать метод Холесского, реализованный в программе МСНВ. Такая программа включена в пример 55.

Существует немало других возможностей ускорения работы Mathcad и облегчения отладки программ. Главное — сохранить основные преимущества Mathcad: наглядность, простоту, удобство работы.

Глава 10

Анимация

Mathcad предусматривает возможность анимации графиков и результатов вычислений путем создания AVI-файлов и вставки их в Mathcad-документ. Воспроизводятся созданные AVI-файлы встроенными в Windows средствами.

10.1. Создание анимации

Mathcad имеет встроенную переменную FRAME, предназначенную только для создания анимационных клипов. FRAME принимает целочисленные значения от 0 до n и должна быть включена в расчетные выражения так, чтобы с изменением FRAME изменялся вид соответствующего графика. Фактически FRAME — это номер кадра.

Переменной FRAME нельзя присваивать какое-либо значение в Mathcad-документе. Диапазон значений FRAME задается в диалоговом окне главного меню командой Tools ▶ Animation ▶ Record (Инструменты ▶ Анимация ▶ Запись). Перед созданием анимационного клипа убедитесь, что в главном меню Mathcad в окне Tools включен режим автоматических вычислений.

Рассмотрим процесс создания анимации:

- ❑ Введите переменную FRAME в математическое выражение, определяющее вид графика (рис. 10.1).
- ❑ Выберите пункт меню Tools ▶ Animation ▶ Record (Инструменты ▶ Анимация ▶ Запись). В разных версиях Mathcad пункт Animation находится в разных местах главного меню, а в Mathcad 14 и 15 присутствует в стандартном меню.
- ❑ После открытия диалогового окна анимации (рис. 10.1) курсором выделите нужный фрагмент изображения.
- ❑ В диалоговом окне задайте общее число кадров и частоту их воспроизведения. Установленное по умолчанию число 10 — слишком большая скорость воспроизведения кадров. Лучше установить 2–3 кадра в секунду, иначе «кино» длится 1 секунду.
- ❑ Щелкните на кнопке Animate (Анимировать) и наблюдайте за созданием кадров. Затем появится окно проигрывателя видеофайлов. В этом окне можно установить размер кадра и скорость воспроизведения (рис. 10.1, *справа*). В появившемся окне Playback можно выбрать Вид (Половинный размер, Обычный размер — в масштабе 1 : 1, Двойной размер) и скорость воспроизведения кадров. Если вы оставили в окне анимации 10 кадров в секунду, то движком надо уменьшить скорость примерно до 30 %. Если в окне анимации вы уже поставили 3 кадра в секунду, то регулятор скорости лучше не трогать, а то скорость движения станет совсем маленькой и эффекта кино не получится. Размеры окна можно изменить и обычным способом, то есть растянув его за угол мышь. Однако при этом пропорции окна могут измениться.
- ❑ Созданный клип сохраните с помощью команды Save As (Сохранить как). Сохраненный файл можно использовать и вне Mathcad-документа.

- ❑ Сохраненный как AVI-файл клип нужно вставить в Mathcad-документ. Для этого в главном меню Mathcad выберите команду **Insert** ▶ **Object** (Вставка ▶ Объект). Откроется окно Вставка объекта. Выберите в этом окне Создать из файла. Выберите нужный файл, щелкнув в диалоговом окне на кнопке Обзор (рис. 10.2).

Желательно нажать кнопку **Связь** в том же окне. Это позволит редактировать AVI-файл непосредственно из Mathcad-документа.

Если в том же окне установить флажок **В виде значка** и нажать **ОК**, то на месте курсора в Mathcad-документе появится значок AVI-файла. Двойной щелчок мышью на этом значке вызовет появление проигрывателя видеофайлов и исполнение анимации.

Если флажок **В виде значка** не устанавливать, то после нажатия кнопки **ОК** на месте курсора в Mathcad-документе появится картинка с первым кадром AVI-файла. Двойной щелчок мышью на этом значке вызовет появление проигрывателя видеофайлов и исполнение анимации.



Рис. 10.1. Пример подготовки анимации графика

Вставленный в документ AVI-файл можно отредактировать. Для этого щелкните на объекте (значке или картинке) правой кнопкой мыши. В появившемся контекстном меню выберите **Связанный объект: Видеозапись** ▶ **Правка**, как показано на рис. 10.3.

В открывшемся окне редактирования (рис. 10.4) выберите пункт **Правка** ▶ **Параметры**. Желательно установить флажок **Автоповтор**, тогда «кино» будет длиться бесконечно долго, до тех пор, пока вы не щелкнете мышью вне объекта.

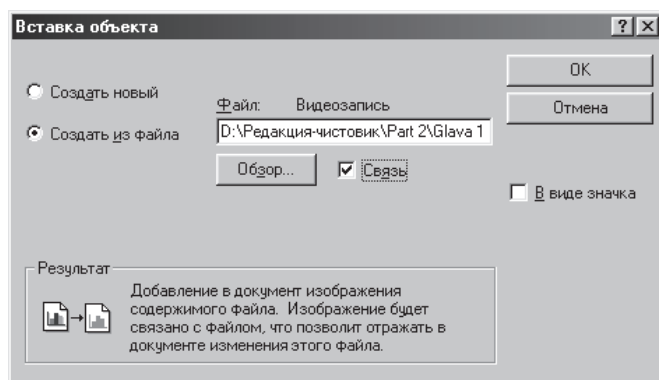


Рис. 10.2. Диалоговое окно вставки объекта

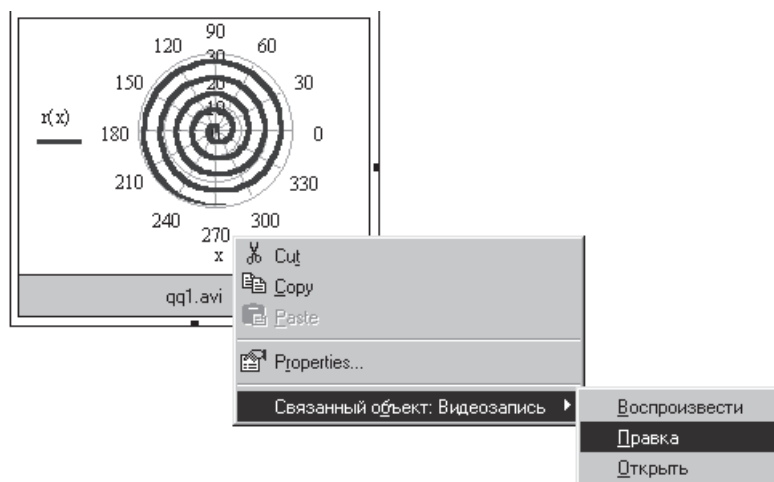


Рис. 10.3. Начало редактирования AVI-файла

Файлы анимации (AVI-файлы) обычно занимают много места на диске. Mathcad позволяет сохранять их в сжатом формате. Перед созданием анимации можно выбрать, какие методы сжатия использовать и сжимать ли файл вообще. Для этого в диалоговом окне Animate (см. рис. 10.1) нужно выбрать Options (Свойства), а в окне Compressor Options (Параметры сжатия) — программу сжатия или команду Полные кадры (без сжатия), рис. 10.5.

ВНИМАНИЕ

При создании анимационных клипов рекомендуется отключить автоматическое масштабирование графика, так как возможны скачки изображения при изменении масштаба. Введите числовые значения на осях графика вручную.

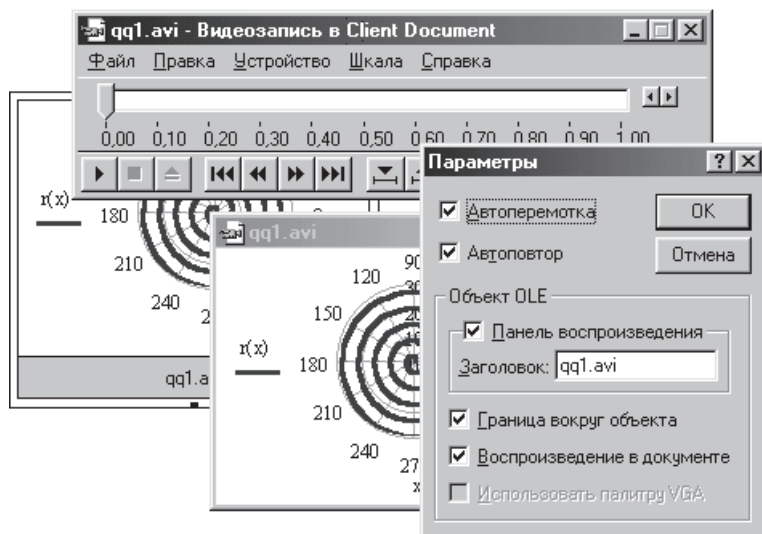


Рис. 10.4. Установка параметров редактирования AVI-файла

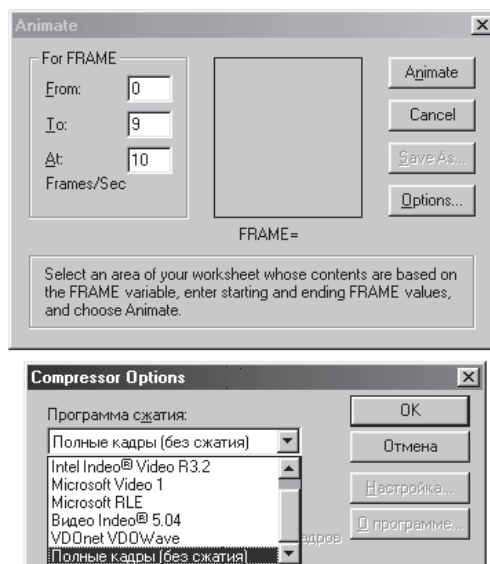


Рис. 10.5. Выбор метода сжатия AVI-файла

Рекомендую вставить анимационный клип в Mathcad-документ не в виде значка, а в виде картинка. После этого график, на основе которого сделан клип, можно удалить. Двойной щелчок мышью в поле объекта приведет к воспроизведению клипа.

10.2. Галерея анимационных клипов

Рисунки 10.6–10.11 иллюстрируют некоторые возможности анимации. Используются различные типы графиков и числовая анимация. Для наглядности на рисунках использовано значение $FRAME = 13$, то есть изображен 13-й кадр анимации. Самостоятельно изготовьте анимационные клипы. Перед анимацией необходимо отключить оператор $FRAME:=13$.

Много интересных примеров анимации объектов (графиков и числовых данных) приведено в файлах помощи Mathcad и в электронной книге (примеры 6 и 26).

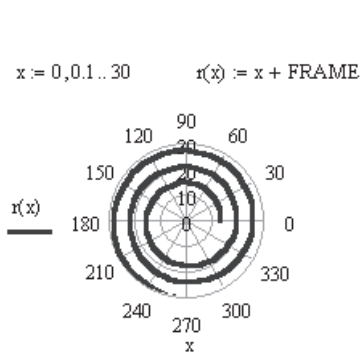


Рис. 10.6. Движение спирали

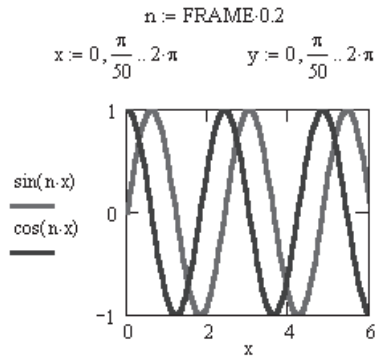


Рис. 10.7. Американские горки

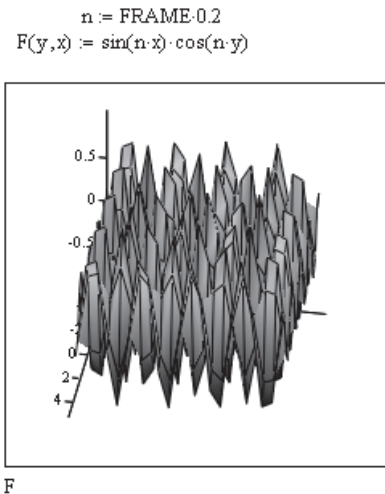


Рис. 10.8. Рост леса синусоид

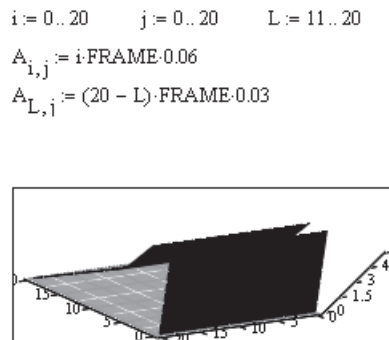


Рис. 10.9. Строительство сарая

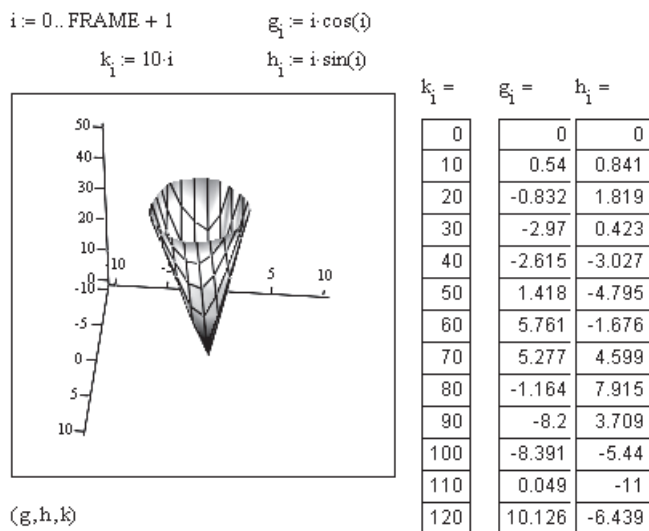


Рис. 10.10. Смерч

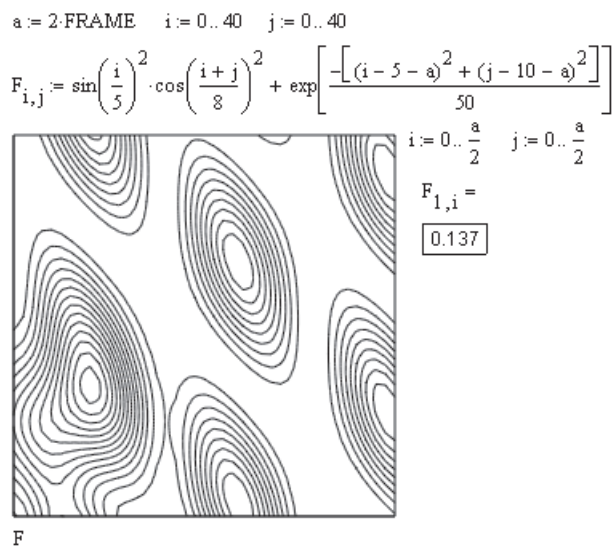


Рис. 10.11. Анимация контурной карты. Установите FRAME = 18

Глава 11

Учет размерностей

В Mathcad возможны вычисления с учетом размерностей используемых величин. Единицы измерения, хотя они и не требуются в выражениях Mathcad, могут помочь упростить ввод данных, обнаружить ошибки и улучшить отображение результатов вычислений.

Присвоение числу размерности эквивалентно его умножению на размерность. Внутренний алгоритм Mathcad умножает число на переводной коэффициент, соответствующий вписанной размерности, и все вычисления выполняет с безразмерными числами, соответствующими СИ (Система интернациональная).

11.1. Выбор системы единиц измерения

В Mathcad встроено большое количество единиц измерения. С ними можно обращаться как со встроенными переменными. Чтобы связать единицу измерения с числом, достаточно умножить число на ее наименование.

Перед началом работы с размерными величинами надо установить систему единиц, в которой вы будете работать:

- ❑ В главном меню Mathcad выберите Tools ▶ Worksheet Options ▶ Unit System (Инструменты ▶ Параметры документа ▶ Система единиц), как показано на рис. 11.1.
- ❑ В открывшемся диалоговом окне выберите систему единиц, например SI (International). Щелкните на кнопке ОК. Если в диалоговом окне выбрать None (Нет), то никакие размерности в расчете не учитываются.

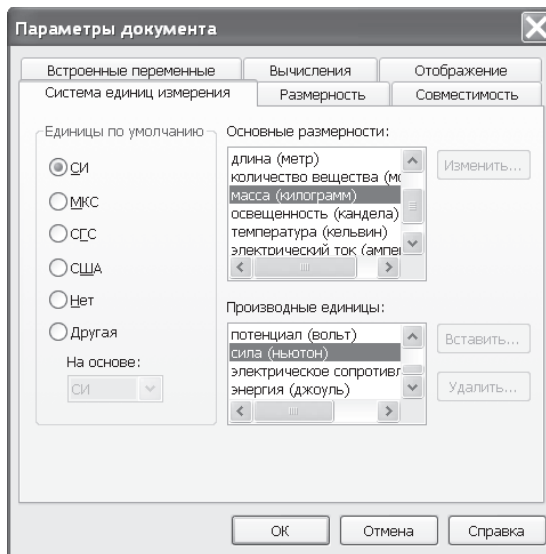


Рис. 11.1. Выбор системы единиц

После этого просмотрите встроенные единицы измерения в выбранной системе. Для этого на стандартной панели Mathcad щелкните на кнопке Insert Unit (Вставить единицу) с изображением мерной кружки. Вначале ничего вставлять не надо, просто познакомьтесь со списком единиц, которые можно использовать в расчетах по умолчанию. В дальнейшем в начале документа при вводе исходных данных для расчета их надо записывать, умножая число на стандартное обозначение размерности, которое можно ввести с клавиатуры либо выбрать в окне Insert Units (Вставить единицы измерения), щелкнув мышью на соответствующем обозначении (рис. 11.2).

Подробные сведения по использованию размерностей в Mathcad можно найти в файлах помощи Mathcad (рис. 11.3).

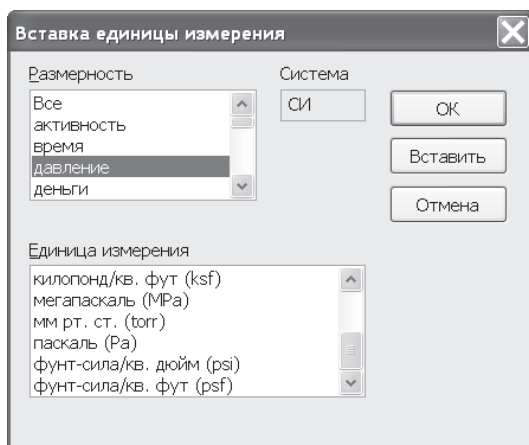


Рис. 11.2. Окно вставки встроенных единиц измерения

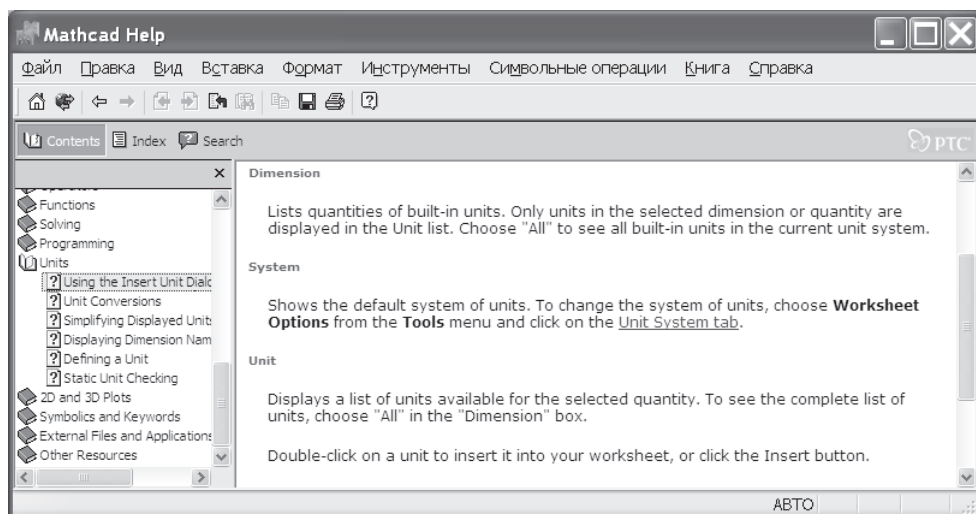


Рис. 11.3. Получение справки по единицам измерения в Mathcad Help

В Mathcad 13 появилась возможность устанавливать смешанную систему размерностей. Для этого нужно в меню Tools ► Worksheet Options ► Units System установить флажок на пункте Custom. Затем выбрать систему, на которой будет базироваться новая система размерностей, например SI или MKS, нажать кнопку Change и выбрать что-нибудь из предложенных размерностей (рис. 11.4). С точки зрения российских пользователей, это бесполезное нововведение, так как предлагается выбрать лишь из размерностей, употребляемых в западном мире, например, добавить фунты, дюймы, галлоны и т. д. Ввести наши производные размерности, тем более на русском языке, невозможно.

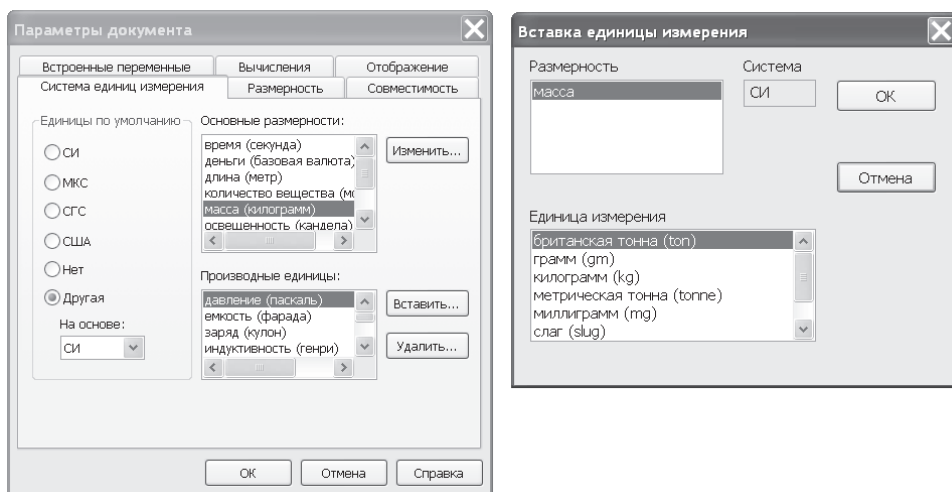


Рис. 11.4. Установка смешанной системы размерностей

11.2. Ввод пользовательских единиц измерения

Обозначать размерности нам привычнее русскими буквами, а это уже нестандартное обозначение размерностей.

Для ввода пользовательских обозначений размерностей нужно:

- ❑ выбрать пункт главного меню Mathcad Tools ► Worksheets Options (Инструменты ► Параметры документа) и перейти на вкладку Dimensions (Размерности), показанную на рис. 11.5;
- ❑ установить флажок Display dimensions (Показывать размерности). Удалить старые размерности и ввести с клавиатуры свои размерности, абсолютно любые;
- ❑ в начале Mathcad-документа ввести производные от исходных размерностей (рис. 11.6). Далее можно вводить исходные данные, умноженные на размерности стандартные (на английском языке) или производные от стандартных. После вычисления результаты расчета будут появляться на экране уже с введенными вами размерностями.

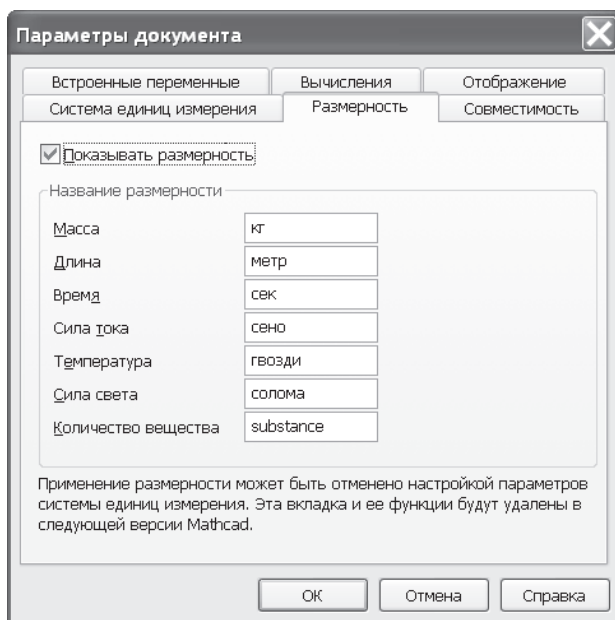


Рис. 11.5. Ввод пользовательских размерностей

Ввод размерностей, производных от базовых

метр := m кг := kg сек := s Н := N т := 1000 · кг

кН := 1000 · Н Р := 3 · кН мин := 60 · сек час := 3600 · s

можно и так мм := mm мин := min час := hr км := km

Ввод величин со стандартными размерностями

a := 2 · m M := 5000 · kg t := 120 · s P := 3000 · N

Ввод величин с производными размерностями

a := 2 · метр M := 5 · т t := 2 · мин P := 3 · кН

Вывод на экран, если в меню Math Option:

Своих размерностей нет

a = 2m M = 5000kg t = 120sec P = 3000kgmsec⁻²

Введены свои размерности масса-кг, длина-метр, время-сек

a = 2метр M = 5000кг t = 120сек P = 3000кгметрсек⁻²

В поле ввода вписаны производные размерности

a = 2000мм M = 5т t = 120сек P = 3кН

Рис. 11.6. Пример ввода размерностей в исходных данных к расчету

11.3. Изменение масштаба результатов расчета

Часто бывает нужно вывести результаты расчета в размерностях, кратных исходным, например, вместо килограммов использовать тонны, вместо метров — кило-

метры, вместо секунд — часы или минуты. Производные размерности вводятся как обычные константы, с использованием оператора присваивания := (рис. 11.6).

ПРИМЕЧАНИЕ

Производную размерность нельзя определить, умножив число на пользовательскую размерность, вписанную на вкладке Tools ▶ Options ▶ Dimensions (Инструменты ▶ Параметры документа ▶ Размерности). Умножать число надо на стандартную размерность, записанную на английском языке, или на другую производную размерность, которой ранее в документе присвоено какое-либо значение.

При выводе результатов расчета после нажатия клавиши (=) появляется число и рядом с ним — место ввода единицы размерности. Если число уже имеет размерность, то рядом с размерностью появляется еще одно место ввода. Сюда можно поместить любую переменную, константу или выражение.

ВНИМАНИЕ

Не забудьте, что всякий раз, когда что-нибудь находится в месте ввода единицы измерения, Mathcad изменяет полученный результат так, чтобы произведение числа и выражения, стоящего на месте размерности, давало правильное значение.

Можно использовать место ввода, чтобы выразить, например, значение угла в долях числа π или вывести числовой коэффициент при каком-то буквенном выражении. На рис. 11.7 рассчитанная величина A выведена двумя способами:

1. Естественным для Mathcad — в единицах, введенных в окне Worksheet Options (Параметры документа).
2. Выведен числовой коэффициент при $\frac{Pa}{t}$.

$$P = 3 \text{ кН} \quad t := 120 \cdot \text{s} \quad a := 2 \cdot \text{m} \quad A := \int_a^{5a} \frac{P \cdot t}{x^2} dx$$

Ответ в размерностях меню Dimension $A = 144000 \text{ кг} \cdot \text{сек}^{-1}$

В поле ввода размерности введено $\frac{P \cdot t}{a} \quad A = 0.8 P \cdot \frac{t}{a}$

Рис. 11.7. Ввод какого-либо выражения на место ввода размерности

Для расчета тригонометрических функций Mathcad требует ввода угловых значений в радианах. При выводе результатов в радианах иногда желательно представить их в градусах. Для этого в поле ввода надо вписать deg (градус). Появится ответ в градусах (рис. 11.8).

Если исходные данные для расчета введены с размерностями, то выводимый результат расчета всегда будет содержать размерность в стандартном обозначении, если вы не вводили пользовательскую размерность, или пользовательскую, если вы ее вводили.

град := deg
asin(0.5) = 0.524
asin(0.5) = 0.524 · rad
asin(0.5) = 30 · deg
asin(0.5) · $\frac{180}{\pi}$ = 30
atan(1) = 0.785
atan(1) = 0.25 · π
atan(1) = 45 · град
atan(1) = 45 · $\frac{\pi}{180}$

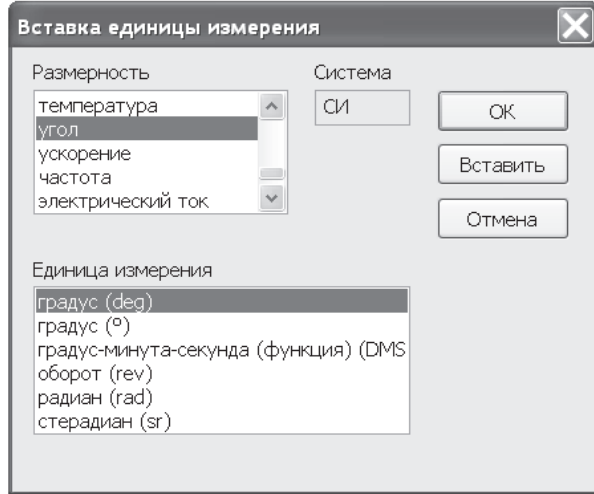


Рис. 11.8. Вывод значения угла в радианах и в градусах

Если где-либо введены производные размерности, например, как на рис. 11.6, и вы хотите видеть результат в этой размерности, то сделайте следующее:

- ❑ установите курсор в поле ввода справа от выведенной размерности и введите производную размерность. При этом исчезнет не только старая размерность, но и сам числовой результат расчета;
- ❑ щелкните мышью вне выражения. Появится новое число, соответствующее введенной вами размерности, и сама размерность (рис. 11.9).

В поле вывода числа в черный квадратик вписать свою введенную размерность--результат изменится.

$$t := 2 \cdot \text{мин} \quad P := 3 \cdot \text{кН} \quad a := 50 \cdot \text{см} \quad g := 9.81 \cdot \frac{\text{метр}}{\text{сек}^2}$$

$$v := g \cdot t \quad M_{\text{изг}} := P \cdot a$$

$$v = 1177.2 \text{ метр сек}^{-1} \quad M_{\text{изг}} = 1500 \frac{\text{кг метр}^2}{\text{сек}^2} \quad \text{размерность, введенная в меню}$$

$$v = 1177 \frac{\text{метр}}{\text{сек}} \quad M_{\text{изг}} = 1500 \frac{\text{кг метр}^2}{\text{сек}^2} \quad \text{размерность форматирована}$$

$$v = 4237.92 \frac{\text{км}}{\text{час}} \quad M_{\text{изг}} = 1.5 \text{ кН} \cdot \text{м} \quad \text{вписана своя размерность}$$

дробная размерность

$$Z := \sqrt[5]{M \cdot \frac{v^2}{a}} \quad Z = 106.743 \text{ кг}^{0.2} \text{ метр}^{0.2} \text{ сек}^{-0.4}$$

Рис. 11.9. Вывод результатов расчета в различных размерностях

В зависимости от вычисляемого выражения размерность может быть как в степени, обозначенной целым числом, так и в дробной степени.

Можно включить автоматический перевод единиц измерения в более простые формы:

- Дважды щелкните мышью на результате вычислений — откроется диалоговое окно Result Format (Формат результата).
- Выберите в нем команду Unit Display (Отображение единиц измерения).
- Установите оба флажка, Format Units (Формат размерностей) и Simplify units when possible (Упрощать размерности, если возможно).
- Если вы щелкнете на кнопке ОК, то к выделенному выражению будет применено упрощение размерностей. При выборе Set as default (Установить по умолчанию) упрощение размерностей будет применено ко всему документу.

11.4. Учет внесистемных размерностей

Иногда бывает удобно вести расчеты в своих собственных единицах измерения, которых нет в системе СИ, например, подсчитать количество мешков, людей, машин и т. д. Такие размерности назовем внесистемными. Их можно обозначить как константы.

$$\begin{aligned}
 m &:= m \text{ человек} := 1 & \text{квартира} &:= 60 \cdot \text{м}^2 & \text{дом} &:= 200 \cdot \text{квартира} \\
 \text{в городе проживает} & M := 10^6 \cdot \text{человек} \\
 \text{в городе} & N := 2000 \cdot \text{дом} \\
 \text{В среднем по городу} \\
 \text{жилплощадь на 1 человека} & S := \frac{N}{M} & S &= 24 \frac{\text{м}^2}{\text{человек}} \\
 \text{Проживает людей} & m := \frac{M}{N} & \text{в одной квартире} & m = 2.5 \frac{\text{человек}}{\text{квартира}} \\
 & & \text{в одном доме} & m = 500 \frac{\text{человек}}{\text{дом}}
 \end{aligned}$$

Рис. 11.10. Учет внесистемных размерностей

На рис. 11.10 приведен пример расчета плотности населения в городе. В качестве единиц измерения взяты человек, квартира, дом. Если квартира измеряется в квадратных метрах, то человек — внесистемная размерность. Размерность плотности заселения города m вписана в черный квадратик ввода рядом с числом.

11.5. Размерность элементов массива

Размерность элементов массива вводится так же, как и размерность чисел, умножением выражения на стандартную размерность на английском языке или производную размерность, введенную ранее. Действия с размерными массивами показаны на рис. 11.11.

$$A := \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 5 \end{pmatrix} \cdot \text{кН} \quad A = \begin{pmatrix} 1000 & 2000 & 3000 \\ 2000 & 3000 & 4000 \\ 3000 & 4000 & 5000 \end{pmatrix} \frac{\text{кг метр}}{\text{сек}^2}$$

$$V := \begin{pmatrix} 20 \\ 40 \\ 60 \end{pmatrix} \cdot \text{мм} \quad V = \begin{pmatrix} 0.02 \\ 0.04 \\ 0.06 \end{pmatrix} \text{метр} \quad V = \begin{pmatrix} 2 \\ 4 \\ 6 \end{pmatrix} \text{см}$$

$$A \cdot V = \begin{pmatrix} 280 \\ 400 \\ 520 \end{pmatrix} \text{кг метр}^2 \text{сек}^{-2} \quad A \cdot V = \begin{pmatrix} 280 \\ 400 \\ 520 \end{pmatrix} \frac{\text{кг метр}^2}{\text{сек}^2}$$

$$A \cdot V = \begin{pmatrix} 0.28 \\ 0.4 \\ 0.52 \end{pmatrix} \text{метр кН}$$

Рис. 11.11. Работа с размерными массивами

Массив A введен в килоньютонах, массив V — в миллиметрах. При выводе результата Mathcad использует размерности, вписанные на вкладке Dimensions (Размерности), — $\text{кг} \cdot \text{м}^2 \cdot \text{с}^{-2}$. После установки флажка Format units (Формат размерностей) размерность на экране меняет вид на $\frac{\text{кг} \cdot \text{М}^2}{\text{с}^2}$. Для ввода производной размерности «кН» надо в дополнительное место ввода вписать «кН», при этом численный результат исчезает. Щелчок мышью вне выражения — и результат пересчитан в метр · кН.

ВНИМАНИЕ

Размерности всех элементов массива должны быть одинаковыми.

$$t := 0..3 \quad w := 0.12 \quad V(t) := w \cdot t \cdot \frac{\text{км}}{\text{hr}}$$

	$V(t) =$		$V(t) =$								
размерность	<table border="1" style="display: inline-table;"><tr><td>0</td></tr><tr><td>0.033</td></tr><tr><td>0.067</td></tr><tr><td>0.1</td></tr></table> метр сек ⁻¹	0	0.033	0.067	0.1		<table border="1" style="display: inline-table;"><tr><td>0</td></tr><tr><td>0.12</td></tr><tr><td>0.24</td></tr><tr><td>0.36</td></tr></table> км час	0	0.12	0.24	0.36
0											
0.033											
0.067											
0.1											
0											
0.12											
0.24											
0.36											
стандартная		введенная									

или вот так $t := 0,40 \cdot \text{мин}..2 \cdot \text{час} \quad w := 0.12 \cdot \frac{\text{метр}}{\text{сек}^2} \quad V(t) := w \cdot t$

$t =$	$t =$	$V(t) =$	$V(t) =$																
<table border="1" style="display: inline-table;"><tr><td>0</td></tr><tr><td>2400</td></tr><tr><td>4800</td></tr><tr><td>7200</td></tr></table> сек	0	2400	4800	7200	<table border="1" style="display: inline-table;"><tr><td>0</td></tr><tr><td>0.667</td></tr><tr><td>1.333</td></tr><tr><td>2</td></tr></table> час	0	0.667	1.333	2	<table border="1" style="display: inline-table;"><tr><td>0</td></tr><tr><td>288</td></tr><tr><td>576</td></tr><tr><td>864</td></tr></table> метр сек ⁻¹	0	288	576	864	<table border="1" style="display: inline-table;"><tr><td>0</td></tr><tr><td>1036.8</td></tr><tr><td>2073.6</td></tr><tr><td>3110.4</td></tr></table> км час	0	1036.8	2073.6	3110.4
0																			
2400																			
4800																			
7200																			
0																			
0.667																			
1.333																			
2																			
0																			
288																			
576																			
864																			
0																			
1036.8																			
2073.6																			
3110.4																			

Рис. 11.12. Ввод и вывод размерной функции

11.6. Учет размерностей в цикле

Дискретная переменная может быть размерной. При использовании ее размерность можно учесть, поместив дискретную переменную выше определения функции и введя размерность в определение функции (рис. 11.12) или записав дискретную переменную с помощью размерных величин. Можно также в дискретной переменной умножить числа на соответствующую размерность.

При выводе результата расчета появляется размерность, введенная пользователем на вкладке Dimensions (Размерности) (на русском языке). Для ввода производной размерности надо вписать ее в место ввода справа от выведенной размерности.

11.7. Построение графиков с учетом размерностей

Графики строятся в стандартных размерностях, приведенных в списке встроенных функций. Ввести пользовательские размерности на графиках нельзя. Для изменения масштаба графика разделите на графике названия аргумента и функции на размерность (рис. 11.13).

Выполните самостоятельно упражнения с размерностями, приведенные на рис. 11.14.

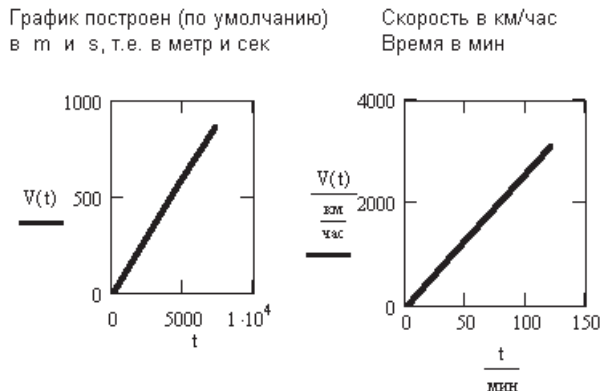


Рис. 11.13. Учет размерности при построении графика

Дано:

$$a := 50 \cdot \text{см} \quad b := 20 \cdot \text{мм} \quad h := 40 \cdot \text{мм} \quad P := 3 \cdot \text{кН}$$

$$q := 10 \cdot \frac{\text{кН}}{\text{м}} \quad M := 20 \cdot \text{кН} \cdot \text{м} \quad E := 2 \cdot 10^5 \cdot \text{МПа}$$

самостоятельно поиграйте с размерностями

$$F := 2 \cdot b^2 \quad J := \frac{b \cdot h^3}{12} \quad W := \frac{b \cdot h^2}{6}$$

$$M := P \cdot a \quad \sigma_{\text{раст}} := \frac{P}{F} \quad \sigma_{\text{изг}} := \frac{M}{W} \quad f := \frac{P \cdot a^3}{3 \cdot E \cdot J}$$

Рис. 11.14. Упражнения для самостоятельной работы с размерностями

11.8. Учет размерностей в программе

Размерности в программном модуле можно учесть, если все выводимые из модуля данные имеют одинаковую размерность. Примеры использования размерных величин в программах приведены в разделе 9.9 и в примерах 5 и 19 электронной книги.

При составлении сложных программ, когда из программного модуля нужно вывести несколько величин (скаляров или массивов) различной размерности, следует выполнять расчет без учета размерностей. В этом случае программный модуль получается компактным, сравнительно простым и понятным. При учете размерностей придется составлять несколько программных модулей и из каждого модуля выводить величины одной размерности. Расчет получается значительно более сложным.

При составлении простых программ, где выводится только одна величина, использование размерностей допустимо и только украшает программу.

11.9. Ограничения на проведение расчетов с учетом размерностей

- ❑ При расчетах с матрицами все элементы используемых в расчетах матриц должны иметь одинаковую размерность.
- ❑ Ряд встроенных функций не могут работать с размерностями. Это функции:

$$H \equiv N \quad m \equiv m \quad \text{МПа} \equiv 10^6 \cdot \frac{H}{m^2} \quad \text{цикл} := 1$$

сталь 40X $\sigma_B := 1000 \cdot \text{МПа}$ $S_K := 2 \cdot \sigma_B$ $\sigma_{СИМ} := 0.5 \cdot \sigma_B$ $N_0 := 3 \cdot 10^6 \cdot \text{цикл}$

Построение усталостной кривой в логарифмических координатах

$$\lg \sigma(x) := \text{interp}(X, Y, x)$$

$$\lg N := 0.. \lg(N_0)$$

$$X := \begin{pmatrix} \lg(1) \\ \lg(N_0) \end{pmatrix} \quad Y := \begin{pmatrix} \lg\left(\frac{S_K}{\text{МПа}}\right) \\ \lg\left(\frac{\sigma_{СИМ}}{\text{МПа}}\right) \end{pmatrix}$$

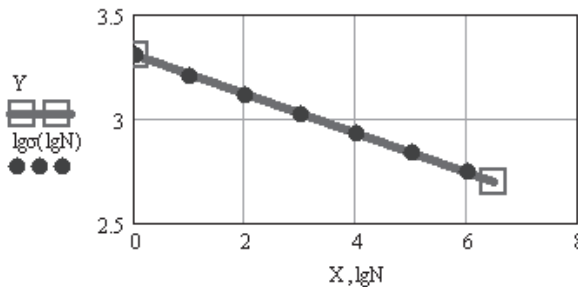


Рис. 11.15. Вычисление логарифма размерного числа

- интерполяции, регрессии, сглаживания;
- Odesolve (решение дифференциальных уравнений);
- ln, log (вычисление логарифма числа).

В таких случаях нужно вводить безразмерные аргументы функции, разделив размерную величину на ее размерность (рис. 11.15).

Примеры инженерных расчетов с учетом размерностей приведены в части IV (примеры 5, 9–20, 47, 49, 53).

11.10. Статическая проверка размерностей

В Mathcad 12, 13 и 14 ужесточен контроль правильности использования размерностей. Применена техника статической проверки размерностей, запрещающая расчет функций, которые в зависимости от значения аргумента могут возвращать результат различной размерности. Например, $f(0) = 1m^2$, а $f(1) = 1m^3$.

В Mathcad 13 встроены функции статической проверки размерностей. До вычислений Mathcad проверяет размерности всех введенных величин слева направо и при наличии ошибок (реальных или кажущихся) выводит сообщение о том, какими размерности должны быть (рис. 11.16).

$$a := 2 \cdot m \quad b := 3 \cdot m$$

$$D := a + b^2$$

Значение имеет единицы измерения: длина^2,
а должно иметь единицы измерения: длина.

Рис. 11.16. Результат статической проверки размерностей

Другой способ анализа размерностей состоит в том, чтобы записать имя функции без указания параметров. В Mathcad 11 в этом случае выводилось слово «function». В Mathcad 13 при написании имени функции без параметров выводится перечень размерностей, использованных в функции (рис. 11.17). На этом рисунке any1, any2 и any3 обозначают размерность первого, второго и третьего параметров, указанных в имени функции. Индекс соответствует степени этой величины в правой части выражения. Если параметром функции является другая функция, то в перечне параметров появляется f (список параметров) (рис. 11.17, выражение для $\Delta 1$).

После знака равенства появляется ожидаемая размерность результата расчета выражения. Для величин, входящих в правую часть выражения, указывается конкретный класс размерности (давление, масса, длина, время). Кроме указанных видов записи в сообщениях встречаются слова *unitless* — безразмерный и $[any1]$ — размерный массив.

Для практических расчетов статическая проверка размерностей — вещь не только бесполезная, но и вредная. В ряде примеров инженерных расчетов, приведенных

$$\sigma(b, h, F) := \frac{6 \cdot F \cdot L}{b \cdot h^2} \quad \Delta(b, h) := \frac{4 \cdot F \cdot L^3}{E \cdot b \cdot h^3} \quad \Delta1(b, h, \sigma) := \frac{4 \cdot F \cdot L^3}{E \cdot b \cdot h^3} \cdot \sigma(b, h, F)$$

$$\sigma = f(\text{any1}, \text{any2}, \text{any3}) \Rightarrow \text{any1}^1 \cdot \text{any2}^2 \cdot \text{any3} \cdot \text{Length}$$

$$\Delta = f(\text{any1} \cdot \text{any2}^3 \cdot \text{Pressure}^1, \text{any2}^4) \Rightarrow \text{any1}^1 \cdot \text{Mass} \cdot \text{Length}^4 \cdot \text{Time}^2$$

$$\Delta1 = f(\text{any1}, \text{any2}, f(\text{any1}, \text{any2}, \text{Force}) \Rightarrow \text{any3}) \Rightarrow \text{any1}^1 \cdot \text{any2}^3 \cdot \text{any3} \cdot \text{Length}^5$$

Рис. 11.17. Статическая проверка размерностей функции без параметров

$$K(y, \phi) := \begin{pmatrix} \Delta r(p, r_0, \delta, y) \cdot 10^{2.8} + r(y) \\ \phi \\ y \end{pmatrix} \begin{array}{l} \text{поверхность цилиндрической} \\ \text{оболочки после деформации} \\ \\ S := \text{CreateMesh}(K, 0, L, 0, 2 \cdot \pi, \text{fmap}) \end{array}$$

$$K1(y, \phi) := \begin{pmatrix} r(y) \\ \phi \\ y \end{pmatrix} \begin{array}{l} \text{поверхность цилиндрической} \\ \text{оболочки до деформации} \\ \\ S1 := \text{CreateMesh}(K1, 0, L, 0, 2 \cdot \pi, \text{fmap}) \end{array}$$

Значение имеет форму: (длина, длина, длина, Unitless), а должно иметь форму: Unit^4 длина.

Рис. 11.18. Фрагмент примера 44 из электронной книги

в электронной книге, функции, в том числе встроенные, хорошо работающие в Mathcad 11, не работают в Mathcad 13 (рис. 11.18). Если в ранних версиях Mathcad ряд функций (regress, Odesolve, CreateMech и др.) допускали использование размерных величин, когда размерность совпадала с базовой в системе СИ, то сейчас с этим строго: в Mathcad 13 никакие размерные величины в таких функциях уже не допускаются.

Подводя итоги этой главы, следует заметить, что учет размерностей несложен для простых задач, где с ними и так все ясно. В сложных расчетах при использовании многих функций и нескольких дискретных переменных учет размерностей может оказаться сложным, а в ряде случаев вообще невозможным. В таких случаях размерность можно указать рядом с результатом расчета в виде текстового комментария (для справки).

