

Лекция 5. Оформление Maxima

5.1 Классические графические интерфейсы Maxima

5.1.1 Графический интерфейс wxMaxima

Для удобства работы сразу обратимся к графическому интерфейсу **wxMaxima**, т. к. он является наиболее дружелюбным для начинающих пользователей системы.

Достоинствами **wxMaxima** являются:

- возможность графического вывода формул;
- упрощенный ввод наиболее часто используемых функций (через диалоговые окна);
- возможность включения графических иллюстраций непосредственно в текст рабочей книги (при использовании формата **wxMaxima**)

5.1.1.1 Рабочее окно wxMaxima

Рассмотрим рабочее окно программы (см. рис. 5.1 и рис. 5.2). Сверху вниз располагаются: текстовое меню программы — доступ к основным функциям и настройкам программы. В текстовом меню **wxMaxima** находятся функции для решения большого количества типовых математических задач, разделённые по группам: уравнения, алгебра, анализ, упростить, графики, численные вычисления. Ввод команд через диалоговые окна упрощает работу с программой для новичков.

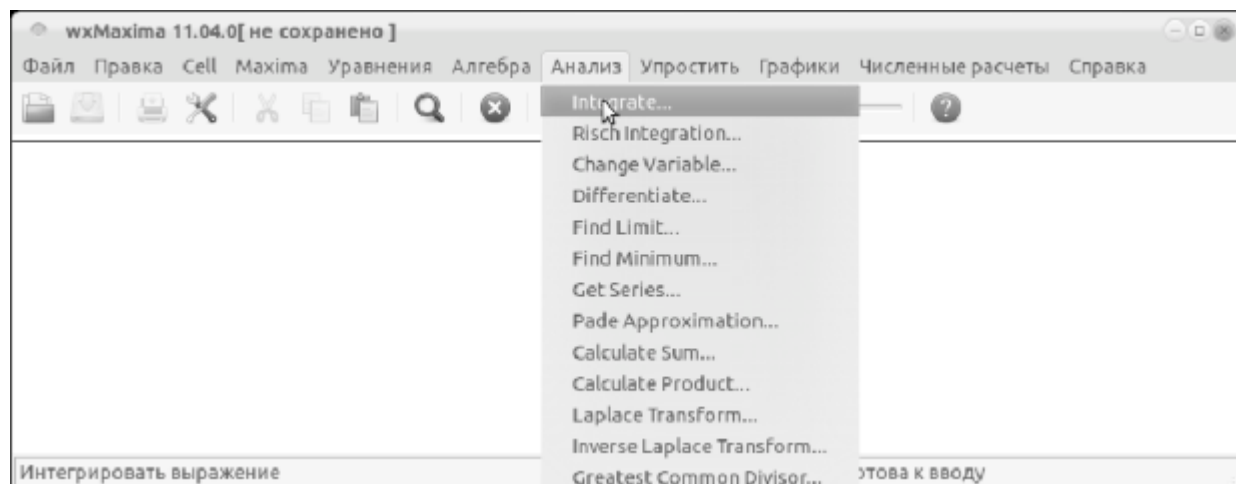


Рис. 5.1. Интерфейс wxMaxima, выбор команды интегрирования.

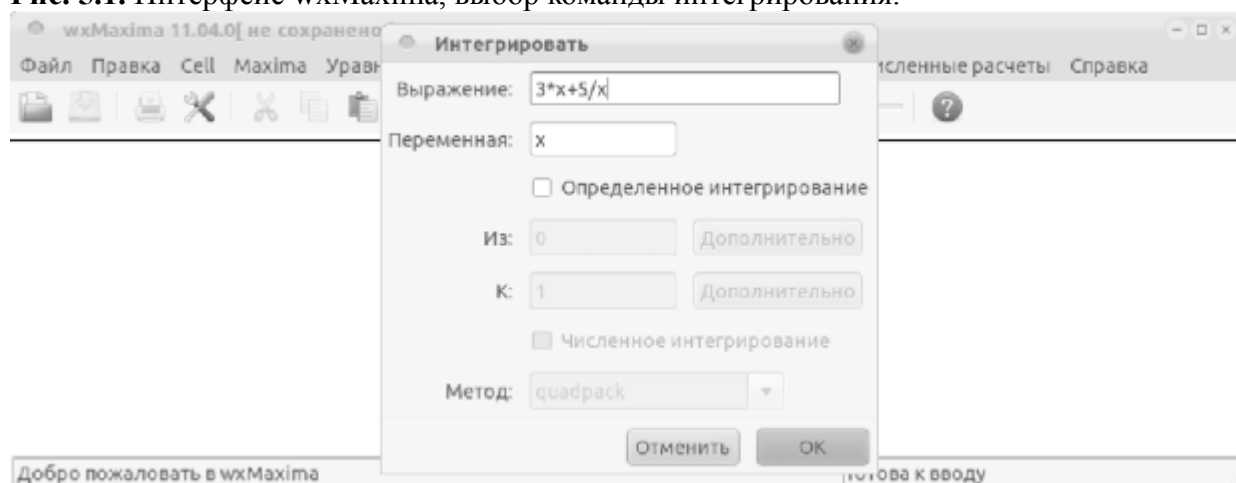


Рис. 5.2. Интерфейс wxMaxima, вычисление интеграла.

Например, пункт меню Анализ/Интегрировать позволяет вычислить определённый или неопределённый интеграл. После ввода необходимых параметров, в рабочем окне мы увидим команду и результат вычисления:

```
(%i1) integrate(3*x+5/x, x);
```

$$(%o1) 5 \log(x) + \frac{3x^2}{2}$$

Пример использования команд меню для вычисления предела

$$\lim_{x \rightarrow 0} \frac{\sin(2x)}{x}$$

представлен на рис. 5.3 и рис. 5.4. Следует отметить, что оболочка **wxMaxima** при вызове команды и соответствующего диалогового окна генерирует текстовую команду, интерпретируемую вычислительным ядром **Maxima**. Передаваемая ядру **Maxima** строка выводится в командное окно аналогично команде, введённой вручную. После генерации и первого выполнения команды (или набора команд) можно дополнять и редактировать автогенерированную команду, рассматривая её в качестве шаблона.

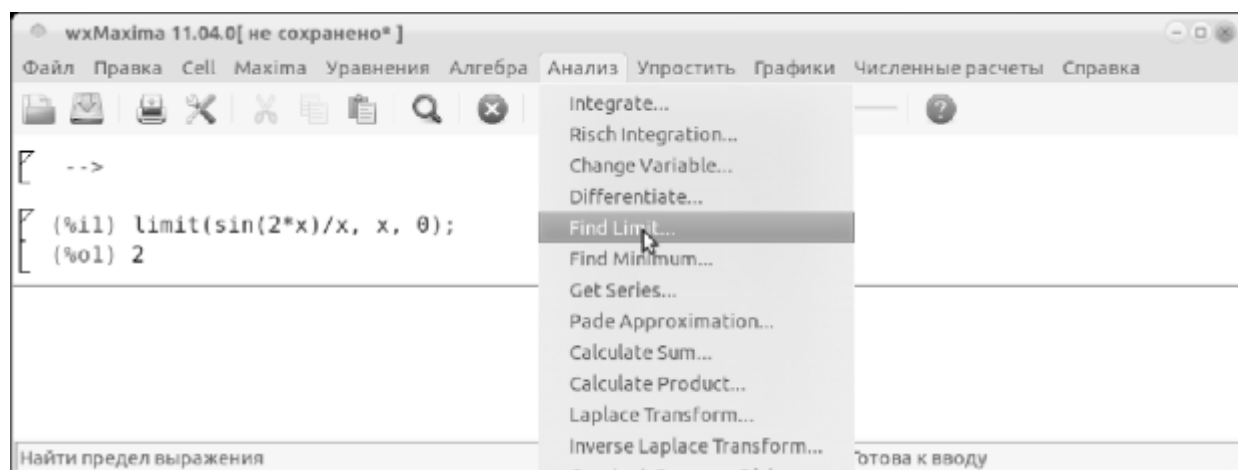


Рис. 5.3. Интерфейс wxMaxima, выбор команды find limit.

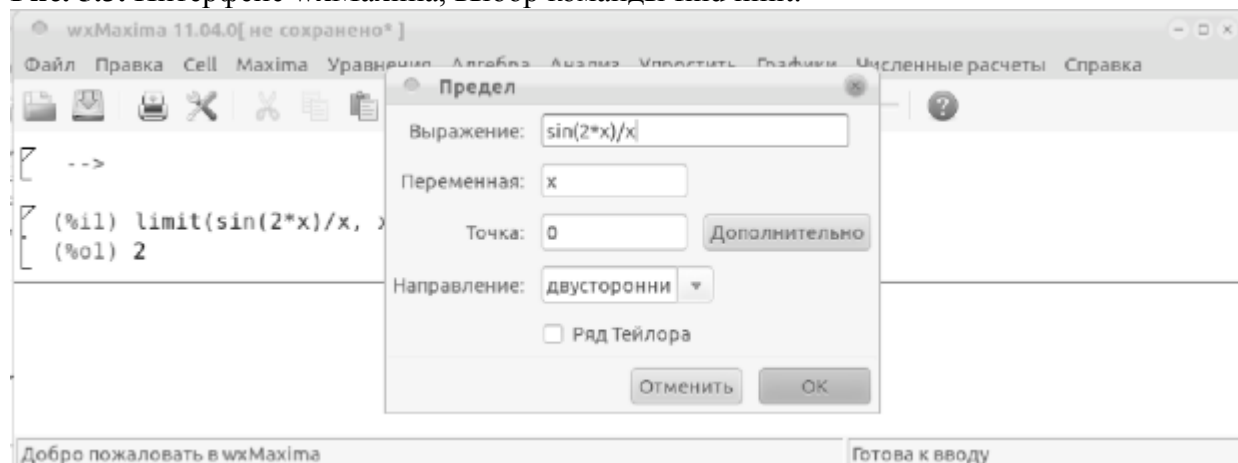


Рис. 5.4. Интерфейс wxMaxima, окно ввода — вычисление предела.

Ниже располагается графическое меню основных команд с пиктограммами, соответствующими наиболее часто используемым функциям для работы с файлами: открыть / сохранить / печать данных, а также функциям правки — копировать / удалить / вставить текст и другие.

