

# **Использование системы wxMaxima для расчета лабораторных работ по физике.**

## Оглавление

Установка и запуск.....	3
Простейшие расчеты и ввод данных.....	4
Функции в Matha.....	8
Циклическая обработка данных.....	10
Сохранение и печать результатов.....	12
Построение дискретных графиков.....	13
Метод наименьших квадратов.....	14
Matha без интерфейса.....	15
Конец.....	16

## Установка и запуск

Программа wxMaxima является графическим интерфейсом к программе Maxima. На данный момент обе программы присутствуют в большинстве дистрибутивов GNU/Linux в качестве опционального ПО. В Ubuntu/Debian дистрибутивах установка может быть произведена следующей командой:

```
sudo apt-get install wxmaxima
```

*Примечание 1: А про то, как устанавливать программу мерзким виндузятникам, я не расскажу.*

*Примечание 2: Обратите внимание на юниксовую концепцию разделения на интерфейс и функционал. В тексте руководства я буду говорить о программах Maxima и wxMaxima, подразумевая разные части одного программного комплекса.*

После этого программа появится в разделе «Обучение» меню вашей ОС. Если она по каким-то причинам не появилась, значит такова ваша судьба. В таком случае я рекомендую вам запускать ее через диалоги запуска, которые в большинстве дистрибутивов вынесены на сочетание клавиш alt-f2. Лично я предпочитаю инициировать этот диалог с панели. А вообще, вы можете расположить на панели любое приложение, которое бы вы хотели, хоть ту же максиму.



Рисунок 1: Диалог запуска на панели.

После запуска программы вы попадете в ее главное окно, которое выглядит следующим образом:

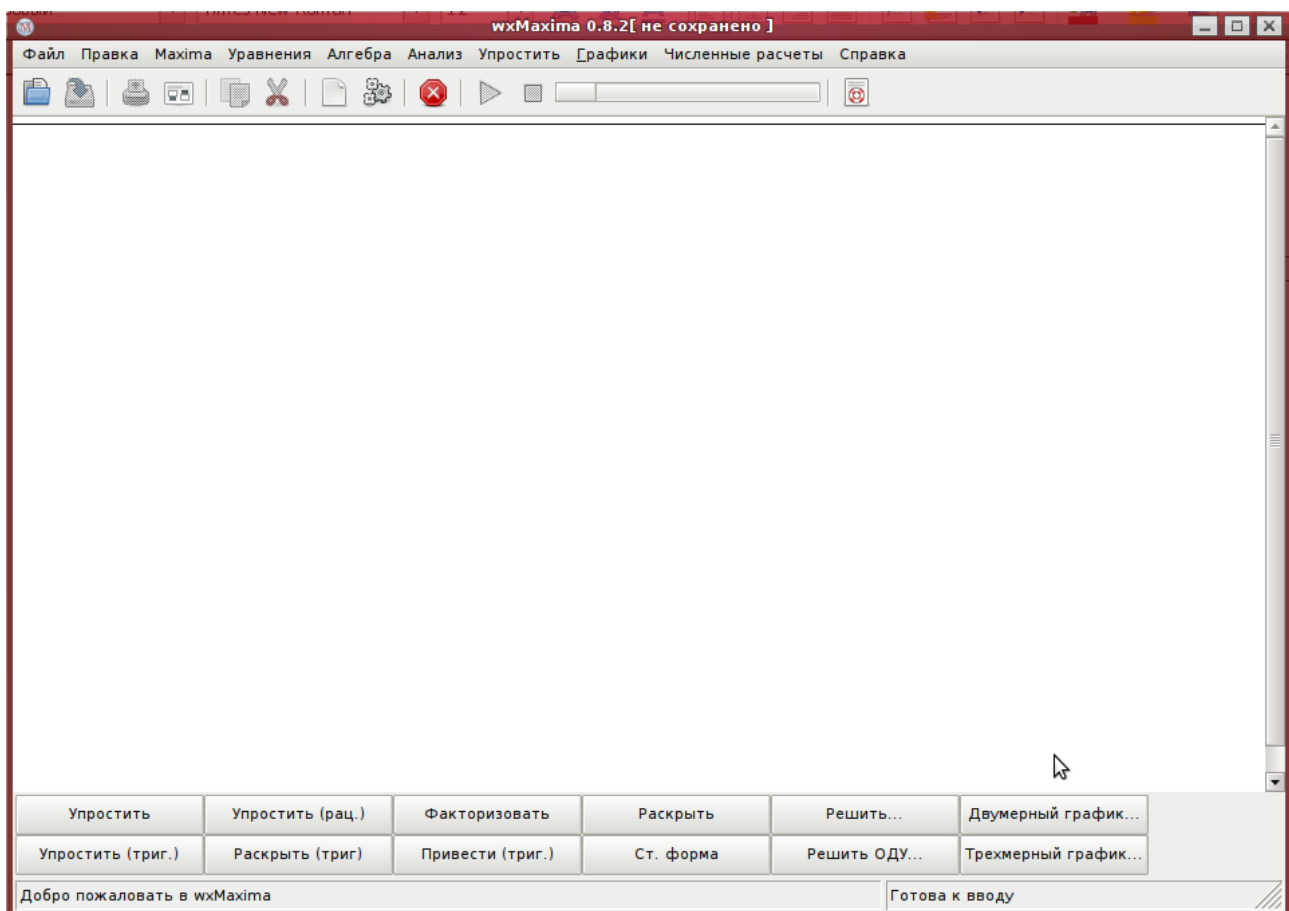


Рисунок 2: Главное окно wxMaxima

Если это окно открылось, то вы прошли самую сложную часть руководства, и можете двигаться далее.

## Простейшие расчеты и ввод данных.

На этом этапе вы уже можете выбросить свой карманный калькулятор, потому что wxMaxima может его заменить. На последующем, сможете выбросить и инженерный. Но пока оставьте, для того чтобы проконтролировать соответствие максимумских алгоритмов правилам математики.

Для начала следует разобраться, как сюда что-то вводить. Нажав на Enter вы обнаружите, что перед вами возникла неведомая фигура, обозначающая здесь приглашение командной строки. Попробуйте ввести какое-нибудь выражение, а затем нажать на Ctrl-Enter, прямо как в аське. И, о чудо, wxMaxima выдает вам ответ!

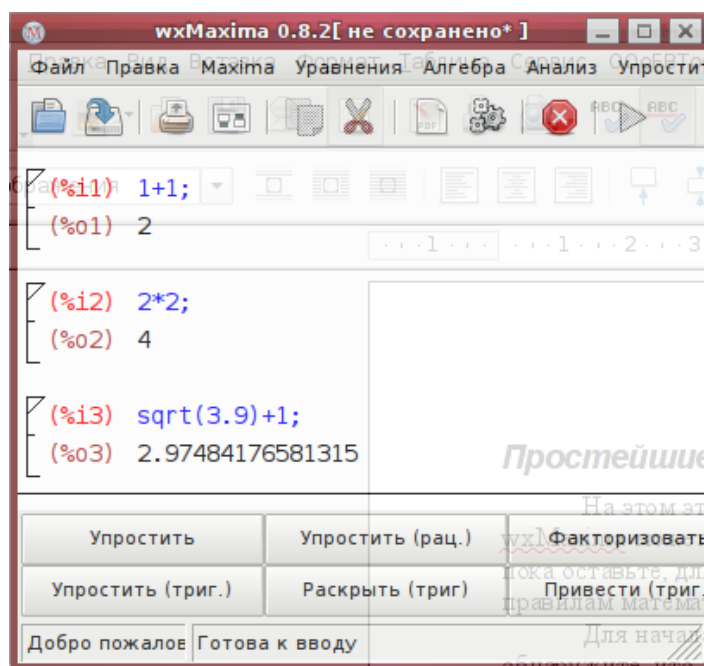


Рисунок 3: Простейшие расчеты в wxMaxima

Если в аське вы привыкли отправлять сообщения по Enter, а не по Ctrl-Enter, то и такую возможность предоставляют настройки программы. Вам необходимо зайти в меню Правка → Настройки и там, на вкладке «Опции» вы увидите последним в списке поле для галочки «Enter evaluates cells». Именно она отвечает за то, чтобы сообщения для максимы отправлялись по Enter.