11.11. Квазиразмерности

Как следует из предыдущих разделов этой главы, учет размерностей очень прост в простых примерах и очень сложен в сложных примерах. Многочисленные обращения к фирме Mathsoft с предложениями об уменьшении числа ограничений на использование размерностей в расчетах остались без ответа. Мне пришла в голову идея, как упростить использование размерностей в расчетах. По сути

дела процесс присвоения числу размерности состоит в умножении числа на некоторый числовой коэффициент. Буквенному обозначению размерности достаточно присвоить численное значение, не умножая его на стандартную размерность.

На рис. 11.19 показан пример учета квазиразмерностей в расчетах. Вначале производится присвоение квазиразмерностям численного значения. Далее записаны исходные данные, умноженные на квазиразмерности. Затем введены расчетные формулы и выведены результаты расчета:

- без учета размерностей, то есть в системе СИ (в ньютонах и в метрах);
- с учетом размерностей, для этого в черный квадратик ввода размерностей (справа от числа) вписана размерность.

На графиках, которые в Mathcad по определению не работают с размерностями и выводят результаты в системе СИ, имена переменной и функции делятся на размерность, как обычно.

Квазиразмерность -- это просто числовой коэффициент при стандартной размерности в системе СИ

$$m := 1 \text{ см} \equiv 0.01 \text{ мм} \equiv 0.001 \text{ кН} \equiv 1000 \text{ кНм} \equiv 1000 \text{ МПа} \equiv 10^6$$

Исходные данные, умноженные на квазиразмерности

$$F := 3 \cdot \text{кН} \quad L := 2 \cdot \text{м} \quad d := 12 \cdot \text{см} \quad E := 2 \cdot 10^5 \cdot \text{МПа} \quad q := 2 \cdot \frac{\text{кН}}{\text{м}}$$

Расчетные формулы

$$J := 0.05 \cdot d^4 \quad W := 0.1 \cdot d^3 \quad M := q \cdot L^2 \quad \sigma := \frac{M}{W} \quad \Delta := \frac{F \cdot L^3}{3 \cdot E \cdot J}$$

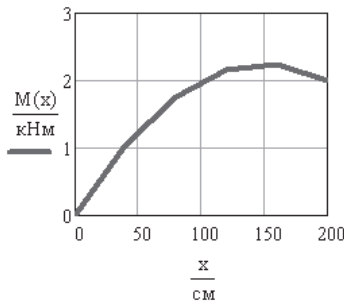
Результаты расчета без учета размерностей, то есть в системе СИ

$$J = 1.037 \times 10^{-5} \quad W = 1.728 \times 10^{-4} \quad M = 8000 \quad \sigma = 46296296 \quad \Delta = 0.004$$

Результаты расчета с учетом квазиразмерностей

$$J = 1036.8 \cdot \text{см}^4 \quad W = 172.8 \cdot \text{см}^3 \quad M = 8 \cdot \text{кНм} \quad \sigma = 46 \cdot \text{МПа} \quad \Delta = 3.858 \cdot \text{мм}$$

Ввод функции и построение графика $M(x) := F \cdot x - \frac{q \cdot x^2}{2} \quad x := 0, \frac{L}{5} \dots L$



$M(x) =$	$M(x) =$	
0	0	· кНм
1040	1.04	
1760	1.76	
2160	2.16	
2240	2.24	
2000	2	

Рис. 11.19. Учет квазиразмерностей в расчетах

ПРИМЕЧАНИЕ

Стандартный ввод размерности — производной размерности присваивается число, умноженное на стандартную размерность.

Ввод квазиразмерности — производной размерности присваивается число.

Использование квазиразмерностей позволяет избежать всех ограничений, перечисленных в разделе 11.9, а также использовать размерности в программах и матричных расчетах.

При использовании квазиразмерностей Mathcad уже не следит за правильностью использования размерностей и вся ответственность за результаты расчета ложится на расчетчика, который сам должен следить за логикой расчета и вписывать в поле ввода ответа правильную размерность.

Большинство примеров в этой книге разрабатывались постепенно в течение ряда лет. Все они могут быть легко перестроены с использованием квазиразмерностей. Необходимости в этом нет, но в дальнейшем я постараюсь использовать в расчетах именно квазиразмерности.

Глава 12

Дополнительные встроенные функции Mathcad

В предыдущих главах рассказывалось о наиболее часто используемых в работе встроенных функциях. В этой главе речь пойдет о функциях, сравнительно редко используемых, поэтому описание функций дано в краткой форме с некоторыми примерами. Дополнительные сведения о работе с этими функциями можно найти в разделе Help (Помощь) главного меню Mathcad. Проще всего найти описание нужной функции, нажав на стандартной панели инструментов Mathcad кнопку Insert Function (Вставить функцию) со значком $f(x)$. Найдите в открывшемся списке нужную функцию и щелкните на ней мышью. Прочитайте внизу описание функции. Для получения дополнительных сведений нажмите кнопку ? (знак вопроса) — откроется раздел справки, посвященный этой функции. В конце каждого раздела есть ссылки на сопутствующие разделы и на примеры использования функций. Совершив обзор встроенных функций. Откроем в стандартном меню Mathcad пункт Insert Function и рассмотрим основные группы функций. Полагаем, что x — вещественное безразмерное число, $x > 0$.

ПРИМЕЧАНИЕ

Для удобства работы с матрицами во всех примерах этой главы $\text{ORIGIN} = 1$.

12.1. Функции для работы с комплексными числами

Комплексные числа в Mathcad можно ввести тремя способами:

- в виде суммы вещественной и мнимой частей, $C := a + bi$;
- в тригонометрической форме, $C := A(\cos(\varphi) + ci \cdot \sin(\varphi))$;
- в экспоненциальной форме, $C := Ae^{ic}$.

Ввод мнимой единицы 1i или 1j или $i := \sqrt{-1}$

Способ ввода комплексного числа

$$C = a + bi = A \cdot (\cos(\varphi) + ci \cdot \sin(\varphi)) = A \cdot e^{i \cdot c}$$

Примеры ввода комплексных чисел

$$a := 3 \quad b := -5 \quad c := 12 \quad A := 120 \quad \varphi := 20 \cdot \text{deg}$$

$$C := a + bi \quad C = 3 - 5i$$

$$C := A \cdot e^{i \cdot c} \quad C = 101.3 - 64.4i$$

$$C := A \cdot [\cos(\varphi) + c \cdot (i \cdot \sin(\varphi))] \quad C = 112.8 + 492.5i$$

$$|C| = 505.3 \quad \text{модуль комплексного числа}$$

$$\arg(C) = 1.3 \quad \arg(C) = 77.1 \cdot \text{deg} \quad \text{аргумент комплексного числа}$$

Рис. 12.1. Ввод комплексных чисел

Чаще всего комплексное число возникает в расчетах как результат вычисления какого-либо выражения. Примеры ввода комплексных чисел приведены на рис. 12.1.

ВНИМАНИЕ

Напомним, для ввода мнимой единицы надо ввести 1i или 1j. После вывода курсора из области числа единица исчезает и Mathcad воспринимает число как комплексное.

Mathcad содержит несколько функций для работы с комплексными числами:

- ❑ $\text{Re}(z)$ — действительная часть комплексного числа z ;
- ❑ $\text{Im}(z)$ — мнимая часть комплексного числа z ;
- ❑ $\text{arg}(z)$ — аргумент комплексного числа z , $-\pi < \text{arg}(z) \leq \pi$;
- ❑ $\text{csgn}(z)$ — функция комплексного знака числа (возвращает либо 0, если $z = 0$; либо 1, если $\text{Re}(z) > 0$ или $\text{Re}(z) = 0$ и $\text{Im}(z) > 0$; либо -1 — в остальных случаях);
- ❑ $\text{signum}(z)$ — возвращает 1, если $z = 0$, и $z/|z|$ — в остальных случаях.

Здесь z — вещественное, мнимое или комплексное число.

Примеры использования указанных функций приведены на рис. 12.2.

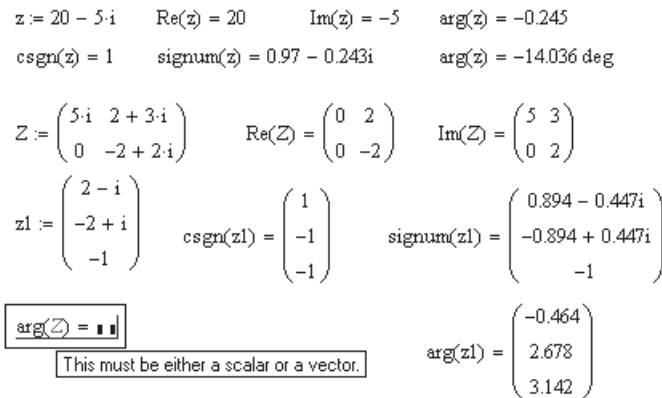


Рис. 12.2. Функции для работы с комплексными числами

Функции Re и Im могут работать со скалярами, векторами и матрицами. Остальные функции с матрицами работать не могут.

Все функции, содержащиеся в двух последующих разделах меню Insert Function, — Curve Fitting (Функции регрессии) и Differential Equation Solving (Функции решения дифференциальных уравнений) — были подробно рассмотрены в главах 2, 6 и 7, поэтому переходим сразу к разделу Expression Type (Тип выражения).

12.2. Тип выражения

Изначально в этом разделе было четыре функции:

- ❑ $\text{IsArray}(x)$ — возвращает 1, если x — матрица или вектор, иначе возвращает 0;

- `IsScalar(x)` возвращает 1, если x — вещественный или комплексный скаляр, иначе возвращает 0;
- `IsString(x)` возвращает 1, если x — строка, иначе возвращает 0;
- `SiUnitsOf(x)` возвращает размерность x , введенную в меню Dimension. Если x — безразмерная величина, то возвращает 1.

В Mathcad 14 к этим функциям добавились еще 2:

- `IsNaN` возвращает 1, если выражение не число, иначе возвращает 0. Может использоваться только при символьных вычислениях;
- `IsPrime` возвращает 1, если число простое, иначе возвращает 0. Может использоваться только при символьных вычислениях.

Смысл этих функций понятен из их описания.

12.3. Доступ к файлам

В разделе File Access (Доступ к файлам) 27 функций. Четыре из них являются чрезвычайно важными при разработке сложных программ. Это функции `READPRN`, `WRITEPRN`, `APPENDPRN` и `READFILE` — функции передачи данных в Mathcad из других файлов и из Mathcad в другие файлы:

- `APPENDPRN("file")` добавляет числовые данные в существующий файл. Здесь "file" — имя файла или путь к файлу, если он находится в другом каталоге;
- `READPRN("file")` считывает числовые данные из файла;
- `WRITEPRN("file")` записывает числовые данные в файл;
- `READFILE("file", "type", [colwidths], [rows], [cols], [emptyfill])` считывает данные из других файлов.

Остальные функции предназначены для обработки графических и музыкальных файлов. Эти функции здесь не рассматриваются. С их описанием и применением можно познакомиться в Mathcad Help и в Resource Center.

Хотя в меню Insert Function отсутствуют старые функции `APPEND("file")`, `READ("file")` и `WRITE("file")`, но Mathcad понимает и допускает их использование.

Если функции `APPENDPRN`, `READPRN` и `WRITEPRN` добавляют, считывают и записывают в файл прямоугольную матрицу (число — частный случай матрицы размерностью 1×1), то функции `APPEND`, `READ` и `WRITE` добавляют, считывают и записывают одно число.

Функции `APPENDPRN`, `READPRN`, `WRITEPRN` и `READFILE` позволяют обмениваться данными с другими программами, создавая вычислительные комплексы из нескольких различных программ. Примеры использования этих функций приведены в главе 2 и на рис. 12.3. Создание вычислительного комплекса показано в четвертой части книги, в примере 55.

ПРИМЕЧАНИЕ

При считывании матрицы функцией READPRN текст (чаще всего заголовки таблиц) игнорируется, принимаются только числа.

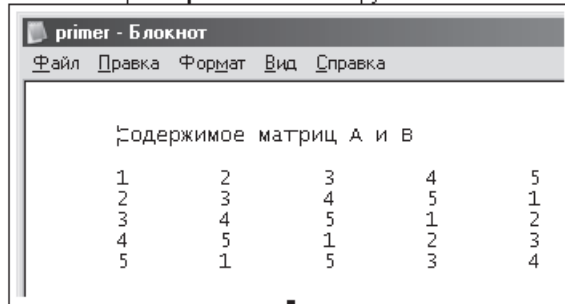
ВНИМАНИЕ

При вводе в название функции имени файла не забывайте ставить кавычки. Имя файла вводите с расширением, так, как оно записано в каталоге. Расширение может быть любым, а не только .pm, или отсутствовать вообще.

На рис. 12.3 показаны запись матрицы A^T в файл primer.prn, добавление к нему матрицы B^T , считывание данных из этого файла в матрицу K (заголовок таблицы при этом пропущен) и присвоение разным элементам матрицы K разных имен. Такая процедура позволяет записывать в один файл и считывать из одного файла много различных данных.

$$\begin{aligned}
 A &:= \begin{pmatrix} 1 & 2 \\ 2 & 3 \\ 3 & 4 \\ 4 & 5 \\ 5 & 1 \end{pmatrix} & B &:= \begin{pmatrix} 3 & 4 & 5 \\ 4 & 5 & 1 \\ 5 & 1 & 5 \\ 1 & 2 & 3 \\ 2 & 3 & 4 \end{pmatrix} & \text{WRITEPRN("primer")} &:= A^T \\
 & & & & \text{APPENDPRN("primer.prn")} &:= B^T \\
 & & & & K &:= \text{READPRN("primer.prn")}
 \end{aligned}$$

Заголовок файла **primer** вписан от руки



Заголовок файла **primer** игнорируется при чтении

$$\begin{aligned}
 K &= \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 4 & 5 & 1 \\ 3 & 4 & 5 & 1 & 2 \\ 4 & 5 & 1 & 2 & 3 \\ 5 & 1 & 5 & 3 & 4 \end{pmatrix} & C1 &:= \text{submatrix}(K, 4, 5, 3, 5) \\
 & & C1 &= \begin{pmatrix} 1 & 2 & 3 \\ 5 & 3 & 4 \end{pmatrix} & C2 &:= C1^{\langle 1 \rangle} & C2 &= \begin{pmatrix} 1 \\ 5 \end{pmatrix} \\
 & & C3 &:= K_{3,4} & C3 &= 1
 \end{aligned}$$

Рис. 12.3. Обмен данными между файлами

Для чтения данных из других файлов в Mathcad 13 появилась новая универсальная функция READFILE. Эта функция объединяет в себе почти все другие существовавшие ранее способы чтения файлов.

Обращение к этой функции:

$$\text{READFILE}(\text{"file"}, \text{"type"}, [\text{colwidths}], [\text{rows}], [\text{cols}], [\text{emptyfill}]).$$

Параметры функции:

- ❑ `file` — имя файла, из которого считываются данные (файл должен быть помещен в тот же каталог, в котором находится и считывающий файл);
- ❑ `type` — тип файла. В кавычках, как и имя файла, надо вписать одно из следующих ключевых слов:
 - `delimited` — колонки данных с любым разделителем между ними. Это файлы с расширениями `.csv`, `.txt`, `.prn`;
 - `Fixed` — таблицы с фиксированной шириной колонок;
 - `Excel` — файлы формата Excel.

Первые два параметра функции `READFILE` являются обязательными, остальные можно пропустить.

- ❑ `Colwidths` — ширина колонок для таблиц с фиксированной шириной колонок. Задается в виде числа считываемых позиций в колонке, если ширина всех колонок одинакова, или в виде вектора чисел, если ширина колонок разная.
- ❑ `Rows` — задается одним числом, указывающим, начиная с какого ряда надо считать данные, или вектором из двух чисел, указывающих, с какого по какой ряд данные надо считать. Если `rows` не задан, считываются все ряды.
- ❑ `Cols` — то же, что для `rows`, но для колонок таблицы данных.
- ❑ `Emptyfill` — указывает Mathcad, какой текст вывести при отсутствии данных в одной из ячеек таблицы. По умолчанию (при отсутствии параметра `emptyfill`) вводится `NAN` (НеЧисло).

Примеры использования функции `READFILE` с необязательными параметрами и без них приведены на рис. 12.4.

В прежних версиях Mathcad для выполнения тех же операций с файлами данных требовалось в главном меню выбрать `Insert` ▶ `Data` ▶ `Data Import Wizard` (Вставить ▶ Данные ▶ Мастер импорта данных) и далее следовать указаниям мастера.

Для сравнения на рис. 12.5 показано считывание данных с использованием `Data Import Wizard`.

`A := archiv\адреса-2005.csv` Выбраны ряды 6 - 8 адресной книги

`A1 := Мои документы\ИРА\Книга1.xls` Документ Excel целиком

$A = \begin{pmatrix} \text{"Mathcad"} & \text{"collabadmin@mathsoft.com"} \\ \text{"Mathsoft"} & \text{"help@mathsoft.co.uk"} \\ \text{"Svetlana Ilyicheva"} & \text{"SvetlanaI@softline.ru"} \end{pmatrix}$

$A1 =$

	1	2	3
1	1	4	0.514
2	1	6	0.539
3	1	8	0.643
4	1	9	0.727
5	1	10	0.724

Рис. 12.4. Импорт данных с помощью функции `READFILE`

$ZZ1 := \text{READFILE}(\text{"адреса-2005.csv"}, \text{"delimited"})$ вся книга целиком
 $ZZ2 := \text{READFILE}(\text{"адреса-2005.csv"}, \text{"delimited"}, \begin{pmatrix} 6 \\ 8 \end{pmatrix})$ строки с 6 по 8
 $ZZ3 := \text{READFILE}(\text{"GU.PRN"}, \text{"delimited"})$ весь файл целиком
 $ZZ4 := \text{READFILE}(\text{"GU.PRN"}, \text{"delimited"}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix})$ строки с 2 по 3 и столбцы с 2 по 3
 $AA := \text{READFILE}(\text{"Книга1.xls"}, \text{"Excel"})$ Документ Excel целиком

$ZZ2 = \begin{pmatrix} \text{"Mathcad"} & \text{"collabadmin@mathsoft.com"} \\ \text{"Mathsoft"} & \text{"help@mathsoft.co.uk"} \\ \text{"Svetlana Ilyicheva"} & \text{"Svetlanal@softline.ru"} \end{pmatrix}$ $ZZ4 = \begin{pmatrix} 3 & 0 \\ 5 & 0 \end{pmatrix}$

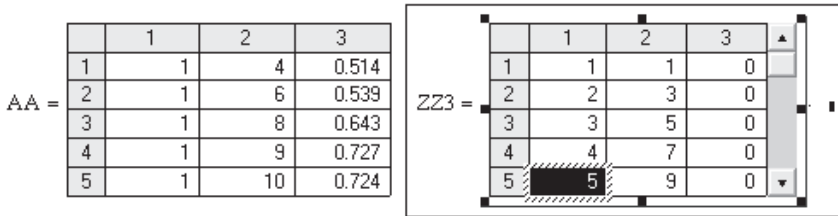


Рис. 12.5. Импорт данных с помощью Data Import Wizard

12.4. Функции поиска значений матрицы

В Mathcad имеется четыре функции поиска значений матрицы:

- ❑ $\text{lookup}(z, A, B)$ в векторе или матрице A ищет заданную величину z и возвращает величины, стоящие в матрице B в той же позиции, что и z в матрице A . Если найдено несколько значений z , то возвращается вектор значений величин в строгом порядке по рядам слева направо (рис. 12.6);
- ❑ $\text{hlookup}(z, A, r)$ в первом ряду матрицы A ищет заданную величину z и возвращает величины в той же колонке в ряду r . Если найдено несколько значений z , то возвращается вектор значений величин (рис. 12.7);
- ❑ $\text{vlookup}(z, A, c)$ в первой колонке матрицы A ищет заданную величину z и возвращает величины в том же ряду в колонке c . Если найдено несколько значений z , то возвращается вектор значений величин (рис. 12.8);
- ❑ $\text{match}(z, A)$ в векторе или матрице A ищет заданную величину z и возвращает индексы, то есть номер строки и номер столбца. Если найдено несколько значений z , то возвращается вектор найденных позиций величин в строгом порядке по рядам слева направо (рис. 12.9).

В названиях функций z — действительное или комплексное число или строка; A и B — матрица или вектор; r и c — целые числа, номер строки и номер столбца. Функции hlookup и vlookup возвращают одно число. С помощью дискретной переменной в приведенных примерах организован поиск по всей матрице.

lookup выводит элементы матрицы D в той же позиции, что и число z в матрице B. По рядам, начиная с первой колонки.

$$D := \begin{pmatrix} 35 & 4 & 7 \\ 4 & 15 & 1 \\ 5 & 13 & 30 \\ 11 & 22 & 33 \\ 2 & 31 & 4 \end{pmatrix} \quad B := \begin{pmatrix} 3 & 4 & 5 \\ 4 & 5 & 1 \\ 5 & 1 & 5 \\ 1 & 2 & 3 \\ 2 & 3 & 4 \end{pmatrix} \quad z := 5 \quad \text{lookup}(z, B, D) = \begin{pmatrix} 5 \\ 15 \\ 7 \\ 30 \end{pmatrix}$$

Рис. 12.6. Пример использования функции lookup

hlookup ищет число z в первом ряду матрицы B и выводит число, стоящее в той же колонке, что и число z, в ряду номер m

$$B := \begin{pmatrix} 3 & 4 & 5 \\ 4 & 5 & 1 \\ 5 & 1 & 5 \\ 1 & 2 & 3 \\ 2 & 3 & 4 \end{pmatrix} \quad z := 5 \quad m := 1..rows(B) \quad \text{hlookup}(z, B, m) = \begin{pmatrix} (5) \\ (1) \\ (5) \\ (3) \\ (4) \end{pmatrix}$$

Рис. 12.7. Пример использования функции hlookup

vlookup ищет число z в первой колонке матрицы B и выводит число, стоящее в том же ряду, что и число z, в колонке номер n

$$B := \begin{pmatrix} 3 & 4 & 5 \\ 4 & 5 & 1 \\ 5 & 1 & 5 \\ 1 & 2 & 3 \\ 2 & 3 & 4 \end{pmatrix} \quad z := 5 \quad n := 1..cols(B) \quad \text{vlookup}(z, B, n) = \begin{pmatrix} (5) \\ (1) \\ (5) \end{pmatrix}$$

Рис. 12.8. Пример использования функции vlookup

match определяет место числа z в матрице B

$$B := \begin{pmatrix} 3 & 4 & 5 \\ 4 & 5 & 1 \\ 5 & 1 & 0 \\ 1 & 2 & 3 \\ 2 & 3 & 4 \end{pmatrix} \quad z := \max(B) \quad z = 5 \\ Z := \text{match}(z, B) \\ i := 1..rows(Z)$$

номер строки номер столбца координаты числа z

$$\begin{pmatrix} (Z_i)_1 \\ (Z_i)_2 \\ (Z_i)_3 \end{pmatrix} = \begin{pmatrix} (Z_i)_2 \\ (Z_i)_1 \\ (Z_i)_3 \end{pmatrix} = Z = \begin{pmatrix} (2,1) \\ (2,1) \\ (2,1) \end{pmatrix} \quad Z = \begin{pmatrix} (3) \\ (1) \\ (2) \\ (2) \\ (1) \\ (3) \end{pmatrix}$$

3	1
2	2
1	3

Рис. 12.9. Пример использования функции match

Функция `match` для каждой найденной позиции числа z выводит две координаты (номер строки и номер столбца) в виде составного массива. Чтобы раскрыть составной массив, нужно:

- ❑ написать имя выражения и нажать клавишу `=`. Появится ответ в виде составного массива, где указаны только размерности вложенных массивов;
- ❑ дважды щелкнуть мышью на составном массиве. Появится меню `Result Format`;
- ❑ в этом меню открыть страницу `Display Options` (Параметры изображения) и на ней установить флажок `Expand nested arrays` (Развертывать вложенные массивы).

На рис. 12.9 показано, как с помощью дискретной переменной вывести отдельно два вектора координат (по строкам и по столбцам).

12.5. Преобразование функций и матриц

Преобразование функции в матрицу или матрицы в функцию необходимо по причине особенностей встроенных функций Mathcad. Это делается, например, если одни функции возвращают матрицу, а вам надо иметь функцию для последующего интегрирования или дифференцирования. Или, наоборот, Mathcad выводит функцию, а вам нужна матрица для последующих матричных преобразований.

12.5.1. Преобразование функции в матрицу

В Mathcad есть встроенная функция `matrix`, которая преобразует функцию в матрицу. Пример ее использования приведен на рис. 12.10.

Обращение к функции:

$$\text{matrix}(m, n, F)$$

возвращает матрицу размерностью $m \times n$, элементы которой — значения функции $F(i, j)$, где $i := 0..m - 1$ и $j := 0..n - 1$.

Функция `matrix` по умолчанию использует `ORIGIN = 0`. Обычно же расчетчики используют `ORIGIN = 1`. Первая строка и первый столбец матрицы обозначаются номером 1. `matrix` значение `ORIGIN = 1` не принимает, поэтому приходится вырезать нулевые значения с помощью функции `submatrix`. Получается громоздко и сложно. Кроме того, значения $x = i$ и $y = j$ — только целочисленные. Конечно, дробные значения аргументов организовать можно, записав x и y как некоторое выражение от i и j , но и в этом случае наложено ограничение: значения аргументов равноотстоят друг от друга.

На рис. 12.10 показано, как получить массив из функции, не используя довольно неудобную функцию `matrix`. С помощью дискретных переменных получается массив значений, равноотстоящих друг от друга. Указывая произвольные векторы x и y , получаем массив значений заданной функции для абсолютно произвольных аргументов, в том числе дробных.

$$\begin{aligned}
 F(x,y) &:= \sin(x \cdot y) & M &:= \text{matrix}(4,4,F) & M &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0.841 & 0.909 & 0.141 \\ 0 & 0.909 & -0.757 & -0.279 \\ 0 & 0.141 & -0.279 & 0.412 \end{pmatrix} \\
 M &:= \text{submatrix}(M,2,4,2,4) & M &= \begin{pmatrix} 0.841 & 0.909 & 0.141 \\ 0.909 & -0.757 & -0.279 \\ 0.141 & -0.279 & 0.412 \end{pmatrix} \\
 \text{аналог функции matrix} & & & & & \\
 n := 3 & \quad i := 1..n & \quad j := 1..n & \quad M_{i,j} := \sin(i \cdot j) & M &= \begin{pmatrix} 0.841 & 0.909 & 0.141 \\ 0.909 & -0.757 & -0.279 \\ 0.141 & -0.279 & 0.412 \end{pmatrix} \\
 \text{лучше, чем функция matrix} & & & & & \\
 x := \begin{pmatrix} 15 \\ 7.4 \\ 12.05 \end{pmatrix} & \quad y := \begin{pmatrix} 25 \\ 2.7 \\ 9.05 \\ 5 \end{pmatrix} & \quad i := 1.. \text{rows}(x) & \quad j := 1.. \text{rows}(y) & & \\
 M_{i,j} := \sin(x_i \cdot y_j) & & M &= \begin{pmatrix} -0.913 & 0.334 & -0.614 & -0.388 \\ 0.347 & 0.905 & -0.84 & -0.644 \\ -0.336 & 0.9 & 0.785 & -0.531 \end{pmatrix}
 \end{aligned}$$

Рис. 12.10. Преобразование функции в матрицу

В Mathcad 13 функция `Matrix` может создавать матрицы, содержащие не только действительные и мнимые числа, как в прежних версиях, но и строки, используя строчные функции, например, `Format` (рис. 12.11).

$$\begin{aligned}
 F(x,y) &:= \sin(x \cdot y) & M &:= \text{matrix}(2,3,F) & M &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0.841 & 0.909 \end{pmatrix} \\
 \text{matrix}[2,3, f(x,y) \leftarrow \sin[(x+1) \cdot (y+1)]] &= \begin{pmatrix} 0.841 & 0.909 & 0.141 \\ 0.909 & -0.757 & -0.279 \end{pmatrix} \\
 \text{matrix}(2,3, f(i,j) \leftarrow \text{identity}(i+1)) &= \begin{bmatrix} (1) & (1) & (1) \\ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} & \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} & \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \end{bmatrix} \\
 \text{matrix}(2,2, f(i,j) \leftarrow \text{format}(\{0\}, \{1\}, i, j)) &= \begin{pmatrix} "0, 0" & "0, 1" \\ "1, 0" & "1, 1" \end{pmatrix}
 \end{aligned}$$

Рис. 12.11. Функция `Matrix` использует и строчные функции

12.5.2. Преобразование матрицы в функцию

Преобразование матрицы в функцию выполняется с использованием встроенных функций интерполяции или регрессии, которые, обрабатывая массивы точек, возвращают функцию.

На рис. 12.12 показано преобразование вектора в функцию. Здесь X — вектор значений аргумента, элементы которого обязательно располагаются в возрастающем порядке, Y — вектор значений функции. В примере использована кубическая сплайн-интерполяция `lspline` со стыковкой сплайнов по прямой линии и интерполяция с регрессией функцией `regress`. Приведенный на этом рисунке график показывает совпадение значений вектора со значениями найденной функции и возможность ее интегрирования и дифференцирования. Там же показано обратное преобразование функции в матрицу.

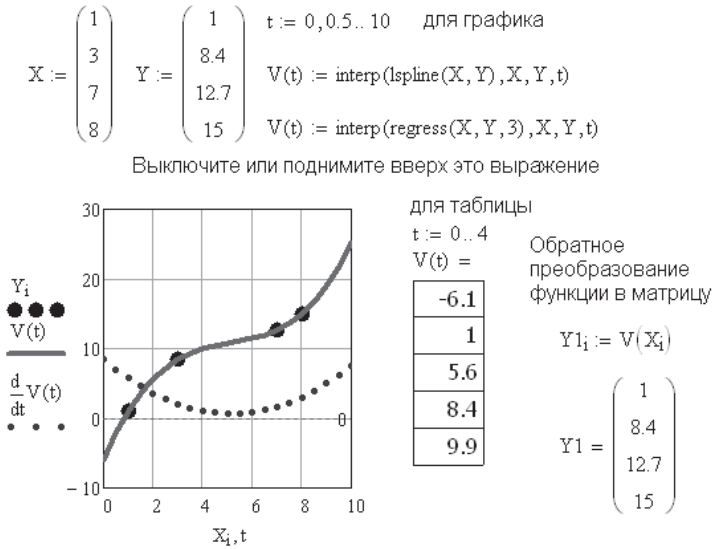


Рис. 12.12. Преобразование вектора в функцию

Преобразование матрицы в функцию двух переменных осуществляется с помощью двухмерной интерполяции (рис. 12.13), выполнение которой объяснялось в главе 7. Вверху на рис. 12.13 приведены исходные данные массива данных. В каждой конкретной задаче он уже создан, и предстоит лишь его интерполяция. Полученную интерполяцией функцию $ZZ(x, y)$ можно интегрировать и дифференцировать. Можно выводить отдельные численные значения найденной функции.

При быстром построении графика поверхности достаточно в шаблон графика вписать имя функции — и график построен (рис. 12.14). Но при этом пределы изменения аргументов берутся по умолчанию, и график получается несопоставимым с графиком исходных значений массива. Чтобы изменить масштаб графика, надо дважды щелкнув мышью на графике, открыть окно форматирования графика и выбрать вкладку `Quick Plot Data` (Данные быстрого графика), а затем ввести пределы изменения координат x и y (наибольшее и наименьшее значения элементов векторов X и Y). Обратите внимание: на левом графике значения координат — это индексы i и j . На правом графике значения координат — это координаты x и y .

Исходные данные: массив значений функции M при значениях аргументов X и Y

$$X := \begin{pmatrix} 1 \\ 3 \\ 7 \\ 8 \end{pmatrix} \quad Y := \begin{pmatrix} 1 \\ 8 \\ 12 \\ 15 \end{pmatrix} \quad M := \begin{pmatrix} 0.8 & 0.9 & -0.5 & 0.6 \\ 0.1 & -0.9 & -0.9 & 0.8 \\ 0.6 & -0.5 & 0.7 & -0.9 \\ 0.9 & 0.9 & 0.9 & 0.5 \end{pmatrix} \quad Y := \begin{pmatrix} 1 \\ 3 \\ 7 \\ 8 \end{pmatrix}$$

Только интерполяция

$XY^{(1)} := X$ $XY^{(2)} := Y$ Двухмерный массив координат точек

$$ZZ(x, y) := \text{interp} \left[\text{cspline}(XY, M), XY, M, \begin{pmatrix} x \\ y \end{pmatrix} \right]$$

Интерполяция с регрессией

Перестройка двухмерных массивов функции и ее аргументов в одномерные (по столбцам)

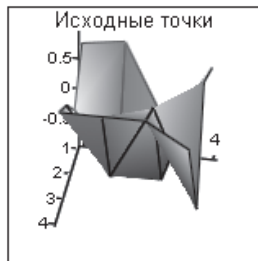
$$XY1 := \begin{pmatrix} 1 & 1 & 1 & 1 & 3 & 3 & 3 & 3 & 7 & 7 & 7 & 7 & 8 & 8 & 8 & 8 \end{pmatrix}^T$$

$$Z := (0.8 \ 0.1 \ 0.6 \ 0.9 \ 0.9 \ -0.9 \ -0.5 \ 0.9 \ -0.5 \ -0.9 \ 0.7 \ 0.9 \ 0.6 \ 0.8 \ -0.9 \ 0.5)^T$$

$$Z_{\text{regress}}(x, y) := \text{interp} \left[\text{regress}(XY1, Z, 4), XY1, Z, \begin{pmatrix} x \\ y \end{pmatrix} \right]$$

$$Z_{\text{loess}}(x, y) := \text{interp} \left[\text{loess}(XY1, Z, 0.8), XY1, Z, \begin{pmatrix} x \\ y \end{pmatrix} \right]$$

Рис. 12.13. Преобразование матрицы в функцию



Действия с функцией

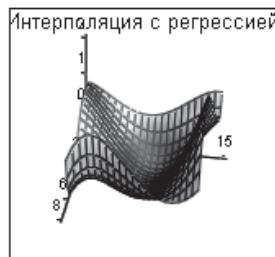
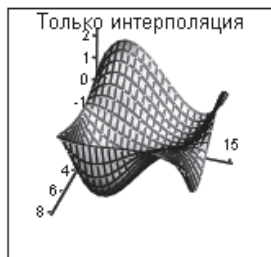
$$a(x, y) := \frac{d}{dx} ZZ(x, y)$$

$$b(x, y) := \frac{d}{dy} Z_{\text{regress}}(x, y)$$

$$a(1, 2) = -0.542$$

$$b(1, 2) = -0.151$$

M



ZZ

Z_{regress}

Рис. 12.14. Результаты преобразования матрицы M (см. рис. 12.13) в функцию

В практических задачах встречаются случаи, когда требуется изменить число элементов вектора чисел или определить элементы вектора при других (заданных) значениях аргумента. Для этого требуется вначале превратить вектор в функцию с помощью встроенной функции *interp*, затем ввести вектор новых значений аргумента и вычислить вектор значений функции для нового вектора аргумента (рис. 12.15).

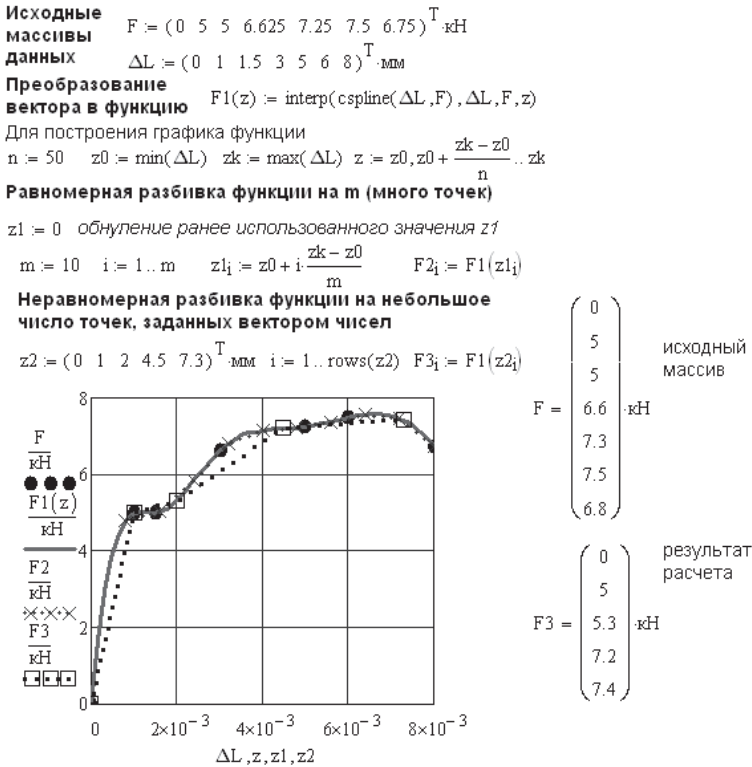


Рис. 12.15. Изменение числа элементов вектора

12.6. Функции преобразования координат

В Mathcad имеется несколько функций преобразования координат:

- $\text{xy2pol}(x, y)$ — преобразование прямоугольных координат в полярные;
- $\text{pol2xy}(r, \theta?)$ — преобразование полярных координат в прямоугольные;
- $\text{xyz2cyl}(x, y, z)$ — преобразование прямоугольных координат в цилиндрические;
- $\text{cyl2xyz}(r, \theta, z)$ — преобразование цилиндрических координат в прямоугольные;

- `xyz2sph(x, y, z)` — преобразование прямоугольных координат в сферические;
- `sph2xyz(x, θ, φ?)` — преобразование сферических координат в прямоугольные.

Эти функции используются иногда для упрощения расчетов, но чаще для построения графиков. Ряд примеров построения графиков с использованием данных функций приведен в главе 15.

ВНИМАНИЕ

Функции преобразования координат не могут работать с размерными величинами, но позволяют вводить угловые координаты как в радианах, так и в градусах. Для ввода углов в градусах надо умножить число на `deg`. Для вывода углов в градусах надо вписать `deg` в поле ввода справа от числа.

Примеры преобразования координат приведены на рис. 12.16–12.21.

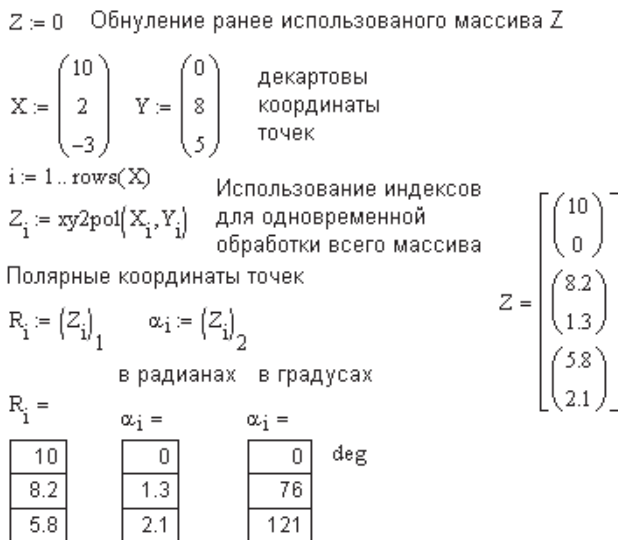


Рис. 12.16. Преобразование декартовых координат в полярные

Функции преобразования координат выводят составной массив, содержащий преобразованные координаты точек. Напомним: для раскрытия составного массива надо дважды щелкнуть на нем мышью и в открывшемся меню **Result Format** (Формат результата) выбрать команду **Display Options** ▶ **Expand nested arrays** (Параметры отображения ▶ Развертывать вложенные массивы). На рис. 12.16–12.21 показаны две различные формы работы используемых функций с векторами. По умолчанию эти функции работают только со скалярами. Для одновременного преобразования массива координат используется либо индексная форма (см. рис. 12.16 и 12.17), либо оператор векторизации (см. рис. 12.18–12.21). Использование оператора векторизации предпочтительно — так проще и элегантнее.



Рис. 12.17. Преобразование полярных координат в декартовы



Рис. 12.18. Преобразование декартовых координат в цилиндрические



Рис. 12.19. Преобразование цилиндрических координат в декартовы



Рис. 12.20. Преобразование декартовых координат в сферические



Рис. 12.21. Преобразование сферических координат в декартовы

Функции преобразования координат используются внутренним алгоритмом Mathcad при построении трехмерных графиков поверхности. Исходная функция двух переменных вводится как параметр функции `CreateMesh`. В качестве еще одного из параметров этой функции, `fmap`, используются функции преобразования координат, рассмотренные ранее. Они преобразуют полярные, цилиндрические или сферические координаты в декартовы, затем по массиву декартовых координат строится график поверхности.

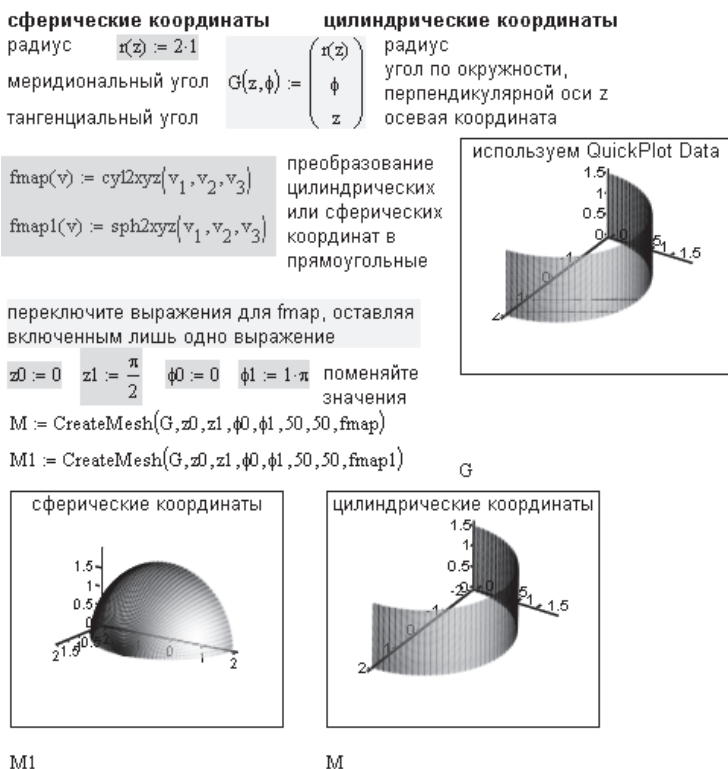


Рис. 12.22. Пример построения графика поверхности по функциям координат

На рис. 12.22 показан пример построения графика поверхности в цилиндрических и сферических координатах. Работая с примером, сделайте радиус r постоянным — получите цилиндр и шар. Тот же результат гораздо проще получить с использованием окна форматирования графика поверхности:

- На математической панели выберите `Graph` ▶ `Surface Plot` (Графика ▶ График поверхности).
- В появившемся шаблоне графика поверхности впишите внизу имя функции `G` (без параметров).
- Дважды щелкните мышью на графике и в окне форматирования `3-D Plot Format` (Формат 3D графика) выберите вкладку `QuickPlot Data` (Данные быстрого графика).

- На этой вкладке выберите систему координат Spherical (Сферическая) или Cylindrical (Цилиндрическая). По умолчанию устанавливается Cartesian (Декартова система координат).
- Впишите пределы изменения для первой (Range1 — область дискретных значений) и второй (Range2) переменных. Установите число линий, образующих сетку (# of Grids).
- Раскрасьте график с помощью вкладки Appearance (Внешний вид).

Как видно из рис. 12.22, оба способа построения графиков дают абсолютно одинаковый результат. Этого и следовало ожидать, ведь внутренний алгоритм Mathcad использует для построения графика поверхности функцию CreateMesh. Выбирая на вкладке QuickPlot Data (Данные быстрого графика) систему координат, мы фактически вызываем одну из описанных в этом разделе функций преобразования координат.

12.7. Функции условия

В Mathcad есть несколько способов записи условия. Наиболее популярные из них были рассмотрены в главах 2 и 9. Повторим еще раз три основных способа записи условия в Mathcad:

- с помощью функции условия if;
- с помощью оператора if с панели программирования;
- с использованием булевых операторов.

Для использования функции условия if нужно:

- записать имя выражения и оператор присваивания (:=);
- на стандартной панели нажать кнопку $f(x)$ и в списке встроенных функций выбрать if, после чего нажать кнопку Insert (Вставить). Появится шаблон функции if с тремя местами ввода;
- заполнить места ввода.

Обращение к функции:

$$\text{if}(\text{cond}, x, y),$$

где cond — условие типа $x \leq L_1$, x и y — значения, возвращаемые функцией. Если условие выполняется, то выражению присваивается значение x , если не выполняется, то — значение y .

Чтобы записать условный оператор с панели программирования, следует:

- записать имя выражения и оператор присваивания (:=);
- вызвать панель программирования Programming Toolbar нажатием соответствующей кнопки математической панели и щелкнуть мышью на кнопке Add line (Добавить линию);
- в верхнем поле ввода ввести выражение для рассматриваемой функции на первом участке диапазона переменной;
- щелкнуть мышью на кнопке if на панели программирования (выражение для изгибающего момента при этом должно быть полностью выделено синим

уголком или взято в скобки); появится место ввода, куда надо вписать условие, например, $x \leq L_1$ или $0 \leq x \leq L_1$;

- в нижнем поле ввода (у вертикальной черты) ввести выражение функции для второго участка и выделить его целиком синим уголком (клавишей пробела);
- щелкнуть мышью на кнопке **Otherwise** (Иначе) на панели программирования или выбрать **if** и вписать условие $x > L_1$.

Использование логических (булевых) операторов состоит в умножении слагаемых заданного выражения на соответствующий логический оператор. Логические операторы вводятся с панели булевых операторов (кнопка **Boolean Toolbar**). Булевы операторы могут возвращать только 0 или 1. Если условие справедливо, то значением оператора является 1, если несправедливо, то 0. В математическом выражении умножение на логический оператор обращает соответствующее слагаемое в нуль или не меняет его значения.

СОВЕТ

Логические операторы $<$, $>$, $=$ (меньше, больше, равно) можно набирать с клавиатуры. Другие операторы, например **NOT**, **AND**, **OR**, **XOR** (нет, и, или), надо вводить с панели **Boolean Toolbar** во избежание возможных математических ошибок.

Пример применения всех трех форм записи условия показан на рис. 12.23.

Функция **if** допускает использование нескольких условий в одном выражении путем вложения условий одно в другое. На рис. 12.23 функция $M(x)$ содержит одно условие, а функция $M_1(x)$ — четыре вложенных друг в друга условия. На практике функция **if** используется редко, так как два других способа записи условия проще и нагляднее.

Для ввода с панели программирования функции $M_1(x)$, включающей в себя 4 условия, необходимо сразу ввести 4 места ввода (4 раза нажать **Add Line**).

При использовании булевых операторов вместо четырех условий введено только три. Четвертое условие, $M_1(x) = 0$ при $x \leq 1$, выполняется автоматически, так как **Mathcad** по умолчанию присваивает нулевое значение всем незадаанным величинам.