Центральную часть рабочего окна **wxMaxima** занимает командное окно (псевдотерминал), в которое вводятся команды системы и выводятся результаты.

В последних версиях интерфейсного пакета **wxMaxima** поддерживается концепция ячеек (cells) в рабочей книге. Ячейка включает либо набор команд **Maxima**, либо результаты их выполнения (в т. ч. графики). Кроме того, по аналогии с Maple и Mathematica **wxMaxima** поддерживает текстовые ячейки (text cells) для пояснений и комментариев, а также ячейки для заголовков и номеров секций (title cells, section cells, subsection cells). Пример книги Maxima с ячейками указанных типов представлен на рис. 5.5. Допускается вставка изображений в рабочую книгу (также в специальные ячейки).

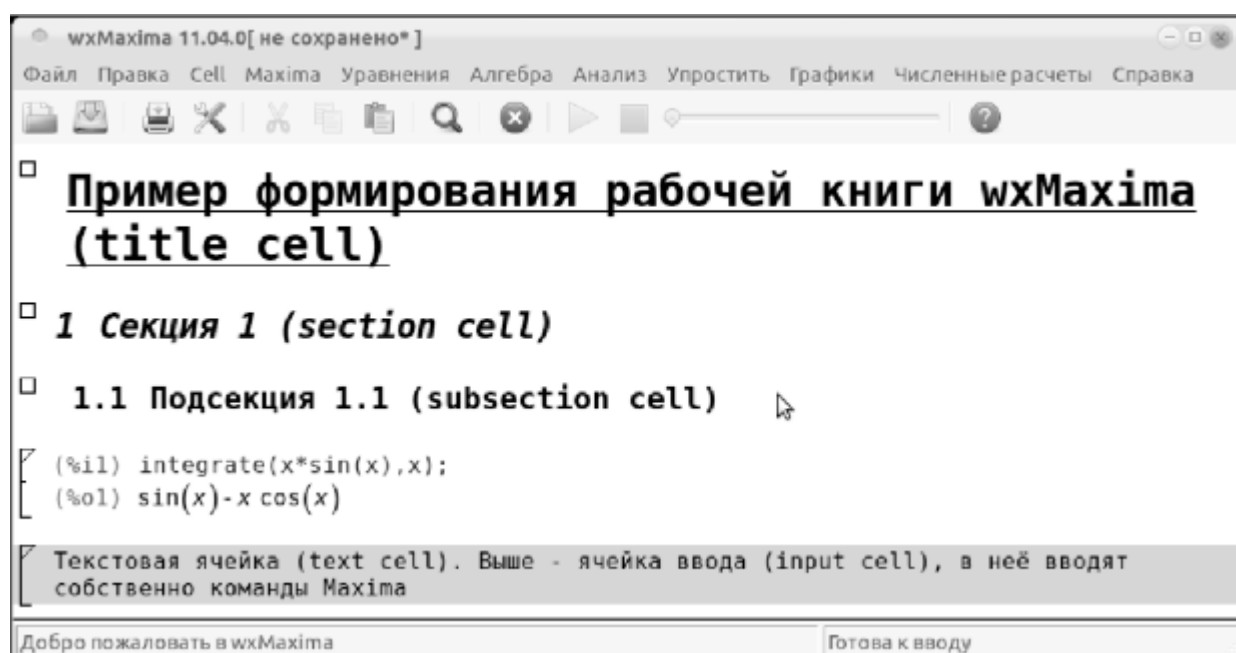


Рис. 5.5. Пример вставки ячеек различных типов в книгу wxMaxima.

При сохранении книги (в формате wxm) в файл выводятся только входные ячейки (input). Поэтому при работе с сохранённым документом не обязательно интерпретировать все ячейки, хотя это возможно — команда Evaluate all cells из меню Cells).

Рабочую книгу **Maxima** можно экспортировать в форматы html или pdflatex.

Интерпретация текущей ячейки, в которой может быть несколько команд, осуществляется после нажатия комбинации клавиш Ctrl+Enter, либо командой меню Cells. Если необходимо предотвратить вывод отклика команды, следует явно завершить её символом \$. Современные версии **wxMaxima** автоматически завершают ввод, если это необходимо, символом ";".

При использовании интерфейса **wxMaxima** можно выделить в командном окне необходимую формулу и вызвав контекстное меню правой кнопкой мыши: скопировать любую формулу в текстовом виде, в формате TEX или в виде графического изображения, для последующей вставки в какой-либо документ. Пример контекстных меню при работе с **wxMaxima** смотри на рис. 5.6, рис. 5.7 и рис. 5.8.

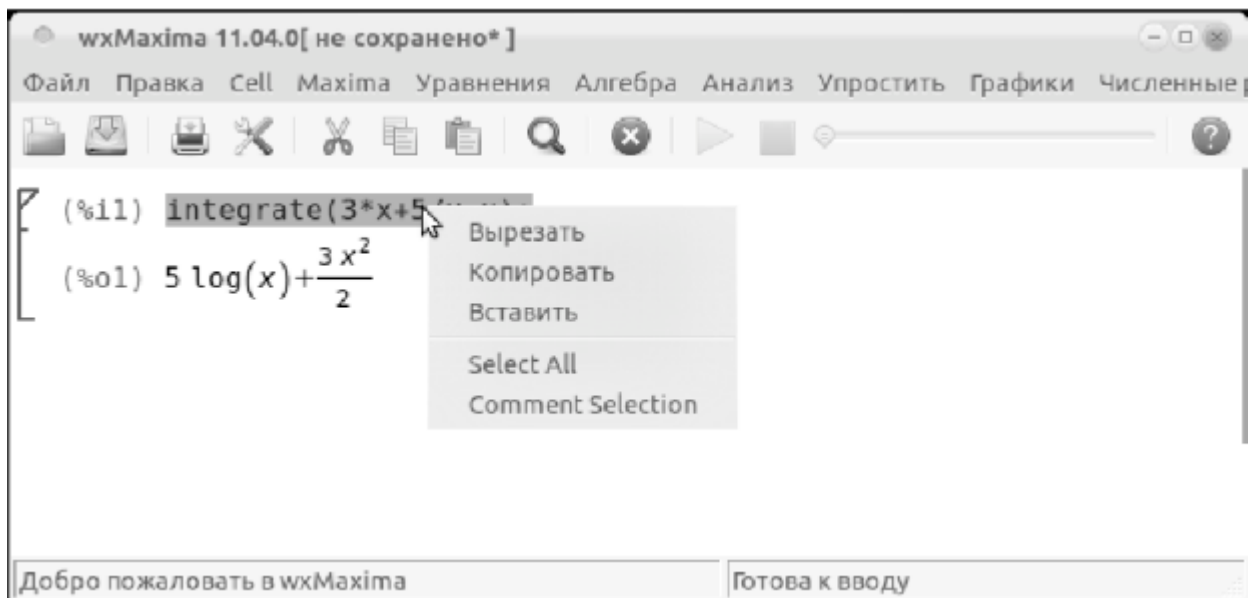


Рис. 5.6. Интерфейс wxMaxima. Контекстное меню строки ввода.

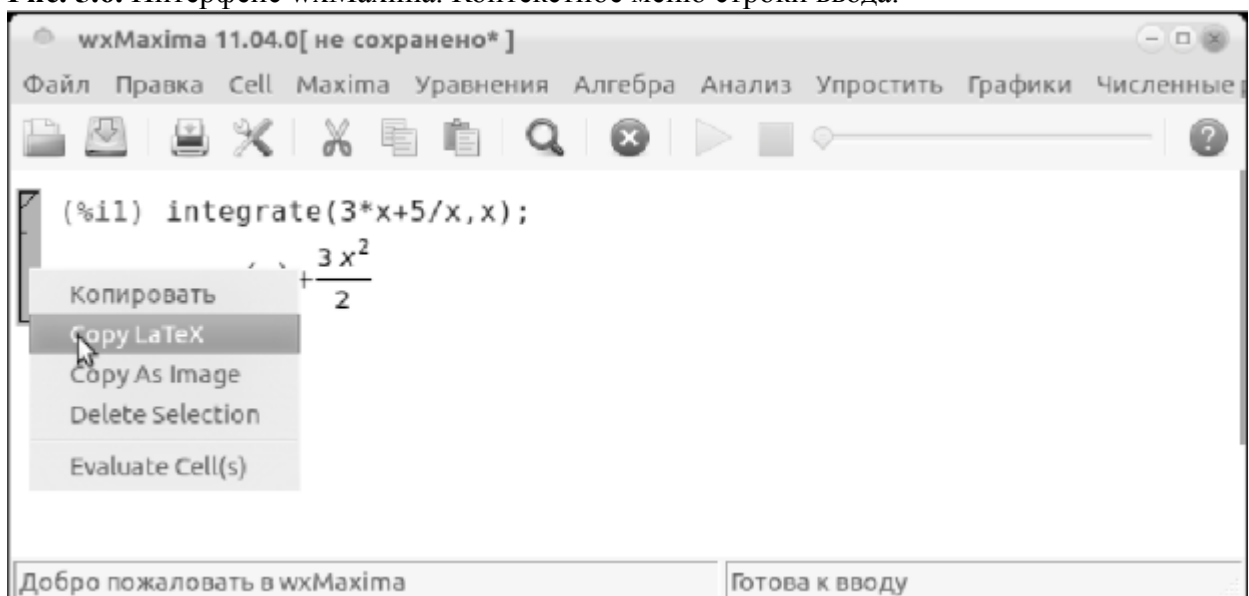


Рис. 5.7. Интерфейс wxMaxima. Контекстное меню ячейки.

Также, в контекстном меню, при выборе результата вычисления, предлагаются ряд операций с выбранным выражением (например, упрощение, раскрытие скобок, интегрирование, дифференцирование и др.).

По умолчанию **wxMaxima** предполагает, что команда, вводимая при помощи кнопки, применяется к последнему выводу (т. е. аргумент команды — $\%$). Все кнопки или пункты меню в верхней или нижней части рабочего окна соответствуют той или иной команде **Maxima**.

Кроме того, **wxMaxima** предоставляет удобный интерфейс к документации по системе **Maxima**.

Меню правка \rightarrow настройки обеспечивает достаточно широкие возможности настройки графического интерфейса **wxMaxima**. Предусмотрены три группы параметров:

- опции, определяющие отдельные особенности выполнения команд;
- опции вызова вычислительного ядра **Maxima**;
- опции, определяющие стиль графического интерфейса (язык, шрифты, цветовую гамму и т.