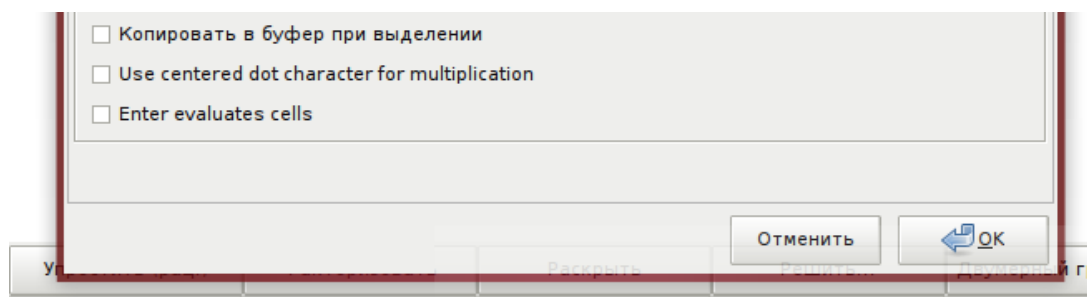


Рисунок 4: Настройка отправки по Enter

Если вы хотите обозначить какое-то значение для более удобного оперирования, вы можете присвоить ему имя. Проще говоря, вы можете задать константу, которую максима будет затем подставлять в выражения автоматически. Делается это следующей не очевидной командой:

```
pi: 3.14
```

Знак ':' является «оператором» присваивания, а левая и правая часть выражения, как можно

догадаться, являются именем и значением константы соответственно. Теперь константу  $\pi$  можно использовать в выражениях.

Что касается числа  $\pi$ , то оно доступно в *maxima* под именем `%pi`. Знак процента означает, что константа служебная. Еще одним примером такой константы может служить мнимая единица - `%i`. Проверьте, если вы возведете `%i` в квадрат, то получите минус единицу.

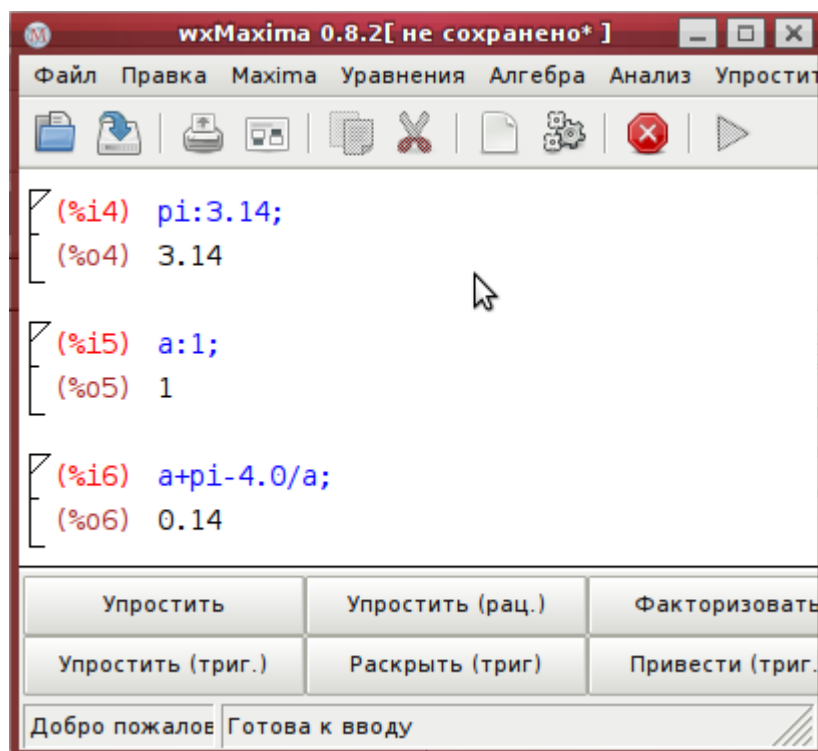


Рисунок 5: Использование констант в максиме.

При расчете лабораторных зачастую приходится использовать массивы данных. Сейчас я расскажу вам, как их вводить. Как их использовать можно будет прочесть далее. Для введения массива можно воспользоваться следующей конструкцией:

```
a: [1,2,3]
```

Вообще говоря, называть эту структуру массивом некорректно. В *maxima* эта структура называется списком, а массив это несколько другое. Но такой терминологией удобнее оперировать, и я буду выражаться то так, то правильно. Доступ к элементам списка напоминает доступ к элементам массива в C-подобных языках программирования:

```
a[1]  
a[3]
```

Как уже поняли некоторые читатели, работа в максиме напоминает программирование на каком-нибудь интерпретируемом языке, и тем, кто ознакомлен с таким программированием будет легче в ней работать. Я постараюсь писать это пособие таким образом, чтобы остальным тоже было понятно. Если же у меня это не получится, переделайте мое руководство и выложите в интернет, ведь оно под лицензией Creative Commons!