Добавим к ним еще три функции, используемые реже. Это функция Хевисайда — Φ , символ Кронекера — δ и функция знака — **sign**:

- $\Phi(x)$ — функция Хевисайда, возвращает 0, если $x < 0$, иначе — 1. Функция Хевисайда включает или выключает слагаемые в выражении в зависимости от значения аргумента.
- $\delta(i, j)$ — символ Кронекера, возвращает 1, если $i = j$, иначе — 0. Символ Кронекера $\delta(i, j)$ используется в операциях с матрицами, включает, выключает или просто изменяет диагональные члены матрицы.
- **sign**(x) возвращает 0, если $x = 0$, возвращает 1, если $x > 0$, и -1 — во всех остальных случаях. Функцию **sign**(x) удобно использовать для перемены знака выражения.

Использование функции if

$$M(x) := \text{if}(x \leq 1, 3, 5 - x)$$

Функция if с четырьмя вложенными условиями

$$M1(x) := \text{if}(x \leq 1, 0, \text{if}(1 < x \leq 2, 2 \cdot x, \text{if}(2 < x \leq 4, 2 \cdot x + 6 - 3 \cdot x, 2 \cdot x + 6 - 3 \cdot x - 2)))$$

Использование панели программирования

$$M1(x) := \begin{cases} 0 & \text{if } x \leq 1 \\ 2 \cdot x & \text{if } 1 < x \leq 2 \\ 2 \cdot x + 6 - 3 \cdot x & \text{if } 2 < x \leq 4 \\ (2 \cdot x + 6 - 3 \cdot x - 2) & \text{otherwise} \end{cases}$$

Все три выражения M1(x)
эквивалентны друг другу

x := 0, 0.2..5

Использование булевых операторов

$$M1(x) := 2 \cdot x \cdot (x > 1) + (6 - 3 \cdot x) \cdot (x > 2) - 2 \cdot (x > 4)$$

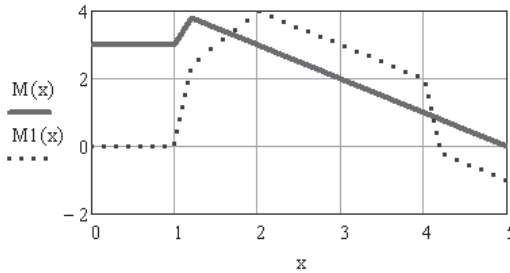


Рис. 12.23. Три способа записи условия в Mathcad

Примеры использования указанных функций приведены на рис. 12.24–12.26.

Использование функции Хевисайда почти аналогично использованию булевых операторов (сравните рис. 12.23 и 12.24). Умножение соответствующего слагаемого на функцию Хевисайда включает или выключает слагаемое в выражении.

Так как функция Хевисайда возвращает 0, то, если аргумент функции равен 0 или меньше 0, записав в качестве аргумента соответствующее выражение, можно наложить на выражение любое ограничение.

На рис. 12.25 показана возможность перемены знака выражения с помощью функции sign, что иногда требуется, если аргумент возводится в четную степень.

На рис. 12.26 показано применение символа Кронекера при использовании в вычислениях тензорной записи. В качестве примера приведен расчет девиатора заданного тензора напряжений

$$[D_{\sigma}] = [T_{\sigma}] - [T_0],$$

здесь $[D_{\sigma}]$ – девиатор тензора напряжений $[T_{\sigma}]$; $[T_0]$ – шаровой тензор напряжений.

Функция Хевисайда включает слагаемое, если выражение в скобках > 0 .
 $\Phi(x)=0$, если $x < 0$. $\Phi(x)=1$, если $x \geq 0$

$$M(x) := 2 \cdot x \cdot \Phi(x - 1) + (6 - 3x) \cdot \Phi(x - 2) - 2 \cdot \Phi(x - 4)$$

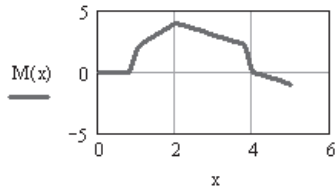


Рис. 12.24. Пример использования функции Хевисайда

Функцию $\text{sign}(x)$ удобно использовать для перемены знака выражения
 $\text{sign}(x)=0$, если $x=0$, $\text{sign}(x)=1$, если $x>0$, $\text{sign}(x)=-1$, если $x<0$

$$A1(x) := (x - 3)^2 \quad A2(x) := (x - 3)^2 \cdot \text{sign}(x - 3)$$

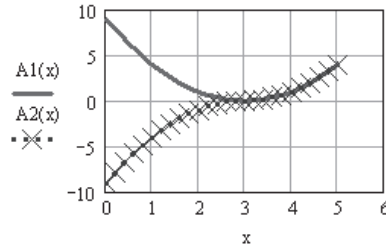


Рис. 12.25. Пример использования функции перемены знака

Символ Кронекера $\delta(i, j)$ используется в операциях с матрицами, включая, выключая или просто изменяя диагональные члены матрицы.
 $\delta_{i,j} = 1$, если $i=j$, иначе $\delta_{i,j} = 0$.

Определение девиатора напряжений

тензор напряжений $i := 1..3 \quad j := 1..3$

$$\sigma := \begin{pmatrix} 15 & -25 & 10 \\ -25 & -10 & 5 \\ 10 & 5 & 20 \end{pmatrix} \quad \sigma_0 := \frac{1}{3} \cdot \sum_{i=1}^3 \sigma_{i,i} \quad \sigma_0 = 8.3$$

среднее напряжение

тензорная запись

$$D_{\sigma_{i,j}} := \sigma_{i,j} - \delta(i,j) \cdot \sigma_0 \quad D_{\sigma} = \begin{pmatrix} 6.7 & -25 & 10 \\ -25 & -18.3 & 5 \\ 10 & 5 & 11.7 \end{pmatrix}$$

шаровой тензор

$$\sigma_0 := \sigma \llcorner T_0 := \text{diag}(\sigma_0) \quad T_0 = \begin{pmatrix} 8.3 & 0 & 0 \\ 0 & 8.3 & 0 \\ 0 & 0 & 8.3 \end{pmatrix}$$

матричная запись

$$D_{\sigma} := \sigma - T_0 \quad D_{\sigma} = \begin{pmatrix} 6.7 & -25 & 10 \\ -25 & -18.3 & 5 \\ 10 & 5 & 11.7 \end{pmatrix}$$

Рис. 12.26. Пример использования символа Кронекера

В тензорной записи

$$D_{\sigma_{i,j}} = T_{\sigma_{i,j}} - \delta_{i,j} \sigma_0,$$

где σ_0 — среднее напряжение; $\delta_{i,j}$ — символ Кронекера.

Как видно из примера, тензорная запись короче и проще, чем матричная.

12.8. Функции округления чисел

Mathcad содержит четыре функции округления чисел:

- ❑ $\text{ceil}(x)$ возвращает ближайшее целое число, большее x («потолок»);
- ❑ $\text{floor}(x)$ возвращает ближайшее целое число, меньшее x («пол»);
- ❑ $\text{trunc}(x)$ возвращает целую часть числа;
- ❑ $\text{round}(x, n)$ возвращает округленное значение числа x с точностью до n знаков после запятой.

Примеры использования этих функций приведены на рис. 12.27.

Функции округления чисел	Функции комбинаторики
$\text{ceil}(19.3) = 20$	$V := (12 \ 16 \ 24 \ 36 \ 8)^T$
$\text{floor}(19.3) = 19$	$\text{combin}(6, 2) = 15$
$\text{round}(19.577, 2) = 19.6$	$\text{gcd}(45, 81, 18, 63) = 9$ $\text{gcd}(V) = 4$
$\text{round}(19.577) = 20$	$\text{lcm}(15, 25, 45, 75) = 225$ $\text{lcm}(V) = 144$
$\text{trunc}(19.377) = 19$	$\text{mod}(225, -50) = 25$
	$\text{permut}(6, 2) = 30$ $\text{permut}(n, k) = \frac{n!}{(n-k)!}$

Рис. 12.27. Функции округления и комбинаций чисел

12.9. Функции комбинаторики и теории чисел

В этом разделе функций Mathcad содержится пять функций:

- ❑ $\text{combin}(n, k)$ возвращает число сочетаний из n целых чисел по k , где $n > k$;
- ❑ $\text{permut}(n, k)$ возвращает число размещений из n целых чисел по k , где $n > k$;
- ❑ $\text{lcm}(A, B, C, \dots)$ возвращает наименьшее общее кратное для чисел A, B, C, \dots ;
- ❑ $\text{gcd}(A, B, C, \dots)$ возвращает наибольший общий делитель для чисел A, B, C, \dots ;
- ❑ $\text{mod}(x, y)$ возвращает остаток от деления на y , записанный со знаком числа x .

В функциях lcm и gcd аргументом может быть как строка чисел, указанных через запятую, так и вектор. Примеры использования этих функций приведены на рис. 12.27.

12.10. Функции тригонометрические, гиперболические, логарифмические и экспоненциальные

Встроенные в Mathcad тригонометрические, гиперболические, логарифмические и экспоненциальные функции приведены на рис. 12.28. Их применение понятно и без пояснений. Обратите внимание на написание некоторых функций, не совпадающее с общепринятым в математике.

В функции $\log(z, b)$ второй параметр b является необязательным. Если он отсутствует, вычисляется десятичный логарифм. Если присутствует, вычисляется логарифм по основанию b .

Обратите внимание на функции $\text{angle}(x, y)$ и $\text{atan}(x, y)$ (угол и арктангенс). Обе они, в принципе, определяют одно и то же: угол между осью абсцисс и радиус-вектором точки с координатами x и y . Но если функция angle вычисляет угол в пределах от 0 до 2π , то функция atan вычисляет угол от $-\pi$ до π . По умолчанию эти функции вычисляют углы в радианах. Чтобы вывести углы в градусах, надо в месте ввода размерности (справа от числового результата) ввести deg (градус).

тригонометрические функции		гиперболические функции	
прямые	обратные	прямые	обратные
$\sin(z) = 0.9$	$\text{asin}(z) = 1.6 - 1.3i$	$\sinh(z) = 3.6$	$\text{asinh}(z) = 1.4$
$\cos(z) = -0.4$	$\text{acos}(z) = 1.3i$	$\cosh(z) = 3.8$	$\text{acosh}(z) = 1.3$
$\tan(z) = -2.2$	$\text{atan}(z) = 1.1$	$\tanh(z) = 1$	$\text{atanh}(z) = 0.5 - 1.6i$
$\cot(z) = -0.5$	$\text{acot}(z) = 0.5$	$\coth(z) = 1$	$\text{acoth}(z) = 0.5$
$\sec(z) = -2.4$	$\text{asec}(z) = 1$	$\text{sech}(z) = 0.3$	$\text{asech}(z) = i$
$\csc(z) = 1.1$	$\text{acsc}(z) = 0.5$	$\text{csch}(z) = 0.3$	$\text{acsch}(z) = 0.5$
$\text{atan2}(1, 3) = 71.6 \text{ deg}$	$\text{angle}(1, 3) = 71.6 \text{ deg}$		
$\text{atan2}(-1, -3) = -108.4 \text{ deg}$	$\text{angle}(-1, -3) = 251.6 \text{ deg}$		
от $-\pi$ до $+\pi$	от 0 до 2π		
Логарифмы и экспонента			
$\exp(z) = 7.4$	$\ln(z) = 0.7$	$\log(z) = 0.3$	$\log(2, 3) = 0.6$

Рис. 12.28. Элементарные математические функции

12.11. Специальные функции

В этом разделе рассмотрим некоторые редко используемые функции. Среди них есть функции активные — вычисляемые и неактивные — невычисляемые, они используются во внутренних алгоритмах Mathcad.

12.11.1. Функции Бесселя

Функции Бесселя по определению являются решениями различных краевых задач для некоторых обыкновенных дифференциальных уравнений. Функции Бесселя первого и второго рода часто возникают как решения волнового уравнения с цилиндрическими граничными условиями.

Mathcad содержит 40 встроенных функций Бесселя. Перечислим некоторые из таких функций.

Обычные функции Бесселя:

- $J_0(x)$ — функция Бесселя первого рода нулевого порядка;
- $J_1(x)$ — функция Бесселя первого рода первого порядка;
- $J_n(m, x)$ — функция Бесселя первого рода m -го порядка;
- $Y_0(x)$ — функция Бесселя второго рода нулевого порядка, $x > 0$;

- $Y_1(x)$ — функция Бесселя второго рода первого порядка, $x > 0$;
- $Y_n(m, x)$ — функция Бесселя второго рода m -го порядка, $x > 0$.

Здесь x — вещественное безразмерное число; m — порядок, целое число, $0 < m < 100$.

Модифицированные функции Бесселя:

- $I_0(x)$ — модифицированная функция Бесселя первого рода нулевого порядка;
- $I_1(x)$ — модифицированная функция Бесселя первого рода первого порядка;
- $I_n(m, x)$ — модифицированная функция Бесселя первого рода m -го порядка;
- $K_0(x)$ — модифицированная функция Бесселя второго рода нулевого порядка, $x > 0$;
- $K_1(x)$ — модифицированная функция Бесселя второго рода первого порядка, $x > 0$;
- $K_n(m, x)$ — модифицированная функция Бесселя второго рода m -го порядка, $x > 0$.

Здесь x — вещественное безразмерное число; m — порядок, целое число, $0 < m < 100$.

Примеры графического представления некоторых функций Бесселя показаны на рис. 12.29 и 12.30.

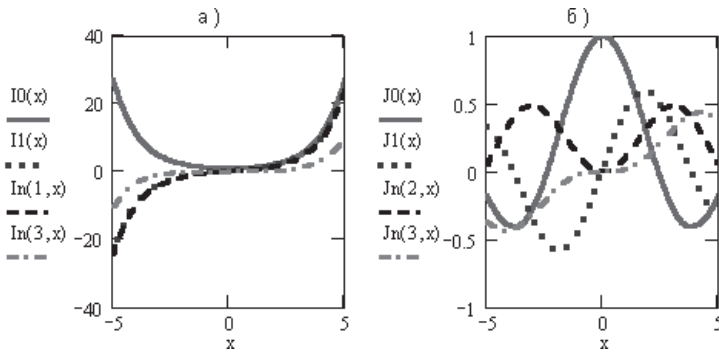


Рис. 12.29. Функции Бесселя первого рода: а — модифицированные; б — обычные

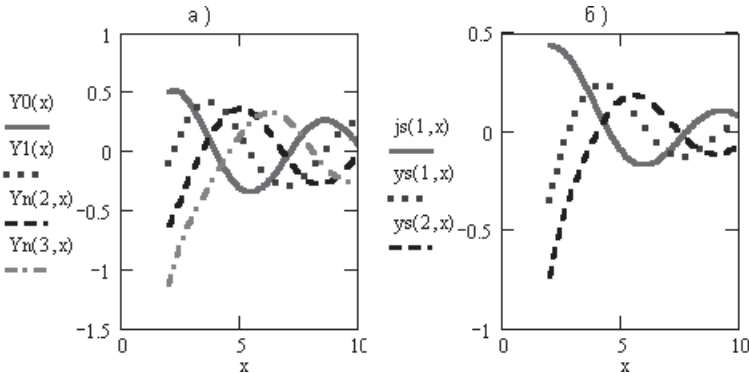


Рис. 12.30. Функции Бесселя: а — второго рода; б — сферические

Функции Эйри являются независимыми решениями ОДУ:

- $Ai(x)$ — функция Эйри первого рода;
- $Bi(x)$ — функция Эйри второго рода.

Здесь x — вещественное безразмерное число.

Комплексная комбинация функций Бесселя — Кельвина является решением соответствующего ОДУ, зависящего от параметра n :

- $bei(n, x)$ — мнимая часть функции Бесселя — Кельвина порядка n ;
- $ber(n, x)$ — действительная часть функции Бесселя — Кельвина порядка n .

Сферические функции Бесселя:

- $js(n, x)$ — сферическая функция Бесселя первого рода порядка n , $x > 0$;
- $ys(n, x)$ — сферическая функция Бесселя второго рода порядка n , $x > 0$.

Здесь n — порядок (целое число).

12.11.2. Другие специальные функции

В Mathcad введено много различных специальных функций. Кроме функций Бесселя, это гамма-функция, гипергеометрические функции, ортогональные полиномы и т. д. Описание этих функций, математические формулы и примеры использования имеются в справочной системе Mathcad. На рис. 12.31 также приведено несколько примеров использования этих функций.

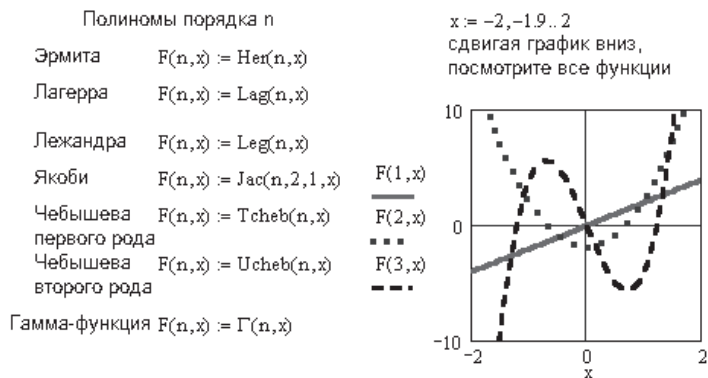


Рис. 12.31. Примеры использования ортогональных полиномов

12.11.3. Дополнительные неактивные функции

При загрузке символьного процессора Mathcad распознает ряд дополнительных специальных функций: интегралы Френеля, интегральные синус, косинус, показательную функцию, дилוגарифм, функцию Дирака. Статус этих функций необычен — они могут фигурировать в результатах символьных операций, но недоступны для обычных численных вычислений. Эти функции, по существу, являются текстовыми записями. Выражения с неактивными функциями Mathcad помещает в буфер обмена, и их можно извлечь оттуда для просмотра. Описание этих функций есть в справочной системе Mathcad.

12.12. Строковые функции

Строковые функции (Strings) используются для вывода стандартных сообщений, возникающих в результате вычислений. Такие функции выполняют действия со строковыми переменными. Строка представляет собой цепочку символов, заключенных в кавычки. В Mathcad имеются следующие функции для работы со строками:

- ❑ $\text{concat}(A, B, C, \dots)$ — строковая переменная, полученная объединением строковых переменных или констант A, B, C ;
- ❑ $\text{num2str}(z)$ — представление числа z ;
- ❑ $\text{str2num}(S)$ — преобразование строкового представления числа S в вещественное число;
- ❑ $\text{str2vec}(S)$ — преобразование в вектор строки S , содержащей записи вещественных чисел в строковом формате;

```

A := "Эта величина"  B := "должна быть"  C := "скаляром"  D := "вектором"
S := concat(A,B,C)      объединение строк
S = "Эта величина должна быть скаляром"

strlen(S) = 34          число символов в строке
search(S,C,0) = 25     число позиций с начала строки S до подстроки C

substr(S,12,8) = "должна"  выделяет подстроку из 7 символов,
                           начиная с позиции 11

V := str2vec(B)        VT = (228 238 235 230 237 224 32 225 251 242 252 32)
                       преобразование строки символов в вектор ASCII-кодов

vec2str(V) = "должна быть"
                       преобразование вектора ASCII-кодов в строку символов

num2str(2.35·106) = "2350000"  преобразование числа
str2num("2350000") = 2350000  в строку и обратно

CWD = "G:\Книга 3\Питер\Глава 12-доп-функции\"
                       полный путь к данному документу

```

Рис. 12.32. Примеры применения строковых функций

- ❑ $\text{str1en}(S)$ — количество знаков в строке S ;
- ❑ $\text{vec2str}(V)$ — строковое представление вектора V ;
- ❑ $\text{lstring}(S)$ возвращает 1, если S — строка;
- ❑ $\text{search}(S_1, \text{Sub}S, m)$ возвращает стартовую позицию подстроки $\text{Sub}S$ в строке S_1 , начиная с позиции m . Возвращает 1, если подстрока не найдена;
- ❑ $\text{error}(S)$ при появлении ошибки возвращает строку S ;
- ❑ CWD возвращает полный путь к папке, где хранится открытый документ.

Примеры действий со строками приведены на рис. 12.32.

При работе со строками первая позиция считается нулевой.

ВНИМАНИЕ

Для того чтобы в строке можно было писать русские буквы, для констант должен быть выбран шрифт, работающий с кириллицей. Если он не был установлен ранее, надо установить уголок курсора на любой численный результат. На панели форматирования появится название стиля Constants (Константы). Нужно открыть список шрифтов и выбрать шрифт, работающий с кириллицей, например Times New Roman Cyr.

Глава 13

Работа с Mathcad-документами

Данная книга рассчитана на людей, знакомых с работой в Windows. Интерфейс Mathcad типичен для Windows-приложений, поэтому не ставится целью описать содержание всех панелей Mathcad. В процессе работы с Mathcad пользователь поневоле вынужден ознакомиться с интерфейсом Mathcad. В данной главе сообщим лишь сведения, необходимые для инженерных расчетов в Mathcad и позволяющие избежать определенные трудности в работе.

13.1. Построение математических выражений

Построение математических выражений было описано в главе 1. Тем не менее один важный абзац из первой главы повторим и здесь.

Имя выражения (все, что стоит слева от знака присвоения значения) может состоять из латинских, русских, греческих и других букв и цифр, знаков подчеркивания (), штриха ($\dot{\quad}$), символа процента (%), знака бесконечности (∞), вводимых с клавиатуры.

Имена переменных и функций не могут начинаться с цифры, знака подчеркивания, штриха, символа процента (%), не могут включать в себя пробелы. Символ бесконечности ∞ может быть только первым символом в имени.

ВНИМАНИЕ

Mathcad воспринимает прописные и строчные буквы как различные идентификаторы, так же как одна и та же буква, изображенная различными шрифтами, — это разные имена.

В Mathcad 13 в меню View ► Toolbars появился пункт Custom Characters (рис. 13.1), выбрав который, можно быстро вставить некоторые символы, встречающиеся в Mathcad-документах, такие как °F, °C, ±, ≈, ||.

Имена переменных и констант в Mathcad могут иметь нижний индекс-комментарий. Для ввода его нужно, выделив имя уголком, нажать клавишу (точка) и ввести с клавиатуры любой комментарий, например, $\sigma_{\text{главное}}$.

Отличить нижний индекс-комментарий от индекса элемента вектора или матрицы в Mathcad 13 позволяет точка около индекса-комментария (сравните V_k и $V_{.k}$), которая видна при выделении выражения.

Построенные математические выражения нельзя форматировать произвольно. Для отдельно взятого выражения нельзя выбрать свой формат.

13.1.1. Стили записи выражений

Математические выражения могут содержать лишь переменные и константы. Соответственно, в Mathcad предусмотрены определения стиля переменных и констант (Variables, Constants).

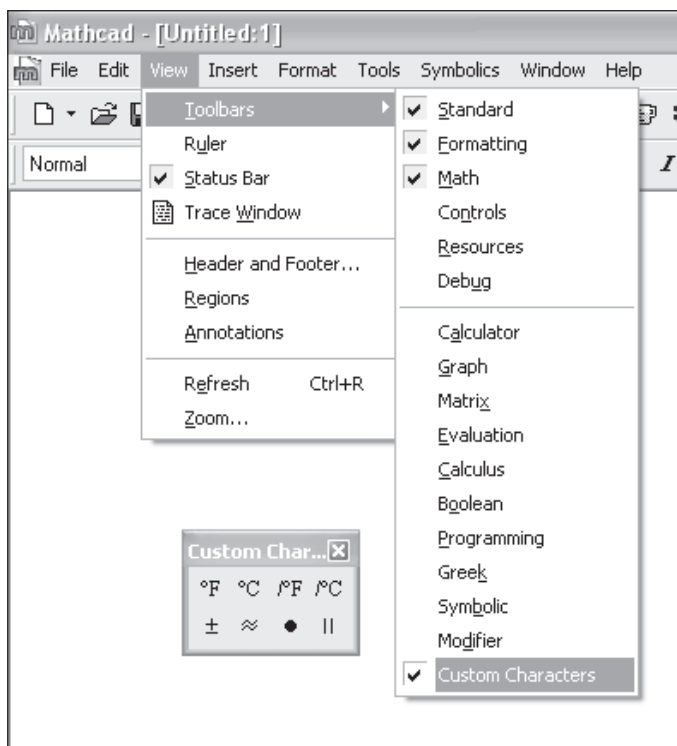


Рис. 13.1. Выбор дополнительных символов в Mathcad 13

ВНИМАНИЕ

Если вы, установив курсор на одном из выражений (или выделив это выражение целиком или частично), с помощью панели форматирования измените наименование шрифта или его размер, то это изменение автоматически будет применено ко всем выражениям, относящимся, соответственно, к переменным или константам.

Такое же изменение формата переменных или констант можно выполнить из главной панели Mathcad, выбрав **Format** ▶ **Equation**. Откроется диалоговое окно **Equation Format** (для переменных) (рис. 13.2). Выберите **Variables** (Переменные) или **Constants** (Константы) ▶ **Modify** (Изменить). В новом окне выберите **Шрифт**, **Начертание**, **размер** ▶ **ОК**. Последнее окно относится к интерфейсу **Windows**, и даже в англоязычной версии пункты меню написаны на русском языке (рис. 13.3).

Итак, Mathcad не разрешает изменять стиль одного отдельно взятого математического выражения. Но «если нельзя, но очень хочется, то ... можно». Для этого надо создать новый стиль и применить его к выбранному выражению.

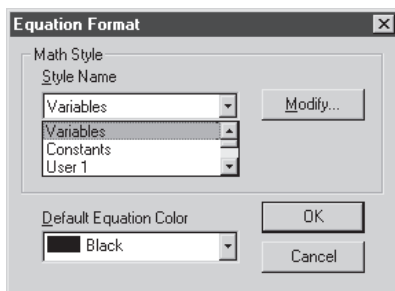


Рис. 13.2. Меню форматирования переменных и констант

Для создания своего стиля надо в главном меню Mathcad выбрать **Format** ▶ **Equation** (Формат ▶ Уравнение). В появившемся окне **Equation Format** (Формат уравнений) выбрать один из стилей **Users** (при желании переименовать его) ▶ **Modify** (Изменить) ▶ **Font** (Шрифт) ▶ выбрать шрифт, начертание, размер ▶ **OK** (рис. 13.3). Аналогично можно создать свой шрифт для работы с текстом (рис. 13.4). Только в меню **Format** вместо **Equation** выбрать **Style**.

А вот стиль вывода результатов расчета для одного-единственного числа создать не удастся. Стили **Users** в этом случае не принимаются, а любое изменение формата числа приводит к аналогичному изменению формата всех чисел в документе.

Обратите внимание на то, что при использовании окна **Equation Format** (Формат уравнений) в наборе шрифтов присутствуют шрифты, работающие с кириллицей (см. рис. 13.3).

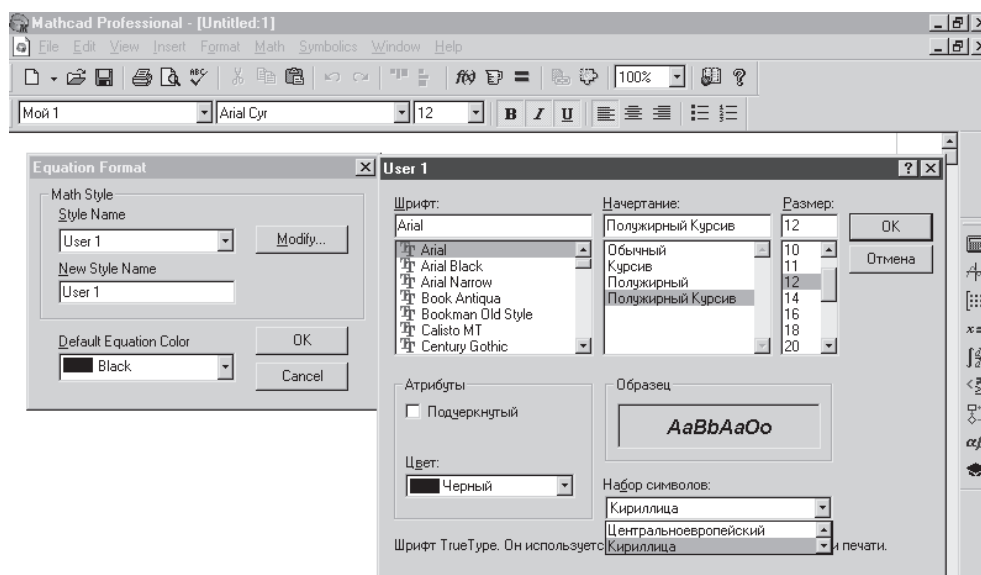


Рис. 13.3. Создание своего стиля для работы с переменными

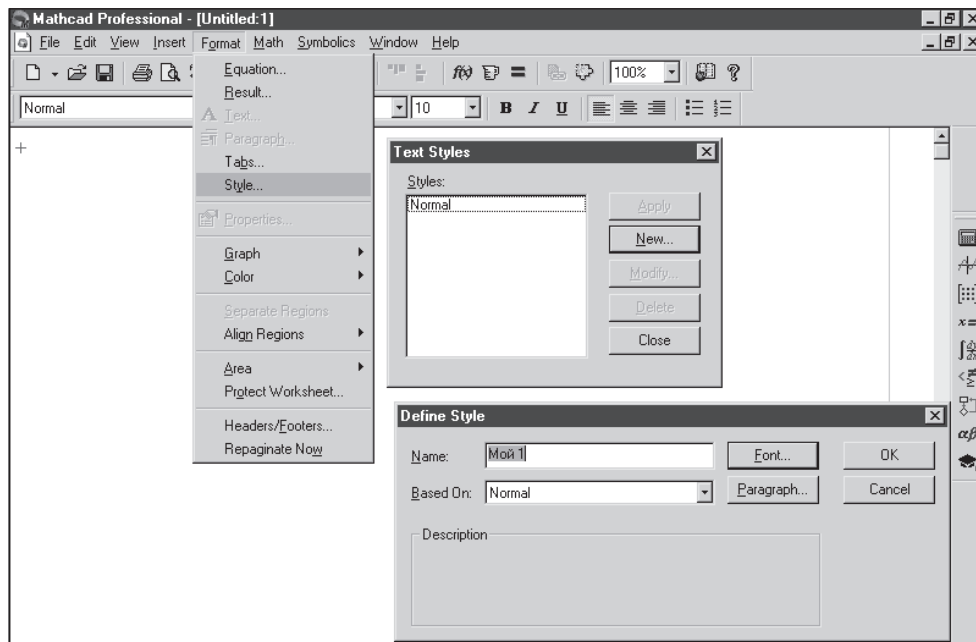


Рис. 13.4. Создание нового текстового стиля

Пример использования нового стиля приведен на рис. 13.5.

стандартные функции стандартные функции

$a := 10$ $\text{atan}(a) = 1.471$ $\text{atan}(a) = 1.471$ $\text{atan}(a) = \dots$

$\text{atan}(a) =$ $\text{atan}(5) = 1.373$

Эта переменная не определена.

Рис. 13.5. Применение вновь созданных стилей

Первое выражение в верхней строке рис. 13.5 создано текстовым стилем Normal, второе — вновь созданным стилем Nov 1.

Во второй и третьей строках для переменных варьируются стиль Variable и вновь созданный Users 1. Для вывода результатов применен единый стиль Constants. Обратите внимание на то, что переменная a , введенная стилем Variable, становится неопределенной для Mathcad при введении ее другим стилем.

Сделанные изменения переменных и констант не касаются результатов расчетов. Для них существует свой способ форматирования.

Если в меню Format (Формат) выбрать Results (Результаты), то откроется окно форматирования результатов расчетов, где можно выбрать Number Format (Формат чисел), Display options (Вид на экране), Units Display (Изображение размерностей), Tolerance (Погрешность) (рис. 13.6).

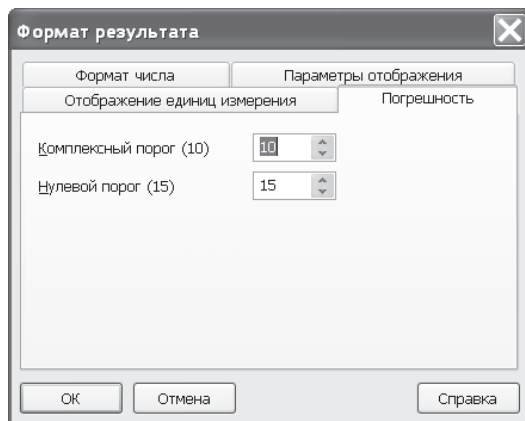


Рис. 13.6. Окно форматирования результатов расчета

Вкладка **Number Format** (Формат числа) предназначена для форматирования чисел — это подробно описано в главе 1.

Вкладка **Display options** (Параметры отображения) предназначена для выбора вида на экране (эти действия описаны в главе 4, посвященной работе с матрицами, поскольку именно вид матриц предлагается изменять в этом пункте меню). Здесь же можно задать вид мнимой единицы на экране: либо i , либо j .

На вкладке **Units Display** (Отображение единиц измерения) можно выбрать вид размерностей на экране (описано в главе 11). Нажатие кнопки **Format Units** (Формат размерностей) изображает отрицательные степени размерностей в виде дроби. Пункт **Simplify units as possible** (Упростить размерности, если можно), как правило, ничего не меняет.

На вкладке **Tolerance** (Погрешность) можно выбрать порог отображения комплексного числа (**Complex threshold**), по умолчанию равный 10. Это означает, что при $\frac{\text{Re}(Z)}{\text{Im}(Z)} > 10^{10}$ число будет изображено как действительное (без мнимой части). При $\frac{\text{Im}(Z)}{\text{Re}(Z)} > 10^{10}$ число будет изображено как чисто мнимое (без действительной части).

Кроме того, на этой вкладке устанавливается порог равенства числа нулю (**Zero threshold**). По умолчанию этот порог равен 15, то есть число, меньшее 10^{-15} , будет изображено как нуль.

Если необходимо выделить цветом или размером букв одно или несколько выражений, используйте пункт **Property** (Свойства), содержащийся в контекстном меню, открываемом при нажатии правой кнопки мыши на нужном выражении. Предварительно нужно выделить нужный объект или группу объектов (рис. 13.7), перечеркнув их курсором.

ПРИМЕЧАНИЕ

Напоминаем, что содержание контекстного меню зависит от выделенного объекта. В нем каждый раз появляются наиболее нужные в данный момент пункты.

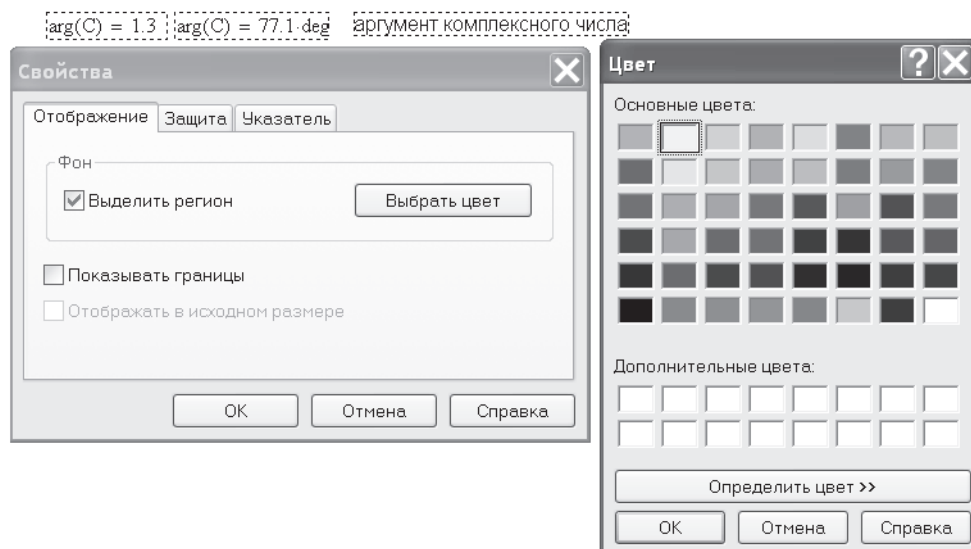


Рис. 13.7. Выбор свойств выделенной группы объектов

На рис. 13.7 предлагается выделить объекты цветом и рамкой.

Вкладка **Protect** (Защитить) позволяет запретить редактирование выделенных объектов, но только в том случае, если включена защита всего документа в целом.

13.2. Аннотация файлов и областей файлов

В Mathcad 12 и 13 есть возможность добавления комментариев в свойства операторов и файлов. Рекомендуется в шаблон документа добавить имя создателя файлов и организации, тогда эти сведения будут присутствовать в каждом созданном документе.

В главном меню Mathcad выберите **File** ▶ **Properties** и в открывшемся окне впишите сведения о создаваемом файле (рис. 13.8). Если необходимые сведения ввести для чистой страницы и сохранить ее как шаблон документа, эти сведения будут присутствовать в каждом создаваемом вами на основе этого шаблона файле. Правда, эти сведения можно прочитать только в оболочке Mathcad. Они не видны из файловых менеджеров, например, Мой компьютер, Проводник (в Windows), Windows Commander.

Чтобы добавить комментарии к отдельным математическим выражениям, в стандартном меню Mathcad выберите **View** и отметьте флажком пункт **Annotation**. Если к каким-либо объектам уже добавлены комментарии, то эти пункты будут выделены цветными скобками. Цвет этих скобок можно определить, выбрав в стандартном меню **Format** ▶ **Color** ▶ **Annotation**. Для ввода аннотации щелкните правой кнопкой мыши на том объекте, где надо ввести аннотацию, в контекстном

меню выберите пункт **Annotate Selection** и в появившемся окне введите текст комментария. После этого такой объект на экране будет виден в круглых скобках выбранного цвета. Чтобы увидеть комментарий к объекту, щелкните на нем правой кнопкой мыши и в контекстном меню выберите пункт **View/Edit Annotation**. На экране появится окно с комментарием (рис. 13.9).

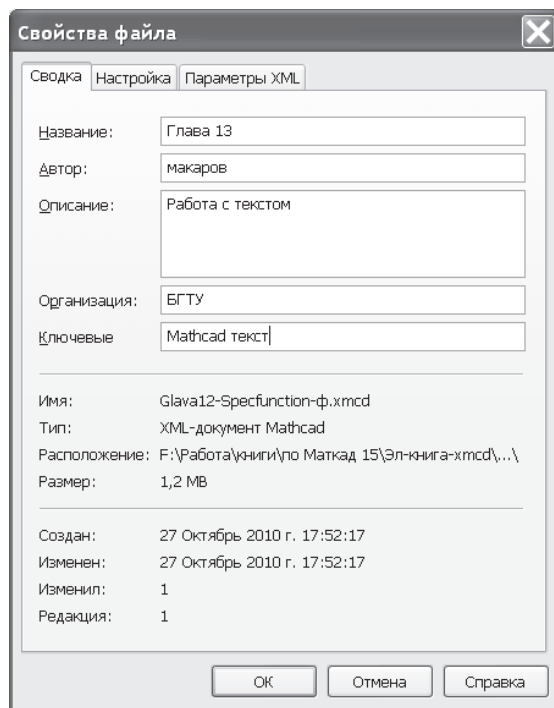


Рис. 13.8. Ввод комментария к документу

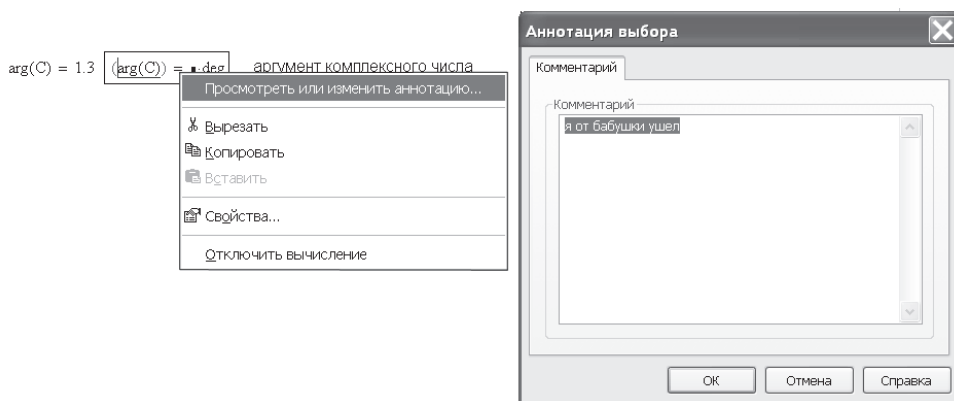


Рис. 13.9. Просмотр комментария к массиву A

13.3. Защита информации в Mathcad

Чтобы включить защиту всего документа в целом, надо в главном меню Mathcad выбрать **Format ▶ Protect Worksheet** (Защитить документ). В появившемся окне предлагается три уровня защиты документа (рис. 13.10):

1. **File** (Файл) не накладывает никаких ограничений на редактирование и копирование любых регионов файла, то есть защита отключена.
2. **Content** (Содержание) — существующие регионы защищены от редактирования. Новые регионы могут быть созданы. Защищенные регионы можно копировать.
3. **Editing** (Редактирование) — созданные регионы не могут редактироваться или копироваться. Новые регионы не могут быть созданы. При нажатии правой кнопки мыши контекстное меню не действует.

Выбор пункта **Password** (Пароль) наиболее эффективно защищает документ. Необходимо ввести пароль два раза и нажать **ОК**.

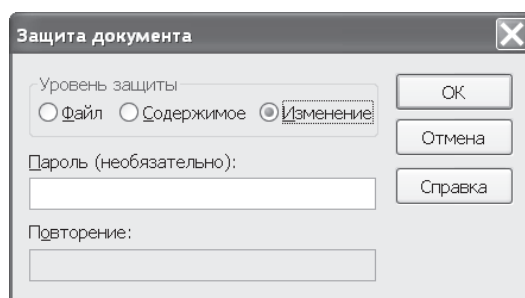


Рис. 13.10. Защита документа от редактирования или копирования

Для снятия защиты надо в главном меню Mathcad выбрать **Format ▶ Unprotect Worksheet** (Снять защиту с документа). Если был введен пароль, появится сообщение с предложением ввести пароль для снятия защиты.

В Mathcad имеется возможность защищать от редактирования только часть регионов. Для этого щелкните правой кнопкой мыши на любом объекте. В открывшемся контекстном меню выберите **Properties ▶ Protect** (рис. 13.11).

По умолчанию пункт **Protect region from editing** (Защитить регион от редактирования) включен у всех выражений. Снимите поставленный флажок, чтобы в дальнейшем при включении защиты в меню **Tools ▶ Protect Worksheet** (Инструменты ▶ Защита документа) объект остался незащищенным и мог быть отредактирован.

Не забудьте сохранить файл, чтобы защита оставалась включенной при открытии файла.

В Mathcad можно использовать установку уровня защиты по умолчанию для всех открываемых документов. Для этого в главном меню Mathcad выберите **Tools ▶ Preference ▶ Script Security** (Инструменты ▶ Параметры ▶ Безопасность скриптов). Откроется окно, показанное на рис. 13.12.

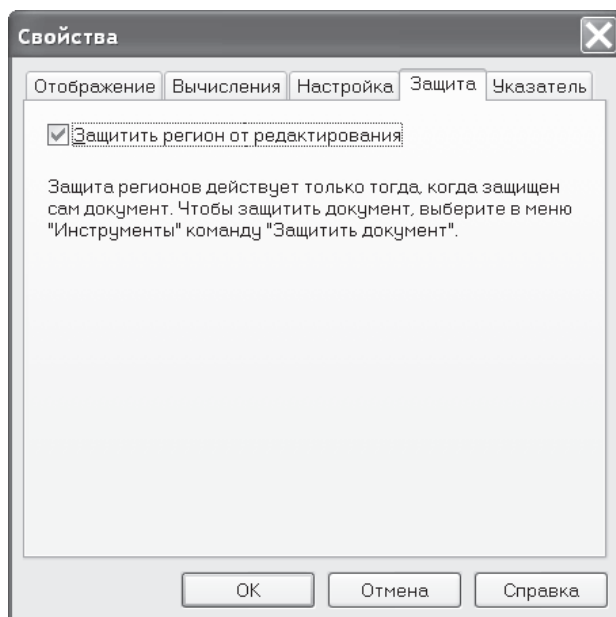


Рис. 13.11. Включение возможности защиты документа

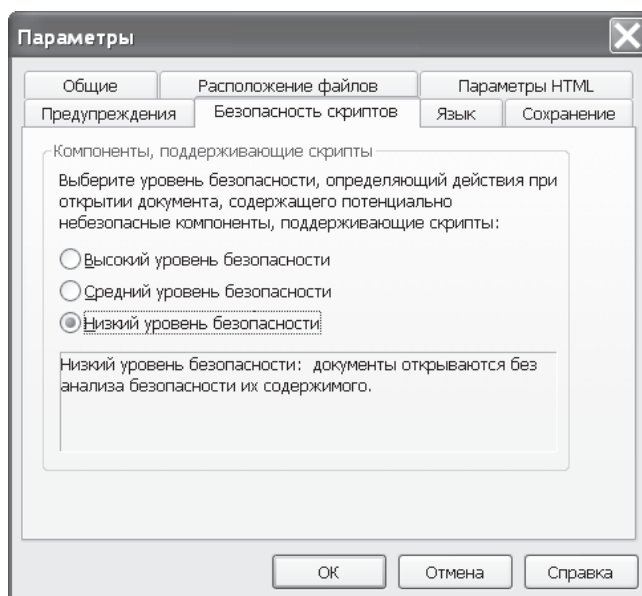


Рис. 13.12. Установка уровня защиты открываемых документов

В открывшемся окне предлагается выбрать уровень безопасности, чтобы определить действия, совершаемые при открытии документов, содержащих защищенные объекты. Отметьте щелчком мыши один из трех уровней защиты:

1. Low Security (Низкий уровень безопасности) — документ открывается без учета уровня защиты документа.
2. Medium Security (Средний уровень безопасности) — при открытии документа, содержащего защищенные регионы, на экран выводится предупреждение.
3. High Security (Высокий уровень безопасности) — защищенные регионы автоматически отключены при открытии документа.

Использование этого меню необязательно при включении защиты документа.

13.4. Закрытые зоны в Mathcad

Закрытая зона — это часть Mathcad-документа, которая присутствует в документе, участвует в расчетах, но не видна на экране. Необходимость в создании такой зоны возникает в двух случаях: для уменьшения размера документа и для сокрытия информации от посторонних глаз.

Чтобы создать закрытую зону, сделайте следующее:

- В главном меню Mathcad выберите Insert ► Area (Вставка ► Область). На экране появятся две горизонтальные линии с указателями слева. Перетащите их мышью в начало и в конец области, которую надо закрыть.
- Щелкните правой кнопкой мыши в выделенной линиями области. В открывшемся контекстном меню выберите Collapse (Свернуть). Выделенная область исчезнет. Останется лишь одна горизонтальная линия с указателем.

Выражения, скрытые в закрытой области, продолжают работать в документе, хотя и не видны на экране. Чтобы открыть закрытую область, нужно сделать двойной щелчок левой кнопкой мыши на линии с указателем. Скрытая область становится открытой и появляется на экране в обрамлении двух линий с указателями.

Чтобы удалить закрытую область, щелкните мышью на одной из линий с указателем, выделив ее, и нажмите клавишу Del (Удалить).