п.).

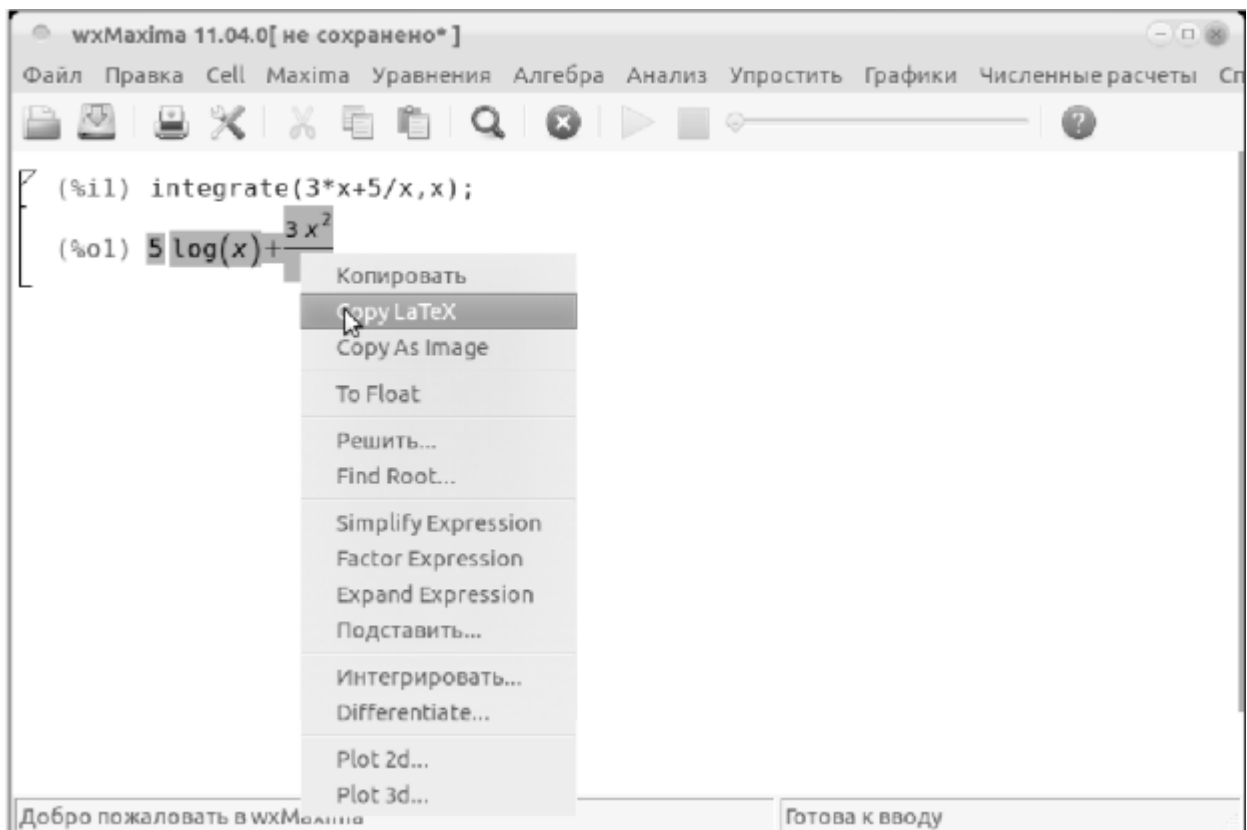


Рис. 5.8. Интерфейс wxMaxima. Контекстное меню строки вывода.

Управление процессом вычислений осуществляется командами пункта главного меню **Maxima**. Пользователю предоставляются следующие возможности:

- прервать вычисления, перезапустить **Maxima**, очистить память;
- просмотреть содержимое памяти (переменные, функции, определения и т. п.);
- изменить формат просмотра результатов.

5.1.2 Графический интерфейс xMaxima

Интерфейс **xMaxima** фактически является специфичным видом веб-браузера, т.к. данный интерфейс предусматривает обмен данными с вычислительным ядром **Maxima** через сокет. Интерфейс отличается простотой (точнее, минимализмом). В последних версиях **xMaxima** при старте открываются одновременно окно браузера системы помощи и консоль команд.

Предполагается, что пользователь владеет командами **Maxima** и макроязыком программирования. Общий вид командного окна **xMaxima** представлен на рис. 5.9. Пункты меню File, Edit, Options позволяют управлять сессией **Maxima**, сохранять и запускать batch- файлы. В рабочую книгу **xMaxima** можно встраивать графики в формате openmath (в зависимости от установки опции plot window). Пример рабочего окна **xMaxima** с простыми графиками представлен на рис. 5.10. График в рабочей книге можно вращать, редактировать, охранять в файл. Как и **wxMaxima**, интерфейс **xMaxima** предоставляет доступ к html-файла помощи по пакету **Maxima**.

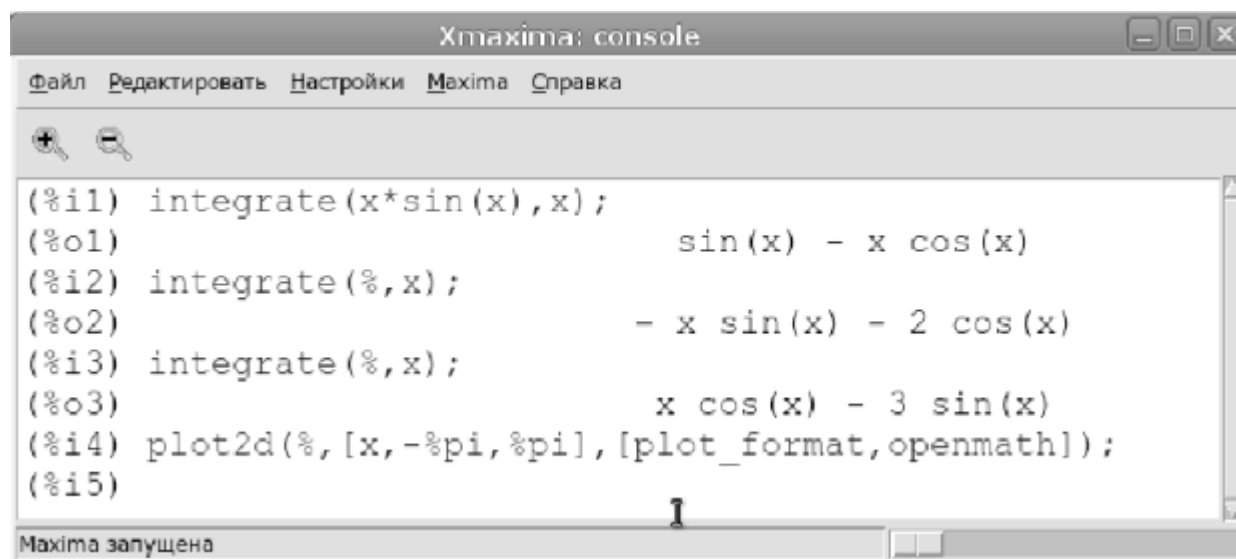


Рис. 5.9. Общий вид рабочего окна xMaxima

5.1.3 Использование редактора TeXmacs в качестве интерфейса Maxima

Широкие возможности работы в **Maxima** и других математических пакетах предоставляет редактор **TeXmacs**. Разработчик позиционирует его как $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -редактор, однако это не совсем так. **TeXmacs** использует собственный внутренний формат, но позволяет экспортировать документы в $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ (при этом полученный $\text{T}_{\text{E}}\text{X}$ -файл очень похож на результат экспорта в .tex документа OpenOffice). **TeXMacS** хорошо локализован и полностью поддерживает русский язык, а также все возможности стандартного текстового процессора