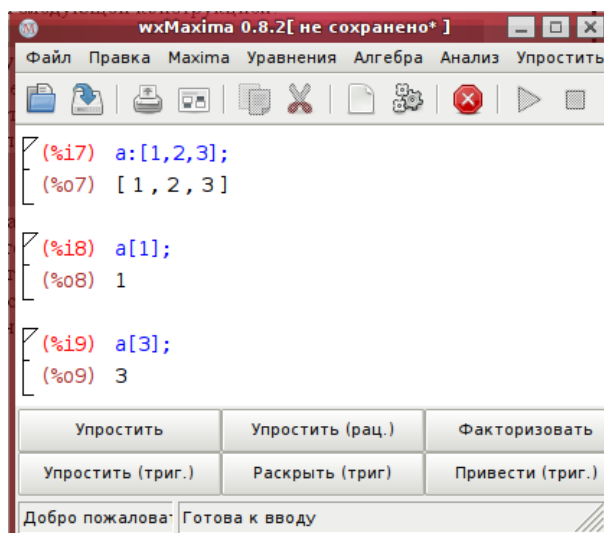


Рисунок 6: Использование списков.

И последнее, о чем я хочу вам рассказать в разделе о данных, это флаг численных вычислений (да-да, именно так тавтологично он здесь и называется :). Этот флаг влияет на представление чисел, которыми вы оперируете. Грубо говоря, его значение по умолчанию означает, что целые числа, которыми вы оперируете, при делении друг на друга предстанут перед вами не в виде иррационального числа с десятичной точкой, а в виде дроби. Если вы его переключите, то все ваши числа будут как будто бы дробными.