Чтобы запретить доступ к закрытой зоне, вначале закройте ее, выбрав Collapse (Свернуть). Затем, щелкнув на линии с указателем правой кнопкой мыши, в контекстном меню выберите Lock (Блокировать). Появится запрос ввести пароль, причем два раза, для надежности (рис. 13.13). После ввода пароля на линии рядом с указателем появятся замок и дата закрытия зоны.

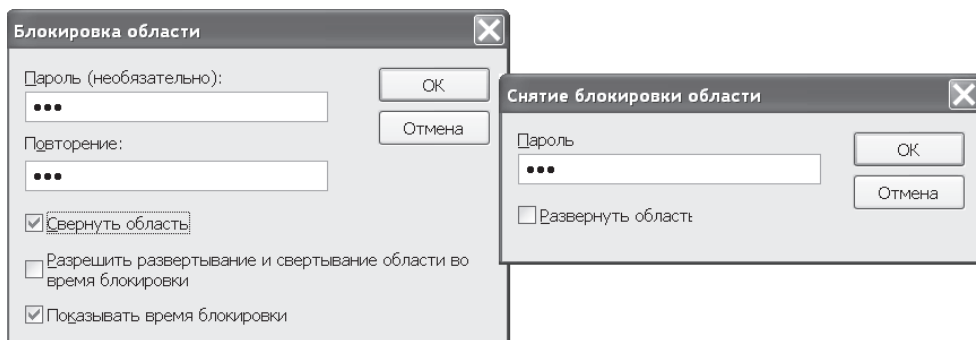


Рис. 13.13. Ввод пароля при заперении и отпирании зоны

Для открытия запертой зоны щелкните правой кнопкой мыши на линии с указателем и в контекстном меню выберите **Unlock** (Развернуть). Появится запрос на ввод пароля. После ввода пароля зона открыта, замки на линии с указателями исчезнут.

Различные варианты зоны показаны на рис. 13.14.

В Mathcad 13 расширены возможности оформления закрытой зоны. Щелкнув правой кнопкой мыши на скрытой зоне, выберите в контекстном меню вкладку **Area** (рис. 13.15). На этой вкладке можно вписать заголовок закрытой зоны, отметить выделение ее линиями и значком, сделать эту зону совсем невидимой.

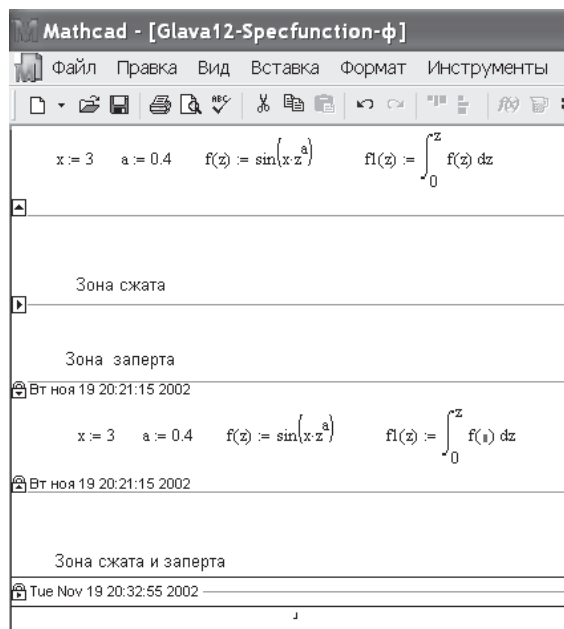


Рис. 13.14. Сжатие и запирание зоны

13.5. Создание вычислительных комплексов

Используя панель программирования, с помощью Mathcad можно составить программу абсолютно любой сложности. Но переводить на Mathcad программы, написанные на других алгоритмических языках, было бы безумием. Mathcad позволяет создавать вычислительные комплексы, объединяющие различные программы. В этом случае нужно лишь дописать к готовым программам недостающие фрагменты на Mathcad.

Есть несколько способов объединения различных программ в один вычислительный комплекс. Рассмотрим два таких способа: с использованием ссылок и гиперссылок.

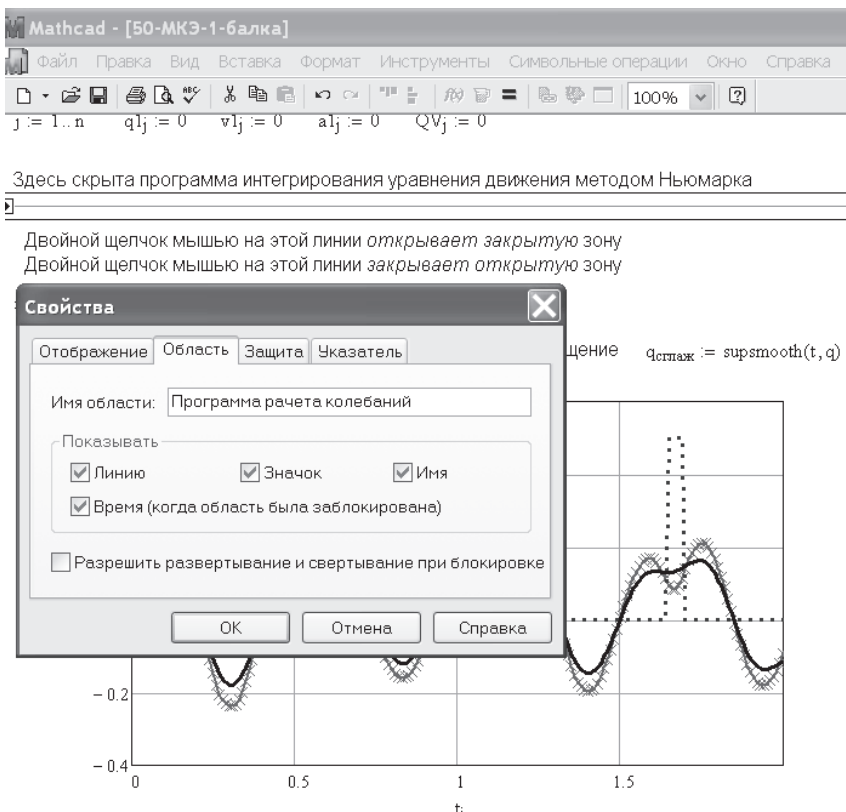


Рис. 13.15. Оформление закрытой зоны

13.5.1. Использование ссылок на другие файлы

В примере 53 электронной книги продемонстрирована возможность объединения нескольких Mathcad-документов в один с помощью ссылки на них. Для этого в основном документе делается ссылка на другой документ. В результате оба документа объединяются в один и работают совместно, хотя второго документа и не видно на экране.

Для создания ссылки надо в главном меню Mathcad выбрать **Insert** ▶ **Reference** (Вставить ▶ Ссылка). В открывшемся окне, выбрав **Browse** (Поиск), укажите путь к документу, который хотите включить в расчет. Нажмите **ОК**. В основном документе появится строка с именем подключенного документа.

Рекомендуется при создании ссылки использовать относительный путь. Указав путь к документу, пометьте флажком пункт **Use relative pass for reference** (Использовать относительный путь для ссылки). При этом в строке документа, указывающей на ссылку, появится буква **(R)**. В этом случае ссылка останется действующей, даже если вы переместите основной документ в другую папку. Важно лишь, чтобы оба документа находились на одном диске.

На рис. 13.16 показано меню вставки ссылки. Сразу под меню показан результат ссылки: строка со ссылкой и начало расчета с использованием ссылки в примере 53.

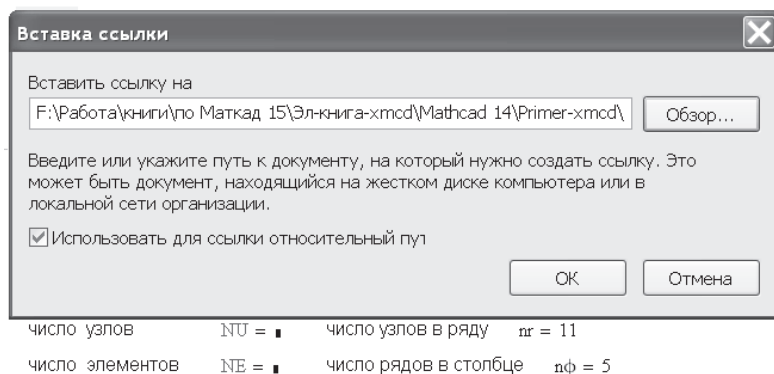


Рис. 13.16. Вставка ссылки на другой файл

При каких-либо изменениях в том или ином документе автоматический пересчет происходит только после сохранения изменений в обоих документах. Иногда для успешной работы комплекса программ после сохранения приходится закрыть и вновь открыть файл.

13.5.2. Использование гиперссылок

В примере 53 для ввода данных и их корректировки использовалось включение программы Matr-ind в основной документ с помощью ссылки (Reference). В примере 55 для корректировки данных используется гиперссылка (Hyperlink).

Гиперссылка позволяет открывать файл, с которым создана связь, в той оболочке, в которой он создан. Для создания гиперссылки нужно:

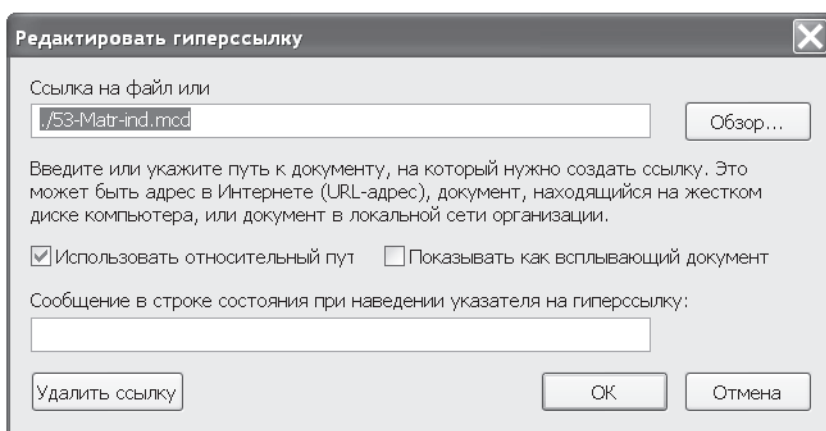
- ❑ выделить слово, с которым будет связан интересующий вас файл, желательно черным цветом. Тогда после установления связи это слово будет выделено жирным шрифтом;
- ❑ в главном меню Mathcad выбрать Insert ► Hyperlink ► Browse ► указать путь к файлу, с которым надо связать выделенное слово ► ОК;
- ❑ в окне Edit Hyperlink установить флажок на пункте Use relative path for hyperlink. В этом случае связь будет работать даже тогда, когда вы переместили каталог со связанными файлами в другое место.

Если вы хотите вызвать на экран не начало, а определенное место выбранного файла, его необходимо задать. Этот процесс описан в разделе 14.2.1.

Двойной щелчок мышью на выделенном слове открывает файл, с которым установлена связь. Теперь этот файл можно отредактировать и сохранить. Для передачи данных в Mathcad-документ надо внутри связанного файла записать их в другой внешний файл, откуда затем считать функцией READPRN.

На рис 13.17 показано создание гиперссылки на файл `Dan.dat`, где находятся исходные данные для расчета на Fortran из примера 55. После создания гиперссылки при подведении курсора к слову, с которым связана гиперссылка, курсор приобретает вид указательного пальца. Двойной щелчок на этом слове вызывает открытие связанного с ним файла в его оболочке, в данном случае Блокнот (рис. 13.18). Теперь можно проверить исходные данные, изменить их для новой задачи, сохранить. Однако в расчеты Mathcad-документа измененные данные еще не попадут.

Документы, присоединенные гиперссылкой, не работают автоматически в Mathcad-документе. Их надо вызвать двойным щелчком мыши на связанном с ними слове и запустить.



Matr-ind coord Miuf gu ссылки

Рис. 13.17. Вставка гиперссылки в примере 55

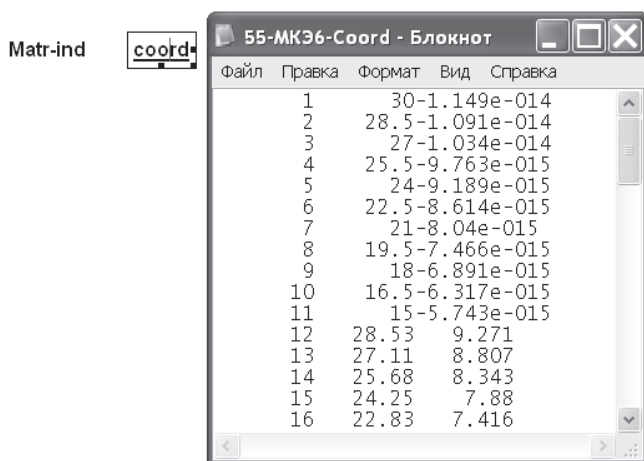


Рис. 13.18. Открытие файла `Coord.prn`

Двойной щелчок на слове «Matr.ind» вызывает запуск и пересчет программы расчета матрицы индексов. Двойной щелчок на слове «Coord» открывает таблицу координат узлов, записанную в этом файле.

Если расчеты в Mathcad должны быть переданы в другую программу для продолжения расчетов или оформления вывода результатов, то в Mathcad-документе нужные результаты расчетов должны быть записаны в отдельный файл функцией WRITEPRN, откуда их «заберет» другая программа.

Подведем итог: ссылка (Reference) работает в Mathcad-документе автоматически, гиперссылка (Hyperlink) требует запуска каждой подключенной программы отдельно двойным щелчком мыши.

Основное предназначение гиперссылок — создание электронных книг. Об этом речь пойдет в главе 17.

13.6. Новые служебные функции Mathcad 13

В Mathcad 13 появилось несколько интересных возможностей работы с документами.

13.6.1. Уменьшение размера файлов

При сохранении файла в текстовом формате XML для уменьшения размера файла Mathcad 13 позволяет выбрать формат XMCD — текстовый или XMCDZ — zip-архив файла (автоматически раскрывается в Mathcad). Размер файла 53-кольцо (из электронной книги) при сохранении в формате Mathcad 2001i (MCD)

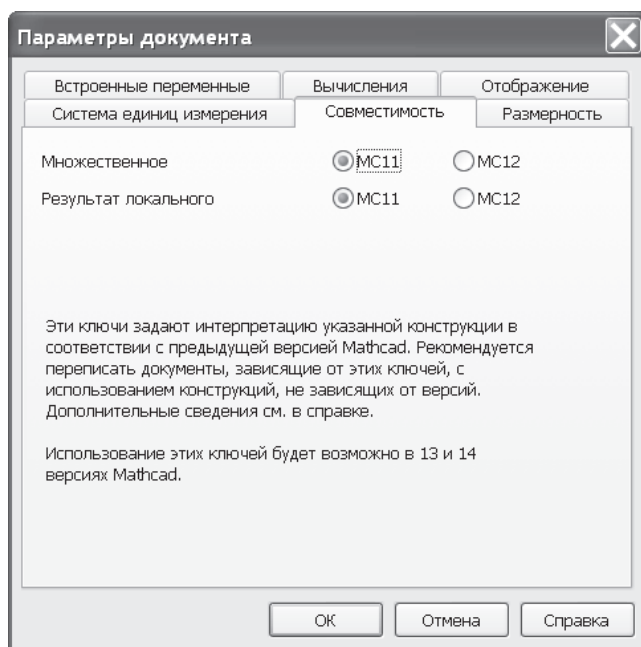


Рис. 13.19. Переключатель версий Mathcad

составил 723 Кбайт, в формате XMCD — 321 Кбайт, в формате XMCDZ — 123 Кбайт.

13.6.2. Переключатель версий Mathcad

В Mathcad 13 в меню Tools ▶ Worksheet Options ▶ Compatibility имеется переключатель совместимости с предыдущими версиями (рис. 13.19). При открытии в Mathcad 13 документа, созданного в Mathcad 11 или в более старых версиях, переключатель автоматически устанавливается на Mathcad 11 (MC11).

Такой переключатель есть и в версии Mathcad 14. Фирма Mathsoft дает пользователям время для постепенного перевода своих файлов к формату Mathcad 13.

13.6.3. Автосохранение

В Mathcad 13 появилась возможность автосохранения файлов в процессе работы с ними. В главном меню выберите Options ▶ Preferences. На вкладке Save можно установить промежуток времени, после которого Mathcad автоматически сохраняет файл в формате XML или XMLZ (в сжатом виде) (рис. 13.20).

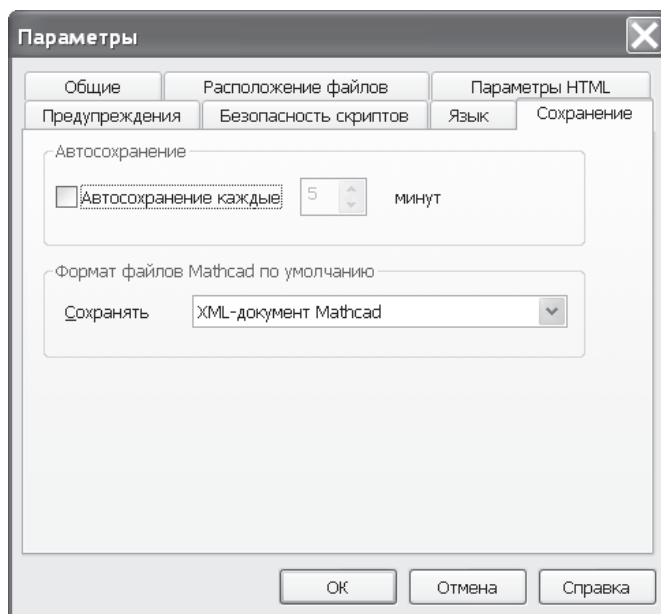


Рис. 13.20. Настройка сохранения файлов

Глава 14

Работа с текстом

Основные положения работы с текстом приведены в главе 1. Здесь же повторим главное из сказанного там. Для ввода текста на русском языке лучше всего, сменив латинский шрифт на русский, набирать текст прямо в математической области. Когда набрано первое слово, при нажатии клавиши пробела область со словом автоматически из математической превращается в текстовую.

В отличие от Word, устанавливающего строгие правила размещения текста на странице, в Mathcad текст можно набирать в любом месте страницы. Размеры текстовой области можно менять обычным для Mathcad способом: щелкнуть мышью на нужной области, подвести курсор к черному прямоугольнику на рамке, окружающей выделенный объект, чтобы появилась стрелка, двигать мышь при нажатой левой кнопке.

Изменение стиля текстовой области не влияет на результаты расчета математических выражений, поэтому можно изменить стиль как всего текста, так и отдельного слова или буквы.

14.1. Изменение стиля текстовых областей

Для изменения стиля всех текстовых областей в документе надо в главном меню Mathcad выбрать команду **Format** ▶ **Style** ▶ **Normal** ▶ **Modify** (**Формат** ▶ **Стиль** ▶ **Обычный** ▶ **Изменить**). В открывшемся окне **Define Style** (**Определение стиля**) нажмите кнопку **Font** (**Шрифт**). Откроется окно **Text Format** (**Формат текста**), где надо задать **Font**, **Font style** (**Вид шрифта**) и **Size** (**Размер шрифта**).

Открывающиеся при изменении стиля текста окна показаны на рис. 14.1 и 14.2.

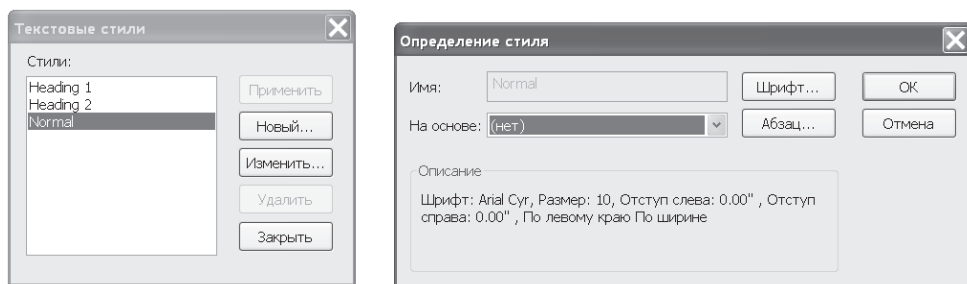


Рис. 14.1. Изменение стиля текста

14.1.1. Установка свойств абзаца

В окне **Define Style** (**Определение стиля**) есть кнопка **Paragraph** (**Абзац**), позволяющая создавать текстовые области, редактируемые средствами, похожими на

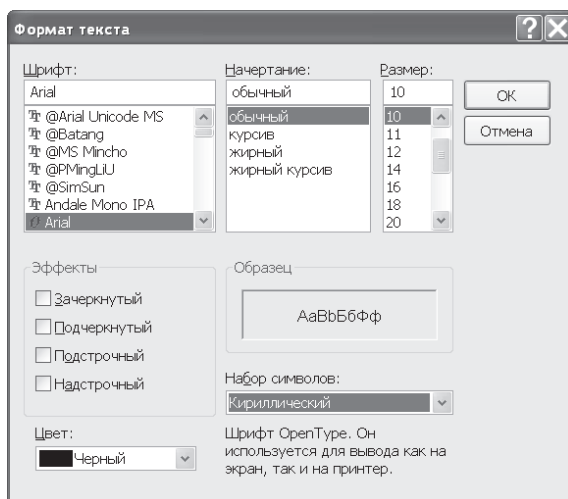


Рис. 14.2. Изменение стиля текста (окончание)

используемые в Word. Средств этих немного, но с их помощью можно создавать хорошие технические отчеты и электронные книги.

Главное назначение окна Paragraph Format (Формат абзаца) — создание электронных книг, сочетающих в себе текст, работающие формулы, которые можно изменять и анализировать, результаты расчета, а также ссылки на другие области документа или другие файлы, легко открывающиеся щелчком мыши на них.

Окно Paragraph Format (Формат абзаца) показано на рис. 14.3.

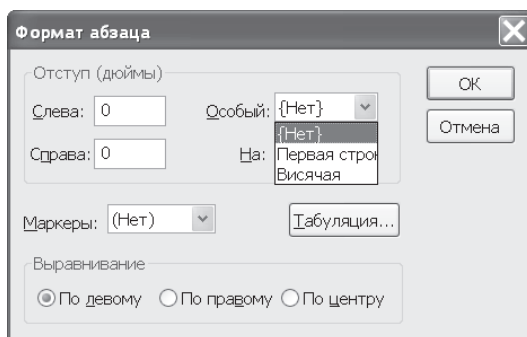


Рис. 14.3. Панель установки параметров абзаца

Слева сверху окна задается отступ (Indent) текста в дюймах (inches) или сантиметрах (centimeters) с левой (left) и с правой (right) стороны.

Справа сверху окна пункт Special (Специальный) задает положение первой строки абзаца:

None — нет сдвига первой строки;

- First Line (Красная строка) — сдвиг первой строки вправо по отношению к остальному тексту;
- Hanging (Висячая строка) — сдвиг первой строки влево по отношению к остальному тексту.

На рис. 14.4 показан образец применения стиля Paragraph (Формат абзаца). Здесь верхняя часть текста записана с использованием пункта First Line (Первая строка). Последний абзац оформлен с применением пункта Hanging (Висячая строка).

Истинные деформации определяются по формуле $\varepsilon_{\text{н}} = \ln\left(\frac{F_0}{F}\right)$

Применение этой формулы требует замера диаметра образца в процессе испытаний, что обычно не делается, поэтому она применена лишь для определения деформации в момент разрыва образца.

- До образования шейки в образце используется другая формула $\varepsilon_{\text{н}} = \ln(1 + \varepsilon)$, не требующая замера диаметра.

- Истинные напряжения определяются по формуле $\sigma_{\text{н}} := \sigma \cdot e^{\varepsilon_{\text{н}}}$, также не требующей замера диаметра.

1. **Результаты расчета** деформаций и напряжений приведены на рисунке 4Р.4, а вид диаграмм условных и истинных напряжений на рис.4Р.6 (слева). Истинные деформации и напряжения в момент разрыва в несколько раз больше условных.

Рис. 14.4. Образцы применения стиля Paragraph (Формат абзаца)

14.1.2. Изменение стиля отдельной текстовой области

Стиль может быть единым для всего текста, если он установлен в начале ввода текста и в дальнейшем не менялся. Можно задать собственный стиль для одного или нескольких абзацев, если, установив курсор на одном из абзацев, правой кнопкой мыши вызвать контекстное меню (рис. 14.5, *слева*), в котором есть пункты Font, Paragraph, Style, выбор которых открывает рассмотренные ранее окна Text Format, Paragraph Format и Text Style. Выберите нужный пункт и поменяйте установки для выбранного объекта.

Если в контекстном меню выбрать пункт Properties (Свойства), как показано на рис. 14.5, *справа*, то, открыв страницу Text (Текст), можно установить параметры:

- Occupy Page Width (Занять всю ширину страницы);
- Push Regions Down As You Type (Сдвигать области вниз, когда вы печатаете).

Кроме того, изменить размер текстовой области можно традиционным способом. Если щелкнуть мышью внутри области, появится рамка с черными квадратиками для изменения размеров. Установите курсор на такой квадратик, подождите до появления стрелки, нажмите левую кнопку мыши и, двигая мышью, растяните или сожмите область.

14.2. Использование гиперссылок

Одним из достоинств электронных документов является возможность быстрого открытия других файлов с помощью гиперссылок. Гиперссылка (Hyperlink) может не только вывести на экран нужный файл, но и показать нужное место открытого документа либо соответствующий сайт в Интернете. Гиперссылки особенно эффективны в больших документах, в обучающих и презентационных программах, в электронных книгах.

На рис. 14.4 в последнем абзаце с выражением «Результаты расчета» связана гиперссылка на пример 13 («Построение диаграммы истинных напряжений»). При двойном щелчке мышью на выделенных словах открывается файл примера 4 в том месте, где находятся результаты расчета истинных напряжений и деформаций.

Процедура создания гиперссылки описана в главе 13 (раздел 13.4.2). Напомним: для создания гиперссылки нужно в меню Insert выбрать пункт Hyperlink (Гиперссылка), нажать кнопку Browse (Обзор) для выбора пути к нужному файлу, затем Use relative path for hyperlink (Использовать относительный путь для гиперссылки) и кнопку ОК.

При двойном щелчке мышью на слове, с которым связана ссылка, откроется выбранный файл, причем на экране видно начало файла.

14.2.1. Переход на метку

Если вы хотите вызвать на экран не начало, а определенное место выбранного файла, его необходимо задать. Для этого

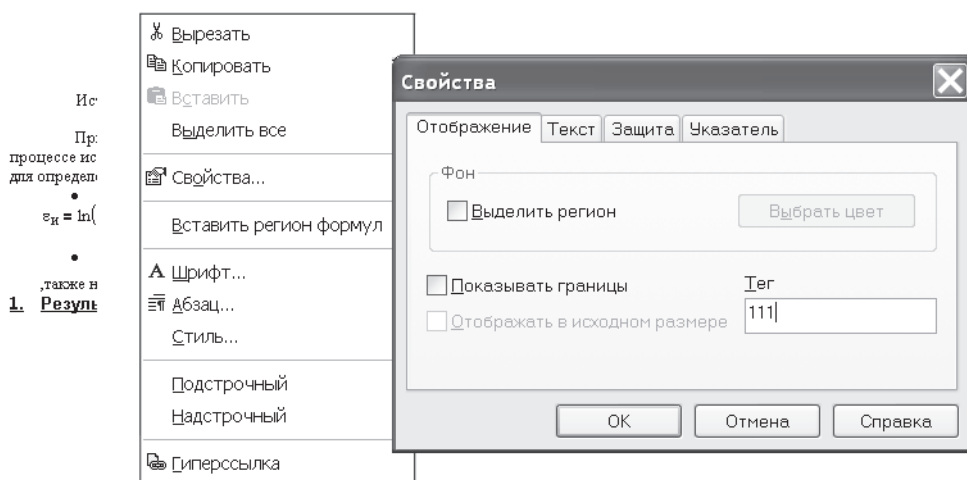


Рис. 14.7. Установка метки на место, с которого откроется файл

- откройте файл, на который будет создана ссылка;
- установите курсор на объект (слово, выражение или график), который при вызове файла будет находиться на экране вверху;
- щелкните на нем правой кнопкой мыши и в открывшемся контекстном меню выберите Properties (Свойства);
- в окне Tag (Метка) впишите имя метки (в нашем случае – 111), как показано на рис. 14.7. Нажмите кнопку ОК.

После создания метки следует создать гиперссылку. Вводя путь к файлу, укажите и имя метки, записав в конце адреса #111 (рис. 14.8). Теперь при двойном щелчке мышью на слове, с которым связана гиперссылка, указанный в адресе файл откроется с того места, на которое установлена метка.

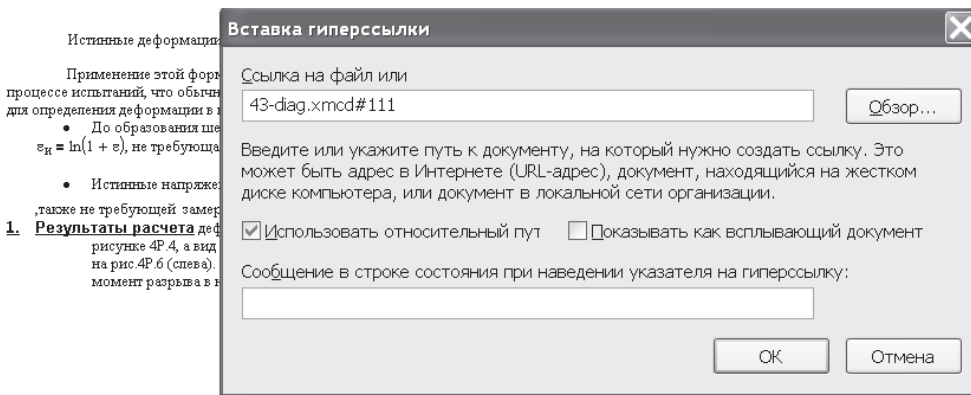


Рис. 14.8. Создание гиперссылки на файл с указанием метки #111

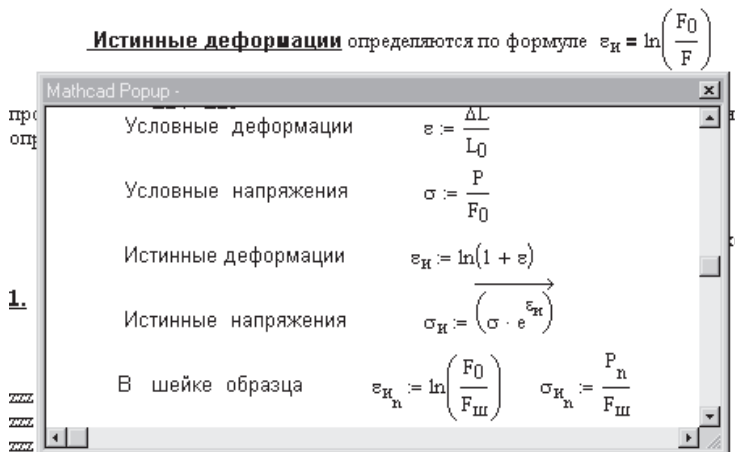


Рис. 14.9. Открытие гиперссылки на всплывающий документ

14.2.2. Всплывающий документ

Если при создании гиперссылки флажком отметить пункт `Display as pop-up document` (Показывать как всплывающий документ), то выбранный файл откроется в отдельном окне (рис. 14.9), при этом метка игнорируется. На этом рисунке гиперссылка связана с выражением «Истинные деформации». Размеры окна можно изменять обычным способом, потянув мышью за угол или за середину стороны окна. Файл, открытый как всплывающий документ, вычисляется, но не редактируется.

Глава 15

Работа с графиками

О построении плоских и пространственных графиков говорилось в главе 1. Повторим общие правила построения графиков, а затем остановимся на особенностях их построения.

15.1. Построение двумерного плоского графика

Рассмотрим построение графика функции:

- ❑ Установите крестообразный курсор в то место, где надо построить график.
- ❑ На математической панели щелкните на кнопке Graph Toolbar (Панель графиков), а на открывшейся панели — на кнопке с изображением плоского графика.
- ❑ В появившемся на месте курсора шаблоне плоского графика введите на оси абсцисс имя аргумента, на оси ординат — имя функции.
- ❑ Щелкните мышью вне шаблона графика. График построен для заданного диапазона изменения аргумента.

Если диапазон значений аргумента не задан, по умолчанию график будет построен в диапазоне значений аргумента от -10 до 10 .

Чтобы в одном шаблоне разместить несколько графиков, надо, набрав на оси ординат имя первой функции, нажать клавишу «запятая» (уголок курсора при этом обязательно должен находиться в конце имени функции). В появившемся месте ввода (черном квадратике) впишите имя второй функции и т. д.

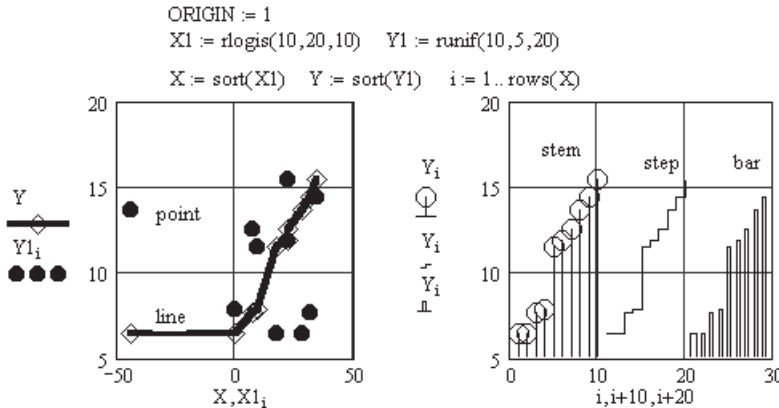


Рис. 15.1. Различные типы изображения точек на графике

Функцию можно задать аналитически или массивами значений аргумента и функции. Для построения плоского графика массивы обязательно должны быть векторами одинаковой размерности.

При задании функции массивами в местах ввода на осях графика надо вписать имена массивов с указанием индексов или без них. На рис. 15.1 массивы чисел созданы генераторами случайных чисел (X_1 и Y_1 — в произвольном порядке, X и Y — отсортированные в порядке возрастания массивы). На этом рисунке показаны различные типы изображения вычисленных значений на графике.

Напомним, что при двойном щелчке мышью в поле графика открывается окно форматирования графика (см. рис. 1.9).

ПРИМЕЧАНИЕ

Это окно (рис. 15.2) для версий Mathcad 11 и более ранних. Окно для версий Mathcad 13 и 14 показано на рис. 1.16. Пункты меню в обоих окнах почти одни и те же. Старое меню, на мой взгляд, было более удобным.

При выборе вкладки Traces (Линии) (рис. 15.2 в Mathcad 11 и рис. 1.16 в Mathcad 15) предоставляется возможность выбрать следующие параметры:

- Line (Вид линии) — solid (сплошная), dot (мелкий пунктир), dadot (штрих-пунктир), dash (крупный пунктир);
- Color (Цвет линии);
- Weight (Толщина линии);
- Type (Тип линии) — line (линия), point (точка), bar (столбик, гистограмма), solidbar (закрашенный столбик), step (ступенька, лестница), stem (палочка с кружочком), draw (рисованная линия), error (позволяет отложить на графике заранее вычисленную ошибку);
- Symbol (Значок, отмечающий вычисленную точку) — none (знак отсутствует), x's (крестик), +'s (плюс), box (прямоугольник), dmnd (ромб), o's (кружок).

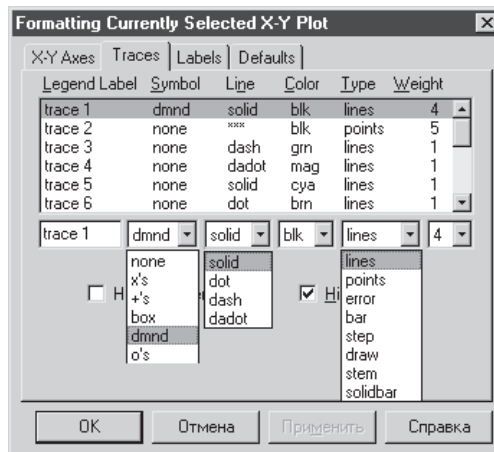


Рис. 15.2. Окно выбора типа линии в Mathcad 11

Чтобы построить график с отложенной ошибкой вычислений (рис. 15.3), нужно наряду с основной функцией вычислить функции ее верхней и нижней ошибки,

присвоив названию функции тип `error`. График на рис. 15.3 построен по результатам испытаний стальных образцов на усталость, приведенным в главе 8 (см. рис. 8.12 и 8.13). Вычисленные верхнее (y_1) и нижнее (y_2) отклонения представляют собой доверительный интервал, соответствующий вероятности разрушения 99 %.

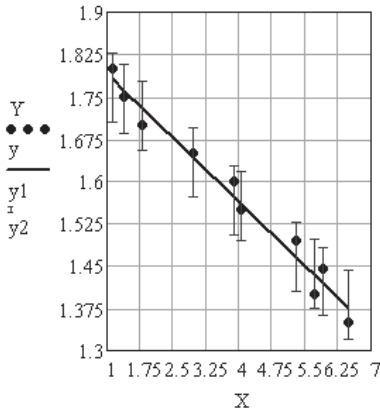


График доверительного интервала при выборе в меню форматирования графиков типа линии `errors`

Рис. 15.3. График с отложенной ошибкой вычислений (сравните с рис. П2.3)

Этот тип кривой представляется совершенно бесполезным. Как видно из рис. 8.13, тип кривой `line` лучше отображает вид интервала вероятности, чем тип `error`.

На рис. 15.4 приведены примеры построения плоских графиков — обычный график зависимости функции от аргумента и параметрический график. Обратите внимание на наличие маркеров на левом графике (см. также рис. 2.16).

`x := 0,0.1..15` `f(x) := x*cos(x)` `fl(x) := x*sin(x)`
`A := 3` `C := 8` `D := -8` маркеры

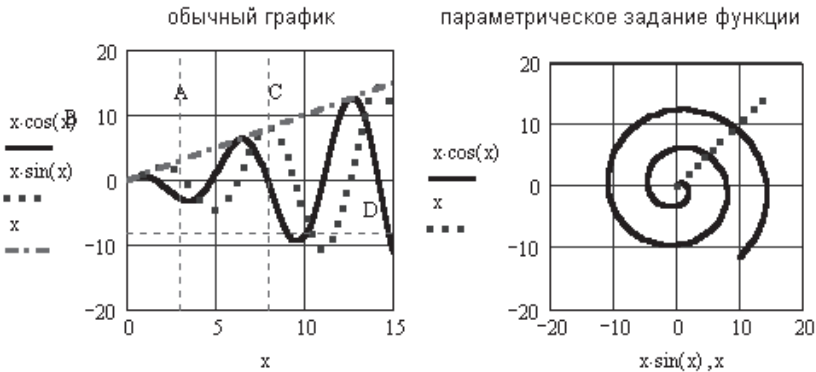


Рис. 15.4. Примеры построения плоских графиков

Для создания маркеров (пунктирных горизонтальных линий на графике) нужно

- ❑ дважды щелкнуть мышью на графике и в открывшемся окне форматирования графиков отметить птичкой **Show Markers** (Показывать метки) по оси X и(или) Y ;
- ❑ щелкнуть мышью вне графика. Окно исчезнет. Около выбранной оси появятся два черных квадратика, в которые можно ввести либо числа, либо имя константы. Имя писать удобнее, так как при перемене значения константы автоматически смещается и метка.

На рис. 15.5 показано увеличение масштаба графика:

- ❑ Поместите курсор в поле графика и нажмите правую кнопку мыши. В открывшемся контекстном меню выберите **Zoom** (Масштаб).
- ❑ Обведите пунктирной линией при нажатой левой кнопке мыши область графика, которую хотите увеличить.
- ❑ В открытом окне **X-Y Zoom** (Масштаб по осям X - Y) нажмите кнопку **Zoom** (Масштабировать). Выделенная область увеличится до размеров всей рамки графика.
- ❑ Если вы хотите вернуться к исходному виду графика, нажмите **Unzoom** (Отменить масштабирование) или **Full View** (Вид полностью). Если надо оставить увеличенное изображение, нажмите кнопку **OK**.

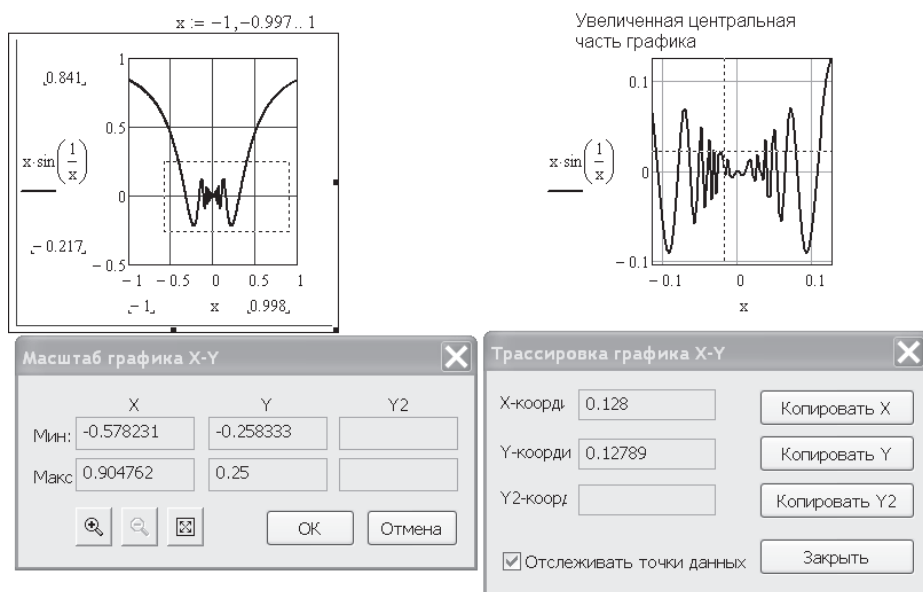


Рис. 15.5. Увеличение масштаба и трассировка графика в Mathcad 15

В контекстном меню, открываемом при нажатии правой кнопки мыши, есть пункт **Trace** (Трассировка). Выбор этого пункта вызывает появление окна трассировки X - Y Trace (рис. 15.5, *справа*).

Щелкните левой кнопкой мыши на графике. На острие курсора появится перекрестье пунктирных линий, перемещающееся при движении курсора при нажатой левой кнопке мыши. Одновременно в окне X-Y Trace бегут цифры, указывающие координаты перекрестья. Установив перекрестье на нужную точку графика, можно скопировать его координаты в буфер обмена и затем вставить в документ для использования в расчетах.

Если поставлен флажок Track Data Points (Отслеживать точки данных), то перекрестье скачет от точки к точке. Если снять этот флажок, движение перекрестья будет плавным.

15.2. Построение полярных графиков

Чтобы построить график функции в полярных координатах, следует:

- в панели графиков Graph Toolbar щелкнуть на кнопке, изображающей полярный график;
- в местах ввода ввести: внизу — угловой аргумент функции, слева — имя функции для построения обычного графика. Для построения параметрического графика нужно ввести в обоих местах ввода имена функций углового аргумента (рис. 15.6).

Форматирование полярных графиков аналогично форматированию плоских графиков. Полярные графики, как и плоские, можно строить в обычных и в логарифмических координатах.

Для построения графика в логарифмических координатах в окне форматирования графика надо отметить флажком пункт Log Scale (Логарифмический масштаб). Пример использования логарифмических координат для плоского графика приведен на рис. 8.13, для полярного графика — на рис. 15.6.

На полярном графике маркеры можно ставить только для радиальной координаты, выбрав в окне форматирования графика пункт Show Markers (Показать метки).

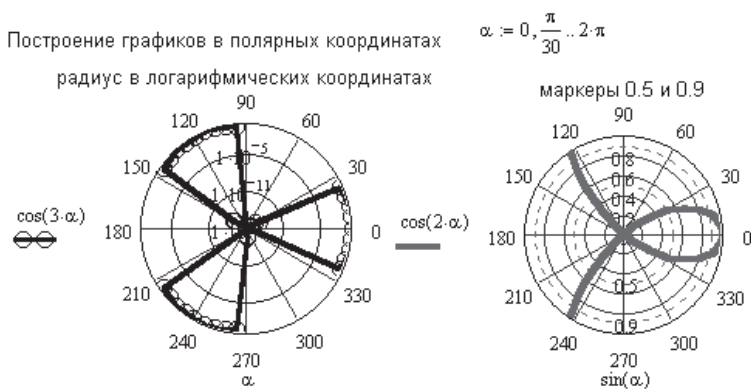


Рис. 15.6. Примеры графиков в полярных координатах

В полярных координатах можно, как и в плоских графиках, использовать изменение масштаба и трассировку, выбрав в контекстном меню пункт Zoom (Масштаб)

или Trace (Трассировка). Заметим, что в логарифмических координатах можно использовать пункт Trace и нельзя использовать пункт Zoom.

15.3. Построение трехмерных графиков

Правила построения трехмерных графиков были описаны в главе 1.

Существует несколько возможностей построения трехмерных графиков:

- с помощью кнопки графиков Graph с математической панели;
- с помощью мастера построения графиков, вызываемого из главного меню Mathcad.

Можно построить трехмерный график, создав матрицу значений функции двух переменных, а также использовать ускоренный метод построения 3D-графиков без создания матрицы значений функции или специальные функции CreateMech и CreateSpace для создания массива значений функции и построения графика.

15.3.1. Построение графика по массиву значений функции

На рис. 15.7 показано построение графика функции двух переменных, заданных массивом значений. При этом два аргумента являются независимыми переменными, а функция — зависимой переменной.

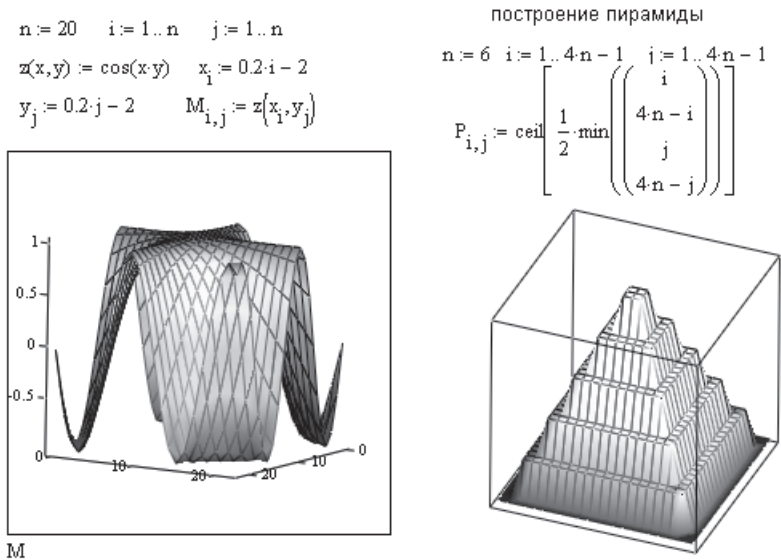


Рис. 15.7. Построение трехмерного графика по заданному массиву значений функции

- Создайте вектор значений функции, предварительно задав векторы значений аргументов функции.

- Установив курсор на то место, где нужно построить график, откройте панель Graph Toolbar (Панель графиков), щелкнув на соответствующем значке математической панели, затем щелкните на кнопке Surface plot (График поверхности). На экране появится шаблон графика.
- В единственном месте ввода введите имя функции (без аргументов). График построен.