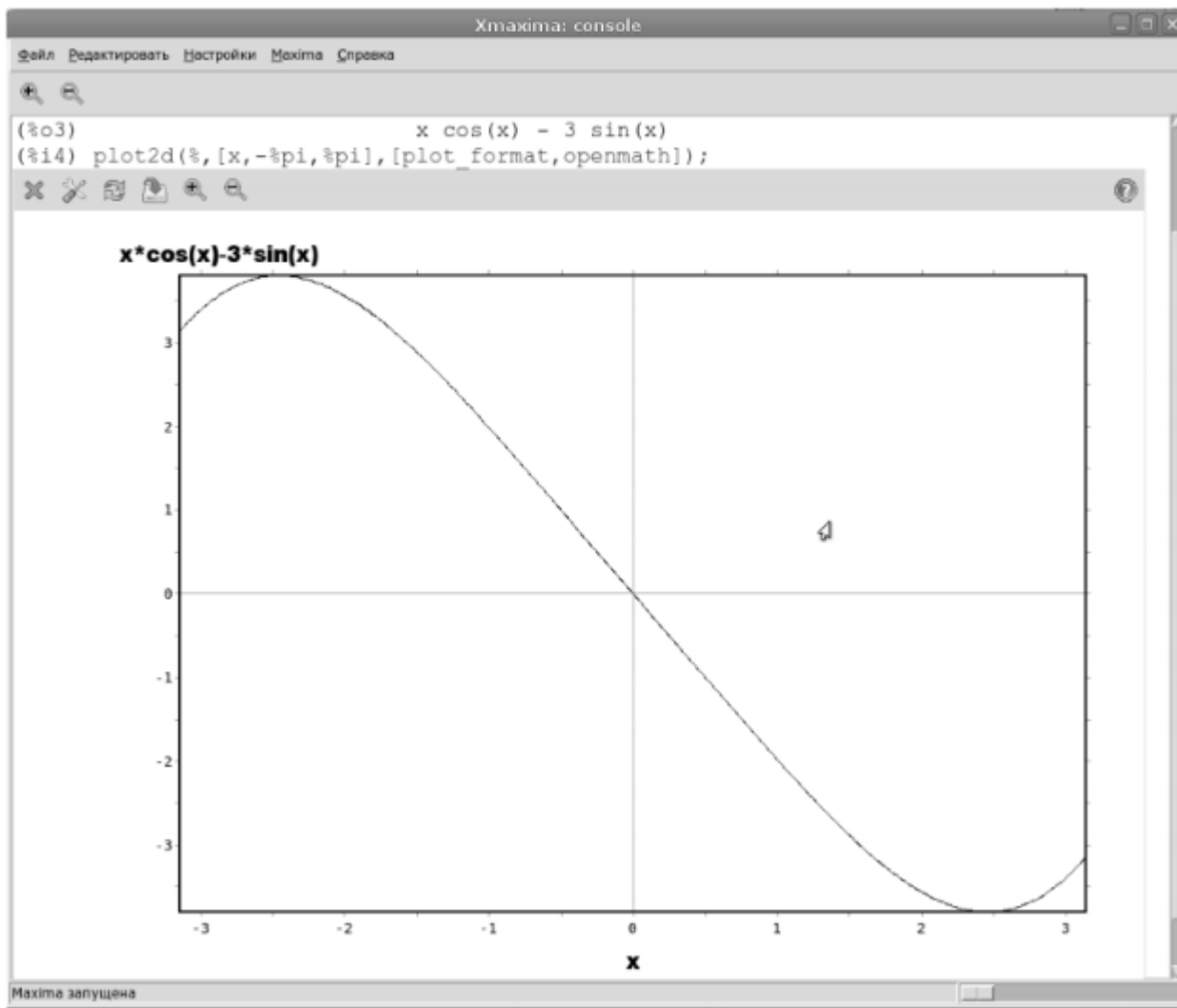


Рис. 5.10. Встроенный график в рабочей книге xMaxima

В **TeXmacs** реализован подход к структуре документа, во многом идентичный \LaTeX , а также возможности ввода и редактирования сложных математических формул. Недостатком редактора является неудачный выбор способа локализации, что затрудняет открытие документов **TeXmacs** при помощи других редакторов (OpenOffice и др.).

Важной особенностью **TeXmacs** является возможность встраивать в текст документа сессии работы с различными математическими пакетами (в т.ч. и **Maxima**). Общий вид рабочего окна **TeXmacs** представлен на рис. 5.11. Последовательность вставки сессии **Maxima** в текст документа показана на рис. 5.12.

Возможность встраивать в текст документа графические иллюстрации, также возможность расщеплять сессию для ввода пояснений и комментариев делает **TeXmacs** весьма привлекательным средством для работы с **Maxima**. В современных версиях **TeXmacs** при запуске сессии **Maxima** в главном меню появляется пункт **Maxima**, в котором предусмотрено выпадающее меню с перечнем основных команд **Maxima**. Недостатками **TeXmacs** являются отсутствие русификации при работе в **Maxima**-режиме, а также проблемы на некоторых дистрибутивах с запуском сессии **Maxima**.

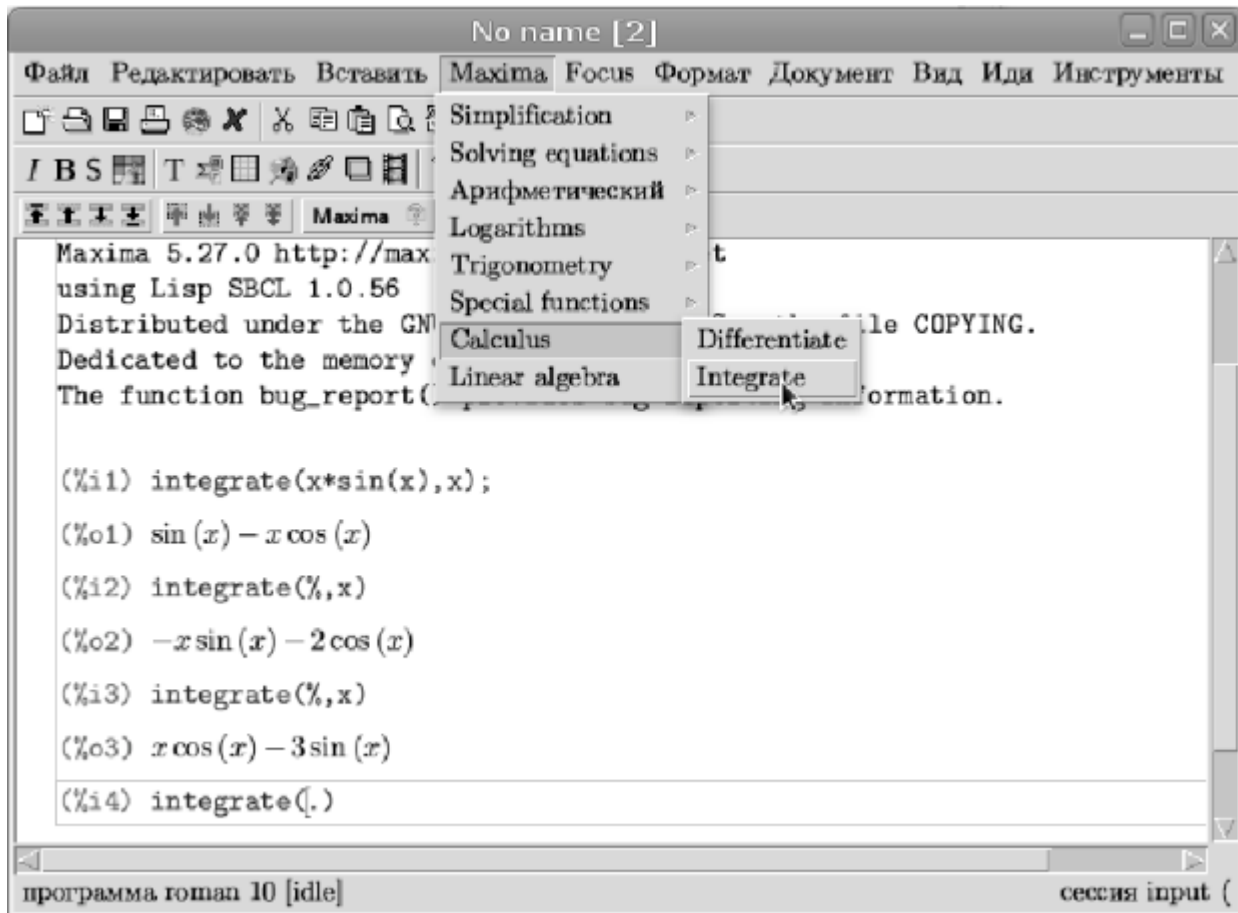


Рис. 5.11. Рабочее окно TeXmacs с запущенной сессией Maxima

Для решения проблем с запуском **Maxima**-сессии из **TeXmacs** возможным решением является редактирование файла `/usr/lib/texmacs/TeXmacs/bin/maxima_detect`, в котором надо ссылку на `#!/bin/sh` заменить ссылкой на `#!/bin/bash` в самом начале файла.

Окончательную версию **TeXmacs**-документов целесообразно представлять в pdf-формате (этот редактор обеспечивает прямой экспорт в pdf). При сохранении документов в формате **TeXmacs** и их последующем редактировании возможно и редактирование полей ввода сессии **Maxima** с пересчётом результатов.

5.1.4 Работа с Maxima из Emacs

Универсальный редактор **Emacs** также может использоваться в качестве front-end к **Maxima**. Для этого предусмотрено несколько режимов: `maxima-mode`, `EMaxima` и `iMaxima`.

Основной режим работы с **Maxima** в **Emacs** — `maxima-mode`. Этот режим запускается клавиатурной комбинацией `M-x-maxima-mode` (обычно нажатием `alt-M-alt-x` и после появления подсказки — набор `maxima`). Этот режим несколько аскетичен (похож на **xMaxima**), но достаточно удобен. Общий вид рабочего окна для данного режима представлен на рис. 5.13. На этом же рисунке видно меню навигации по текущей сессии, позволяющее показывать необходимый участок сессии, сохранять часть результатов в протокол, повторять ввод уже использовавшихся в данной сессии команд и т.п.

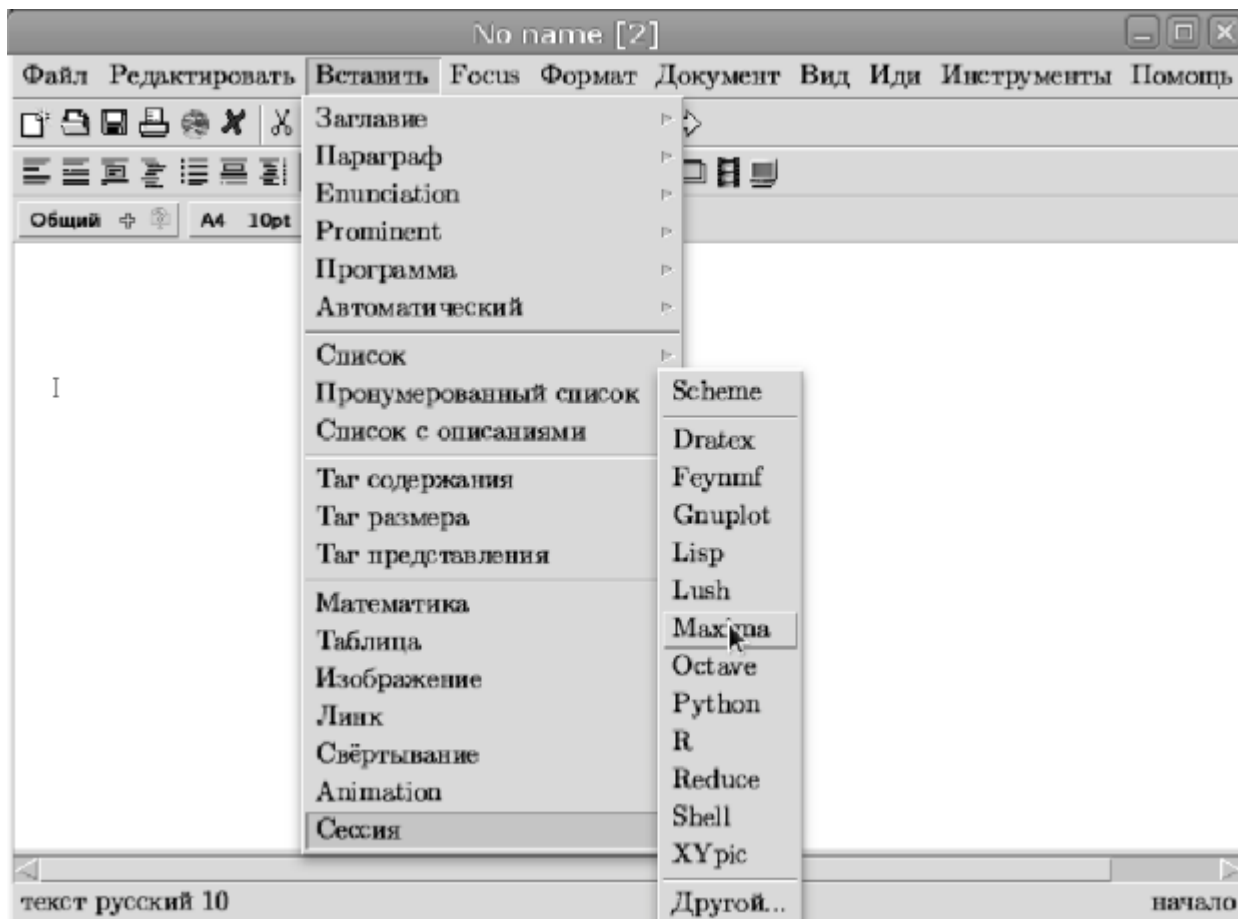


Рис. 5.12. Запуск сессии Maxima в текущем документе TeXmacs

Графики в рабочую книгу, открытую в **Emacs**, не встраиваются. Сохранение копии рисунка должно выполняться средствами gnuplot или openmath.