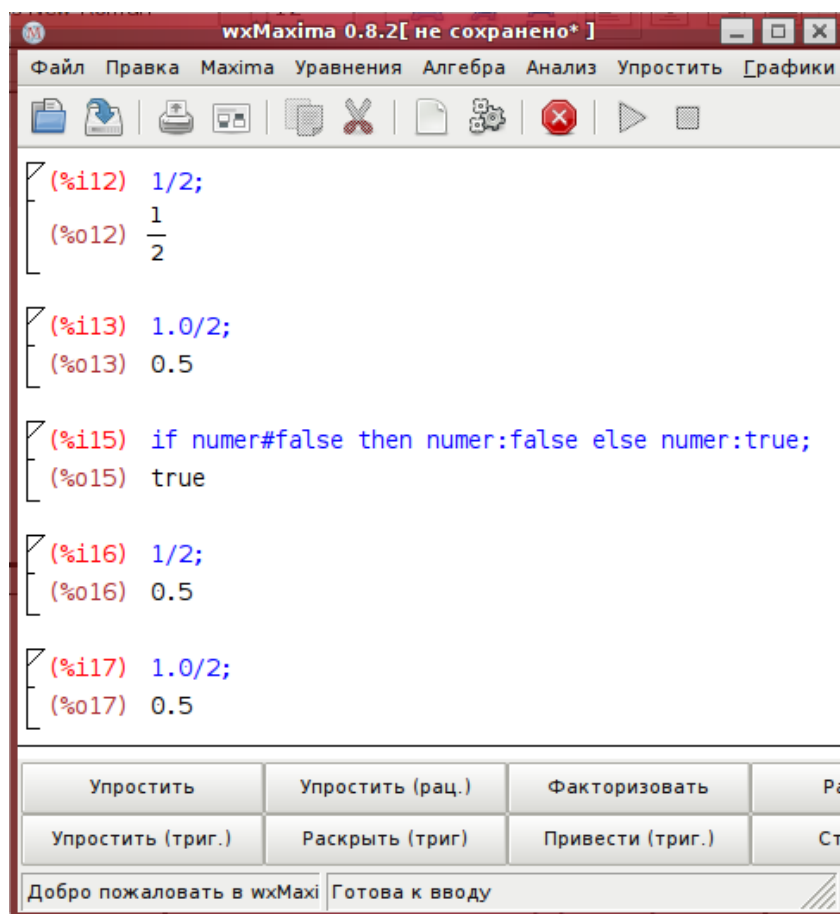
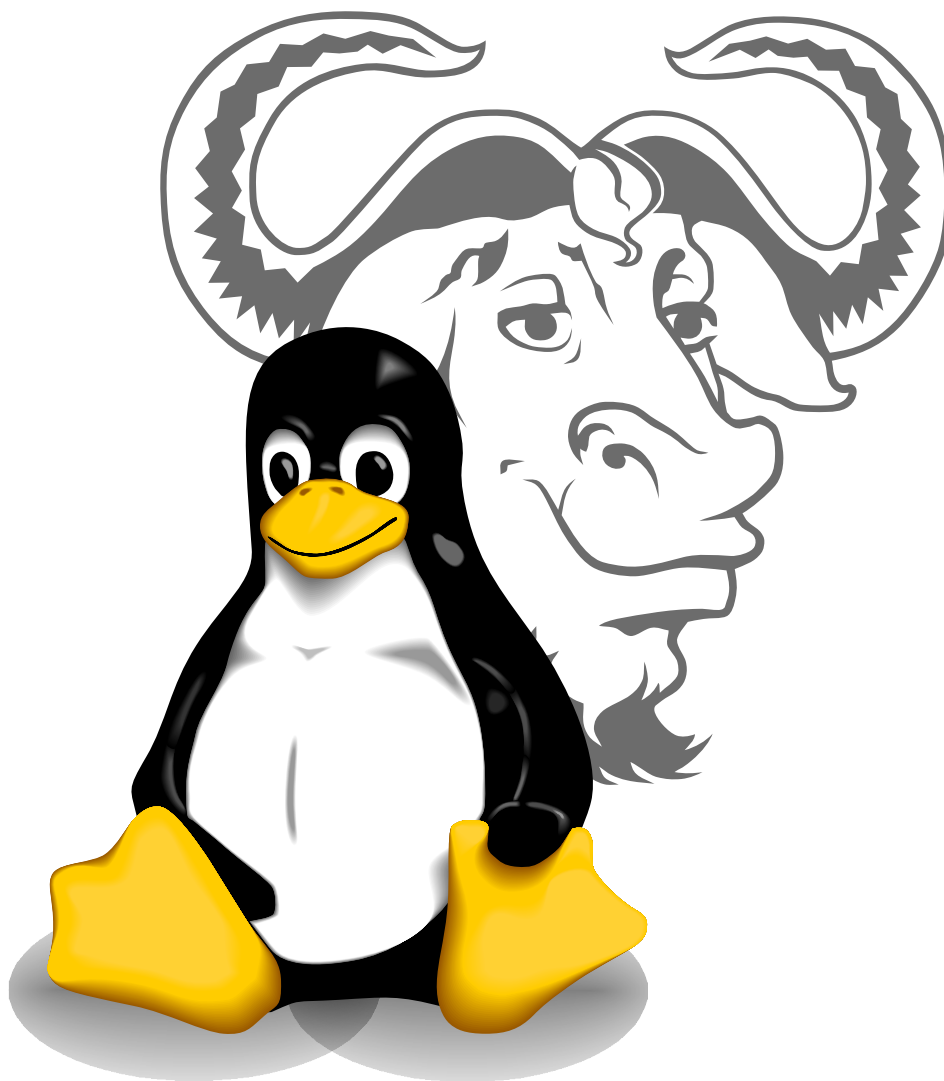


Рисунок 7: Влияние флага численных вычислений.

Вот и все. Теперь у вас есть отличный калькулятор для выражений любой сложности. Если вы хотите, чтобы максима для вас стала чем-то большим, то не переключайте канал, после рекламы вас ждет продолжение!



## **Используйте свободные операционные системы класса GNU/Linux!**

GNU/Linux это:

- Свобода настройки и модификации на любом уровне.
- Надежность и стабильность, пока руки не зачесутся.\*
- Лучший антивирус.
- Отличный способ повысить свою компьютерную грамотность.
- Способ не платить за ПО не нарушая закона.
- Как бы одна и та же операционная система, и много разных, одновременно.

Сделайте правильный выбор сейчас! GNU/Linux — твоя эволюция.