Появившийся график больше похож на скелет из отдельных линий. Чтобы сделать из него почти произведение искусства, надо дважды щелкнуть мышью в поле графика. Откроется окно форматирования графика. О содержании этого окна рассказано в главе 1. Чуть позднее мы вернемся к форматированию графиков, а пока продолжим разговор о способах создания графиков.

15.3.2. Быстрое построение графика

На рис. 15.8 показан самый простой и удобный способ построения графика:

- Запишите функцию двух переменных.
- Установив курсор на то место, где нужно построить график, откройте панель графиков Graph Toolbar, затем щелкните на кнопке графика поверхности Surface plot. На экране появится шаблон графика.
- В единственном месте ввода введите имя функции (без аргументов). График построен.

В Mathcad есть возможность построения нескольких графиков на одном шаблоне. Для этого в единственном месте ввода нужно вписать через запятую названия всех функций (рис. 15.8, *справа*).

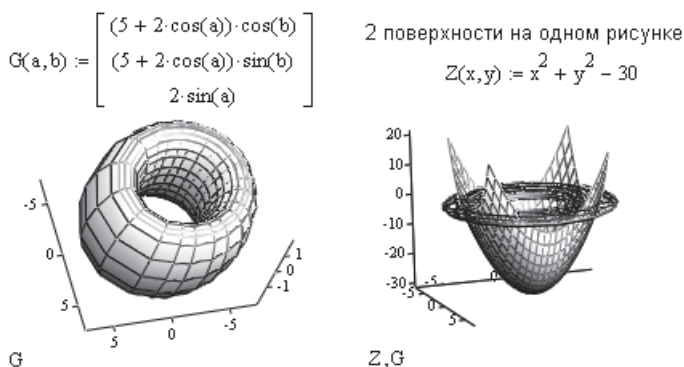


Рис. 15.8. Пример быстрого построения графиков

При быстром построении графиков Mathcad по умолчанию выбирает значения обоих аргументов в пределах от -5 до $+5$ и число контурных линий 20. Для изменения этих значений дважды щелкните мышью на графике и в открывшемся окне форматирования выберите вкладку Quick Plot Data (Данные быстрого графика). Измените пределы значений переменных и нажмите кнопку ОК.

На рис. 15.9 показан тот же график, что и на рис. 15.8, но после установки новых пределов изменения аргументов. Для этого предназначены области окна **Range 1** — для первого аргумента и **Range 2** — для второго аргумента. Введите свои численные значения аргументов: **start** — начальное значение, **end** — конечное значение, **# of Grids** — число линий сетки, покрывающей поверхность. В принципе все линии с поверхности можно удалить. Для этого на вкладке **Appearance** (Внешний вид) выберите в группе **Line options** (Параметры линии) переключатель **No Lines** (Без линий).

После ввода новых пределов значений бочонок оказался разрезанным двумя сечениями.

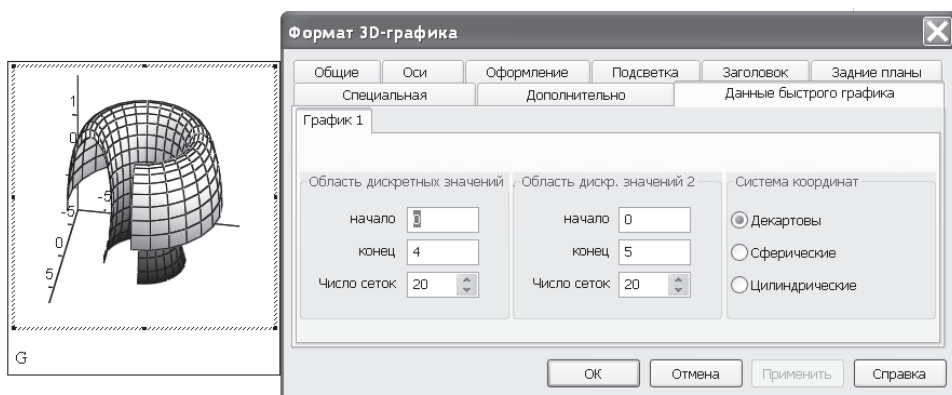


Рис. 15.9. Тот же график, что и на рис. 15.8, после изменения пределов значений аргументов

На той же вкладке **Quick Plot Data** (см. рис. 15.9) пункт **Coordinate System** (Система координат) позволяет быстро построить график в декартовой (**Cartesian**), сферической (**Spherical**) или цилиндрической (**Cylindrical**) системе координат. Для быстрого перестроения графика в другой системе координат надо выбрать переключатель, соответствующий одной из трех систем, и щелкнуть на кнопке **OK**. График перестроен.

На рис. 15.10 показан пример построения графика функции в двух проекциях. В поле ввода имя функции указано два раза. Двойным щелчком мыши открываем окно форматирования графика. На вкладке **General** (Общие) внизу (см. рис. 1.12) есть раздел **Display as** (Показать как), в котором для каждой функции можно выбрать тип графика.

В нашем примере выбраны: для первой функции — **Surface plot** (График поверхности), для второй функции — **Contour plot** (График контуров). Каждый график форматируется отдельно. Для контурного графика (рис. 15.11) в окне форматирования на вкладке **Special** (Специальная) можно указать, нужны ли линии уровня. Если линии нужны, то следует указать вид этих линий, их число, проставлять ли цифровые значения на графике. Установите флажки **Fill** (Заливка поля графика цветом), **Draw Lines** (Рисовать линии), **Auto Contour** (Автоматический контур).

тический контур), Numbered (С указанием чисел). Можно снять флажок Auto Contour и задать число линий контура (Number).

$$f(x, y) := \sin(x) \cdot \cos(y)$$

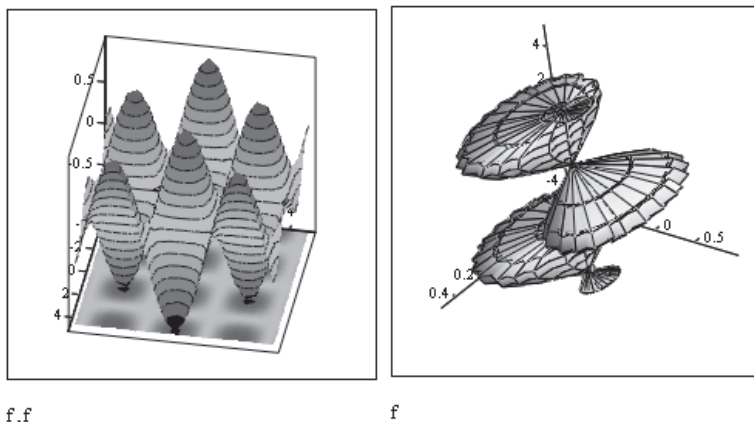


Рис. 15.10. Еще один пример быстрого построения графиков

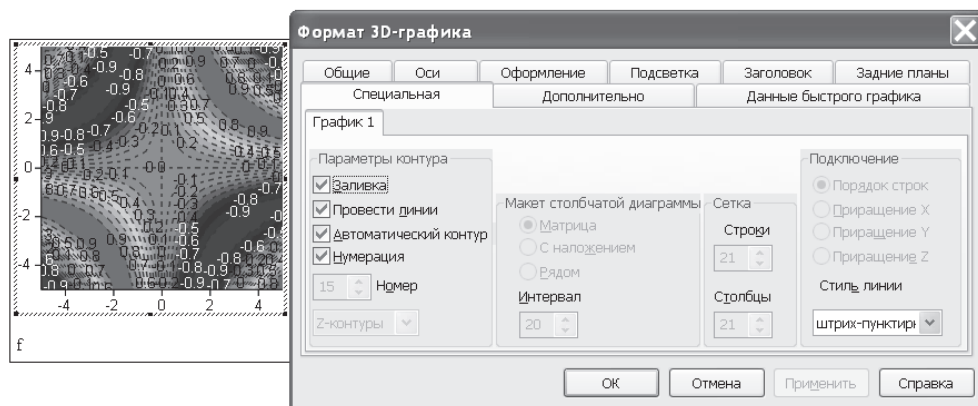


Рис. 15.11. Пример форматирования контурного графика

В списке Line Style можно выбрать вид линий: solid (сплошные), dotted (мелкий пунктир), dashed (крупный пунктир), da-dot (штрих-пунктир).

Вернемся к рис. 15.10. На этом рисунке линии уровня не показаны. Все флажки, кроме Fill, сняты. Проекция на горизонтальную плоскость выглядит просто как тень предмета в полдень. На этом же рисунке справа показана смена системы координат для вывода графика. На вкладке Quick Plot Data (Данные быстрого графика) (см. рис. 15.9) выбрана сферическая система координат (Spherical). Это означает, что Mathcad принимает исходную функцию $f(x, y)$ как заданную в сферических координатах и для построения графика поверхности их надо преобразовать в прямоугольные.

15.3.3. Создание массива значений функции

Для быстрого построения графика функции двух переменных Mathcad использует встроенную функцию `CreateMech`. Эту функцию может использовать и пользователь Mathcad для создания массива значений функции.

`CreateMech` принимает выражения для исходной функции, заданной в виде одной функции двух переменных или функции-вектора из трех элементов или трех отдельных функций, параметрически задающих функцию двух переменных, поверхность которой надо построить.

`CreateMech` возвращает массив координат x, y, z параметрически заданной функции. По этим значениям и строится график поверхности.

Обращение к функции:

`CreateMech(G (или F, или f_1, f_2, f_3), $s_0, s_1, t_0, t_1, sgrid, tgrid, fmap$),`

где $G, F (f_1, f_2, f_3)$ — исходная функция двух переменных, возможный вид которых показан на рис. 15.12; s_0, s_1, t_0, t_1 — пределы изменения переменных, по умолчанию от -5 до 5 ; $sgrid, tgrid$ — число расчетных точек по интервалу изменения переменных s и t (число линий в сетке, покрывающей график). По умолчанию $sgrid = 20, tgrid = 20$; $fmap$ — трехэлементная векторная функция (или три функции), осуществляющая пересчет исходных координат (полярных, цилиндрических или сферических) в декартовы. По умолчанию (если параметр $fmap$ отсутствует) исходные координаты — декартовы.

Если вас устраивают значения, принимаемые по умолчанию, в параметрах функции `CreateMech` можно указать только имя функции (см. рис. 15.12).

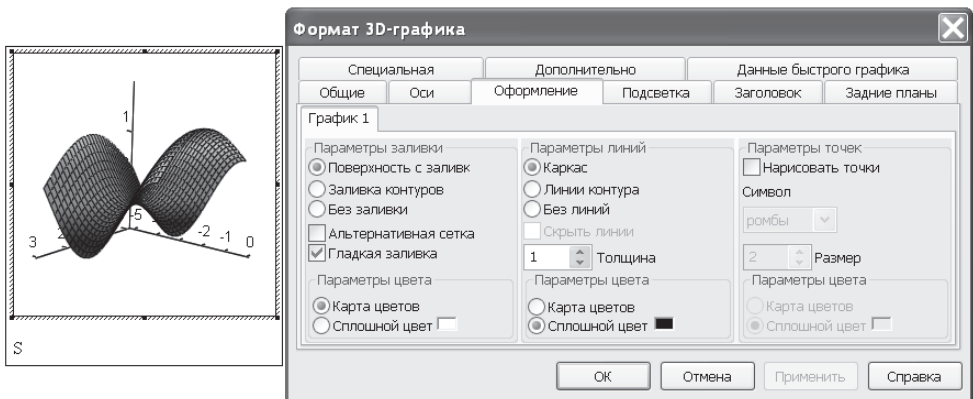


Рис. 15.12. Оформление графика поверхности

ПРИМЕЧАНИЕ

Включать функцию `CreateMech` в документы нерационально. Внутренний алгоритм Mathcad использует эту функцию по умолчанию. Фактически вкладка `Quick Plot Data` (Данные быстрого графика) служит для установки параметров функции `CreateMech`.

Пришло время рассмотреть раскраску графика. Когда вы вписали в шаблон трехмерного графика имя функции и щелкнули мышью вне поля графика, на экране появился черно-белый график функции. Чтобы раскрасить его, двойным щелчком мыши в поле графика откройте окно форматирования графика и выберите вкладку Appearance (Оформление) (см. рис. 15.12).

В разделе Fill Options (Параметры заливки графика) по умолчанию отмечен пункт No Fill (Без заливки). Установите переключатель Fill Surface (Поверхность с заливкой), а в разделе Color Options (Параметры цвета) установите флажок Colormap (Карта цветов). Если хотите нанести на график расчетные точки, установите в разделе Point Options (Параметры точек) флажок Draw Points (Нарисовать точки).

В разделе Line Options (Параметры линий) установите переключатель Wireframe (Каркас). В этом же разделе можно выбрать толщину линий (Weight) и цвет линий (Color Options). Остальные пункты этого меню, как и всех других, опробуйте в процессе работы с графиками.

На рис. 15.13 приведен еще один пример использования функции CreateMech. Этот график изображен в двух проекциях. В окне форматирования на вкладке General (Общие) установлено Show Box (Показать рамку), на вкладке Axes Style (Вид осей) — Corner (В углу) и Equal Scales (Равные шкалы по осям).

Каждый из двух графиков форматируется отдельно. Перейдите на вкладку внизу Plot 2 (График 2). Там (в данном примере) отражаются те же установки, что и для графика 1.

В разделе View (Вид) можно выбрать углы поворота изображенной поверхности вокруг всех трех осей:

- Rotation (Поворот) — вращение вокруг оси Z . Значения углов от -360° до 360° ;
- Tilt (Наклон) — вращение вокруг оси X после вращения вокруг оси Z . Углы могут принимать значения от 0° до 180° ;
- Twist (Изгиб) — вращение вокруг оси Z после вращений вокруг Z и X . Углы могут принимать значения от -360° до 360° .

Поменяйте углы и посмотрите на вид графика.

Этот способ вращения хорош, если надо поточнее развернуть поверхность. Однако обычно используют более простой способ, описанный в главе 1. Следует установить курсор в поле графика, а затем выполнить следующие действия:

- двигать мышь при нажатой левой кнопке. График поворачивается;
- двигать мышь при нажатой левой кнопке и нажатой клавише Ctrl. График приближается или удаляется;
- двигать мышь при нажатой левой кнопке и нажатой клавише Shift. Начинается вращение графика (анимированное). Для остановки вращения надо щелкнуть мышью в поле графика.

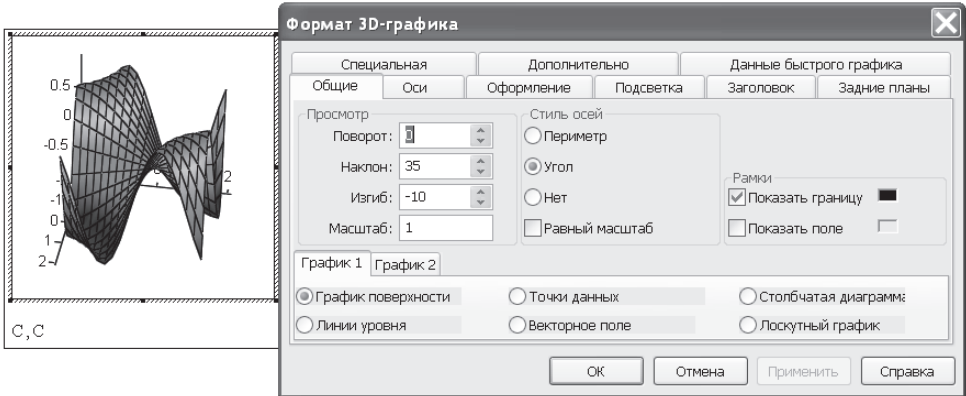


Рис. 15.13. Вкладка General (Общие) окна форматирования трехмерных графиков

15.3.4. Поверхности тел вращения

Аналитически тела вращения, как правило, описываются в сферических или цилиндрических координатах. Mathcad строит трехмерные графики в декартовых координатах. В таких случаях необходимо преобразовать исходные координаты в декартовы. Для таких преобразований можно использовать аналитические формулы или встроенные функции `su2xyz` и `sph2xyz`.

На рис. 15.14 показано построение поверхности шара с использованием аналитического преобразования сферических координат в прямоугольные без применения функции `CreateMesh`.

$$\begin{aligned}
 n &:= 40 & i &:= 1..n & j &:= 1..n & \alpha_i &:= i \cdot \frac{\pi}{n} \\
 \beta_j &:= j \cdot 2 \cdot \frac{\pi}{n+5} & X_{i,j} &:= \sin(\alpha_i) \cdot \cos(\beta_j) \\
 Y_{i,j} &:= \sin(\alpha_i) \cdot \sin(\beta_j) & Z_{i,j} &:= \cos(\alpha_i)
 \end{aligned}$$

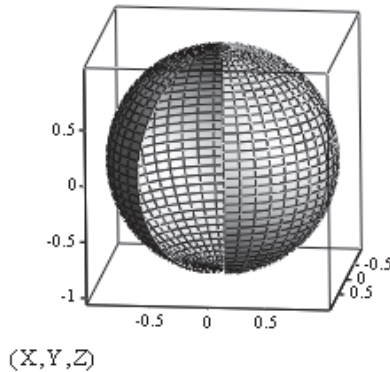
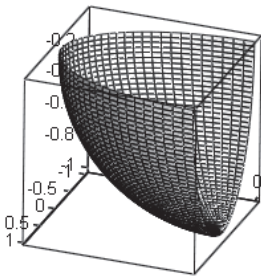


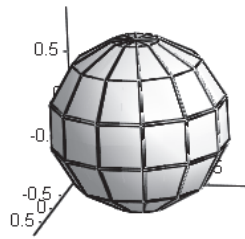
Рис. 15.14. Построение шара с использованием аналитического преобразования координат

$$G(\theta, \phi) := \begin{pmatrix} 1 \\ \theta \\ \phi \end{pmatrix} \begin{matrix} \text{поменяйте местами fmap} \\ \text{fmap}(v) := \text{cyl2xyz}(v_1, v_2, v_3) \\ \text{fmap}(v) := \text{sph2xyz}(v_1, v_2, v_3) \end{matrix} \quad \text{fmap}(r, \alpha, z) := \begin{pmatrix} r \cdot \cos(\alpha) \\ r \cdot \sin(\alpha) \\ z \end{pmatrix}$$

$\theta_0 := 0 \quad \theta_1 := \pi \quad \phi_0 := 1 \cdot \pi \quad \phi_1 := 1.5 \cdot \pi$ поменяйте значения углов θ и ϕ
 $M := \text{CreateMesh}(G, \theta_0, \theta_1, \phi_0, \phi_1, 50, 50, \text{fmap}) \quad M1 := \text{CreateMesh}(G, \text{fmap})$



M



M1

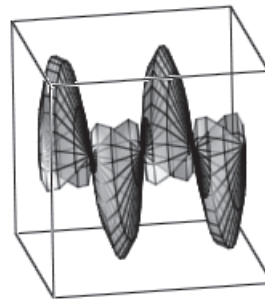
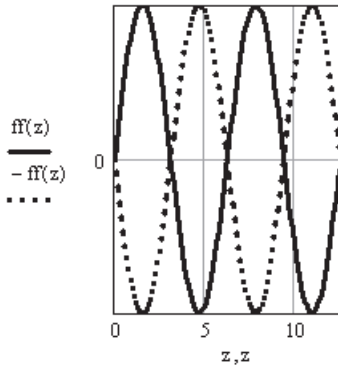
Рис. 15.15. Построение поверхности тела вращения с вырезом

На рис. 15.15 показано построение поверхности шара с применением функции `CreateMesh`. В качестве функции преобразования координат `fmap` прилагаются аналитическое преобразование цилиндрических координат в декартовы функцией `cyl2xyz` и преобразование сферических координат в декартовы функцией `sph2xyz`. Поменяйте эти выражения местами.

$$ff(\phi) := \sin(\phi) \text{ уравнение образующей} \quad \text{fmap}(v) := \text{cyl2xyz}(v_1, v_2, v_3)$$

$$K(z, \phi) := \begin{pmatrix} ff(z) \\ \phi \\ z \end{pmatrix} \quad \begin{matrix} a := 4 \cdot \pi & b := 1.2 \cdot \pi \end{matrix} \quad \text{Поменяйте пределы a и b}$$

$$S := \text{CreateMesh}(K, 0, a, 0, b, \text{fmap}) \quad z := 0, 0.1 .. a$$



Тело вращения с вырезом

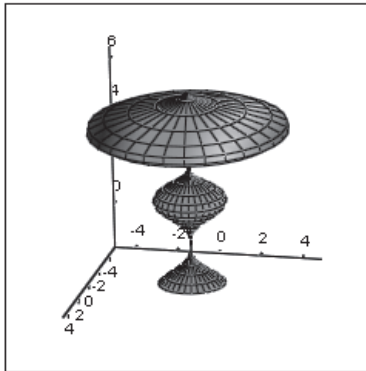
Рис. 15.16. Поверхность вращения плоской кривой

В этом примере функция CreateMech применена с параметрами и без параметров. Если использована функция без параметров, получается замкнутая фигура, а если с параметрами — есть возможность, меняя пределы изменения углов, модифицировать размеры выреза.

На рис. 15.16 показана возможность построения поверхности вращения по уравнению ее образующей. На рис. 15.17 подобное построение гораздо проще выполнено с помощью вкладки Quick Plot Data (Данные быстрого графика).

то же самое гораздо проще выполнить с помощью вкладки Quick Plot Data

$$f(x) := x \cdot \sin(x)^2 \quad S(u, v) = \begin{pmatrix} u \\ f(u) \cdot \cos(v) \\ f(u) \cdot \sin(v) \end{pmatrix}$$



S

$$x := -3.14, -3..2 \cdot \pi$$

график образующей

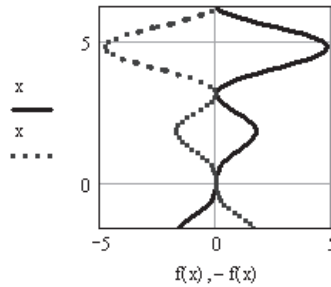


Рис. 15.17. График поверхности, построенный с помощью вкладки Quick Plot Data (Данные быстрого графика)

СОВЕТ

Не используйте в расчетах функцию CreateMech. Те же параметры можно и нужно вводить с помощью вкладки Quick Plot Data.

15.3.5. Построение сложных фигур

Построение пространственной кривой с применением функции CreateSpace

Встроенная функция CreateSpace принимает трехэлементную векторную функцию одной переменной и возвращает декартовы координаты пространственной кривой.

Обращение к функции:

$$\text{CreateSpace}(F, t_0, t_1, tgrid, fmap),$$

где F — трехэлементная векторная функция одной переменной; t_0 и t_1 — пределы изменения переменной, по умолчанию от -5 до 5 ; $tgrid$ — число точек кривой, по умолчанию 20 ; $fmap$ — функция преобразования координат. Параметр $fmap$ необязателен. При его отсутствии входная функция по умолчанию считается заданной в декартовых координатах.

Применение функции CreateSpace показано на рис. 15.18.

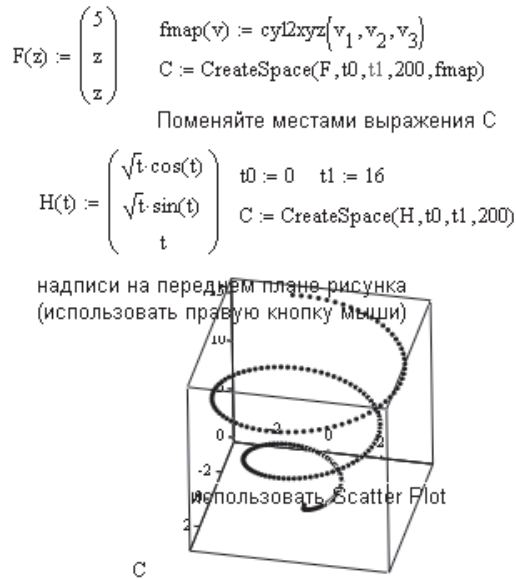


Рис. 15.18. Построение пространственной кривой

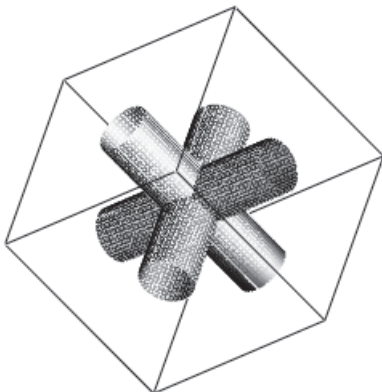
Пересекающиеся фигуры

На рис. 15.19 и 15.20 показаны возможности создания в Mathcad сложных пересекающихся фигур. Обе фигуры получены путем формирования массивов координат точек фигур по формулам. На этих фигурах поупражняйтесь в форматировании рисунков.

Пересечение цилиндров

$$n := 40 \quad i := 1..n \quad j := 1..n$$

$$X_{i,j} := \cos\left(\frac{2 \cdot \pi \cdot i}{n}\right) \quad Y_{i,j} := \sin\left(\frac{2 \cdot \pi \cdot i}{n}\right) \quad Z_{i,j} := \frac{j - 20}{5}$$



(X,Y,Z),(Z,Y,X),(Y,Z,X)

Рис. 15.19. Пересечение трех цилиндров

$$\text{ORIGIN} := 0 \quad N := 60 \quad i := 0..N$$

$$\phi_i := i \cdot 2 \cdot \frac{\pi}{N} \quad j := 0..N \quad \alpha_j := j \cdot 2 \cdot \frac{\pi}{N}$$

$$x_{i,j} := (5 + 2 \cdot \cos(\phi_i)) \cdot \cos(\alpha_j)$$

$$y_{i,j} := (5 + 2 \cdot \cos(\phi_i)) \cdot \sin(\alpha_j) \quad z_{i,j} := 2 \cdot \sin(\phi_i)$$



Рис. 15.20. Два связанных кольца

15.3.6. Построение многогранников

В Mathcad имеется возможность быстрого построения поверхностей многогранников. Всего можно построить 80 многогранников. Описание многогранников, которые Mathcad может вывести на экран, можно найти в Центре документации: Resource Center ▶ QuickSheets and Reference Tables ▶ Reference Tables ▶ Geometry Formulas ▶ Polyhedra (Центр ресурсов ▶ Таблица справочных данных ▶ Геометрические формулы ▶ Многогранники).

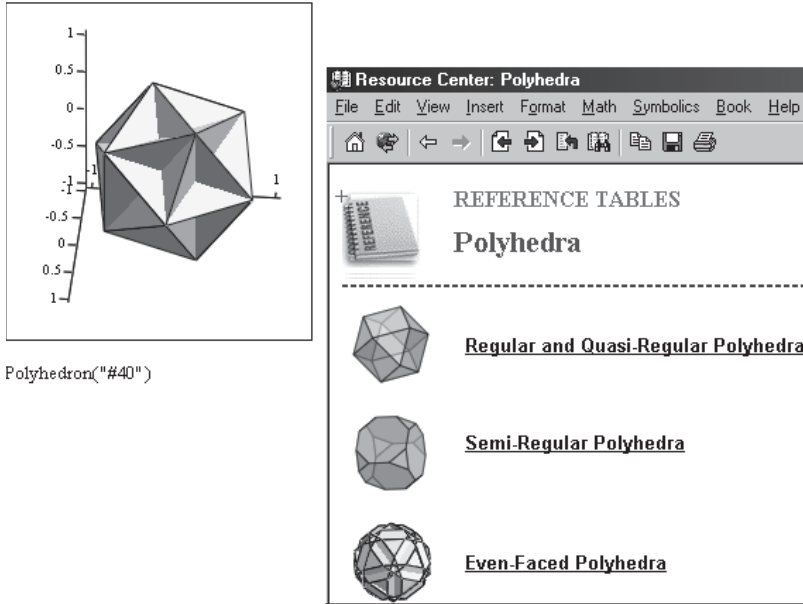


Рис. 15.21. Построение поверхностей полиэдров

Для построения полиэдра выведите на экран шаблон 3D-графика и в поле ввода напишите Polyhedron ("#"«номер от 1 до 80»). Примеры построения многогранников приведены на рис. 15.21 и 15.22.

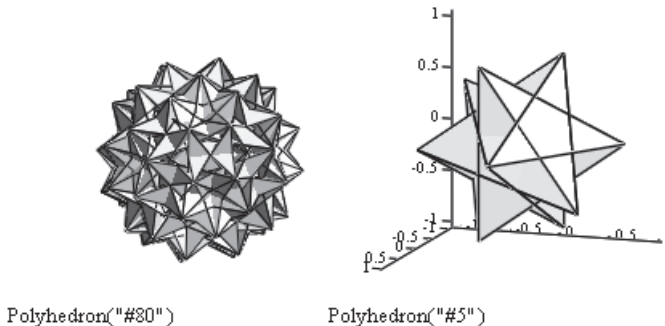


Рис. 15.22. Примеры построения поверхностей многогранников

15.3.7. Быстрое изменение типа графика

В заключение главы рассмотрим некоторые типы графиков, довольно редко используемых на практике, но тем не менее имеющих в арсенале Mathcad.

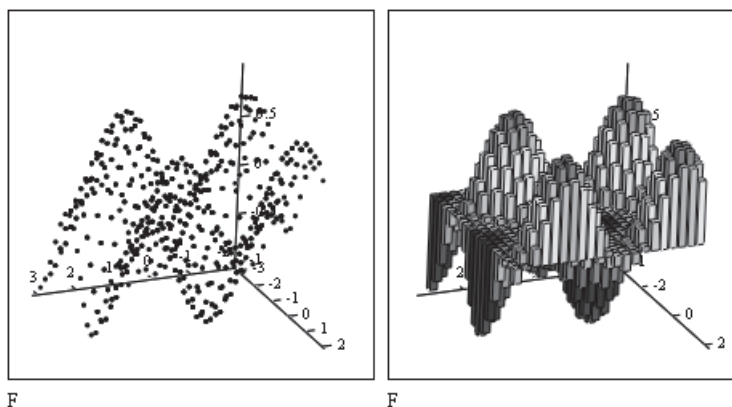


Рис. 15.23. Примеры построения точечного (слева) и столбчатого (справа) графиков

В окне форматирования 3D-графиков на вкладке **General** внизу (см. рис. 15.13) имеются шесть кнопок перестроения графиков:

- Surface Plot** — график поверхности;
- Contour Plot** — линии уровня;
- Data Points** — точки данных, расположение расчетных точек в пространстве (рис. 15.23, *слева*);
- Vector Field Plot** — векторное поле (рис. 15.25);
- Bar Plot** — столбчатая диаграмма (рис. 15.23, *справа*);
- Patch Plot** — лоскутный график, похожий на точечный график, каждая точка в нем имеет вид лоскутка (рис. 15.24).

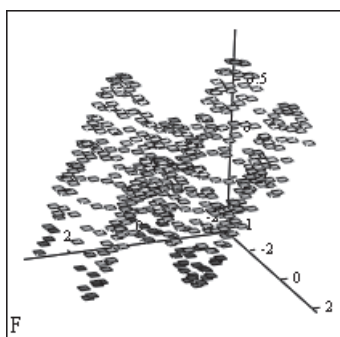


Рис. 15.24. График поверхности из лоскутков (Patch Plot)

Построив один из графиков, можно легко перестроить его в другой, отметив флажком соответствующий тип графика на вкладке **General**, за исключением **Vector Field Plot**.

15.3.8. График векторного поля

Векторное представление функции создается построением коротких стрелочек — векторов. Стрелочка направлена в сторону возрастания высоты поверхности (увеличения значений функции). Плотность стрелок зависит от скорости изменения функции, то есть от ее градиента. Длина стрелочки пропорциональна модулю вектора. Такого вида графики удобно приводить при расчете различных полей: электромагнитных, тепловых, гравитационных и т. д.

```

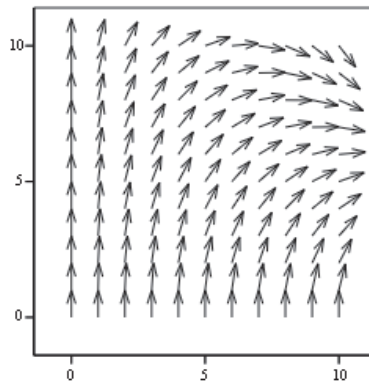
y := 0  обнуление прежнего значения массива y
F1(x,y) := sin(x·y)  F2(x,y) := cos(x·y)  n := 10  m := 10

i := 0..n  j := 0..m  x1 :=  $\frac{\pi}{2 \cdot n} \cdot i$   yj :=  $\frac{\pi}{2 \cdot m} \cdot j$ 

H1i,j := F1(x1,yj)  H2i,j := F2(x1,yj)  Gi,j := H1i,j + i·H2i,j

Поменяйте функцию F и число точек n и m

```



(H1,H2) Вместо (H1,H2) поставьте G

Рис. 15.25. Векторное поле функции, проекции которой — H_1 и H_2

При построении векторного поля требуется задать двухмерный массив значений функции в заданных точках. На рис. 15.25 показано векторное поле — поток функции G , у которой проекции на осях X и Y — это H_1 и H_2 соответственно. Иногда удобно представить функцию как матрицу комплексных чисел, где одна из проекций вектора — его вещественная часть, другая проекция — мнимая часть. Работая с примером, поменяйте функцию и количество точек.

15.4. Особенности форматирования графиков в новых версиях Mathcad

15.4.1. Редактирование двумерных графиков

Двойной щелчок мышью в поле двумерного графика вызывает появление окна форматирования графика. В Mathcad 13 в этом окне изменилась вкладка Traces (Трассировка). На рис. 15.26 показана вкладка Traces. Поскольку окно резко

увеличилось в размерах, не помещающаяся в окне часть вкладки «подклеена» к рис. 15.26. Столбцы в окне прежних версий стали строками в Mathcad 13. Отметим новый столбец Symbol Frequency (Частота символов), который позволяет показывать на графике не все расчетные точки, а каждую вторую, третью и т. д.

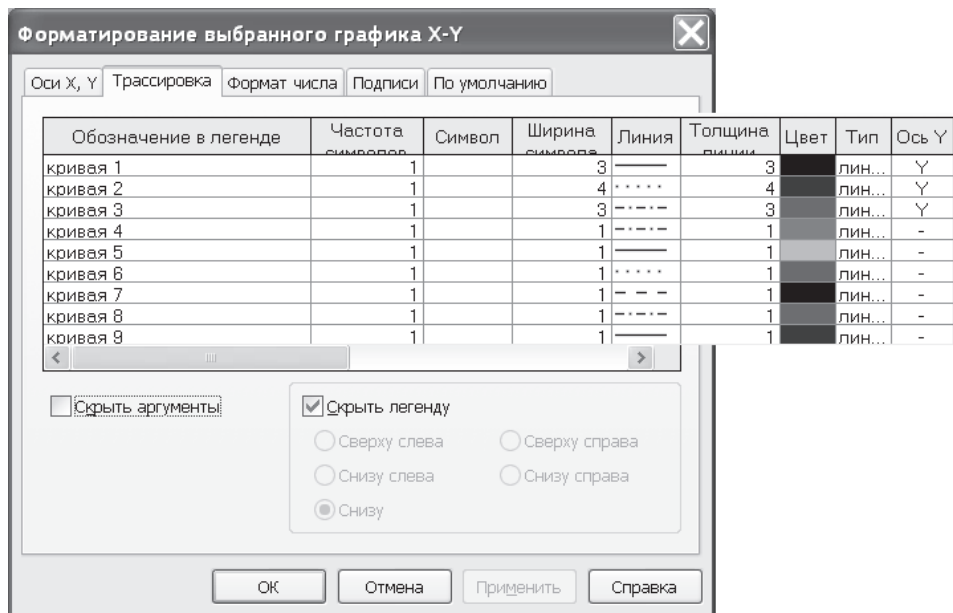


Рис. 15.26. Окно форматирования двумерного графика в Mathcad 15

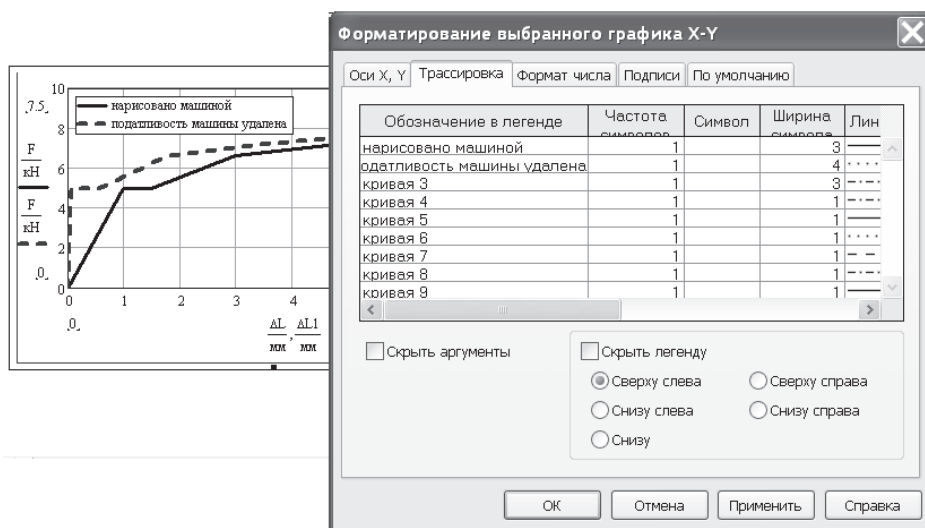


Рис. 15.27. Ввод комментария к графику

Появилась возможность выбора местоположения комментария к графику (рис. 15.27). Комментарий вводится на вкладке Trace (Трассировка). Вместо слов «кривая №» можно вписать свой текст. В нижней левой части вкладки можно выбрать его положение (внизу, сверху, слева, справа).

В Mathcad 12 появилась возможность указывать на графике численные значения на двух вертикальных осях Y (слева и справа от графика). На рис. 15.28 показана вкладка X-Y Axes окна форматирования графика. Для введения второй шкалы Y справа от графика вначале нужно отметить флажком пункт Enable secondary Y axis (Включить дополнительную ось Y), затем закрыть окно форматирования. Затем в шаблоне графика в появившемся справа месте ввода вписать имя функции, снова открыть окно форматирования на вкладке Trace (Трассировка) и установить нужные параметры графика.

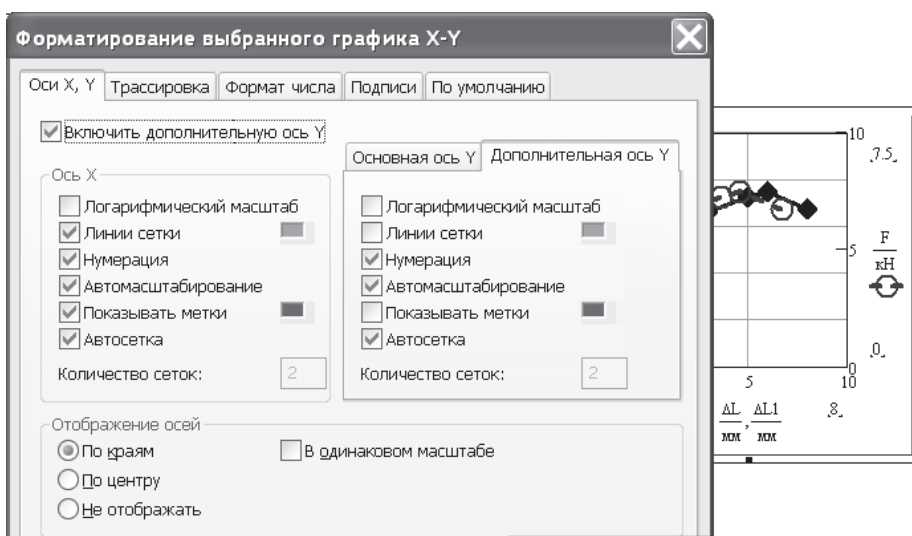


Рис. 15.28. Две шкалы по оси Y на плоском графике

15.4.2. Форматирование двумерных графиков

В Mathcad 14 в окне форматирования двумерных графиков появилась новая вкладка, позволяющая изменять формат чисел (рис. 15.29).

На рис. 15.30 приведен пример форматирования чисел, выводимых на графике. Для начала график был укрупнен путем вписывания нужных пределов чисел на оси X . Затем двойным щелчком мышью вызвано окно форматирования, в котором выбрана вкладка Number Format. Для наглядности выбран совсем уж экзотический вариант — представление чисел в виде дробей. Не самое нужное нововведение, но оно есть.

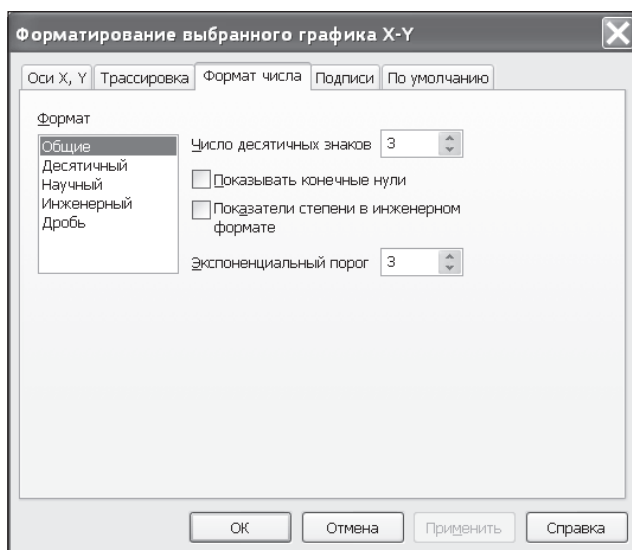


Рис. 15.29. Вкладка форматирования чисел в окне форматирования плоских графиков

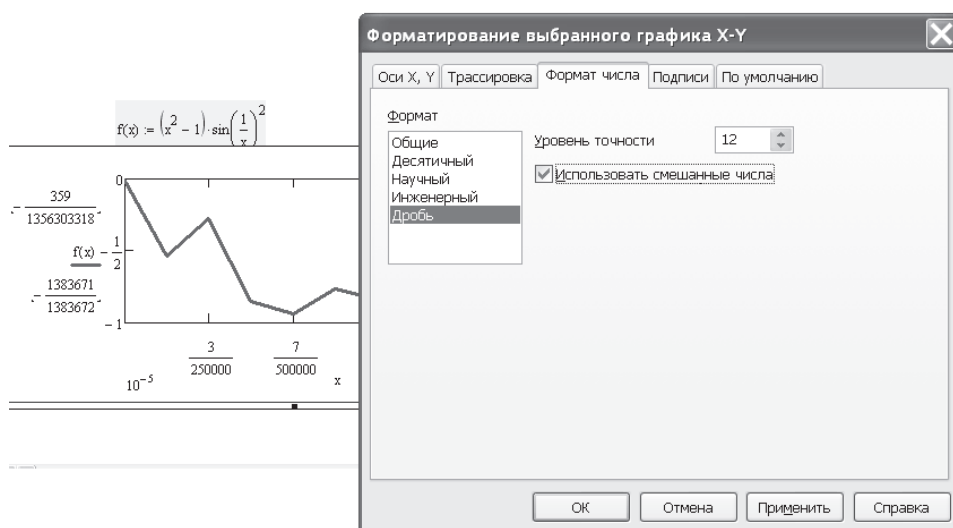


Рис. 15.30. Экзотика. Изменение формата чисел на осях графика

15.4.3. Функции logspace и logpts

В Mathcad 13 появились две функции для работы с графиками с использованием логарифмической шкалы.

Обращение к ним:

logspace(min, max, npts);

logppts(minexp, dec, dnpts),

где min и max — минимальное и максимальное значения диапазона чисел; dec — число декад (значащих цифр между ними); npts и dnpts — общее число точек и число точек в каждой декаде.

Функция logspace создает вектор чисел, распределенных по логарифмическому закону между min и max.

Функция logppts создает вектор чисел, распределенных по линейному закону внутри каждой декады числового ряда. Общее число точек равно произведению dec × dnpts. Пример создания таких массивов показан на рис. 15.31.

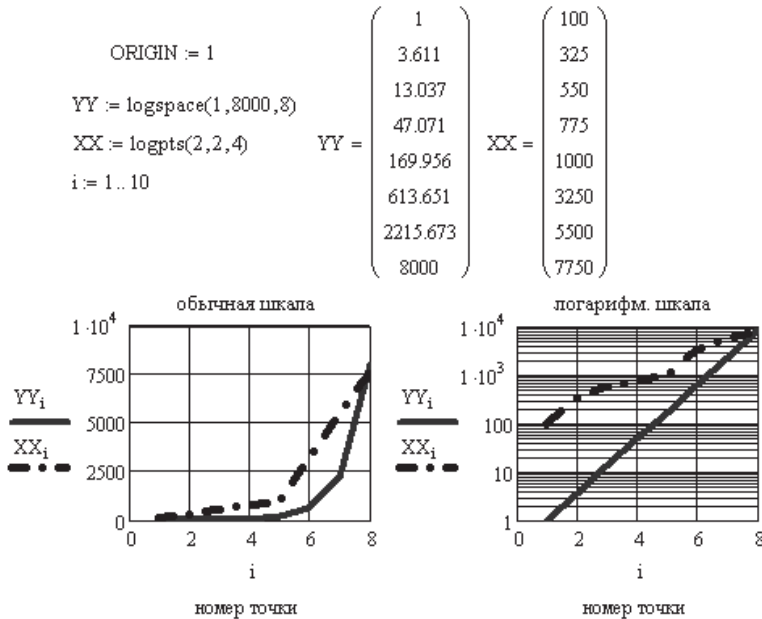


Рис. 15.31. Сравнение векторов чисел, созданных функциями logspace и logppts

Глава 16

Справочная информация в Mathcad

В меню Help Mathcad есть три источника справочной информации:

- ❑ собственно Help Mathcad (Справка), вызываемый клавишей F1 или непосредственно из меню Help;
- ❑ электронная книга qscheet, содержащая несколько разделов (в разных версиях они разные);
- ❑ Интернет.

Рассмотрим кратко эти источники информации.