Интерфейс EMaxima — скорее не самостоятельный режим, а надстройка над режимом $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, которая наверняка понравится тем, кто использует **Emacs** для редактирования $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -документов. В отличие от режима **Maxima**, который предназначен для обычного изолированного запуска полноценной **Maxima**-сессии, здесь речь идет о возможности вставлять отдельные команды **Maxima** и, естественно, результаты их вычислений, прямо в редактируемый $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ документ. Запуск режима осуществляется командой EMaxima-mode (M-x emaxima).

В простейшем случае с использованием EMaxima можно создать ячейку **Maxima** комбинацией C-c C-o ("open cell"), ввести в ней любую команду или набор команд **Maxima** в текстовой нотации и получить результат вычисления этой команды либо в обычном текстовом виде нажатием C-c C-u c, либо в $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -виде с помощью C-c C-u C (т. е. Ctrl-c Ctrl-u Shift-c). Здесь "u c" происходит от "update cell"; а смежные команды, генерирующие вывод в простой текстовой форме и в форме $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, всегда привязаны в EMaxima к одинаковым строчной и заглавной буквам соответственно. Пример работы с EMaxima представлен на рис. 5.14, где показаны результаты создание ячейки с **Maxima**-кодом и результаты дополнения ячейки (команды можно выбирать из меню EMaxima в верхней части рабочего окна).

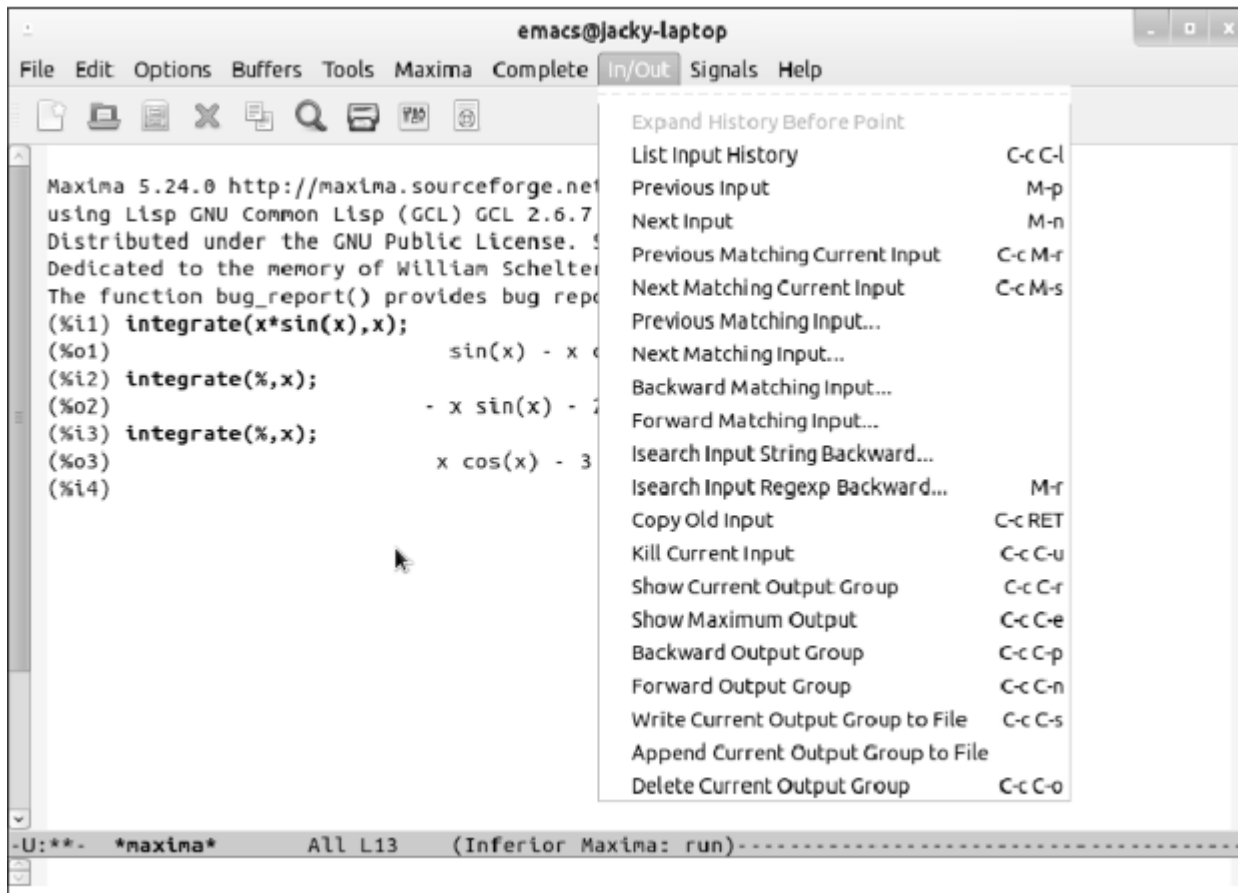


Рис. 5.13. Рабочее окно Emacs с запущенной сессией Maxima

Использовать интерфейс EMaxima удобно при создании объёмных документов в LATEX математического характера, которые предполагают включение результатов символьных вычислений.

Последний Emacs-интерфейс к Maxima — iMaxima — отличается от остальных самостоятельным (а не посредством LATEX -документа, как в EMaxima) графическим представлением математических формул. Собственно, именно для этого он и создан, и его отличие от Maxima-mode заключается именно в возможности графического отображения TEX -кода, генерируемого Maxima.

Этот режим можно настроить таким образом, чтобы внутри него запускался режим Maxima (т. е. Maxima-Emacs), и пользоваться всеми командами последнего и их клавиатурными привязками. Т.е. фактически режим iMaxima в таком варианте можно рассматривать как графический интерфейс уже над Maxima-Emacs; именно это может добавить дополнительной привлекательности последнему. В отличие от всех рассмотренных выше интерфейсов, iMaxima — сторонний проект, разрабатываемый отдельно. Для его установки необходимо дополнительно установить пакет breqn, отвечающий за перенос строк в математических формулах в формате LATEX . Инструкцию по установке самой iMaxima и breqn можно найти на сайте проекта.

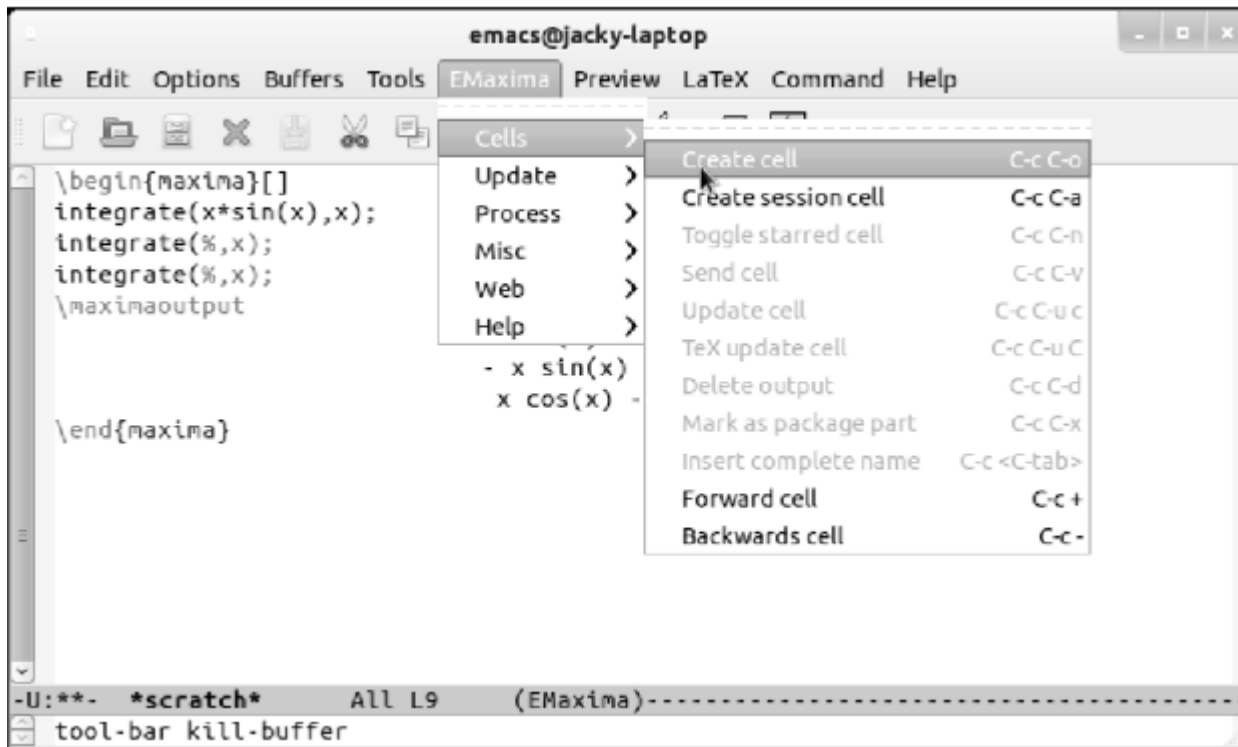


Рис. 5.14. Рабочее окно Emacs с запущенной сессией EMaxima

5.2 Работа с Maxima в KDE: интерфейс Cantor

Интерфейс пользователя **Cantor** состоит из трёх частей:

- Панель вкладок, при помощи которой можно переключаться между документами;
- Панель справки, где будет показано описание команды, если ввести "? команда";
- Панель текущего документа с меню команд, напоминающим интерфейс **wxMaxima**.