\* (А когда зачесутся, никакой мастдай не спасет :))

## Функции в Maxima.

В Максима реализована возможность задания математических функций. Но я начну с того, что расскажу о встроенных функциях. Как правило, эти функции записываются аналогично математике.

```
sin(3.14)
```

Только вот математика эта американская, а не отечественная. Поэтому привычные нам со школы  $\text{tg}$  следует заменять на  $\text{tan}$ . Вот список тех функций, которые я смог обнаружить самостоятельно:

Функция в Maxima	Функция в математике
$\sin(x)$	Синус.
$\cos(x)$	Косинус.
$\tan(x)$	Тангенс.
$\sinh(x)$	Гиперболический синус.
$\cosh(x)$	Гиперболический косинус.
$\tanh(x)$	Гиперболический тангенс.
$\log(x)$	Натуральный логарифм.
$\text{abs}(x)$	Модуль.
$\text{atan}(x)$	Арктангенс
$\text{asin}(x)$	Арксинус
$\text{acos}(x)$	Арккосинус

*Примечание: А вы знали, что если взять арксинус/арккосинус от числа больше 1, то у вас получится комплексное значение?*

Наверняка встроенных функций куда больше. Если вам еще что-то нужно, то попробуйте обратиться к руководству по wxMaxima/Maxima. Там много интересного для начинающего любителя численных вычислений.

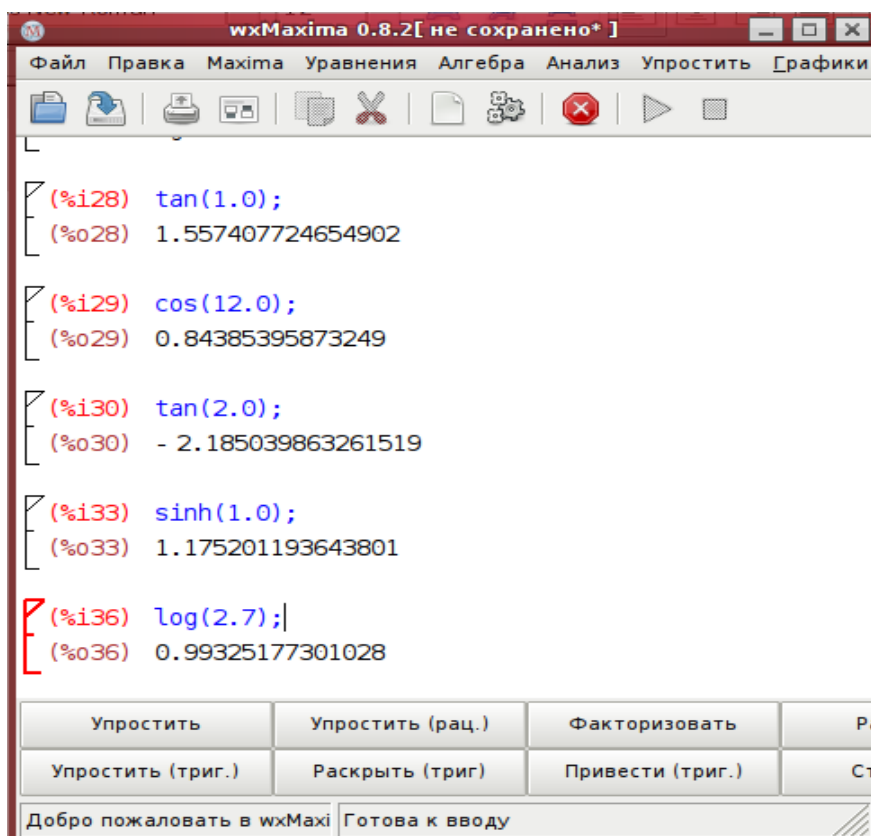


Рисунок 8: Встроенные функции в Maxima.



Если вы введете функцию, которая программе не известна, то она покажет вам точно такую же строку, как вы ввели. Но будьте внимательны! Если у вас установлен флаг numeric в дефолтное положение, то она поведет себя точно так же и с заданной функцией. Так что если вы намерены все же получить свой ответ, то переключите флаг, или передавайте функции вещественный параметр.