16.1. Меню Help (Помощь)

При выборе пункта Help в главном меню Mathcad открывается подменю с несколькими пунктами (рис. 16.1):

- ❑ Mathcad Help (Справка по Mathcad) — система справок по всем вопросам работы в Mathcad;
- ❑ Developer's Reference (Руководство для разработчиков) — дополнительные справки для разработчиков приложений Mathcad;
- ❑ Author's Reference (Руководство для авторов) — дополнительные главы справки для авторов электронных книг;

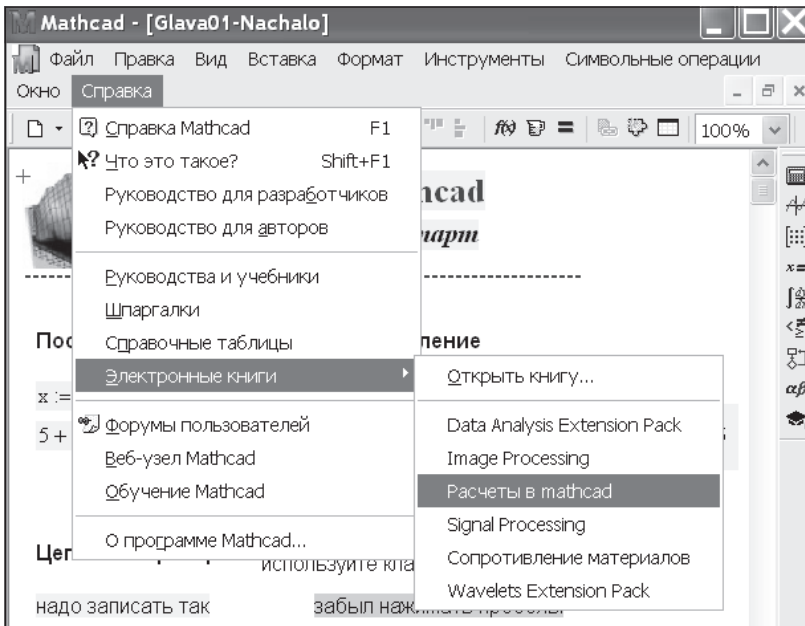


Рис. 16.1. Вызов справки в Mathcad

- ❑ Tutorials (Руководства и учебники) — сборник статей, обучающих работе в Mathcad;
- ❑ QuickSheets (Шпаргалки) — сборник примеров расчетов в Mathcad;
- ❑ Reference Tables (Справочные таблицы) — таблицы справочных данных;
- ❑ E-books (Электронные книги) — перечень электронных книг, включенных в оболочку Mathcad, в том числе две электронные книги;
- ❑ перечень интернет-адресов для связи с фирмой Mathsoft.

Во всех ранних версиях Mathcad (до Mathcad 11) все перечисленные пункты, кроме Mathcad Help, были объединены в один пункт Resource Centre (Центр документации).

Если в какой-либо момент работы в Mathcad вам понадобится помощь, выберите в главном меню команду Help ► Mathcad Help, либо щелкните на кнопке со знаком вопроса (?) на стандартной панели Mathcad, либо воспользуйтесь клавишей F1. Откроется окно справочной системы Mathcad (рис. 16.2).

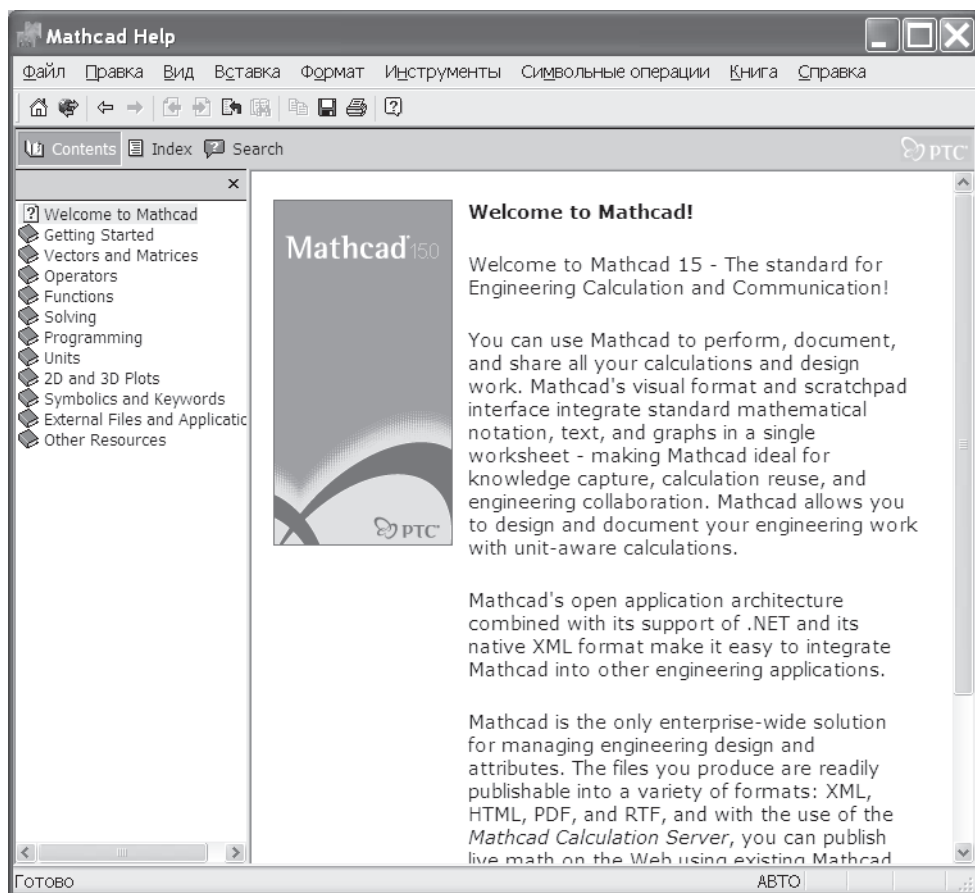


Рис. 16.2. Окно справки Mathcad

Справка в Mathcad является контекстно-зависимой, то есть ее содержание зависит от того, из какого места документа Mathcad она вызвана. Например, при установке курсора на имени функции `polyroots` и нажатии клавиши F1 на экране появляется окно Mathcad Help, открытое на месте описания функции `polyroots` (см. рис. 16.2). Окно справочной системы Mathcad построено в характерном для Windows стиле и состоит из двух частей. Слева отображаются названия статей, справа — их текст. Справочная система насыщена гиперссылками, находящимися в тексте статей и обеспечивающими переход от одной статьи к другой. Для перехода от одной статьи к другой установлены кнопки навигации Back (Назад), Forward (Вперед), Home (Домой).

В тексте статей часто встречаются кнопки QuickSheet Example, вызывающие пример расчета, связанный с содержанием справочной статьи.

Статей гораздо больше, чем заголовков в содержании. Для вывода большинства из них на экран потребуется один-два перехода по гиперссылкам.

В конце почти каждой статьи есть кнопка Related Topics (Статьи, близкие по смыслу). Щелчок мышью на этой кнопке открывает перечень статей на сходные темы, связанных гиперссылками с данными статьями.

16.2. Перевод справочной системы Mathcad

Все статьи справочной системы Mathcad написаны на английском языке. Если английский текст вам непонятен, воспользуйтесь какой-либо системой-переводчиком, например Prompt.

Выделите черным цветом фрагмент текста, который хотите перевести, или в главном меню Mathcad выберите команду Edit ▶ Select all (Редактировать ▶ Выделить все). Затем скопируйте выделенный текст в буфер (рис. 16.3). В меню Prompt выберите пункт Перевод Clipboard (Перевод содержимого буфера обмена). В появившемся окне выберите направление перевода: Англо-русский перевод — и нажмите кнопку ОК. Машинный перевод, конечно, примитивный, но зато быстрый, и, в общем, становится понятно, о чем идет речь (рис. 16.4).

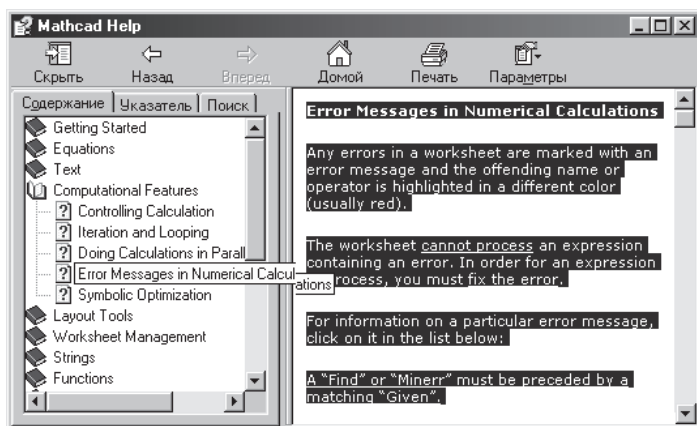


Рис. 16.3. Выделение текста для перевода

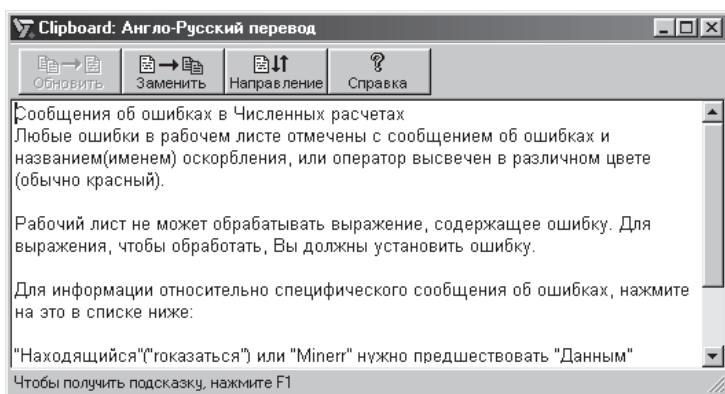


Рис. 16.4. Перевод текста, выделенного на рис. 16.3

В Mathcad Help все статьи подготовлены в текстовом редакторе, и поэтому электронный переводчик обрабатывает их без проблем.

Сложнее обстоит дело со статьями из электронных книг. Они подготовлены как часть электронной книги в Mathcad и не могут быть переведены текстовым электронным переводчиком. Эти статьи вначале надо перенести в текстовый редактор.

Выделите текст статьи электронной книги. Скопируйте его в буфер обмена и вставьте в пустой документ Mathcad. Поскольку электронная книга создана в Mathcad, то только в Mathcad ее можно преобразовать в текстовый файл.

В меню File (Файл) выберите команду Save as (Сохранить как), в открывшемся окне впишите какое-либо имя и выберите тип файла Rich Text Format File (*.rtf), затем нажмите кнопку Сохранить. Созданный документ стал текстовым файлом, и теперь он открывается в Word.

Из документа Word текст скопируйте в буфер обмена, затем вызовите электронный переводчик и выберите команду Перевод Clipboard ► Англо-русский перевод. Примеры перевода текста из электронной книги Resource Center приведены на рис. 16.6–16.9.

ПРИМЕЧАНИЕ

Электронный переводчик Prompt 7 переводит тексты, созданные в различных Windows-приложениях, включая Mathcad, после копирования их в буфер обмена и не требует преобразования в текстовый файл.

16.3. Электронная книга qscheet

Электронная книга qscheet представляет собой сборник статей и примеров, поясняющих работу операторов и встроенных функций Mathcad, а также примеров решения различных математических, физических, инженерных задач с помощью Mathcad.

Чтобы вызвать эту электронную книгу на экран, следует в меню Help выбрать нужный раздел (см. рис. 16.1). На экране появится титульный лист этой электронной книги (рис. 16.5).

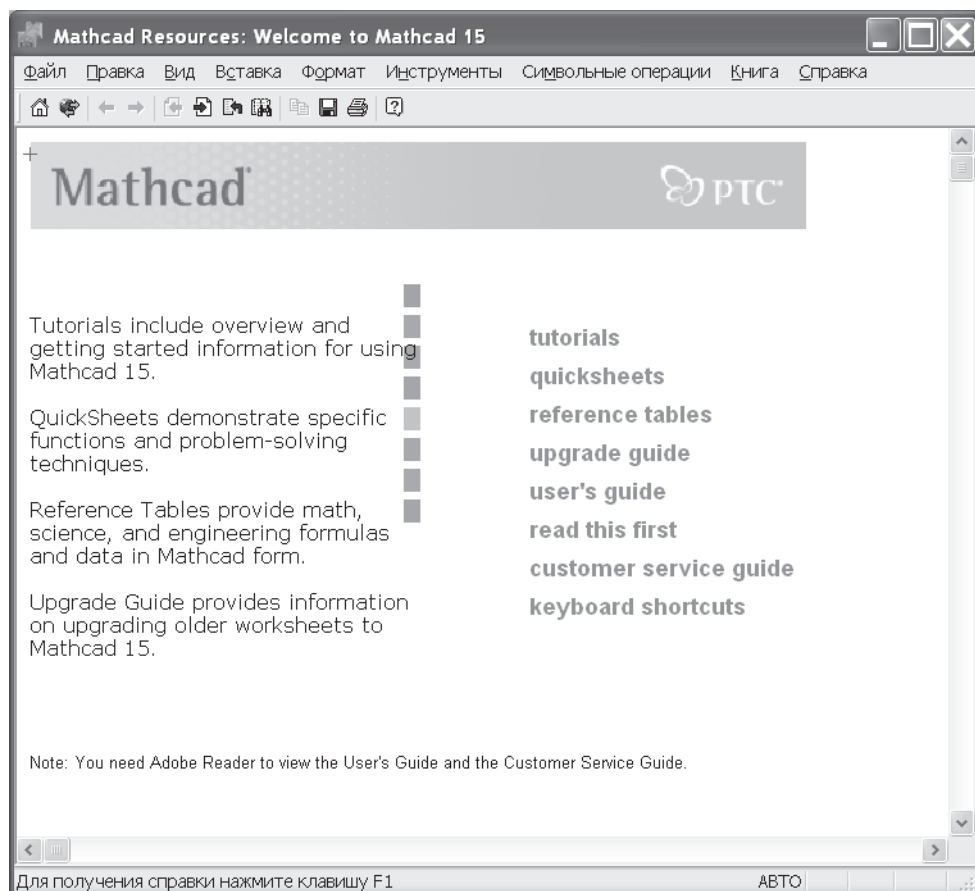


Рис. 16.5. Заглавная страница электронной книги qsheet

На титульном листе все заголовки являются гиперссылками. Щелчок мышью на них открывает соответствующий раздел книги. Всего разделов три:

- Tutorial (Учебник) — информация о возможностях системы и основные приемы работы с Mathcad;
- QuickSheets — сборник примеров;
- Reference Tables — справочные таблицы.

Передвигаться по разделам электронной книги очень легко. В начале каждого раздела и подразделов приводится оглавление, выполненное в виде гиперссылок. При наведении на гиперссылку курсор приобретает вид руки с вытянутым указательным пальцем. При щелчке мышью на соответствующей гиперссылке пользователь переходит в нужный раздел электронной книги.

Особенностью электронных книг в Mathcad является выполнение расчетов прямо на страницах книги, причем пользователь может изменять любые параметры и наблюдать изменение результатов расчета. Правда, сохранить результаты пересчета прямо в электронной книге нельзя, поскольку в ней включена защита от записи. Зато нужные фрагменты книги можно скопировать в свой документ.

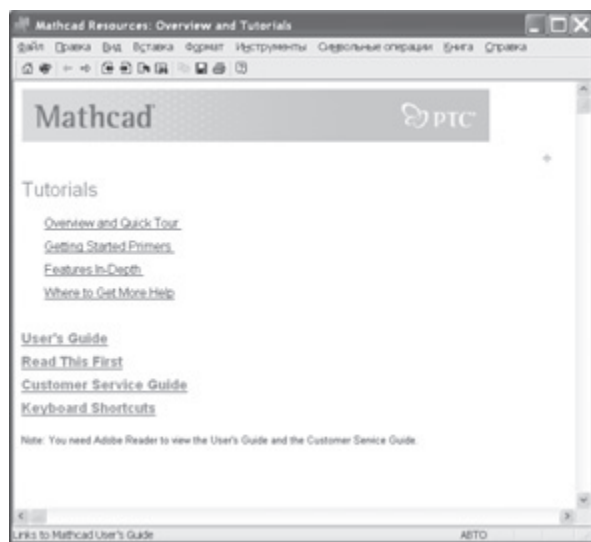
Чтобы скопировать фрагмент книги, надо выделить его пунктирной рамкой, обведя курсором при нажатой левой кнопке мыши, а затем скопировать в буфер обмена и оттуда вставить в нужное место своего документа.

Есть и более простой способ копирования, типичный для всех Windows-приложений. Можно выделенный фрагмент перетащить по экрану в свой документ при нажатой левой кнопке мыши.

16.3.1. Обзор и учебник

В разделе Tutorial (Учебник) пользователь найдет основные сведения по данной версии Mathcad на английском языке. В Mathcad 15 включен раздел Users Guide на русском языке. Чтобы открыть его, выберите пункты Справка ► Руководства и учебники ► Users Guide. На рис. 16.6 показано оглавление этого раздела.

Начало учебника (рис. 16.7) посвящено обучению основам работы в Mathcad и похоже на главу 1 данной книги. В последующих разделах идет углубленное изучение Mathcad и приведено большое количество примеров, как правило, очень простых.



Содержание

Сведения о данном руководстве	1
Другие документы	1
Другие ресурсы	1
Техническая поддержка	21
1: Знакомство с Mathcad	31
"Что такое Mathcad?"	34
2: Присутствие в работе с Mathcad	45
Работа с объектами Mathcad	45
Объекты	47
Панель вычисления	47
Создание и редактирование	81
График	111
Скриншоты, печать, печать на принтере	121
3: Интерактивные ресурсы	151
Ресурсы Mathcad	151
Формы пользовательской Mathcad	202
Другие ресурсы	203
4: Работа с математическими выражениями	231
Вставка математических выражений	238
Область выражений	251
Правила выражений	303
Математические ссылки	355
5: Переменные, диаграммы и массивы	381
Создание массивов	389
Итерационные расчеты	402
Диаграммы и диаграммы сетки	441
Создание массивов	449
Работа с массивами	494
6: Работа с текстом	521
Вставка текста	529
Скрытие текста и абзаца	534
Стиль текста	534
Уравнения в тексте	527
Технические инструменты	527

Рис. 16.6. Первая страница учебника и оглавление User Guide на русском языке

16.3.2. Примеры и справочные таблицы

В разделе Quickschets электронной книги сосредоточено большое количество примеров, в основном из математики и физики, с использованием большинства встроенных функций и различных возможностей Mathcad.

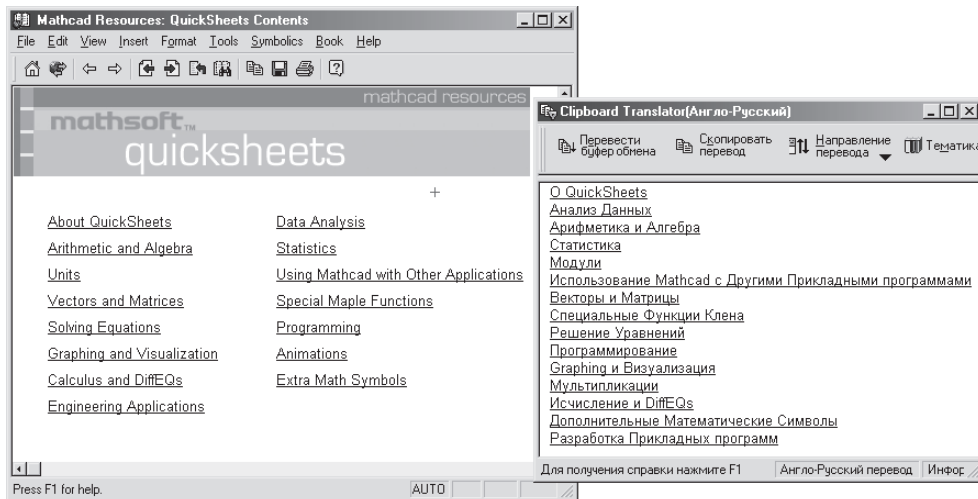


Рис. 16.7. Оглавление раздела Getting Started Tutorial с переводом

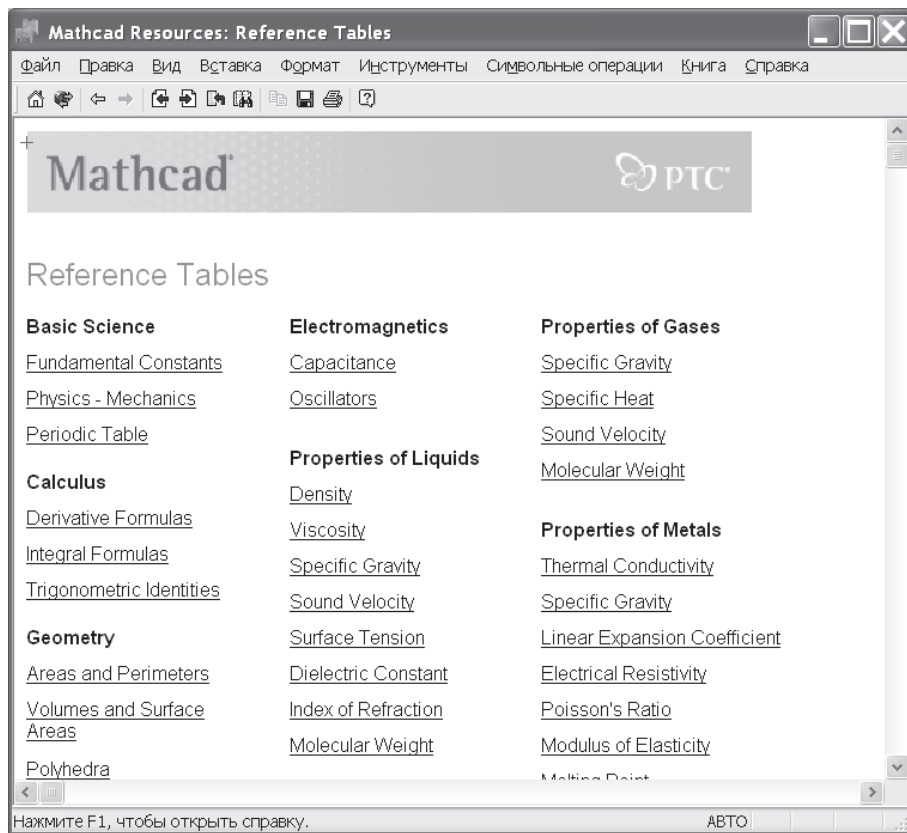


Рис. 16.8. Первая страница раздела Reference Tables

Раздел Reference Tables (Справочные таблицы) содержит простейшие математические формулы, сведения о свойствах различных материалов и т. п. (рис. 16.8). Сведения эти бессистемные и в общем мало кому нужные. Правда, в разделе Geometry Formulas (Геометрические формулы) есть хоть и не очень нужный, но весьма любопытный раздел Polyhedra (Многогранники), содержащий графики поверхностей 80 многогранников (правильных). Об использовании этого раздела шла речь в главе 15 (см. рис. 15.20 и 15.21).

16.4. Mathcad и Интернет

Для работы в Интернете Mathcad имеет встроенный браузер, позволяющий пользоваться услугами Интернета, не выходя из программы. Кнопки доступа в Интернет из среды Mathcad находятся в выпадающем меню кнопки Help стандартного меню Mathcad (см. рис. 16.1):

- ❑ User Forums — форумы пользователей Mathcad;
- ❑ Mathcad.com — открытие начальной страницы фирмы Mathsoft;
- ❑ Consulting & Training — обучение Mathcad.

Встроенный в Mathcad браузер обладает рядом недостатков, поэтому для работы с сайтом PTC Mathcad лучше пользоваться другими браузерами (например, Internet Explorer). Запустите браузер и наберите в его окне поиска ключевую

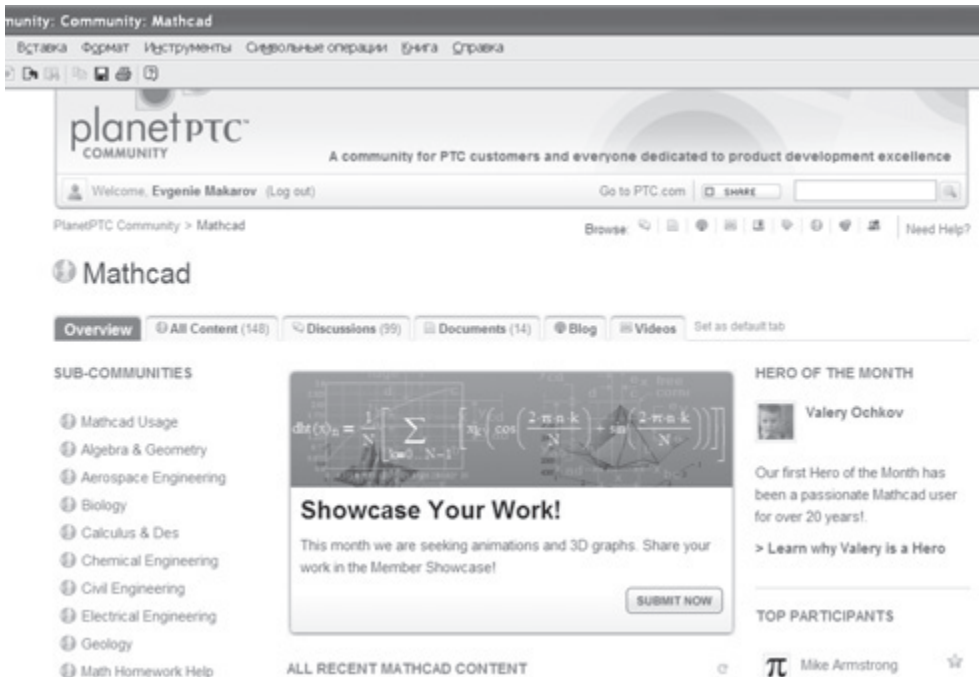


Рис. 16.9. Страница сайта форума Mathcad, открытая в браузере Mathcad

фразу *PTC Mathcad*, и вы получите большой список тематических ссылок. Можно самостоятельно ввести следующие адреса:

- ❑ <http://www.ptc.com/> — начальная страница фирмы PTC, которая в 2003 году купила фирму Mathsoft и в настоящее время продолжает работу по совершенствованию системы Mathcad.
- ❑ <http://www.ptc.com/appserver/mkt/products/home.jsp?k=3901> — помимо кратких описаний программных продуктов, выпускаемых фирмой PTC, здесь можно найти полезные утилиты (как платные, так и бесплатные) и сотни файлов с примерами их использования, последние новости и другую полезную информацию.
- ❑ <http://communities.ptc.com/community/mathcad> — сайт, на котором пользователи из разных стран обсуждают вопросы, связанные с работой в Mathcad (рис. 16.9).

16.4.1. Техническая поддержка Mathcad

На сайте PTC Mathcad работает служба технической поддержки (ссылка Support на главной странице сайта). Зарегистрированные пользователи имеют полный доступ к документации по системе Mathcad и другой полезной информации, возможности незарегистрированных пользователей ограничены.

В целом Mathcad неплохо интегрирован с Интернетом. Фирма Mathsoft много работает именно в этом направлении. Большинство изменений, появившихся в последних версиях Mathcad, в первую очередь касаются работы в Интернете.

Глава 17

Создание электронных книг

Создание электронных книг — своего рода высший пилотаж в среде Mathcad. Электронная книга сочетает в себе текст и примеры вычислений в Mathcad. Большое количество ссылок обеспечивает быстрое и удобное перемещение по книге, поиск нужного фрагмента. Все приведенные примеры работают, допускают изменение параметров расчета с мгновенным пересчетом примера. Изменения в электронной книге нельзя сохранить в ней, но любой фрагмент книги можно скопировать в другой файл Mathcad и сохранить там.

В Mathcad 15 включено пять электронных книг:

- Quick Sheets — примеры использования операторов и встроенных функций Mathcad. Эта книга содержит три раздела:
 - Tutorials — обучение работе в Mathcad;
 - Quick Sheets — примеры расчетов в Mathcad;
 - Reference Tables — справочные таблицы;
- Data Analysis Extension Pack — анализ наборов данных;
- Image Processing — цифровая обработка изображений в Mathcad;
- Signal Processing — обработка сигналов в Mathcad;
- Wavelets Extension Pack — волновые преобразования массивов данных.

Перечисленные электронные книги можно вызвать на экран из главного меню Mathcad, выбрав пункт Справка ► Электронные книги (рис. 17.1). В оболочку Mathcad я добавил и три свои электронные книги:

- «Инженерные расчеты в Mathcad» — она содержит все, что вы видите на страницах этой книги;
- «Справочник по инженерным расчетам в Mathcad» — электронный конструктор для сборки собственных программ из готовых фрагментов. Этот справочник выпущен фирмой «Новый диск» на компакт-диске под названием «Mathcad — конструктор программ»;
- «Сопротивление материалов» — подготовка нового компакт-диска с примерами решения прочностных задач.

Наличие на компьютере встроенной в оболочку Mathcad нужной электронной книги очень полезно и удобно. Благодаря системе вложенных друг в друга оглавлений и системе ссылок можно практически мгновенно найти нужный материал, скопировать необходимый фрагмент в буфер обмена и вставить его в собственную программу.

Создание электронных книг предусматривает следующие действия:

1. Создание необходимых Mathcad-файлов.
2. Подготовку содержания электронной книги.

3. Добавление электронных ссылок из содержания на соответствующие файлы.
4. Создание НВК-файла, определяющего порядок следования файлов в электронной книге.
5. Создание индексов для поиска нужного фрагмента книги.
6. Проверку правильности индексов, НВК-файла и Mathcad-файлов, включенных в электронную книгу.

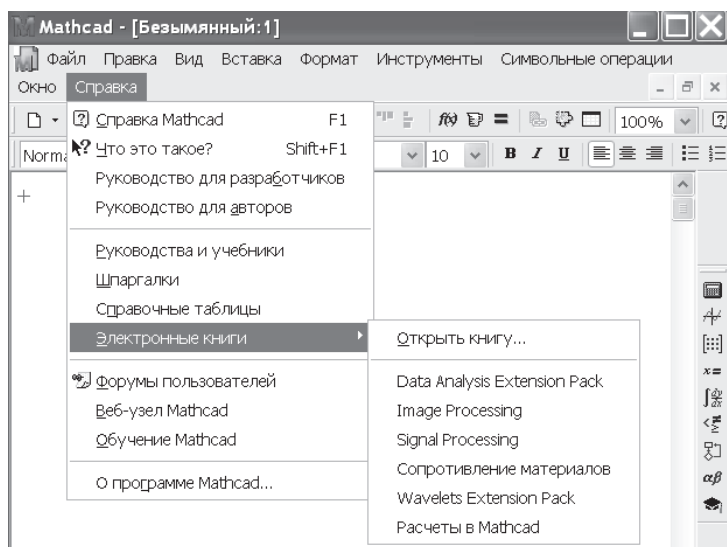


Рис. 17.1. Окно выбора электронных книг

17.1. Создание Mathcad-файлов для электронной книги

Все файлы, включаемые в электронную книгу, должны быть помещены в одну папку, имеющую то же название, что и электронная книга. Наличие вложенных папок допускается, но не рекомендуется, так как при этом некоторые функции электронной книги станут недоступными. Если в тексте предполагается появление всплывающих окон со вспомогательным материалом, то содержание этих окон как раз и надо размещать во вложенной папке под именем **Popups**.

Полезно перед началом подготовки электронной книги создать свои пользовательские стили, отличающиеся от стандартных цветом, шрифтом и т. д. Это удобно для придания единообразия всем создаваемым документам.

На экране компьютера люди читают текст значительно медленнее, чем на листах обычной книги, поэтому для облегчения чтения примите к сведению несколько советов:

- Текст на экране должен быть сравнительно крупный и четкий, черный на белом фоне. Ключевые слова лучше выделить другим шрифтом или цветом.

Обычный текст лучше печатать полужирным шрифтом размером 12 пунктов. Жирный текст читать тяжелее, поэтому его используйте для отдельных ключевых слов. Ссылки желательно подчеркнуть и выделить цветом.

- ❑ Математические области в документе должны быть упорядочены, выровнены, снабжены краткими текстовыми комментариями. Желательно, чтобы на экране можно было видеть одновременно весь законченный по смыслу фрагмент вычислений. Основные уравнения, а также исходные данные, которые можно менять, желательно выделить цветом.
- ❑ Включайте в текст больше заголовков, разделяя ими разные по смыслу фрагменты вычислений.
- ❑ При вводе текста в Mathcad-документ избегайте использования клавиши Enter для перехода на другую строку. Набирайте текст подряд, а затем, подводя курсор к краю рамки, установите нужный размер текстовой области. В этом случае достигается наилучшая видимость на мониторах различных размеров.

17.2. Подготовка содержания электронной книги

Закончив работу с Mathcad-файлами, подготовьте оглавление электронной книги. Поместите его в файл под названием TOC.mcd. Оглавление должно быть как можно более подробным. Рекомендуется делать вложенные оглавления, тогда основное оглавление не будет чрезмерно большим. Почти каждому крупному пункту основного оглавления будет соответствовать связанное с ним ссылкой вложенное оглавление. Пример основного и вложенного оглавлений показан на рис. 17.2.

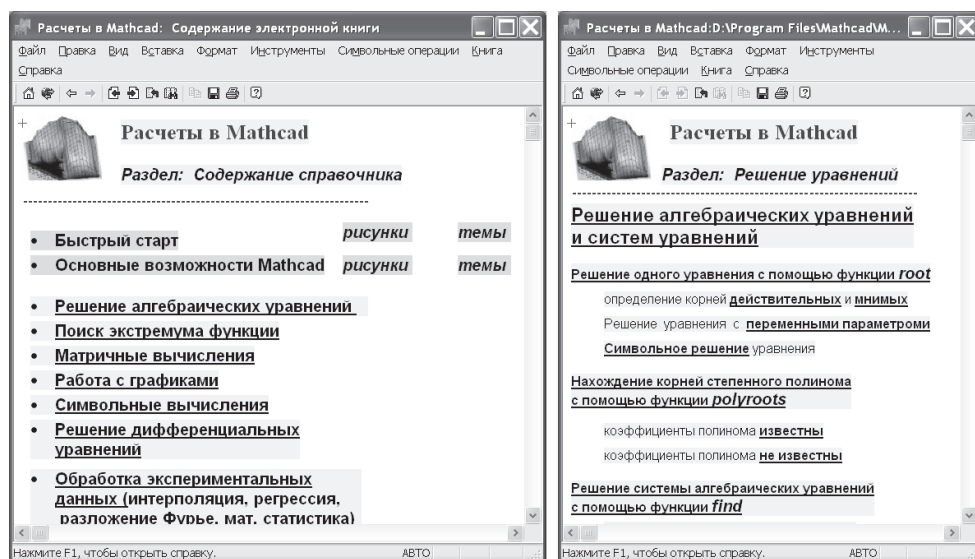


Рис. 17.2. Основное и одно из вложенных оглавлений электронной книги

17.3. Создание системы ссылок между файлами

Для быстрого просмотра содержания электронной книги надо установить ссылки между файлами. В главе 13 (раздел 13.4) описывается порядок создания гиперссылок (Hyperlink) между файлами. Напомним его.

Гиперссылка позволяет открывать файл, с которым создана связь, в той оболочке, в которой он создан. Для создания гиперссылки нужно:

- ❑ выделить слово или фрагмент текста, с которым будет связан интересующий вас файл, желательно, черным цветом. Тогда после установления связи выделенное слово будет выделено жирным шрифтом и подчеркнуто;
- ❑ в главном меню Mathcad выбрать Insert ► Hyperlink ► Browse (Вставка ► Гиперссылка ► Обзор) ► указать путь к файлу, с которым надо связать выделенное слово ► ОК;
- ❑ в окне Edit Hyperlink (редактировать гиперссылку) установить флажок на пункте Use relative path for hyperlink (Использовать относительный путь для гиперссылки). В этом случае связь будет работать даже в том случае, если вы переместили каталог с электронной книгой в другое место;
- ❑ двойной щелчок мышью на выделенном слове открывает файл, с которым установлена связь.

Если вы хотите вызвать на экран не начало, а определенное место выбранного файла, то его необходимо задать (этот процесс описан в разделе 14.2.1):

- ❑ откройте файл, на который будет создана ссылка;
- ❑ установите курсор на объект (слово, выражение или график), который при вызове файла будет находиться сверху на экране;
- ❑ щелкните на нем правой кнопкой мыши и в открывшемся контекстном меню выберите Properties (Свойства);
- ❑ в окне Tag (Метка) впишите имя метки (в нашем случае — 931), как показано на рис. 17.3. Нажмите кнопку ОК.

После создания метки следует создать гиперссылку, как указано ранее. Вводя путь к файлу, укажите и имя метки, записав в конце адреса, например, #931 (рис. 17.4). Теперь при двойном щелчке мышью на слове, с которым связана гиперссылка, указанный в адресе файл откроется с того места, на которое установлена метка.

Чем больше установлено гиперссылок, тем быстрее пользователь сможет найти нужный материал в электронной книге.

17.4. Создание НВК-файла

НВК-файл играет ту же роль, что и переплет в обычной книге. Он устанавливает порядок следования файлов в электронной книге.

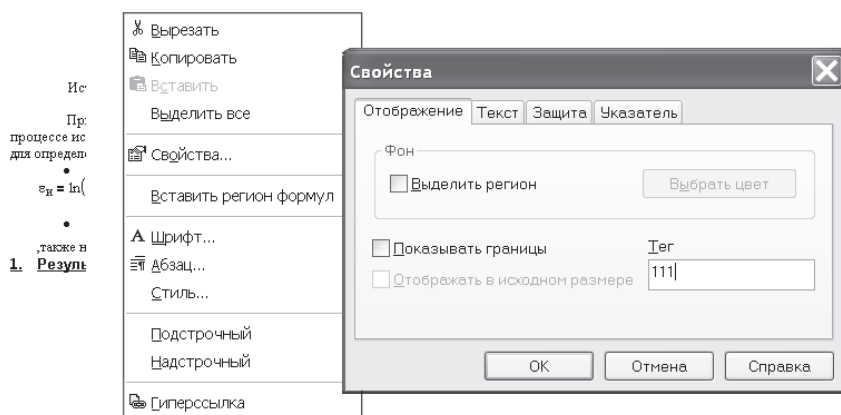


Рис. 17.3. Установка метки на место, с которого откроется файл

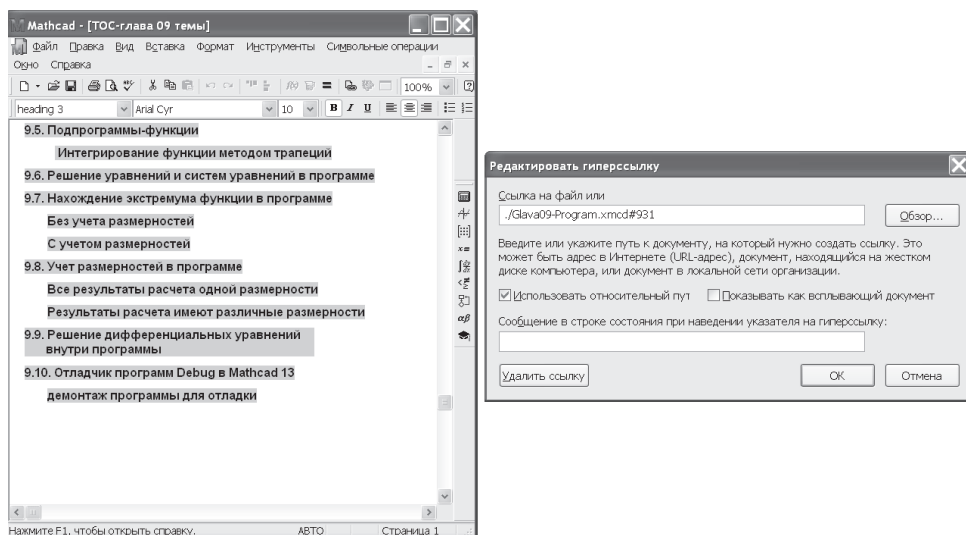


Рис. 17.4. Установка гиперссылки между файлами

НВК-файл — это простой текстовый файл, который обеспечивает:

- порядок обзора или последовательность Mathcad-файлов в вашей книге;
- Mathcad-файлы, соответствующие оглавлению, индексу и открытой странице;
- текст, появляющийся в области заголовка окна электронной книги при открытии любого раздела;
- заголовок книги, как замечено в области заголовка и меню Help ► Open Book;
- сообщения, которые появляются, когда читатель нажимает гиперссылку.

Для создания НВК-файла можно использовать любой текстовый редактор, например Блокнот Windows или Microsoft Word.

В начале НВК-файла должны присутствовать четыре ключевых слова:

- Version;
- Title;
- Splash;
- TOC (должно быть набрано прописными буквами).

Пример НВК-файла показан на рис. 17.5.

```

;Электронная книга Mathcad
;Copyright 2005 Макаров All rights reserved.
.version 2000
.title Инженерные расчеты в Mathcad
.path электронная книга2
;
splash splash.mcd
TOC TOC.mcd : Содержание справочника
Glava01-Nachalo Glava01-Nachalo.mcd : Построение выражений и графиков в Mathcad
Glava02-Vozmojnosti Glava02-Vozmojnosti.mcd : Основные возможности Mathcad
Glava03-Uravnienie Glava03-Uravnienie.mcd : Решение уравнений
Glava04-Matrix Glava04-Matrix.mcd : Работа с матрицами
Glava05-Symbol Glava05-Symbol.mcd : Символьные вычисления
Glava06-odesolve Glava06-odesolve.mcd : Решение дифференциальных уравнений
Glava06-старые Glava06-старые.mcd : Решение дифференциальных уравнений
Glava06-частные Glava06-частные.mcd : Решение дифференциальных уравнений
Glava07-эксперимент Glava07-эксперимент.mcd : Обработка экспериментальных данных
Glava08-Statistica Glava08-Statistica.mcd : Математическая статистика
Glava09-Program Glava09-Program.mcd : Программирование
Glava10-Animation Glava10-Animation.mcd : Анимация
Glava11-Razmernost Glava11-Razmernost.mcd : Учет размерностей
Glava12-Specfunction Glava12-Specfunction.mcd : Встроенные функции
Glava15-Graphic Glava15-Graphic.mcd : Построение графиков

```

Рис. 17.5. Пример НВК-файла

Строки НВК-файла, начинающиеся с точки с запятой, игнорируются Mathcad и могут использоваться как комментарии.

Каждый Mathcad-файл внутри НВК-файла записывается в три столбца:

- в первом столбце пишется имя файла без расширения;
- во втором столбце — имя файла с расширением .mcd;
- в третьем столбце пишется двоеточие (:), пробел (лучше несколько пробелов) и заголовок файла, который должен появиться в области заголовка при просмотре электронной книги.

Для перехода между столбцами используйте клавишу табуляции (Tab).

При сохранении НВК-файлу присваивается имя, совпадающее с названием папки, в которой хранятся все Mathcad-файлы. После него нужно вписать расширение .hbk. Например, все файлы нашей электронной книги хранятся в папке с названием Primer, и НВК-файл имеет имя Primer.hbk.

17.5. Вложенные папки

Если в электронной книге очень много файлов, то можно использовать вложенные папки. В этом случае в верхней части НВК-файла нужно указать путь к каждой папке ниже заголовка и версии.

К папкам, содержащим всплывающие окна (Popups), указывать путь не надо, так как эти файлы не включаются в НВК-файл.

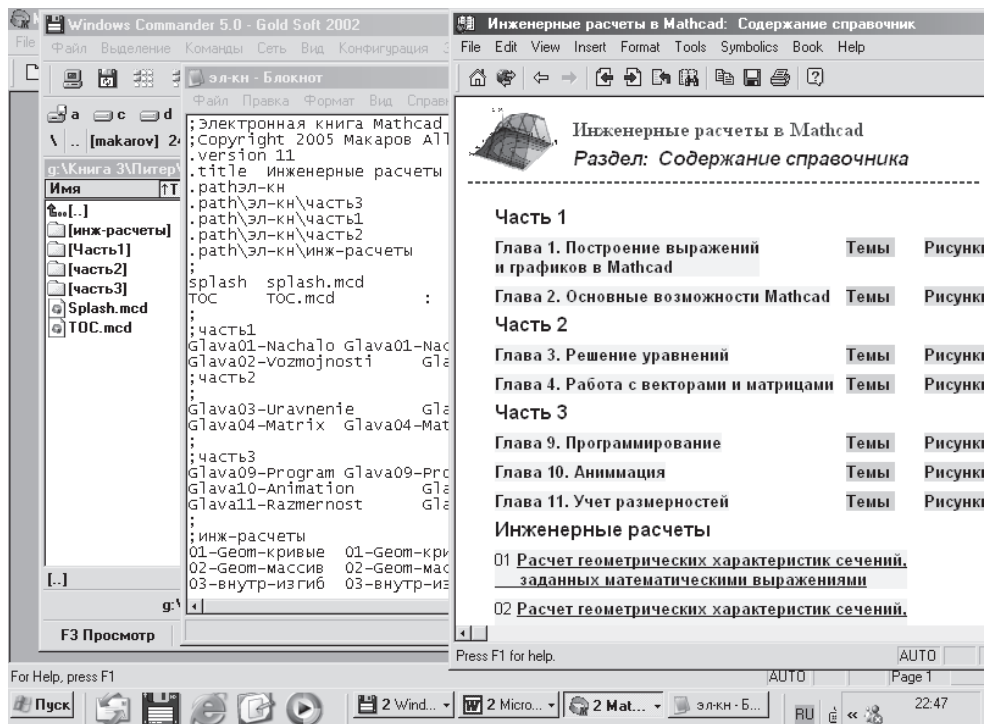


Рис. 17.6. Вложенные папки в электронной книге

Если все Mathcad-файлы содержатся в одной папке, указывать путь к папке не обязательно. На рис. 17.6 показан вариант НВК-файла для электронной книги с вложенными папками.

ВНИМАНИЕ

При наличии вложенных папок становится невозможной навигация по электронной книге с помощью стрелок \leftarrow и \rightarrow , но все ссылки работают нормально.

17.6. Проверка электронной книги

Для проверки НВК-файла необходимо перезагрузить Mathcad в режиме НВК. Для этого в окне Windows нажмите кнопки Пуск \blacktriangleright Выполнить. В появившемся окне Запуск программы выберите Обзор и укажите путь к файлу mathcad.exe. Далее через пробел введите /hbkmode (рис. 17.7, *слева*), затем ОК. Mathcad будет загружен в режиме НВК. На экране появится окно, изображенное на рис. 17.7, *справа*. Выберите Check Handbook, укажите путь к проверяемому НВК-файлу, нажмите ОК. Результаты проверки автоматически записываются в файл ErrorLog.txt в тот же каталог, где находится НВК-файл. Вид этого файла показан на рис. 17.8.

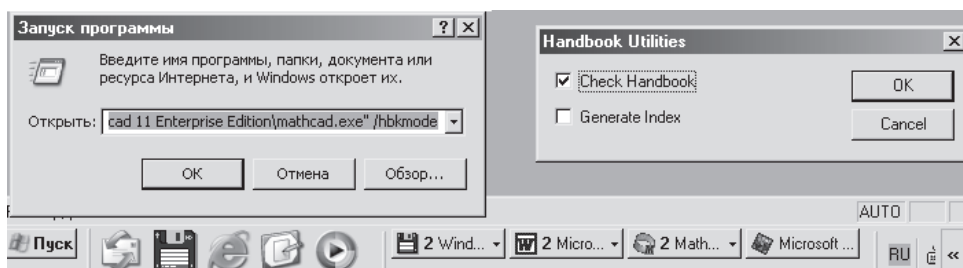


Рис. 17.7. Запуск программы проверки электронной книги

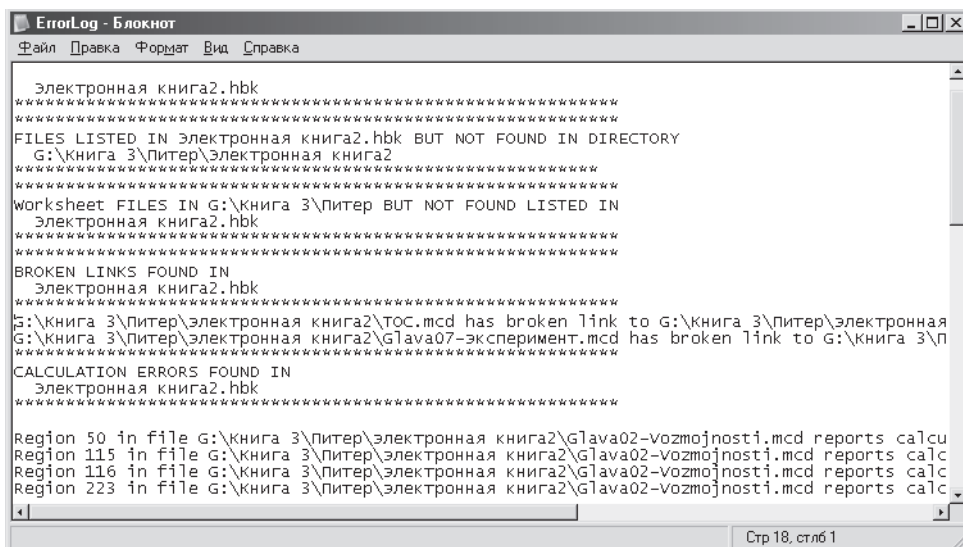


Рис. 17.8. Файл результатов проверки электронной книги

Файл ErrorLog сообщает о следующих ошибках:

- файлах в папках электронной книги, которые не перечислены в НВК-файле;
- файлах, перечисленных в НВК-файле, которых нет в папках электронной книги;
- неправильно установленных гиперсвязях;
- вычислительных ошибках в каждом Mathcad-документе электронной книги.