Пакет **Cantor** рассчитан на рабочий стол KDE, поэтому графический интерфейс написан с использованием библиотек Qt.

5.2.1 Документ Cantor

В **Cantor** вы работаете с документом. В нём можно вводить выражения, производить вычисления и видеть результаты. Аналогично **wxMaxima** или **Emacs**, документ **Cantor** включает ячейки, содержащие команды текущего пакета и результаты их выполнения. Наряду с командами и результатами, можно вводить и комментарии.

Набор доступных в выражениях команд зависит от используемой системы компьютерной алгебры, поэтому полезно знать синтаксис конкретной системы. Если вы знаете название команды, можно посмотреть её описание, введя "? команда". Чтобы посмотреть примеры документов **Cantor**, выберите пункт меню Файл → Загрузить примеры... и загрузите документы, опубликованные другими пользователями.

5.2.2 Настройка

В меню Настройка можно настроить внешний вид текущего документа. Параметр Показывать результаты с помощью LATEX влияет на то, в каком виде будут показаны результаты вычислений. Если он включён, результат будет обработан системой LATEX для создания визуально понятных

$$3x^2 \cdot \sqrt{2} \cdot x + \frac{2}{3}$$

формул. Например, $3 \cdot x^2 \cdot \sqrt{2} \cdot x + 2/3$ превратится в

Подсветка синтаксиса повышает читаемость кода, выделяя цветом ключевые слова и парные скобки. Если включить параметр Автодополнение, **Cantor** будет показывать возможные продолжения вводимой вами команды при нажатии клавиши Tab. Если существует только одно продолжение команды, при нажатии клавиши Tab название команды будет автоматически введено полностью.

Параметр Показывать номера строк позволяет добавить нумерацию введённых выражений. Нумерацию можно использовать для подстановки предыдущих результатов в новое выражение.

5.2.3 Прочие возможности Cantor

Документы **Cantor** сохраняются во встроенном формате cws. Конечный документ пользователя можно сохранять в формате $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$. При работе с **Maxima** в качестве backend графические документы можно встраивать в документ, при помощи контекстного меню они сохраняются в формат eps.

Опцию Встраивать график в файл можно отключить, при этом графики будут воспроизводиться программой gnuplot в отдельном окне. При наличии pdf-псевдопринтера можно распечатать документ **Cantor** в формате pdf (в т. ч. включённые в текст графики и комментарии).

5.3 Интегрированная среда Sage

Sage (англ. "Мудрец") — система компьютерной алгебры покрывающая много областей математики, включая алгебру, комбинаторику, вычислительную математику и матанализ. Первая версия **Sage** была выпущена 24 февраля 2005 года в виде свободного программного обеспечения с лицензией GNU GPL. Первоначальной целью проекта было "создание открытого программного обеспечения альтернативного системам Magma, Maple, Mathematica, и MATLAB". Разработчиком **Sage** является Уильям Стейн — математик Университета Вашингтона (Официальный сайт: <http://sagemath.org>).

Многочисленные возможности **Sage** включают:

- Интерфейс notebook для просмотра и повторного использования введённых команд и полученных результатов, включая графики и текстовые аннотации, доступный из большинства современных веб-браузеров. Доступно защищённое соединение через протокол HTTPS, когда конфиденциальность имеет значение. Так же **Sage** может выполняться как локально, так и удалённо.
- Интерфейс ввода на основе командной строки, с использованием мультипарадигменного языка IPython.
- Поддержка параллельных вычислений с использованием как многоядерных процессоров, так и многопроцессорных систем и систем распределённых вычислений.
- Внутренняя инфраструктура на python, поддерживающая взаимодействие с математическими пакетами на python: SymPy, SciPy и NumPy.
- Различные статистические библиотеки функций, использующие функциональность R и SciPy.
- Возможность построения плоских и трёхмерных графиков для функций и данных.
- Средства работы с матрицами и массивами данных с поддержкой разрежённых массивов.
- Набор инструментов для добавления собственного пользовательского интерфейса к вычислениям и приложениям.
- Сетевые инструменты для соединения с базами данных SQL, поддержка сетевых протоколов,

включая HTTP, NNTP, IMAP, SSH, IRC, FTP.

Sage — сам по себе мощное средство благодаря многочисленным объектно-ориентированным возможностям и большому объёму возможностей, реализованному на python для решения всевозможных задач. Однако следует учитывать, что основная идея **Sage** — интеграция всевозможных математических пакетов, как открытых, так и проприетарных. Наряду с функцией интеграции, **Sage** включает достаточно развитые собственные возможности — многочисленные функции и структуры данных. Преобразование результатов, полученных, например, в **Maxima**, к структурам **Sage** может оказаться достаточно сложной задачей.

5.4 Построение графических иллюстраций: пакет draw

В **Maxima** имеется несколько альтернативных библиотек для отображения графиков функций, наборов точек, трехмерных тел, градиентов и т.д. По умолчанию используется библиотека Plot, но для решения некоторых задач может оказаться удобнее библиотека Draw.

Варианты использования команды plot2d рассмотрены выше, поэтому ниже иллюстрируются возможности draw. Библиотека draw построена на интерфейсе Maxima-gnuplot. Библиотека включает три основные функции, доступные на уровне **Maxima**: draw2d, draw3d, draw. Перед использованием draw необходимо загрузить командой load("draw").

Рассмотрим несложный пример. На графике (рис. 5.15) показана кривая $y = \exp(x)$. График построен с использованием функции *draw2d*. Функции, заданные явно, указываются командой *explicit*. Для каждой функции указывается имя переменной и пределы изменения абсциссы. Пределы ординаты выбираются автоматически. Команда построения графика:

```
(%i4) draw2d(grid=true,xlabel = "Time",
           ylabel = "Population", explicit(exp(u),u,-2,2))$
```

На графике показана сетка (grid=true), а также метки осей (xlabel и ylabel).

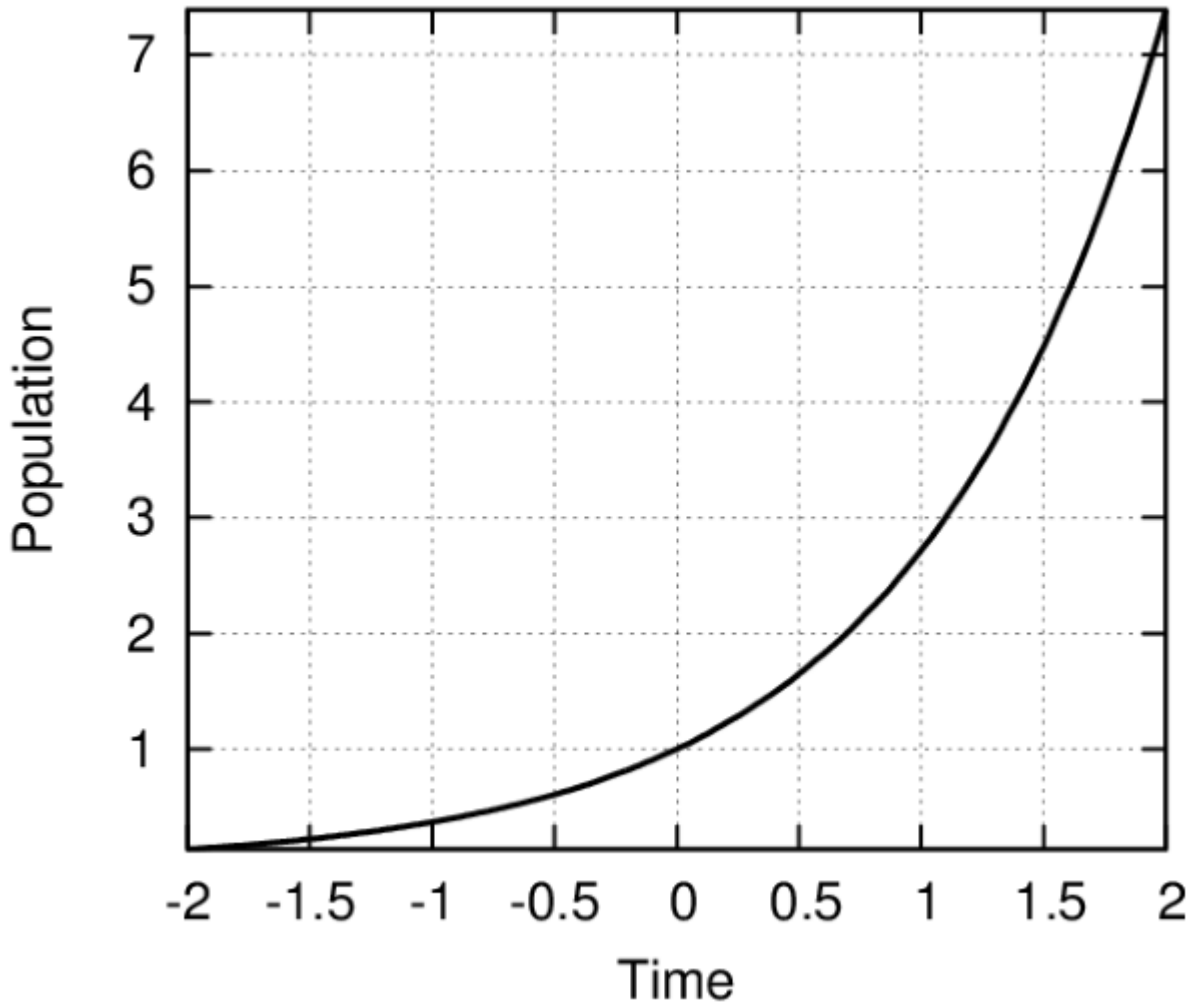


Рис. 5.15. График, построенный с помощью функции `draw2d`

Вывод графика на печать организуется при помощи указания типа терминала. Возможные варианты — `screen` (экран по умолчанию), `png`, `jpg`, `eps`, `eps_color`, `gif`, `animated_gif`, `wxt`, `aquaterm`.