После проверки и исправления отмеченных ошибок надо снова запустить Mathcad в режиме НВК, добиваясь безошибочной работы электронной книги.

17.7. Создание поисковых индексов

Для поиска нужного фрагмента текста, содержащего ключевое слово или фразу, можно создать систему поисковых индексов. Работа эта кропотливая и длительная. В каждом из множества файлов надо отыскать все характерные слова и отметить их индексом в свойствах объекта, тогда в электронной книге открывается система поиска фрагментов по ключевым словам (рис. 17.9).

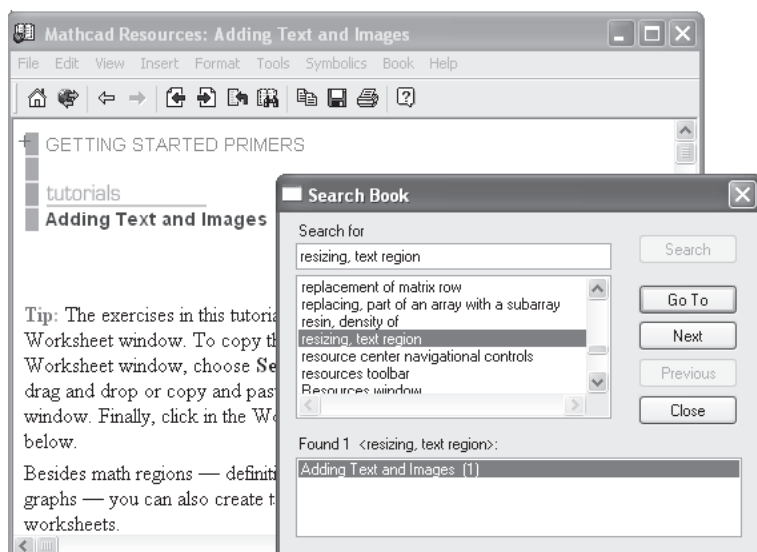


Рис. 17.9. Система поиска в электронной книге

Для создания системы поисковых индексов надо запустить Mathcad в авторском режиме (Authoring mode). Для этого в стартовом окне Windows нажмите кнопки Пуск ► Выполнить. В появившемся окне Запуск программы (см. рис. 17.7, *слева*) выберите Обзор и укажите путь к файлу mathcad.exe. Далее через пробел введите /a, затем нажмите ОК. Mathcad будет загружен в авторском режиме. На экране появится окно, изображенное на рис. 17.7, *справа*. Выберите Generate Index, нажмите ОК.

После запуска Mathcad откройте нужный файл, выберите объект, который вы хотите пометить индексом (любой — текстовый или математический). Нажав на

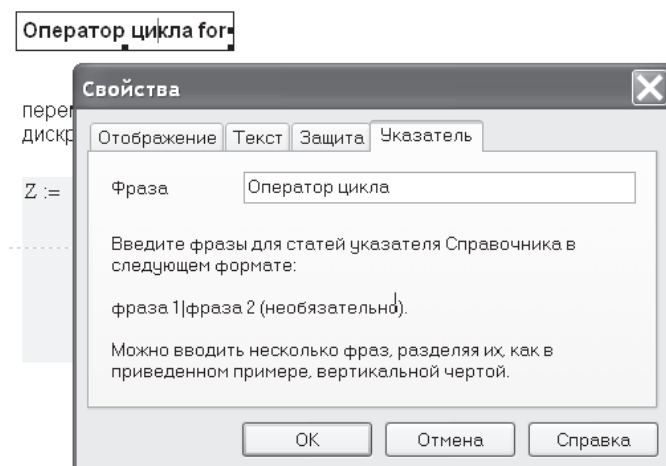


Рис. 17.10. Ввод индекса в Mathcad-файл

него правой кнопкой мыши, выберите в контекстном меню Properties (Свойства). В окне свойств (рис. 17.10) появится новый раздел Указатель (Index). Впишите фразу, при выборе которой в электронной книге данный файл будет открываться на этом объекте. Можно вписать несколько различных фраз. Они должны разделяться вертикальной чертой (без пробела).

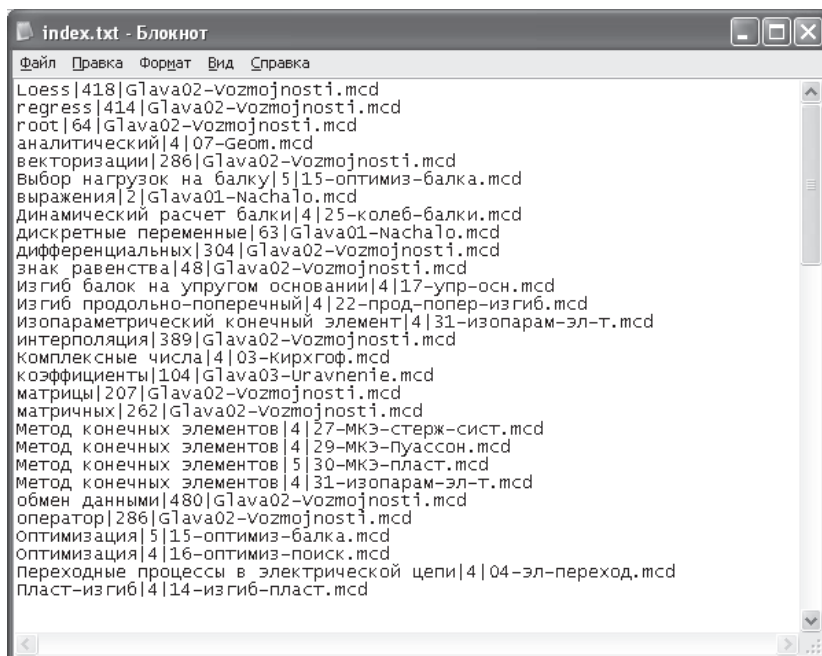


Рис. 17.11. Ввод индексов в электронную книгу

После того как будут созданы все индексные входы, можно приступить к автоматической генерации индексов. Запустите Mathcad в режиме НВК. В диалоговом окне (см. рис. 17.7), появившемся на экране, выберите Generate Index и нажмите ОК. В результате работы Mathcad в той же папке, где находится и ваш НВК-файл, появится файл Index.txt (рис. 17.11). Откройте его в Word для сортировки.

Выделите весь файл. В меню Таблица выберите Преобразовать в таблицу. В появившемся диалоговом окне задайте число столбцов — 3, разделитель — другой и укажите в качестве него вертикальную черту.

Теперь надо отсортировать индексы. В меню Таблица выберите Сортировка. В появившемся окне укажите Сортировать — столбец 1, тип — текст, затем — столбец 3, тип — текст, затем — столбец 2, тип — число. Нажмите ОК.

Далее таблицу нужно вновь преобразовать в текст, выбрав разделитель — вертикальную черту. Сохранить файл как Только текст.

Заключительная часть — обработка индексов. В главной папке Mathcad найдите файл Newdict.exe и скопируйте его в ту же папку, где находятся НВК-файл и файл Index.txt. В MS-DOS в командной строке напечатайте последовательно

имена трех файлов, например, Newdict.exe Index.txt Primer.hbk. Нажмите Enter. В результате действия программы в том же каталоге появятся два файла, out.dct и out.rfs.

Если в файле Index.txt есть ошибки, то Newdict.exe, останавливает выполнение программы и указывает номер строки, где находится ошибка. Выполняйте программу Newdict.exe до тех пор, пока все ошибки не будут устранены.

После этого переименуйте файлы out.dct и out.rfs. Вместо out напишите имя вашей электронной книги, например, Primer.dct и Primer.rfs. Переместите DCT- и RFS-файлы в тот же каталог, где находится и ваш НВК-файл.

ПРИМЕЧАНИЕ

При любых изменениях в одном из файлов электронной книги или в НВК-файле необходимо повторить процедуру создания индексов.

Ошибки часто возникают при наличии пустых линий или пробелов в индексе или в НВК-файле, а также из-за несоответствия имен файлов в индексном файле и в НВК-файле или отсутствия нужных имен.

Для устранения индексных ошибок приходится многократно возвращаться к редактированию индексных входов в файлах электронной книги.

Процедура создания системы поисковых индексов настолько сложная и длительная, что встает вопрос: а зачем все это надо?

Навигация по электронной книге и система ссылок прекрасно работают и без поисковых индексов. Единственно, что отсутствует, — это поиск фрагмента по заданному слову (или фразе). Для успешного поиска пользователь электронной книги должен ввести ключевое слово (или фразу) точно так, как вы ее вписали в индекс. Если он ошибся хоть в одной букве, поиск не удастся.

На мой взгляд, овчинка выделки не стоит. Электронная книга прекрасно работает и без системы поисковых индексов.

Глава 18

Работа с электронной книгой

Электронная книга «Расчеты в Mathcad» — это не просто справочник и интерактивные иллюстрации к основной книге. Это мощный инструмент создания собственных программ пользователя.

Для создания собственной программы надо вначале составить формульный алгоритм решения поставленной задачи, то есть последовательный список формул и этапов расчета. Затем, пользуясь системой вложенных оглавлений, отыскать в электронной книге нужные фрагменты Mathcad-файлов, решающих подобные задачи, и вставить их в свой рабочий документ. Нужно подправить вставленные фрагменты в соответствии с условиями своей задачи, связать их между собой по смыслу, ввести свои исходные данные и оценить результаты расчета.

Далее приводится ряд примеров подготовки программ пользователя с применением электронной книги.

18.1. Обработка экспериментальных данных

В результате какого-то эксперимента получены два массива данных: массив значений аргумента X и массив значений функции Y . Для последующих расчетов требуется получить функцию $y(x)$. Физический смысл этих массивов может быть любым.

Пусть измерена зависимость перемещения x автомобиля от времени t . Надо найти скорость перемещения, то есть производную dx/dt , и работу, затраченную на движение автомобиля. Введите в Mathcad-документ свои исходные данные (массивы экспериментальных замеров).

Массивы опытных данных надо превратить в функцию перемещения от времени. Для этого можно использовать интерполяцию или регрессию. Начнем с интерполяции. В электронной книге откройте раздел Глава 7. Обработка экспериментальных данных ▶ Темы ▶ кубическая интерполяция. Вам понадобится всего одна строчка оттуда, но я рекомендую скопировать в свой документ большой фрагмент раздела Интерполяция вместе с исходными данными для того, чтобы, подумав и проверив в работе, выбрать нужный вариант (в данном случае вариант интерполяции).

Интерполяция отрезками прямых линий в данном случае не годится, так как уравнение прямой линии не имеет второй производной и не позволяет найти ускорение тела, хотя позволяет искать интеграл функции. Кубическая сплайн-интерполяция позволяет выполнить все необходимые расчеты. Копируем в область своих расчетов строчку с функцией `lspline`, в которой шивка сплайнов (отрезков) в граничных точках происходит по прямой линии (рис. 18.1).

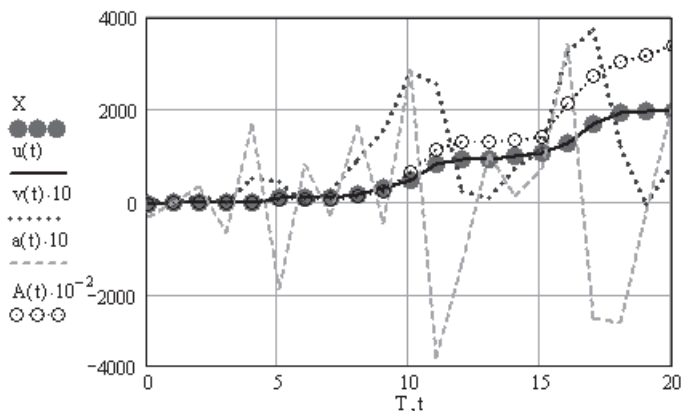


Рис. 18.2. Результаты расчета

Можно использовать сочетание функций сглаживания и интерполяции. Для этого выберите в электронной книге раздел Глава 7. Обработка экспериментальных данных ▶ Темы ▶ 7.3. Функции сглаживания. Однако в нашем случае отклонение от расчетных точек при сглаживании получается неправдоподобно большим. Тем не менее в ряде других случаев сглаживание оказывается полезным.

Если экспериментальные данные не набор случайных точек, а предположительно плавная кривая, то лучше использовать функции регрессии для преобразования массива в функцию. На рис. 18.3 набор данных преобразуется в полином 5-й степени с помощью функции `regress` (скопируйте из электронной книги

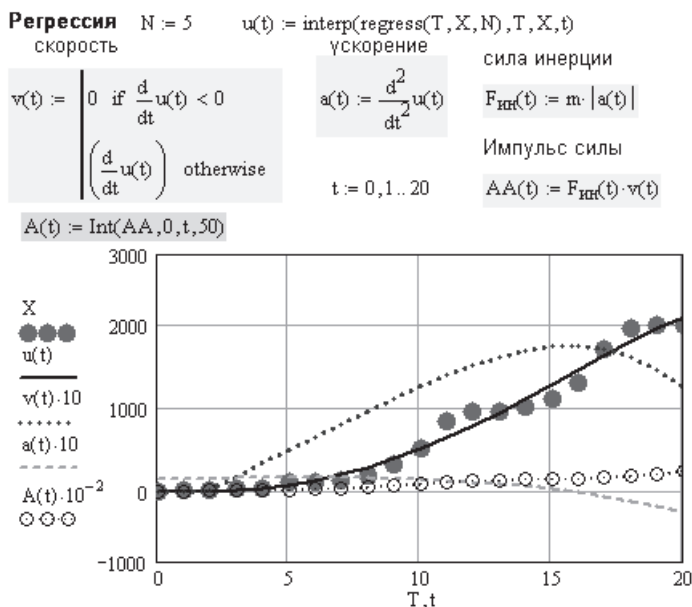


Рис. 18.3. Результаты расчета с использованием функции регрессии

нужный фрагмент из раздела Глава 7. Обработка экспериментальных данных ▶ Темы ▶ 7.2.1. Одномерная регрессия). Расчетные формулы надо скопировать ниже функции регрессии, иначе не произойдет пересчета формул после регрессии.

Функцию регрессии `loess` использовать в расчетах не удастся, так как нельзя вычислить интеграл от найденной функции.

Результаты расчета озадачивают. Неправдоподобно гладко выглядят графики скорости и ускорения, а ведь автомобиль движется рывками и с остановками. Соответственно, и затраченная энергия двигателя на разгоны и торможения $A(t)$ оказалась в 15 раз меньше, чем при использовании интерполяции. Все логично, но будьте бдительны, Mathcad ведь не вникает в суть проблемы — он всего лишь математический инструмент.

18.2. Обработка цифрового сигнала

Предположим, вы имеете файл с записью цифрового сигнала. Надо очистить сигнал от случайных помех. Для этого воспользуемся разложением в ряд Фурье.

В новый Mathcad-документ введите свой массив цифрового сигнала, прочитав его из файла с помощью функции `Readprn`. В электронной книге выберите Глава 7. Обработка экспериментальных данных ▶ Темы ▶ 7.4.1. Преобразование Фурье. Скопируйте открывшийся пример в свой документ.

Посмотрите график своего сигнала (рис. 18.4), введя в поле ввода графика имя, присвоенное массиву данных. Введите это имя в функции прямого преобразования Фурье `sfft`. Посмотрите на графике (рис. 18.5) значения коэффициентов ряда Фурье. Введите свои значения диапазона выбираемых коэффициентов k_1 и k_2 . После обратного преобразования Фурье с учетом выбранного ряда коэффициентов вы увидите (рис. 18.6) график очищенного сигнала. Варьируя значения k_1 и k_2 , подберите нужный вам вид очищенного сигнала. Графика исходного сигнала вы не увидите, так как у вас его просто нет.

Обработка цифрового сигнала

Здесь скрыто формирование массива опытных данных



```
S := READPRN("18-2.pm")
```

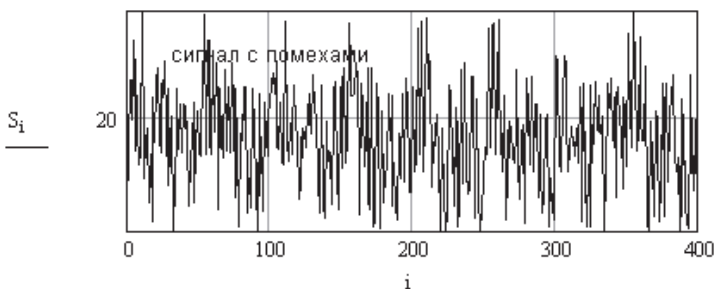
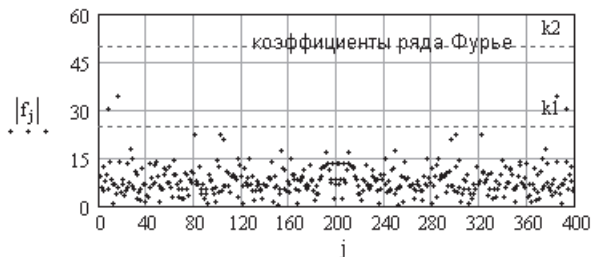


Рис. 18.4. График сигнала, требующего очистки от помех

прямое преобразование $f := \text{cfft}(S)$

построить ряд Фурье, значения коэффициентов которого лежат между $k1$ и $k2$

$k1 := 25$ $k2 := 50$ поменяйте $k1$ и $k2$ $j := 0..400$



выбор коэффициентов $g_j := \text{if}(k1 \leq |f_j| \leq k2, f_j, 0)$ **обратное преобразование** $h := \text{icfft}(g)$

Рис. 18.5. Процедура очистки сигнала от помех

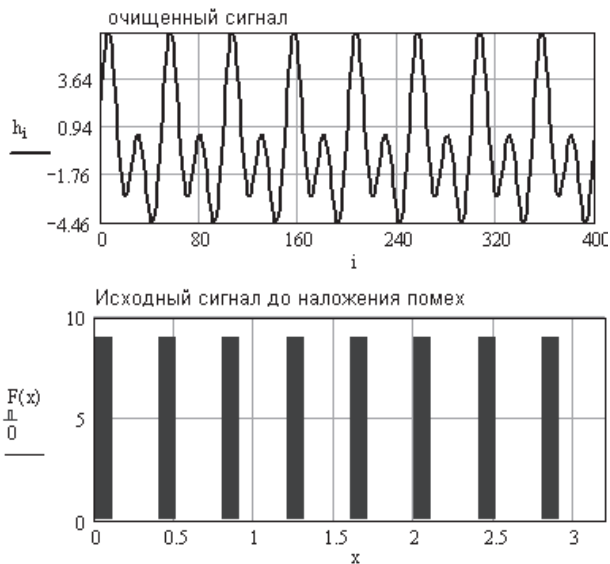


Рис. 18.6. Сравнение очищенного сигнала с исходным

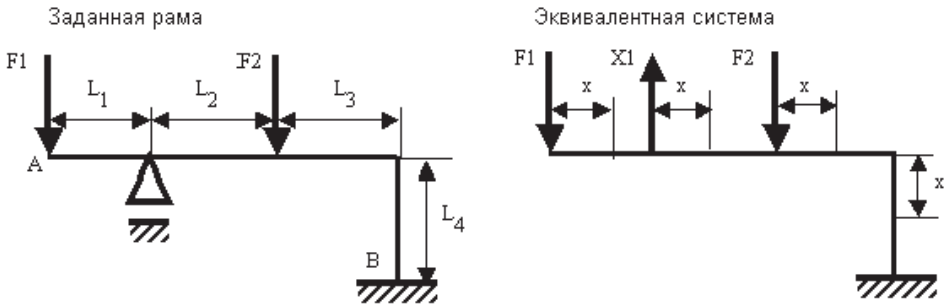
18.3. Оптимизация нагрузки на балку

Балка на двух опорах нагружена тремя силами, две из которых приложены к концам балки. Определить положение третьей силы (силы F_2), при котором максимальный прогиб балки будет наименьшим. Исходные данные для расчета показаны на рис. 18.7.

Похожая задача решается в примере 5 (Движение двух машин на мосту) в электронной книге. Посмотрите этот пример. Он весьма сложный, связан с программированием и решением системы уравнений с переменными параметрами. Пойдем простейшим путем, чтобы не демонстрировать «высший пилотаж» в Mathcad, а показать простоту составления программ пользователя.

Расчет на прочность 1 раз статически неопределимой плоской рамы

ORIGIN := 1



Каноническое уравнение метода сил $X_1 \cdot \delta_{1,1} + \Delta_{1,F} = 0$

Уравнение одно, в программе везде заменить вектор X_1 на число X (измененные блоки голубого цвета)

Рис. 18.7. Схема балки и исходные данные для расчета

Рассчитаем максимальный прогиб балки при заданном положении силы F_2 . Расчет выполним с учетом размерностей, поэтому из любого примера, приведенного в электронной книге, скопируем в свой документ производные размерности. Далее введем исходные данные.

Расчет начнем с определения реакций опор и построения эпюры изгибающих моментов. Откроем в электронной книге пример 12. Определение внутренних усилий при изгибе балки. Скопируем оттуда определение поперечной силы Q и изгибающего момента M , а также определение опорных реакций. Копируем оттуда же определение прогиба балки методом начальных параметров. Поскольку в нашем случае отсутствует распределенная нагрузка, то слагаемые, учитывающие ее, удаляем. После отладки программы (а она состоит в возможной корректировке некоторых обозначений) для достижения краткости и элегантности программы отдельные фрагменты можно спрятать в закрытых зонах (рис. 18.8).

Получив функцию прогиба f от положения выбранного сечения x , найдем максимальный прогиб балки (по абсолютной величине). Для его определения воспользуемся функцией F_{\max} из электронной книги Глава 9. Программирование ▶ Темы ▶ 9.8. Нахождение экстремума функции в программе. Поскольку надо найти максимальный по абсолютной величине прогиб, то вначале находим модуль f_1 от функции прогиба f , затем в обращение к функции записываем имя f_1 (см. рис. 18.8).

Задаем величину L_2 и выводим результаты расчета на график (рис. 18.9).

Далее требуется определить, при каком значении L_2 максимальный прогиб будет минимальным. Как отмечено в начале этого примера, для Mathcad это очень сложная проблема.

Здесь скрыто определение опорных реакций и внутренних усилий Q и M



Здесь скрыто определение прогиба балки методом начальных параметров



здесь скрыта программа Fmax для определения maxM и max прогиба $M(x) := |M_z(R_A, R_B, L_2, x)|$



$M_{\max} := F_{\max}(M, L_H, L_K, N)$ $f(x) := |f(x)|$ $\Delta_{\max} := F_{\max}(f, L_H, L_K, N)$

Рис. 18.8. Расчетная часть документа со скрытыми зонами

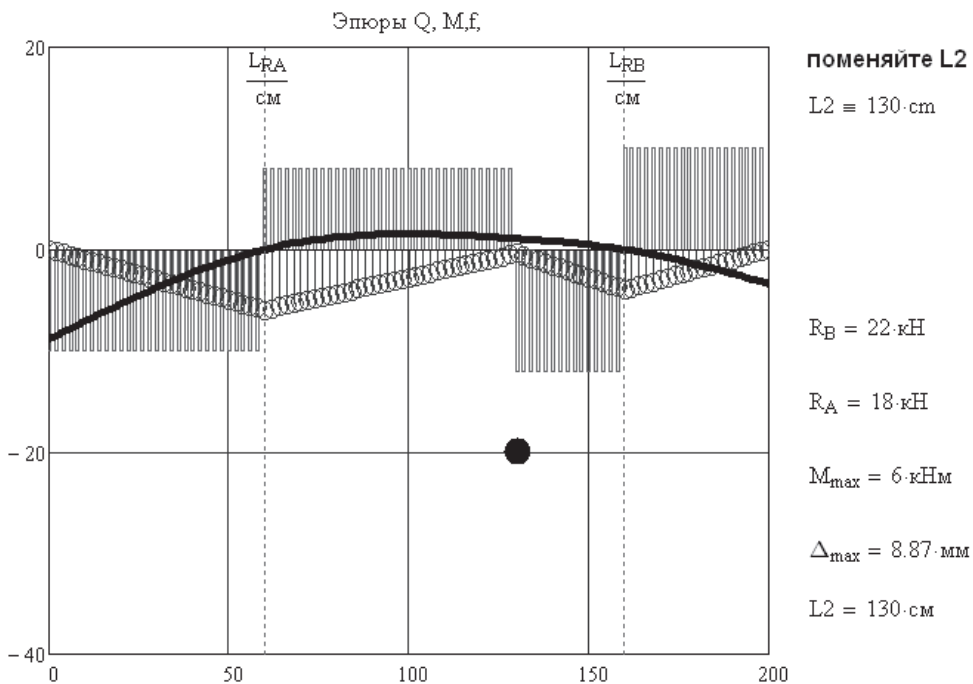


Рис. 18.9. Оптимальные результаты расчета балки

Выполним почти цирковой трюк. Анимлируем этот график при движении силы F_2 вдоль балки. В область анимации включаем заданное значение L_2 и расчетное значение максимального прогиба Δ_{\max} . В таком случае от кадра к кадру будут изменяться и график, и выводимые числовые значения.

Уменьшим скорость проигрывания клипа, выбрав в меню Record Animation скорость 1 кадр в секунду. Наблюдая за клипом, запоминаем значение L_2 , при котором значение Δ_{\max} минимально. Выключите значение Frame (см. рис. 18.7), введите нужное значение L_2 и внимательно рассмотрите результаты оптимального расчета.

К такому примитивному трюку приходится прибегать вследствие несовершенства системы Mathcad, которая крайне неуклюже работает с вычислительными блоками Given–find (minerr, Odesolve, maximize, minimize), не допуская свободной вариации параметров функций и использования вычислительного блока в программных модулях.

Часть IV

Примеры инженерных расчетов в Mathcad

Содержит примеры решения инженерных задач в Mathcad, которые иллюстрируют различные приемы работы в Mathcad.

Скачайте с сайта издательства www.piter.com или с сайта автора www.emakarov.nsknet.ru архив электронной книги Эл-книга.zip. В этом архиве находится электронная книга «Расчеты в Mathcad» с программами для 56 примеров сложных инженерных расчетов. Подробное описание этих примеров и пояснения к ним находятся в этом же архиве. Они записаны в формате *pdf*, для просмотра файлов необходимо установить на компьютере Acrobat Reader любой версии, а еще лучше FoxitReader, который работает значительно быстрее, чем AcrobatReader.

Распакуйте архив и откройте файл Примеры-все.pdf (рис. П.1). Выберите в левом верхнем углу окна пункт **Закладки** — откроется список всех 56 примеров. Щелкните мышью на любом примере и читайте пояснения к нему.

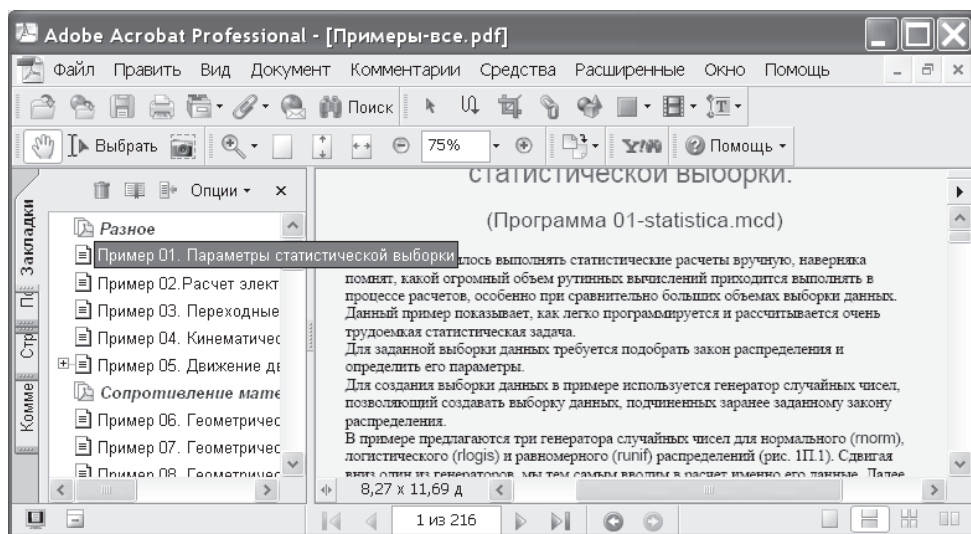


Рис. П.1. Выбор примера в окне Acrobat Reader

Mathcad-программы всех примеров находятся в электронной книге «Расчеты в Mathcad». Порядок установки электронной книги описан в файле *ReadMe*, находящемся в папке Эл-книга. После ее установки откройте Mathcad и в главном меню Mathcad выберите **Справка** ► **Электронные книги** ► **Расчеты в Mathcad** (**Help** ► **E-books** ► **Расчеты в Mathcad**). В открывшейся электронной книге выберите **Примеры инженерных расчетов в Mathcad** и далее название интересующего вас примера (рис. П.2).

Работать с примером можно и внутри электронной книги, однако, чтобы сохранить произведенные изменения, надо выделить все (клавиши **Ctrl+A**), скопировать в буфер обмена (клавиши **Ctrl+C**), вставить в новый Mathcad-документ (клавиши **Ctrl+V**), сохранить файл (клавиши **Ctrl+S**). В диалоговом окне сохранения файла ввести имя файла и выбрать версию Mathcad, в которой вы хотите сохранить файл.

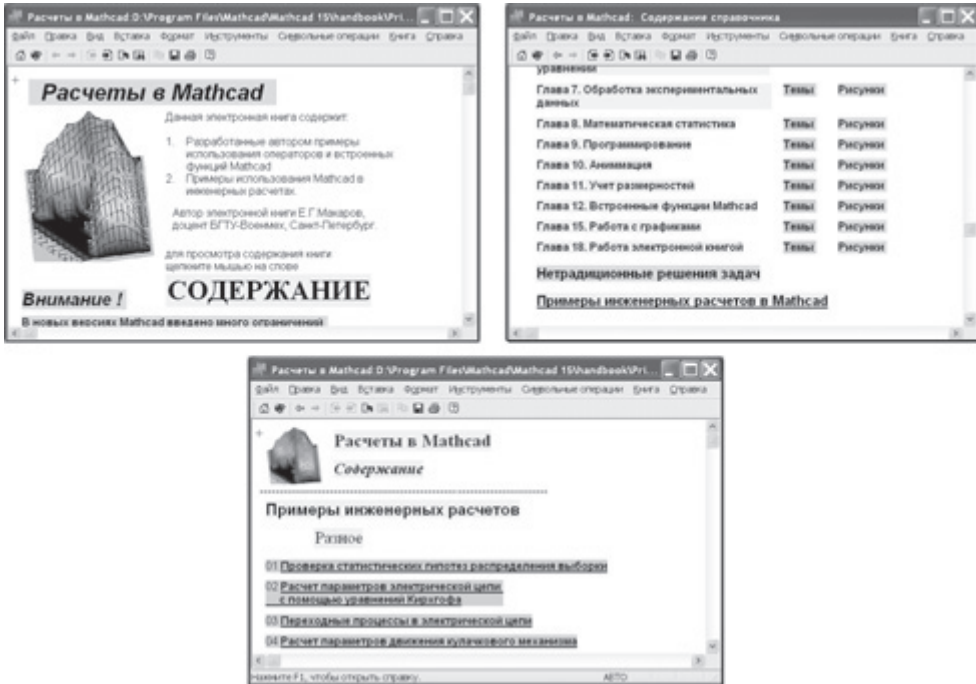


Рис. П.2. Выбор нужного примера в электронной книге

Далее приведем основные сведения (схемы и условия задачи) для всех 56 примеров, находящихся в электронной книге.

Пример 1. Проверка статистических гипотез

Для заданной выборки данных требуется подобрать закон распределения и определить его параметры.

Выдвигаются три гипотезы закона распределения и определяются функции вероятности для них.

Предполагается, что выборка данных подчиняется:

- нормальному распределению (функция вероятности $pnorm$);
- логистическому распределению ($plogis$);
- равномерному распределению ($punif$).

Оценка трех гипотез производится с помощью коэффициентов корреляции, взвешенных квадратов отклонений, методом Пирсона и Романовского, методом максимального правдоподобия.

Пример 2. Расчет электрической цепи переменного тока

В цепи (рис. П.3) действует источник синусоидальной ЭДС. Требуется определить токи и напряжения, действующие на всех участках цепи; проверить баланс мощностей и найти показания приборов.

Расчет ведется двумя методами: по уравнениям Кирхгофа и методом эквивалентных преобразований.

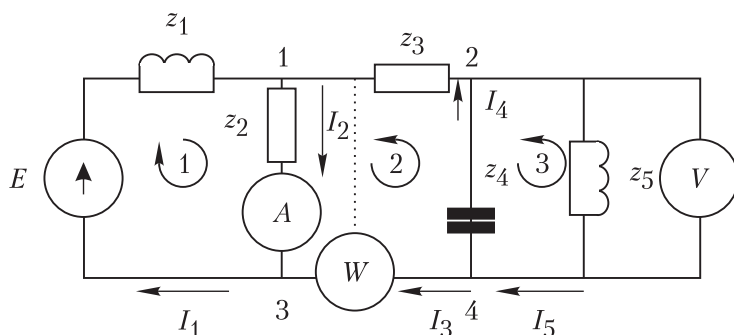


Рис. П.3. Схема электрической цепи переменного тока

Пример 3. Переходные процессы при замыкании-размыкании электрической цепи

В примере рассматривается цепь, содержащая источник тока E , катушку индуктивности L , два сопротивления, R_1 и R_2 , конденсатор C и рубильник (рис. П.4). Определить распределение токов и напряжений в ветвях электрической цепи во времени.

В программе рассматриваются: решение дифференциального уравнения второго порядка и решение системы дифференциальных уравнений первого порядка.

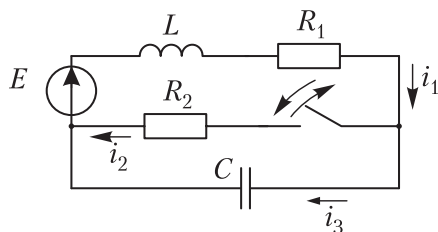


Рис. П.4. Схема электрической цепи

Пример 4. Кинематический анализ кулачкового механизма

Заданы массивы углов поворота кулачка и соответствующих им углов поворота коромысла. Для определения динамических нагрузок на конструкцию требуется определить скорость и ускорение коромысла.

В программе расчета массивы преобразуются в функции, которые затем дифференцируются. Для сглаживания экспериментальных замеров используется разложение в ряд Фурье.

Пример 5. Движение двух машин по мосту

Две машины массой m_1 и m_2 въезжают на мост с разных сторон со скоростями V_1 и V_2 . Определить прогиб моста под нагрузкой и наибольшие напряжения в балках моста в зависимости от положения машин на мосту (рис. П.5).

В программе определяются нагрузки на мост, максимальные напряжения и прогиб балок моста в зависимости от времени. Рассмотрены три варианта расчета: использование программирования с подпрограммами без учета и с учетом размерностей, а также расчет без использования программирования. Для демонстрации результатов расчета используется анимация.

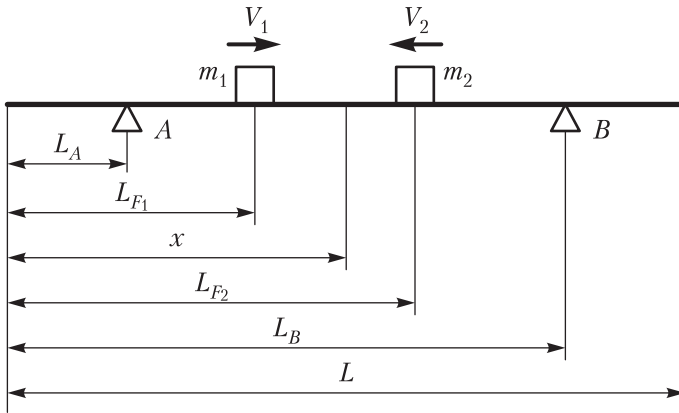


Рис. П.5. Схема движения двух машин по мосту

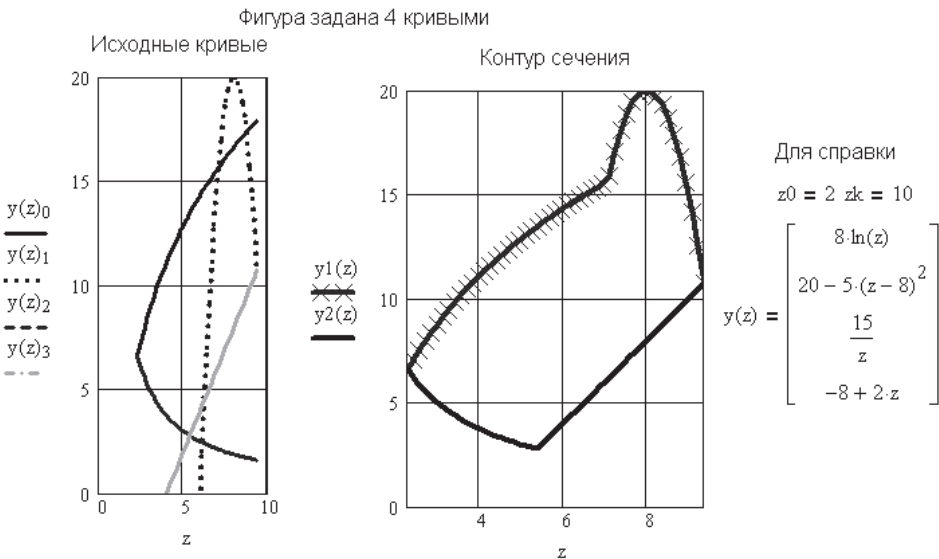


Рис. П.6. Сечение, заданное аналитически

Пример 6. Расчет моментов инерции сечения, заданного аналитически

Задано сечение, ограниченное четырьмя кривыми. Найти главные центральные моменты инерции фигуры (рис. П.6), образованной этими кривыми, и положение главных осей инерции.

Расчет ведется по формулам сопротивления материалов путем непосредственного интегрирования.

Пример 7. Расчет моментов инерции сечения, заданного набором точек на контуре

Сечение задано десятью точками (5 точек – верхний контур, 5 точек – нижний контур). Найти главные центральные моменты инерции, образованной всеми точками фигуры, и положение главных осей инерции.

Массив точек на контуре сечения преобразуется в функцию. Далее выполняется интегрирование, как и в примере 6.

Пример 8. Расчет геометрических характеристик составного сечения

Рассматривается сечение из 7 элементов (рис. П.7). Для каждого элемента вводится ширина b , высота h , площадь A , координаты центра тяжести Y и Z в виде элементов вектора для выполнения последующих однотипных операций. Вводится код элемента K и признак наличия выреза $K1$.

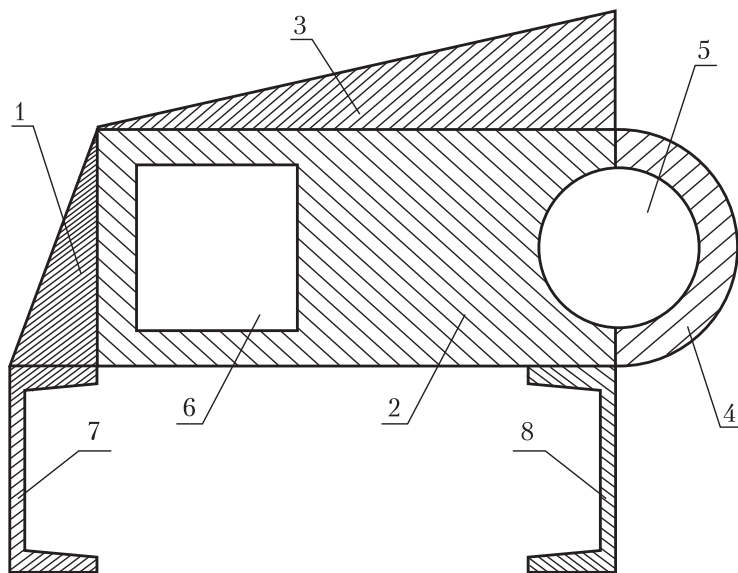


Рис. П.7. Сечение из семи элементов

Пример 9. Определение внутренних усилий при растяжении стержня

Для стержня, нагруженного тремя сосредоточенными и двумя распределенными нагрузками, построить эпюру продольных усилий. Чтобы не загромождать рис. П.8, на нем показаны только две нагрузки вместо пяти.

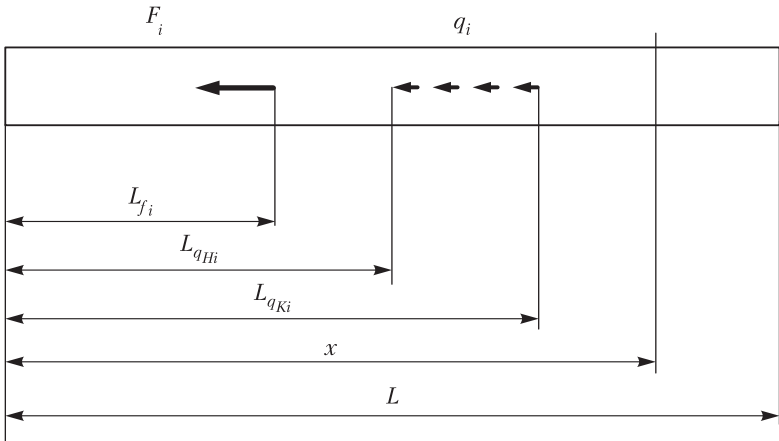


Рис. П.8. Условная схема растяжения стержня

Пример 10. Определение внутренних усилий при кручении стержня

Для стержня, закручиваемого двумя парами сил и двумя распределенными моментами, построить эпюру крутящих моментов. Чтобы не загромождать рис. П.9, на нем показаны только две нагрузки из четырех.

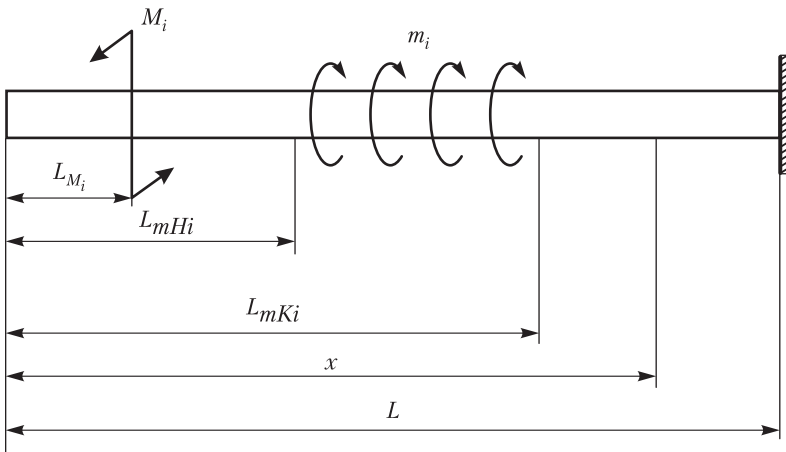


Рис. П.9. Условная схема кручения стержня

Пример 11. Определение внутренних усилий при изгибе консольной балки

Для консольной балки, нагруженной тремя силами, двумя парами сил и двумя распределенными нагрузками, построить эпюры поперечных сил и изгибающих моментов. Чтобы не загромождать рис. П.10, на нем показаны только три нагрузки из семи. Средствами Mathcad определяются максимальные значения внутренних усилий.

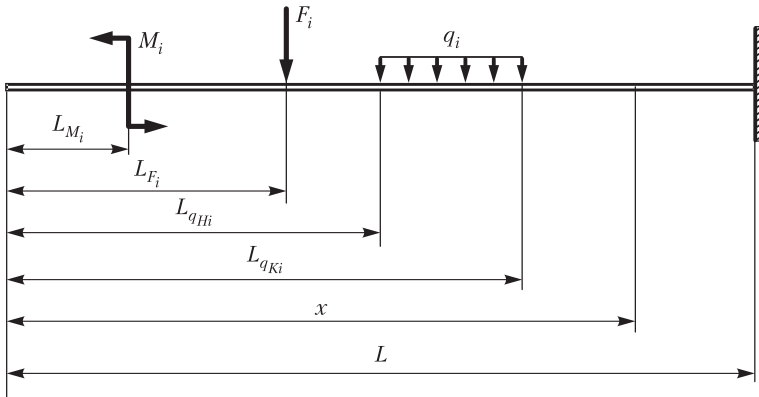


Рис. П. 10. Условная схема нагружения консольной балки

Пример 12. Определение внутренних усилий при изгибе балки на двух опорах

Для балки на двух опорах, нагруженной тремя силами, двумя парами сил и двумя распределенными нагрузками, построить эпюры поперечных сил и изгибающих моментов.

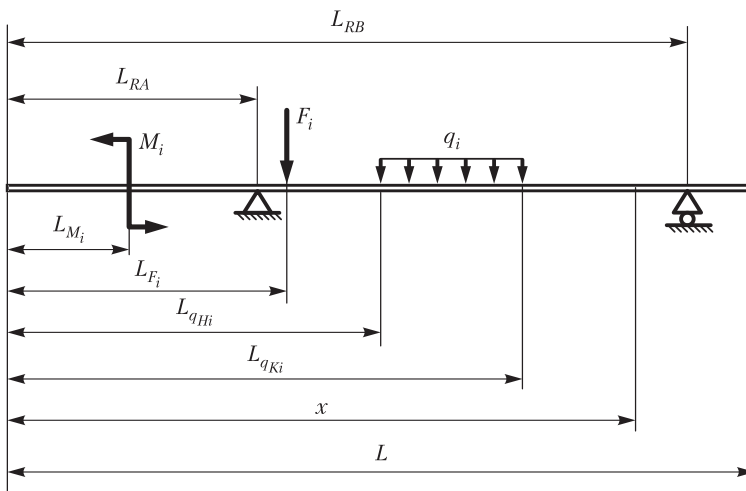


Рис. П. 11. Условная схема нагружения балки на двух опорах

щих моментов. Чтобы не загромождать рис. П.11, на нем показаны только три нагрузки из семи.