Команда для вывода графика на печать имеет вид (указан терминал `eps` — `encapsulated postscript`):

```
(%i6) draw2d(terminal=eps,grid=true,xlabel = "Time",
           ylabel = "Population", explicit(exp(u),u,-2,2))$
```

По умолчанию рисунок сохраняется в файл `maxima_out.eps`; указание имени файла для вывода осуществляется командой `file_name = "имя файла"`.

Построим аналогичный график (рис. 5.16), но с выводом кривых $y = \exp(x)$ и $y = \exp(-x)$ в одних осях с сохранением графика в файл `draw_2.eps`. Необходимая команда:

```
(%i7) draw2d(terminal=eps, file_name="draw_2",grid=true,
           xlabel = "x",ylabel = "y",explicit(exp(u),u,-2,2),
           explicit(exp(-u),u,-2,2))$
```

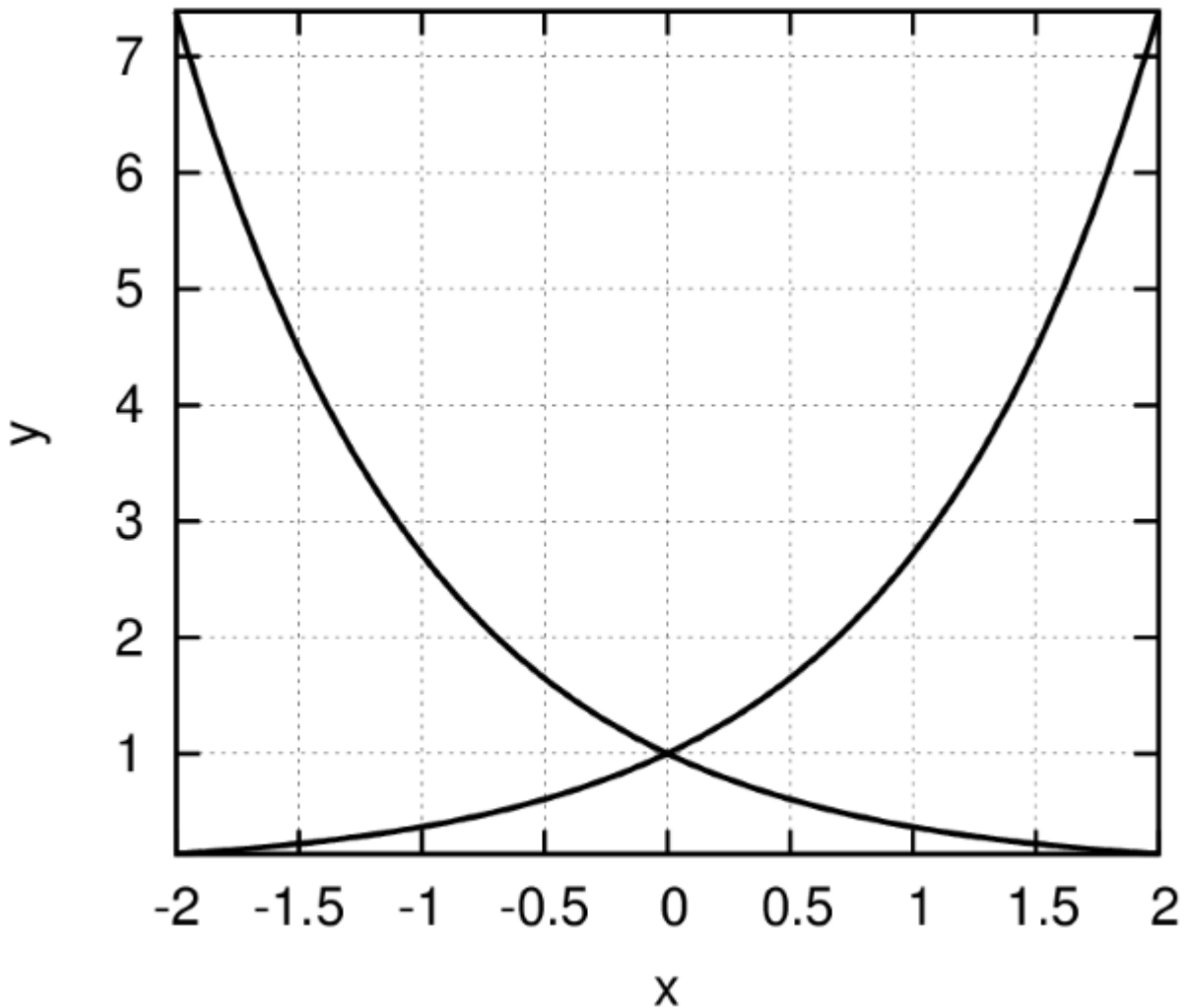


Рис. 5.16. График двух функций (использована функция draw2d

С помощью пакета draw можно строить и графики функций, заданных неявно. В этом случае функция задаётся командой implicit, например (результат построения — на [рис. 5.17](#)):

```
(%i1) load(draw)$
(%i2) draw2d(grid = true, title = "Two implicit functions",
line_type = solid, key = "y^2=x^3-2*x+1",
implicit(y^2=x^3-2*x+1, x, -4,4, y, -4,4),
line_type = dots, key = "x^3+y^3 = 3*x*y^2-x-1",
implicit(x^3+y^3 = 3*x*y^2-x-1, x,-4,4, y,-4,4))$
```

На графике хорошо видно, что кривые проведены разными линиями (одна сплошная, другая точечная). Для указания типа линии использована опция line_type=тип линии (возможные значения — solid и dots).

Помимо графиков неявных функций, при помощи draw могут быть построены и графики параметрических функций или функций, заданных в полярных координатах. В этих случаях вместо команд explicit или implicit используются команды parametric и polar соответственно. Пример графика функции в полярных координатах — на рис. 5.18. Соответствующая команда:

```
draw2d(user_preamble = "set grid polar", nticks = 200,
xrange = [-5,5], yrange = [-5,5], color = blue, line_width = 3,
title = "Hyperbolic Spiral", polar(10/theta,theta,1,10*pi))$
```

Two implicit functions

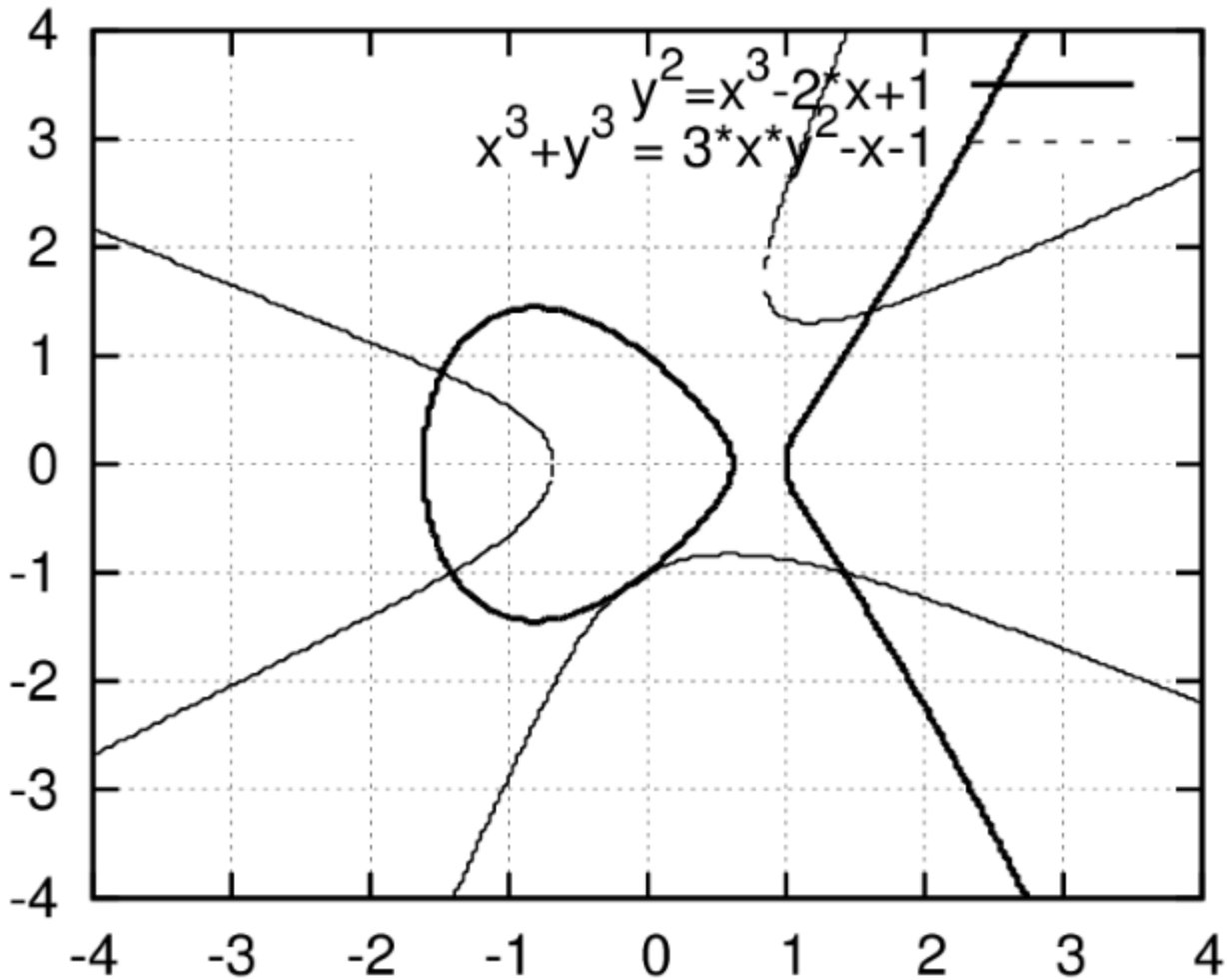


Рис. 5.17. График двух функций, заданных неявно

В последнем примере указаны параметры построения графика: интервалы изменения x и y , равные $xrange$ и $yrange$, толщина линии $line_width$ и её цвет $color$. Кроме того, важным параметром является опция $user_preamble$. Эта опция указывает команды `gnuplot`, определённые пользователем и выполняющиеся перед построением данного графика.

Для использования меток осей и заголовков на русском языке необходимо в $user_preamble$ или в специальном файле `.gnuplot` (этот файл содержит команды `gnuplot`, выполняющиеся при старте программы) указать русскую кодировку командой `set encoding koï8r` или украинскую кодировку `set encoding koï8u`. Кроме того, часто оказывается необходимым указать и шрифт для вывода заголовка или меток.

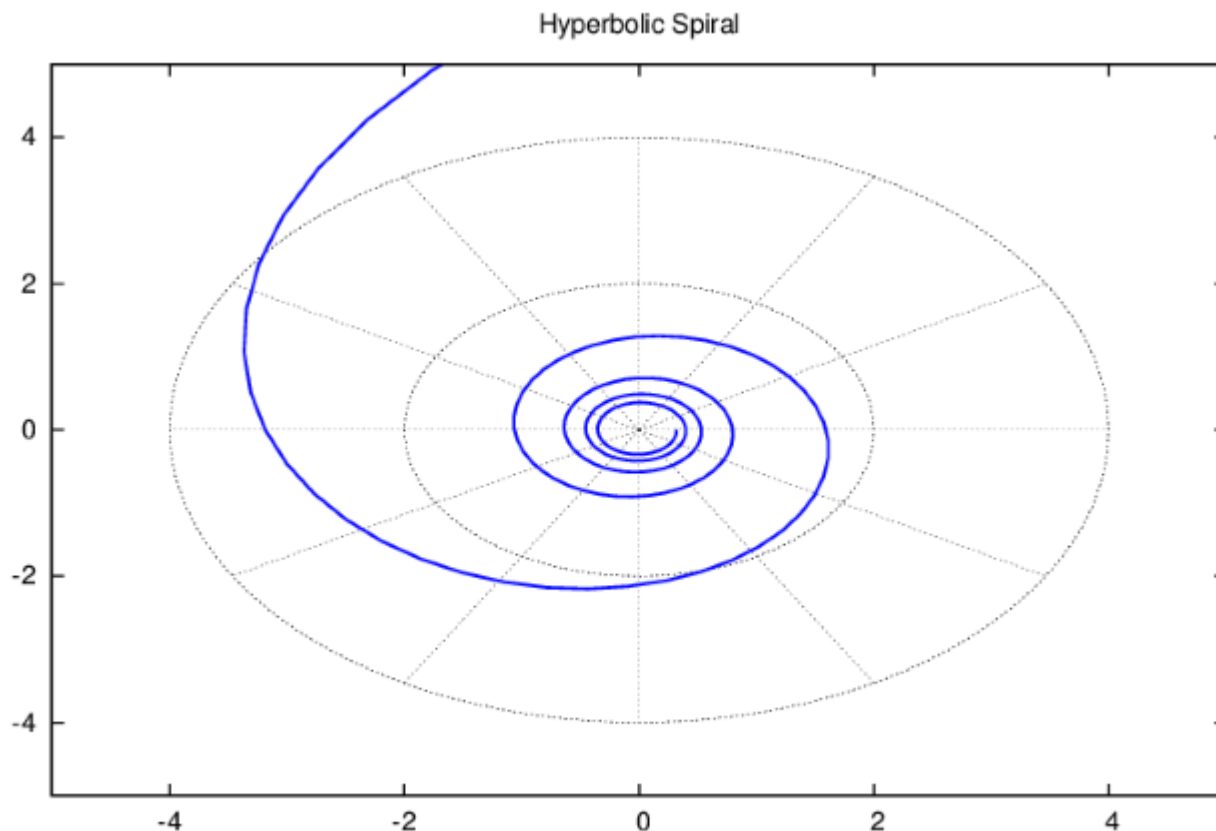


Рис. 5.18. График функции в полярных координатах

Функция *draw3d* позволяет строить трёхмерные графики. **При-** мер:

```
(%i8) draw3d(zlabel = "Z variable", ylabel = "Y variable",
explicit(sin(x^2+y^2), x, -2, 2, y, -2, 2), xlabel="X variable")$
```

Метка оси *z* указывается командой `zlabel=имя`. Вывод графика на печать аналогичен указанному выше. Пример (с указанием, помимо меток осей, и названия графика командой `title=имя`) приведен на рис. 5.19.

Очевидно, что при помощи функции *draw3d*, можно строить и окрашенные поверхности (либо полутоновые). Для этого в качестве аргумента функции *draw3d* указывается опции *enhanced3d* (указывает на построение трёхмерной окрашенной поверхности) и *palette* (`palette=color` — цветная поверхность, `palette=gray` — оттенки серого). Пример поверхности, окрашенной оттенками серого — на рис. 5.20. Необходимая команда:

```
(%i12) draw3d(terminal=eps, surface_hide = true, enhanced3d = true,
palette=gray, explicit(20*exp(-x^2-y^2)-10, x, -3, 3, y, -3, 3))$
```

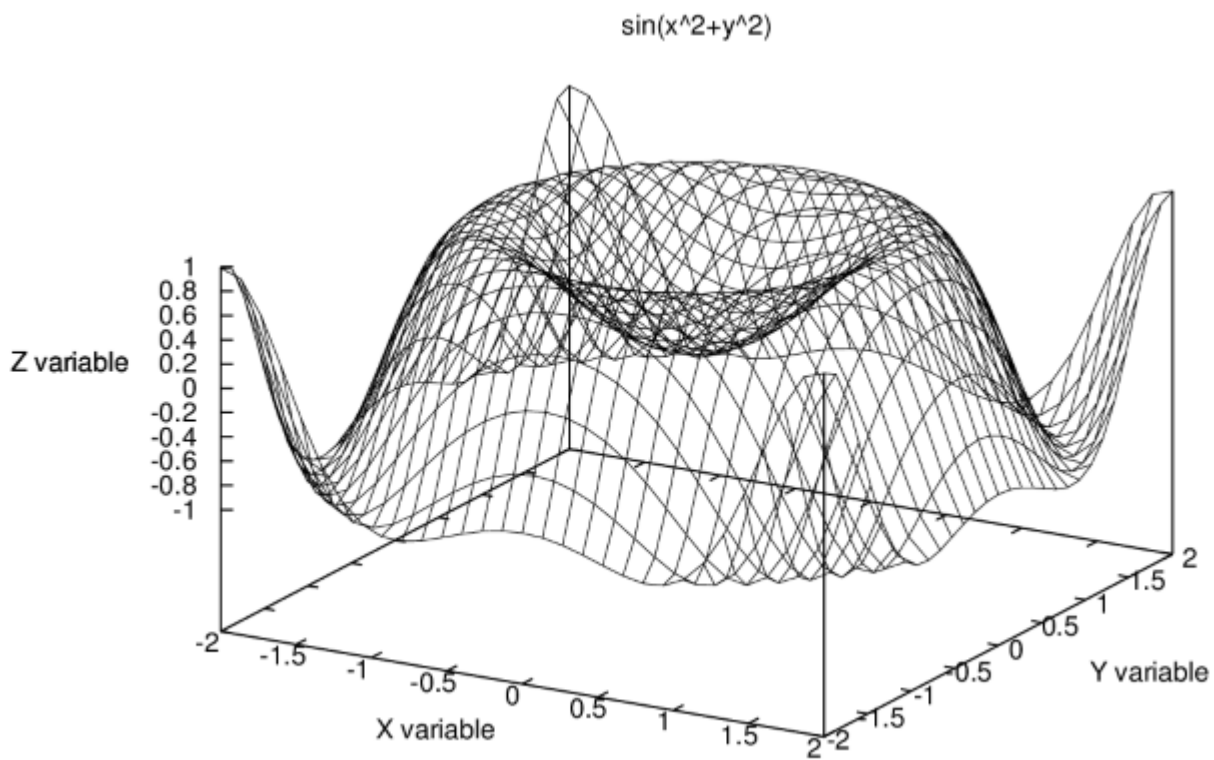


Рис. 5.19. Поверхность, построенная с помощью функции draw3d

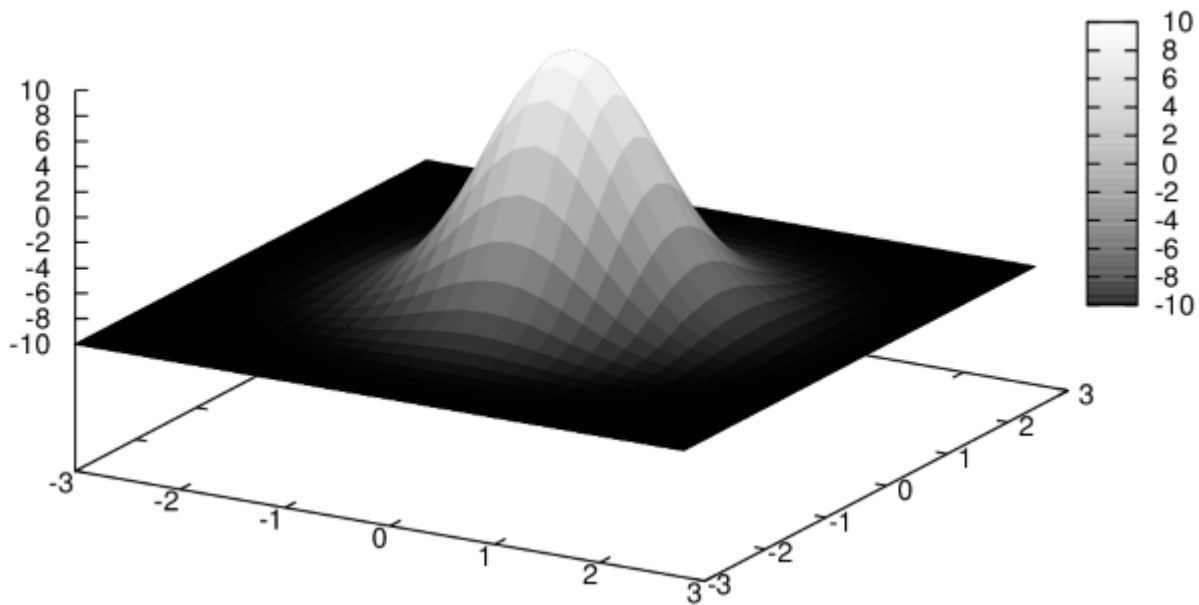


Рис. 5.20. Окрашенная поверхность (использована функция draw3d).

Пакет draw позволяет и строить несколько графиков на одном рисунке, а также предоставляет ряд других полезных возможностей, но для их использования необходимо ознакомиться с документацией, поставляемой с пакетом **Maxima**.