Пример 13. Растяжение статически-определимого стержня

Для стержня переменного сечения (рис. П.12), нагруженного тремя силами и двумя распределенными нагрузками, требуется найти распределение напряжений и перемещений по длине стержня.

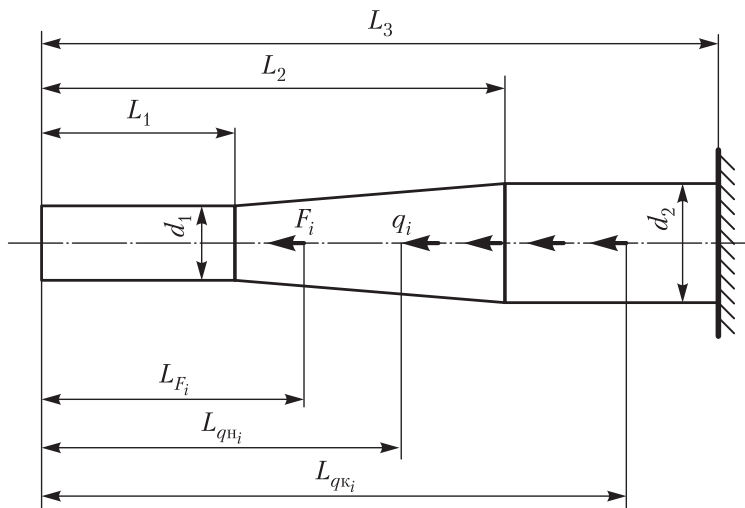


Рис. П.12. Условная схема растяжения стержня

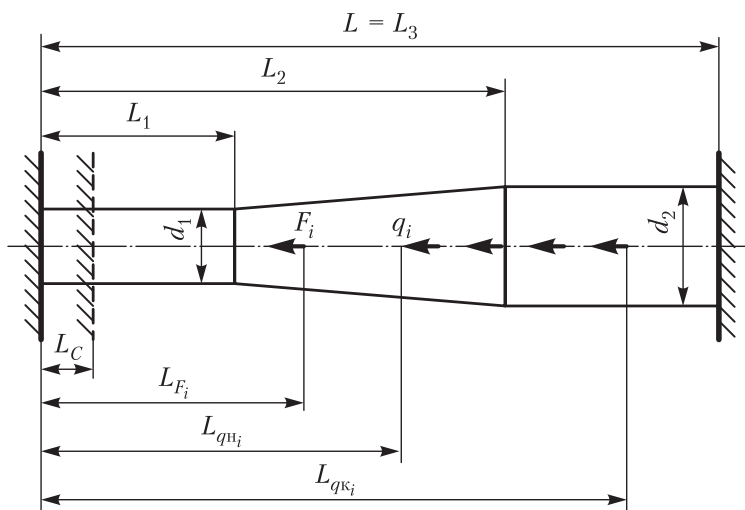


Рис. П.13. Растяжение статически-неопределимого стержня

Пример 14. Растяжение статически-неопределимого стержня

Для статически-неопределимого стержня переменного сечения (рис. П.13), нагруженного тремя силами и двумя распределенными нагрузками, требуется найти распределение напряжений и перемещений по длине стержня. Учесть изменение температуры Δt и монтажную погрешность $\Delta_{\text{монт}}$.

Пример 15. Растяжение-сжатие статически неопределимой шарнирно-стержневой системы

Для статически-неопределимой шарнирно-стержневой системы (рис. П.14), нагруженной тремя силами и двумя распределенными нагрузками, требуется найти напряжения в стержнях и удлинение стержней. Учесть изменение температуры Δt и монтажную погрешность $\Delta_{\text{монт}}$.

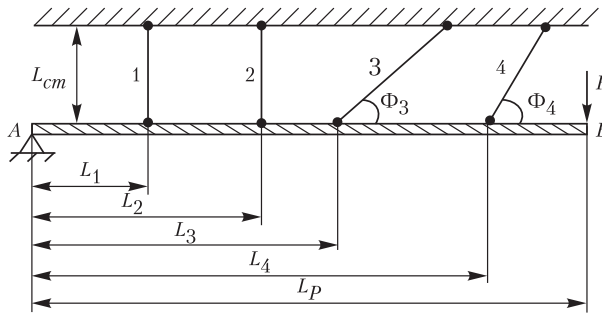


Рис. П.14. Растяжение шарнирно-стержневой системы

Пример 16. Расчет простейших соединений

Из условия прочности определить прочные размеры тяги, штифта и сварного соединения (рис. П.15).

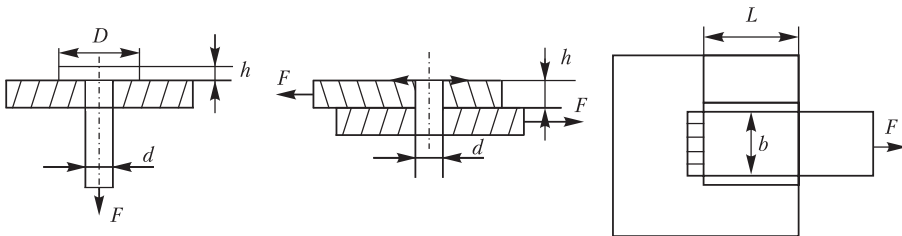


Рис. П.15. Схемы для расчета тяги, штифта и сварного соединения

Пример 17. Расчет статически определимого стержня на кручение

Для заданного ступенчатого стержня (рис. П.16), нагруженного пятью различными нагрузками, определить распределение максимальных касательных напряжений и угла закручивания по длине стержня.

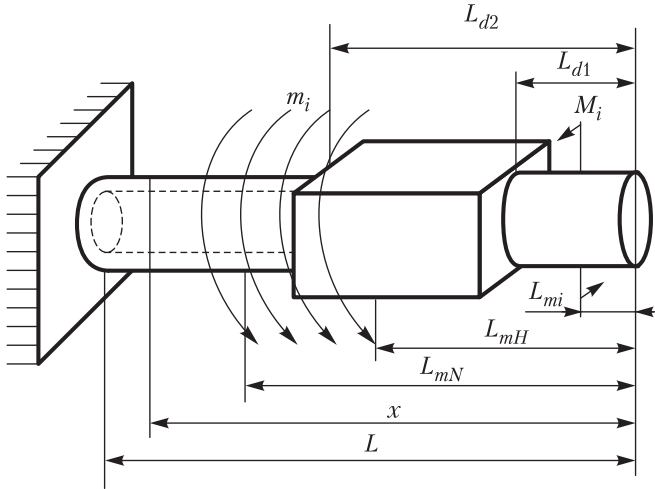


Рис. П. 16. Условная схема нагружения ступенчатого стержня

Пример 18. Расчет статически неопределимого стержня на кручение

Из условий прочности и жесткости на кручение определить диаметр статически-неопределимой трубы (рис. П.17).

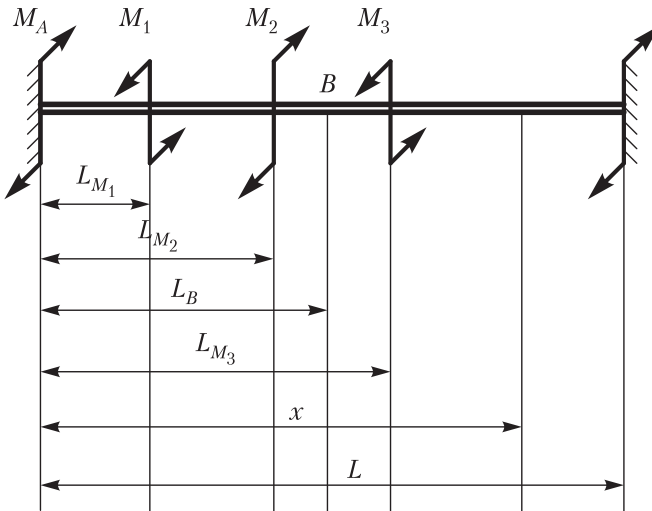


Рис. П. 17. Кручение статически-неопределимого стержня

Пример 19. Расчет на изгиб консольной балки

Для заданной балки коробчатого поперечного сечения (рис. П.18) определить максимальные напряжения и построить упругую линию балки.

В процессе расчета строятся эпюры нормальных и касательных напряжений в опасном сечении балки. Перемещения балки при изгибе определяются тремя способами: путем непосредственного интегрирования уравнения упругой линии, методом начальных параметров и с помощью интеграла Мора.

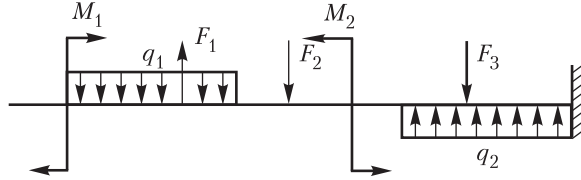


Рис. П.18. Схема нагружения консольной балки

Пример 20. Расчет на изгиб балки на двух опорах

Для заданной балки на двух опорах коробчатого поперечного сечения (рис. П.19) определить максимальные напряжения и построить упругую линию балки.

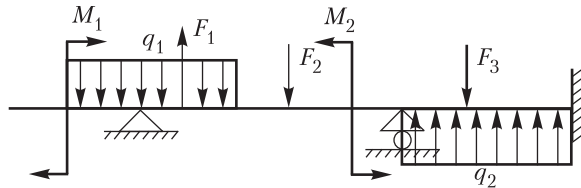


Рис. П.19. Расчетная схема балки на двух опорах

Пример 21. Расчет на изгиб балок на упругом основании

В примере приведен расчет трех балок на упругом основании (рис. П.20): балки с одной силой посередине и сплошным упругим основанием, балки с тремя силами на сплошном упругом основании, балки с тремя силами и двумя участками упругого основания.

В программе функцией *Odesolve* решается дифференциальное уравнение 4-го порядка. Сосредоточенные силы заменяются нагрузками, распределенными на малом участке балки.

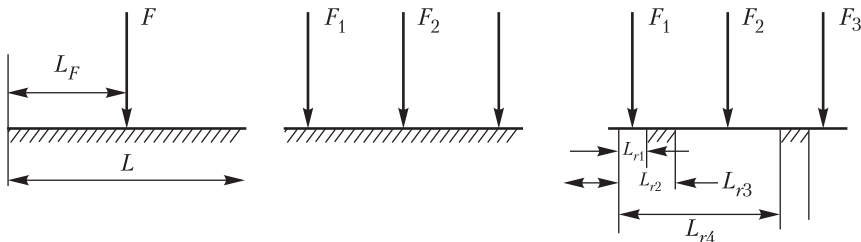


Рис. П.20. Схемы балок на упругом основании

Пример 22. Учет пластических деформаций при растяжении-сжатии

Для заданной статически-неопределимой шарнирно-стержневой системы (рис. П.21) найти зависимость предельной нагрузки F от перемещения точки приложения силы.

В расчетах используются три вида аппроксимации диаграммы деформирования: диаграмма Прандтля, диаграммы с линейным и степенным упрочнением.

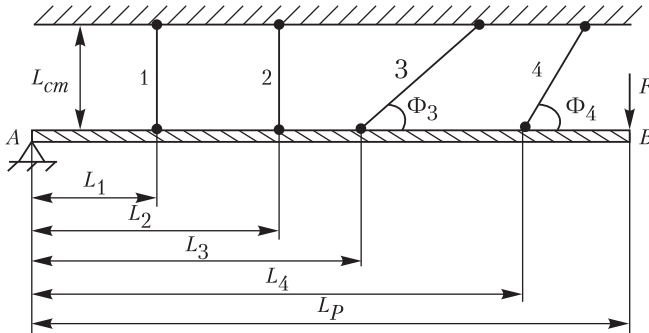


Рис. П.21. Схема шарнирно-стержневой системы

Пример 23. Учет упругопластических деформаций при изгибе

В примере определяется распределение деформаций и напряжений в поперечном сечении балки при изгибе, а также остаточных напряжений при упругопластических деформациях. В расчетах используются три вида аппроксимации диаграммы деформирования: диаграмма Прандтля, диаграммы с линейным и степенным упрочнением.

Пример 24. Учет пластических деформаций при кручении

В примере определяется распределение деформаций и напряжений в поперечном сечении стержня при кручении, а также остаточных напряжений при упругопластических деформациях. Для проведения расчетов диаграмма деформирования материала балки при растяжении $\sigma - \varepsilon$ перестраивается в диаграмму сдвига $\tau - \gamma$. В расчетах используются три вида аппроксимации диаграммы деформирования: диаграмма Прандтля, диаграммы с линейным и степенным упрочнением.

Пример 25. Расчет прямого стержня на сложное сопротивление

Для заданного прямого стержня переменного поперечного сечения (рис. П.22), нагруженного 23 нагрузками, определяется максимальное эквивалентное напряжение и строятся графики перемещений по шести направлениям (трем линейным и трем угловым).

Это большая сложная программа, включающая в себя фрагменты примеров 9–12.

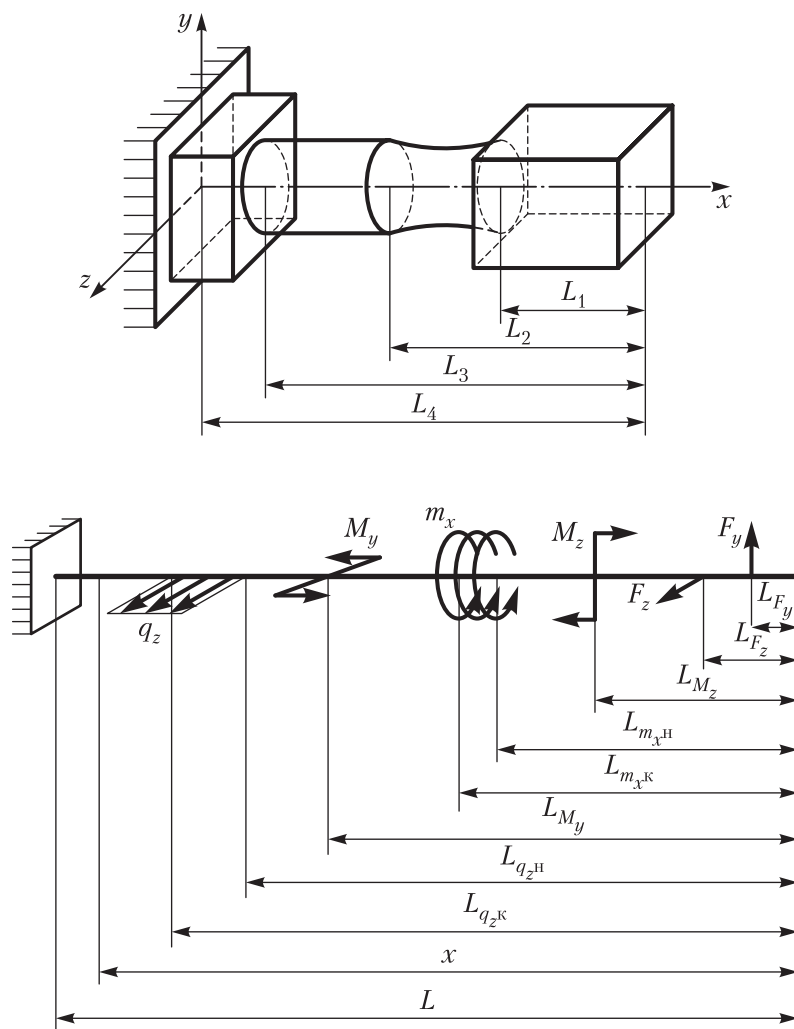


Рис. П.22. Стержень переменного сечения (вверху) и условная схема его нагружения (внизу)

Пример 26. Расчет пространственного стержня на сложное сопротивление

Для заданного пространственного стержня (рис. П.23) подобрать прочные размеры поперечных сечений и горизонтальное перемещение конца стержня.

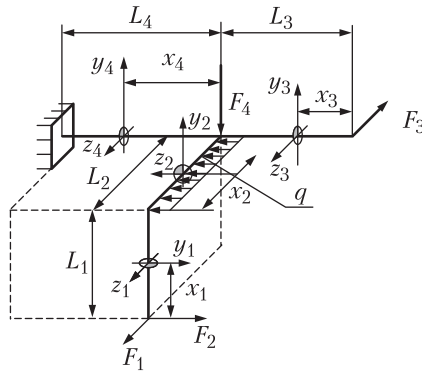


Рис. П.23. Схема нагружения пространственного стержня

Пример 27. Определение перемещений систем из нескольких элементов

Для заданной системы, состоящей из двух балок, связанных пружиной и стержнем (рис. П.24), определить вертикальное перемещение под силой и угол поворота сечения на опоре *D*.

Перемещения определяются с помощью интеграла Мора.

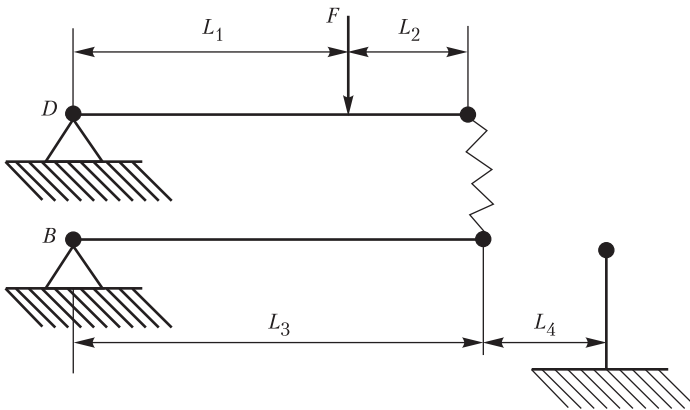
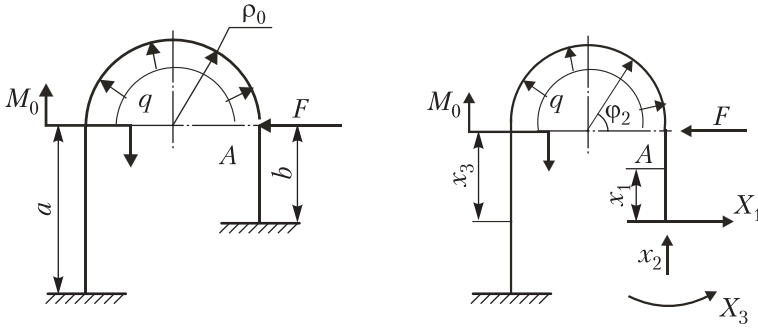


Рис. П.24. Схема нагружения сложной системы

Пример 28. Раскрытие статической неопределимости плоской рамы

В примере методом сил раскрывается статическая неопределимость плоской рамы (рис. П.25), определяется одно из перемещений и выполняется проверка правильности решения.



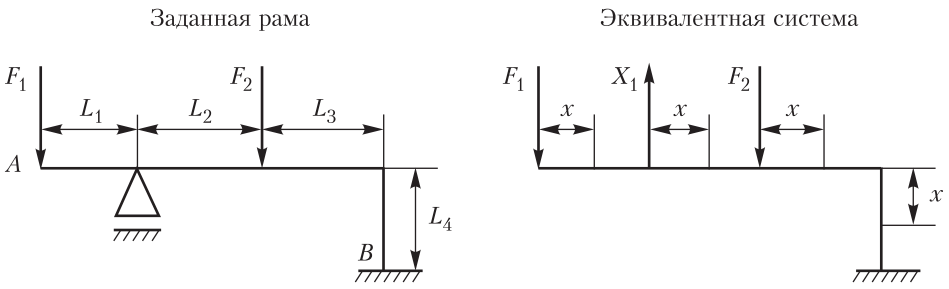
Заданная система

Эквивалентная система

Рис. П.25. К расчету плоской рамы

Пример 29. Расчет один раз статически неопределимой системы

Расчет такой системы (рис. П.26) почти аналогичен примеру 28, но потребовалась новая система индексов. В конце примера произведен расчет рамы на прочность.



Заданная рама

Эквивалентная система

Рис. П.26. Расчетные схемы рамы

Пример 30. Расчет пространственной статически неопределимой рамы

Для заданной, пять раз статически неопределимой пространственной рамы (рис. П.27) требуется раскрыть статическую неопределимость и сделать проверку правильности решения, выбрать опасное сечение и определить диаметр стержней рамы из условия прочности.

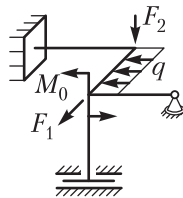


Рис. П.27. Схема пространственной рамы

Пример 31. Расчет многопролетной балки с одной заделкой на изгиб

Расчет ведется методом сил. Количество опор, как и количество нагрузок, действующих на балку, может быть любым (рис. П.28).

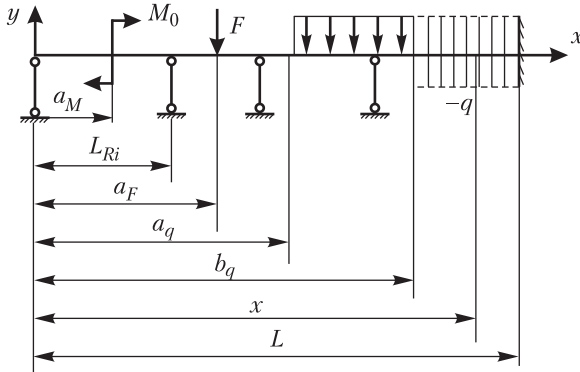


Рис. П.28. Схема многопролетной балки с заделкой

Пример 32. Расчет многопролетной балки на изгиб

Программа расчета такой балки (рис. П.29) близка к программе, приведенной в примере 31. Но в предыдущем примере основная система (балка с заделкой) не требует определения опорных реакций, в этом примере основная система (балка на двух опорах) требует многократного определения опорных реакций.

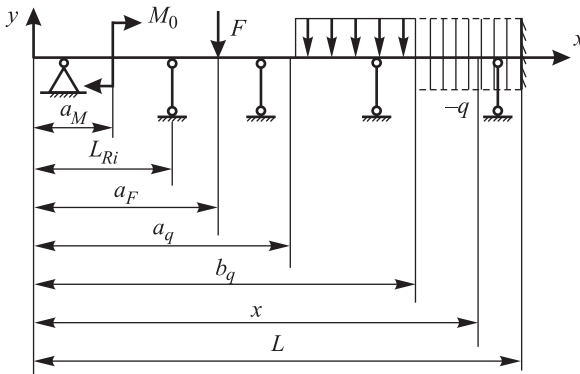


Рис. П.29. Схема многопролетной балки

Пример 33. Расчет на устойчивость сжатого стержня

Для стержня круглого сечения, нагруженного сжимающей осевой силой (рис. П.30), определить диаметр стержня из расчета на устойчивость.

Расчет ведется методом последовательных приближений с использованием программирования.

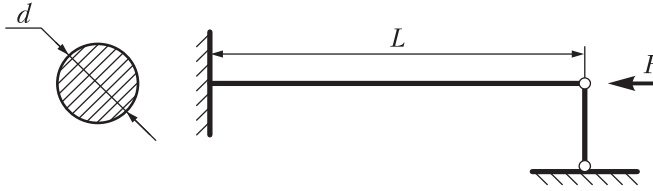


Рис. П.30. Схема сжатого стержня

Пример 34. Определение критических нагрузок на сжатый стержень энергетическим методом

Определить нагрузку, соответствующую потере устойчивости для ступенчатого стального стержня с двумя заделками и промежуточной шарнирной опорой, нагруженного двумя силами (рис. П.31).

Расчет ведется по формуле Рэлея, в которой уравнение упругой линии аппроксимируется полиномом степени m .

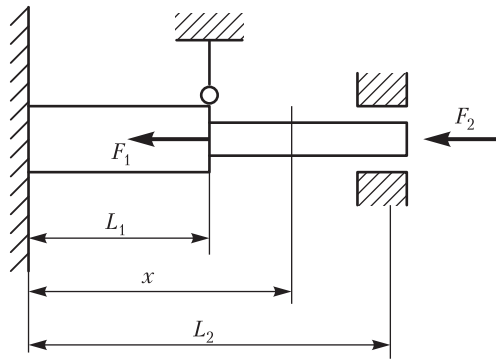


Рис. П.31. К расчету ступенчатого сжатого стержня на устойчивость

Пример 35. Продольно-поперечный изгиб балок

Для двух балок, нагруженных продольной и поперечной нагрузками (рис. П.32), определить прогиб балок и максимальные напряжения.

Расчет ведется двумя способами с решением дифференциальных уравнений второго и четвертого порядка.

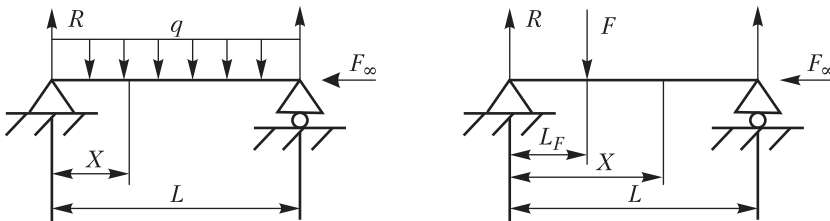


Рис. П.32. Расчетные схемы нагружения балок

Пример 36. Расчет консольной балки на удар и колебания

Для консольной балки массой m_0 (рис. П.33) определить:

- максимальные динамические напряжения при падении на балку груза массой m в точку с координатой x_m ,
- собственную частоту колебаний с учетом массы балки m_0 и сосредоточенной массы m в точке x_m ,
- перемещение $u(t)$ точки x_m балки под действием возмущающей силы $F(t)$, приложенной к этой точке.

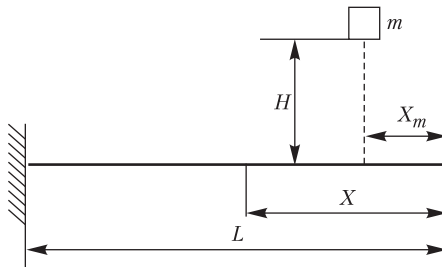


Рис. П.33. Консольная балка под действием ударной нагрузки

Пример 37. Расчет балки на двух опорах на удар и колебания

Для балки на двух опорах массой m_0 (рис. П.34) определить те же величины, что и в примере 36.

Расчет почти такой же, как в примере 36, но требующий определения опорных реакций для всех видов нагружения.

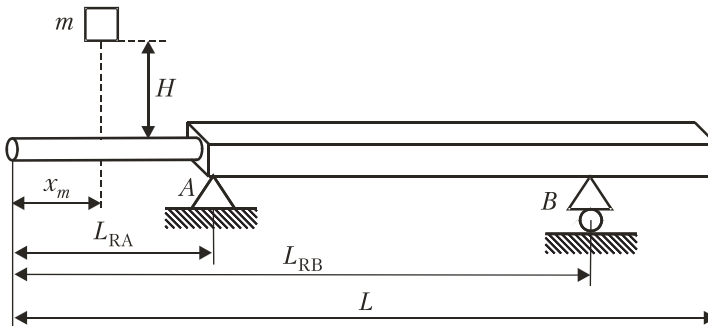


Рис. П.34. Балка на двух опорах под действием ударной нагрузки

Пример 38. Расчет сложной системы на удар и устойчивость

Для заданной системы, состоящей из двух балок, связанных пружиной и стержнем (рис. П.35), определить вертикальное перемещение под силой и угол поворота сечения на опоре D .

Фактически этот пример является продолжением примера 27, но в данной задаче определяются не статические, а динамические напряжения и перемещения.

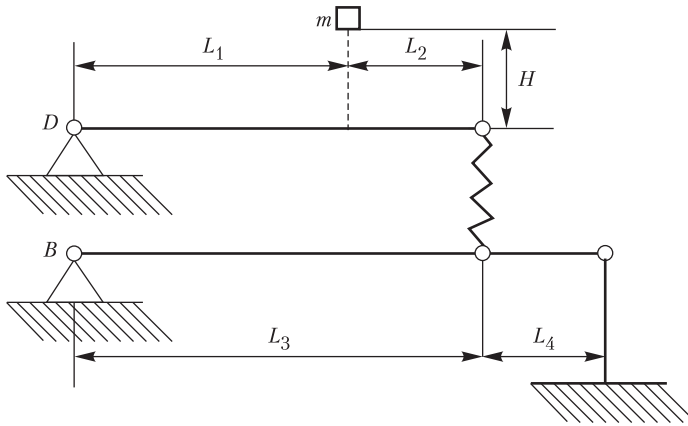


Рис. П.35. Схема ударного нагружения сложной системы

Пример 39. Статистическая обработка результатов эксперимента

В программе приведены результаты испытаний на усталость серии из 10 стальных образцов. Требуется построить усталостную кривую и определить доверительный интервал для вероятности неразрушения образцов $p = 99\%$.

Пример 40. Расчет болта на усталостную прочность

Определить диаметр болта (рис. П.36), испытывающего нагружение асимметричным циклом растяжения из расчета: на статическую прочность, по пределу выносливости, на долговечность $N = 10^5$ циклов.

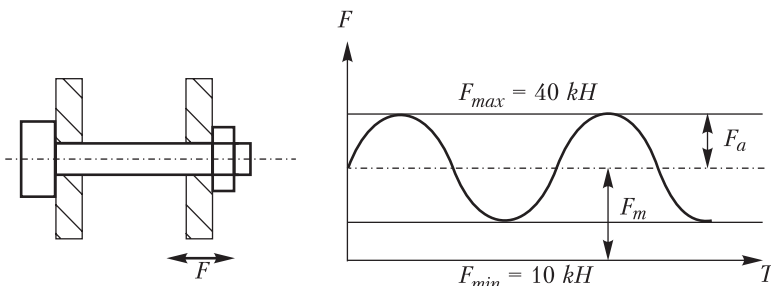


Рис. П.36. Схема болтового соединения и характеристики цикла нагружения

Пример 41. Расчет долговечности детали при нестационарном нагружении

Определить срок службы автомобильной оси из стали 18ХНМА (рис. П.37). Обработка циклограммы нагружения позволила свести процесс нагружения оси

к повторению блока нагружения. Определить, сколько блоков нагружения выдержит ось до разрушения.



Рис. П.37. Нестационарное нагружение автомобильной оси

Пример 42. Расчет долговечности полосы с трещиной

В полосе из стали 20X (рис. П.38) обнаружена боковая трещина длиной $L = 2$ мм. Полоса испытывает пульсирующий цикл нагружения. Как характеристика материала приводится экспериментально найденная зависимость

$$\frac{dL}{dN} = 1,5 \cdot 10^{-13} (\Delta K)^3.$$

Определить число циклов N подрастания трещины до критической длины $L_{кр}$.

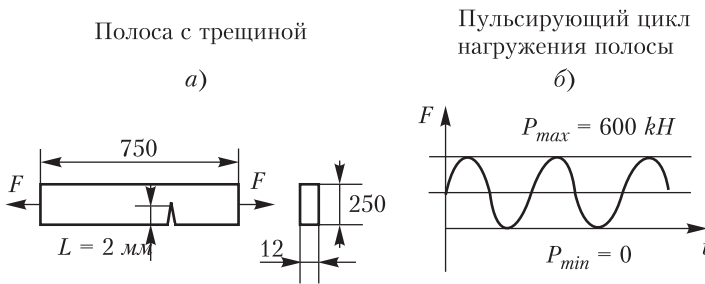


Рис. П.38. Схема полосы с трещиной и цикл ее нагружения

Пример 43. Построение диаграммы истинных напряжений

Указанная диаграмма строится по результатам испытаний стандартного образца на растяжение. По замерам образца до и после испытаний в расчет вводятся начальные длина L_0 и диаметр образца d_0 , а также диаметр образца в шейке после разрыва d_k

С нарисованной испытательной машиной диаграммы в координатах нагрузка — перемещение снимаются координаты нескольких характерных точек. Далее рассчитываются условные и истинные напряжения и деформации, определяется зависимость напряжений от деформаций при линейной и степенной аппроксимации диаграммы деформирования.

Пример 44. Расчет тонкостенной оболочки

Для оболочки, состоящей из трех частей: конической, цилиндрической и сферической (рис. П.39), при заданных габаритных размерах и внутреннем давлении найти толщину стенки.

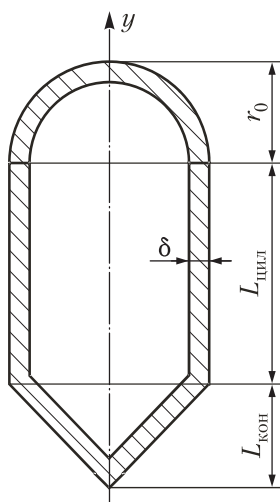


Рис. П.39. Схема цилиндрической оболочки

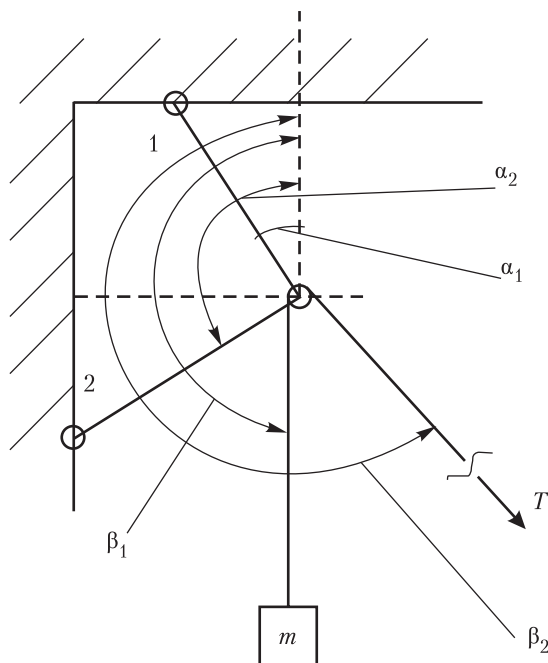


Рис. П.40. Схема нагружения кронштейна

Пример 45. Оптимизация расчета кронштейна

Для кронштейна из двух шарнирно соединенных стержней (рис. П.40) требуется определить:

- площади сечений стержней из условия прочности,
- углы наклона стержней 1 и 2 из условия равенства напряжений в стержнях,
- углы 2 и 2 из условия минимального веса кронштейна.

Пример 46. Оптимальный расчет сферической крышки

Сосуд высокого давления герметично закрыт сферической крышкой (рис. П.41). Давление в сосуде p , диаметр сосуда $2r_0$, толщина крышки d неизвестна, но должна удовлетворять условию прочности. Требуется определить радиус кривизны крышки R , при котором масса крышки $m(R)$ будет минимальной.

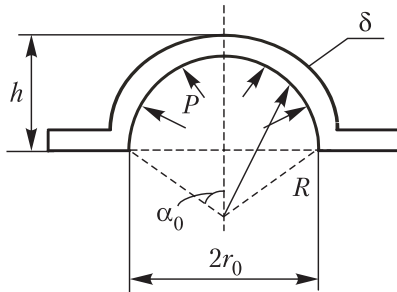


Рис. П.41. Сферическая крышка под внутренним давлением

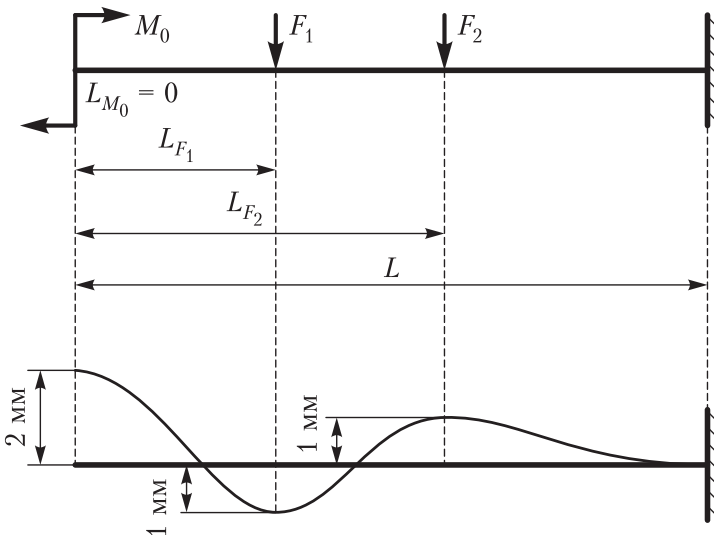


Рис. П.42. Перемещения консольной балки и нагрузки, которые надо определить

Пример 47. Подбор нагрузок по перемещениям балки

Для консольной балки прямоугольного сечения (рис. П.42) заданы перемещения в трех точках после деформации. Требуется подобрать нагрузки, действие которых на балку вызовет появление заданных перемещений.

Пример 48. Подбор нагрузок на балку, удовлетворяющих граничным условиям

В этой задаче заданы равенство нулю прогиба и угла поворота в заделке $L_1 = 0$, прогиб $= 0,5$ см, изгибающий момент $M = 10$ кг см в сечении $L_2 = 50$ см и поперечная сила $Q = 50$ кг на свободном конце балки $L_3 = 100$ см (рис. П.43). В принципе можно задать любые граничные условия. Определяются нагрузки, удовлетворяющие заданным условиям.

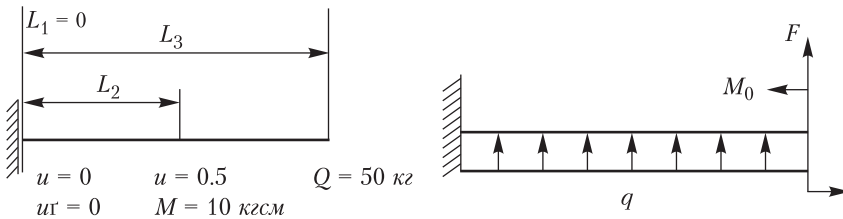


Рис. П.43. Балка, граничные условия и нагрузки на балку

Пример 49. Поиск оптимальных размеров сечения балки

Консольная балка прямоугольного сечения нагружена одной силой на свободном конце. Требуется подобрать размеры прямоугольного сечения балки из условий, что максимальное напряжение $\sigma_{\max} = [\sigma_{\text{дон}}]$, перемещение конца балки $\Delta \leq [\Delta] = 2$ мм. На размеры сечения тоже наложены ограничения $2 \cdot \text{см} < b < 6 \cdot \text{см}$ и $1 \cdot \text{см} < h < 10 \cdot \text{см}$.

Пример 50. Динамический расчет балки матричным методом перемещений

Для тонкой линейки из оргстекла, защемленной одним концом (рис. П.44), определить: 1) собственные частоты, 2) отклик балки на воздействие вынуждающих сил. Статический и динамический расчет ведется методом конечных элементов с разбивкой на семь плоских стержневых элементов.

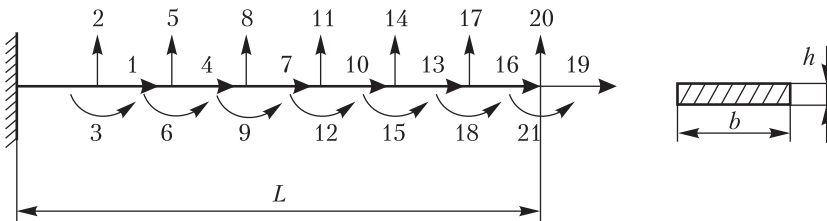


Рис. П.44. Разбивка балки на конечные элементы и нумерация перемещений в ней

Пример 51. Динамический расчет плоской рамы методом конечных элементов

Расчет аналогичен предыдущему. Отличается только ориентацией стержней-элементов (рис. П.45).

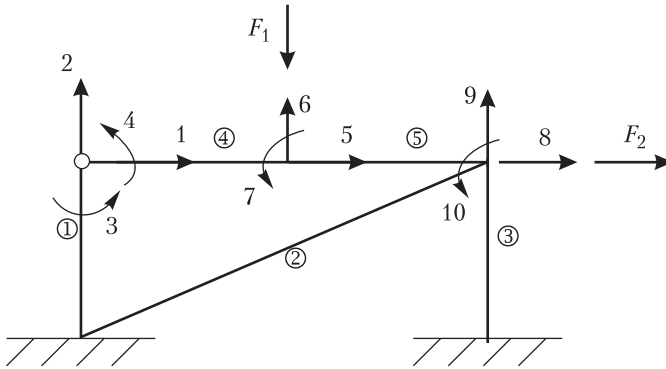


Рис. П.45. Разбивка плоской рамы на конечные элементы и нумерация перемещений в ней

Пример 52. Расчет пространственной стержневой системы

Расчет аналогичен предыдущему. Отличается только ориентацией стержней-элементов в пространстве (рис. П.46).

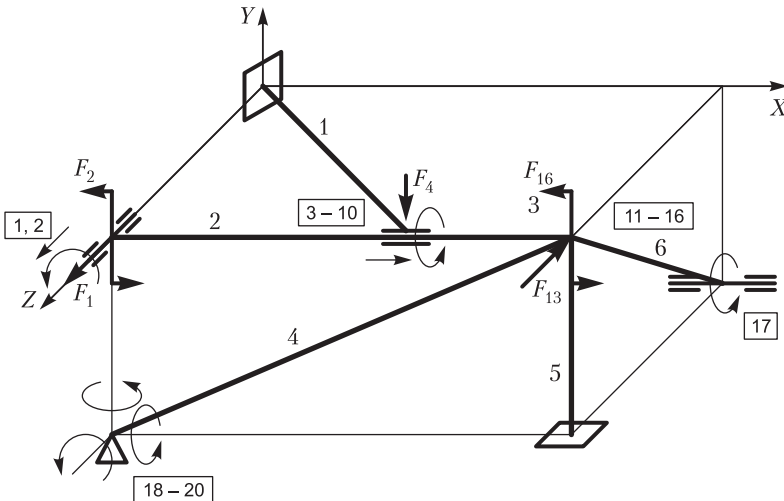


Рис. П.46. Разбивка пространственной рамы на конечные элементы и нумерация перемещений в ней

Пример 53. Расчет кольца методом конечных элементов

В данном примере решается плоская задача теории упругости. Кольцо, нагруженное двумя силами, имеет две оси симметрии, поэтому для повышения точности

расчета рассматривается одна четвертая часть кольца (рис. П.47). Четверть кольца разбиваем на треугольные конечные элементы с шестью степенями свободы.

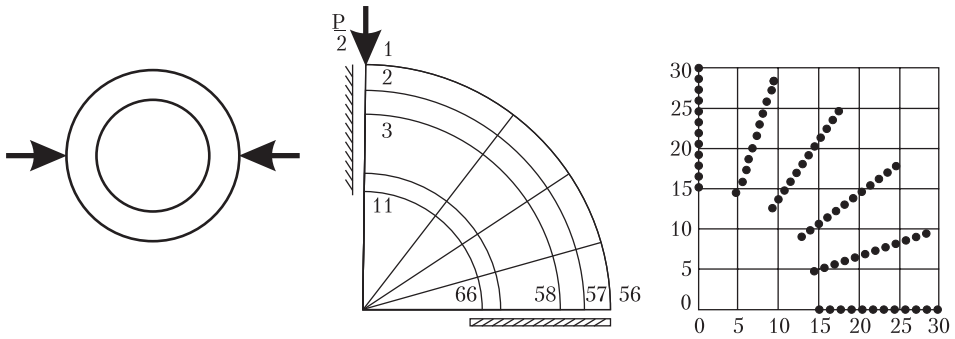


Рис. П.47. Расчетная схема кольца под нагрузкой и выбранные для расчета узлы

Пример 54. Решение уравнения Пуассона при кручении стержня

Для стержня заданного сечения (рис. П.48) определить: момент инерции и момент сопротивления при кручении, распределение касательных напряжений в сечении.

В данном примере рассматривается решение уравнения Пуассона методом конечных элементов для областей произвольной формы и произвольной разбивки на элементы. Распределение касательных напряжений при кручении стержня произвольного сечения подчиняется уравнению Пуассона.

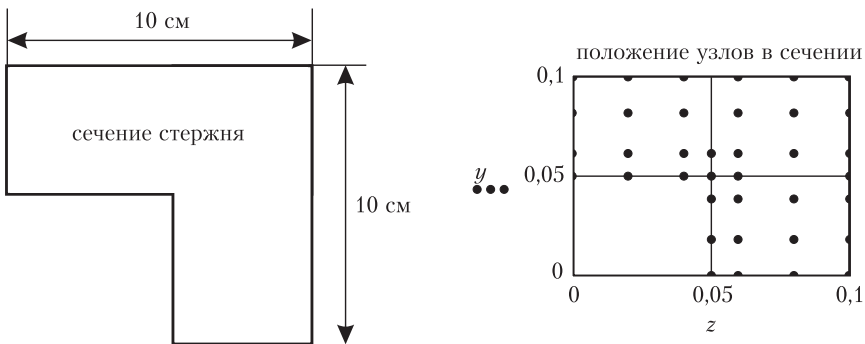


Рис. П.48. Поперечное сечение стержня и выбранные для расчета узлы

Пример 55. Решение плоской упругопластической задачи методом конечных элементов

Данный пример иллюстрирует возможность и удобство программирования на Mathcad любой сколь угодно сложной задачи.

Для кольца под нагрузкой (см. рис. П.47) рассчитывается распределение напряжений и деформаций. Влияние пластических деформаций учитывается методом переменных параметров упругости.

Пример 56. Использование прямоугольного квадратичного элемента в методе конечных элементов

Для длинной толстой трубы под давлением 700 атмосфер найти распределение напряжений в поперечном сечении.

Для расчета используется прямоугольный квадратичный элемент с восемью узлами, что позволяет достичь требуемой точности расчета даже при использовании одного конечного элемента (рис. П.49).

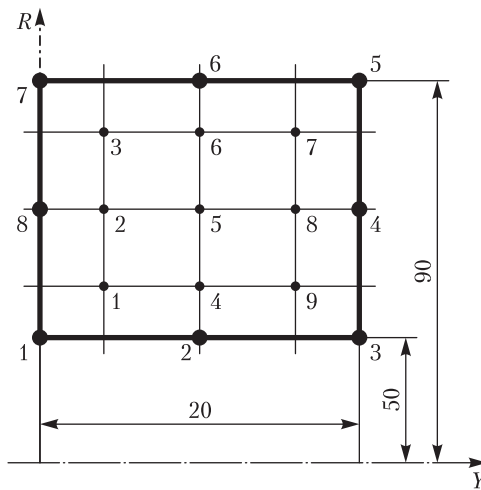


Рис. П.49. Прямоугольный квадратичный элемент с восемью узлами и девятью точками интегрирования

Евгений Макаров

Инженерные расчеты в Mathcad 15. Учебный курс

Заведующий редакцией
Руководитель проекта
Ведущий редактор
Художественный редактор
Корректор
Верстка

*А. Кривоцов
А. Юрченко
Ю. Сергиенко
Л. Адуевская
И. Тимофеева
Л. Харитонов*

ООО «Мир книг», 198206, Санкт-Петербург, Петергофское шоссе, 73, лит. А29.
Налоговая льгота — общероссийский классификатор продукции ОК 005-93, том 2; 95 3005 — литература учебная.
Подписано в печать 26.01.11. Формат 70х100/16. Усл. п. л. 32,250. Тираж 2000. Заказ 0000.
Отпечатано по технологии СтР в ОАО «Печатный двор» им. А. М. Горького.
197110, Санкт-Петербург, Чкаловский пр., 15.