Теперь о том, как задавать собственные функции. Как и в математике, функция может быть определена выражением. Чтобы задать функцию, вы должны воспользоваться следующим оператором:

```
f(x) := x/2
```

После определения вы можете использовать ее также, как и встроенные функции:

```
f(3)
```

У функции также может быть несколько параметров, которые задаются и передаются через запятую. Пример вы можете увидеть на следующем скриншоте.

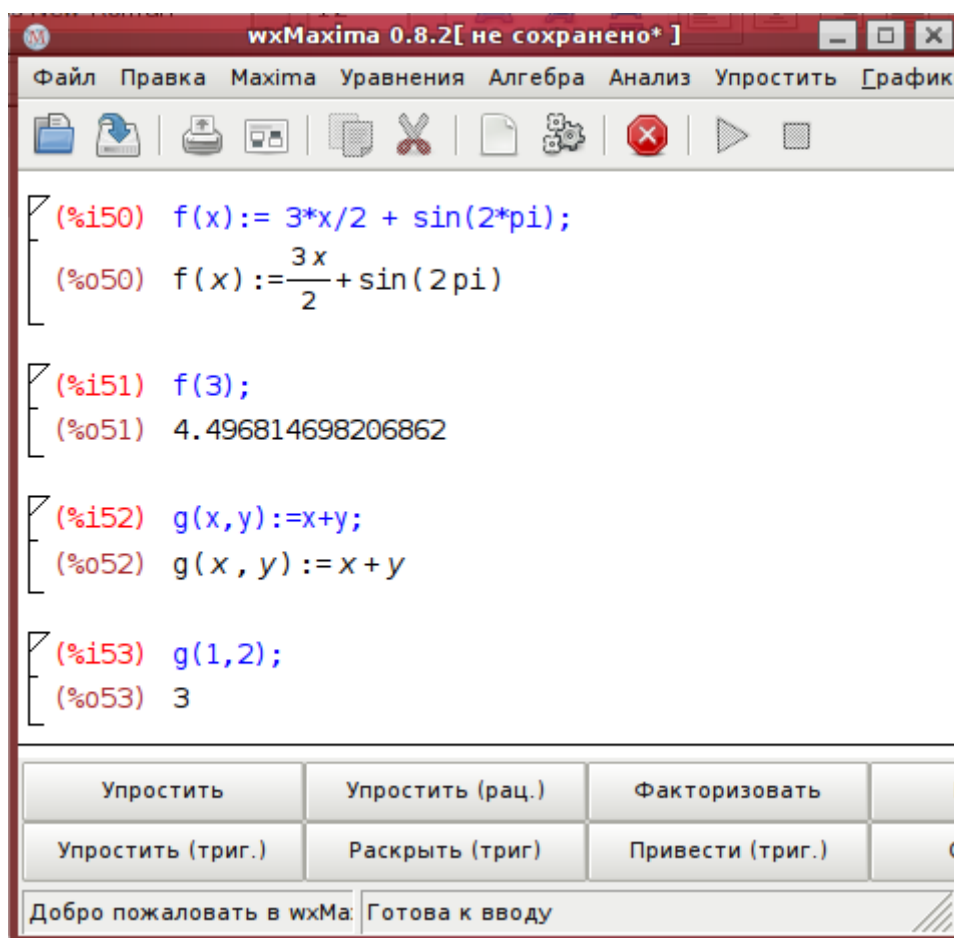


Рисунок 9: Собственные функции в wxMaxima

Как вы видите, ничего сложного. Используйте функции для упрощения ваших расчетов. Если вы последовательно посчитаете свои данные используя функцию, то у вас уже получится красивая табличка.

## Циклическая обработка данных.

Это пожалуй самая сложная часть руководства, потому что она использует малопонятные непрограммистам циклы. Но если вы будете аккуратно вводить команды и не ошибаться, то все будет пучком.

Допустим, у вас имеется список  $A$ , который вы ввели по правилам, приведенным в разделе о вводе. Допустим, в нем находятся амплитудные значения тока. Тогда, чтобы получить действительные значения, вам необходимо каждое из них поделить на  $\sqrt{2}$ .

```
for I in A do ldisp(I/sqrt(2))
```

По порядку. Здесь `for` — это ключевое слово, обозначающее цикл.  $I$  это временная переменная, которая соответствует одному из элементов списка.  $A$  это массив, который уже был введен вами ранее. Ключевое слово `do` говорит максиме что надо делать, проходя массив. Проход массива осуществляется поочередно, то есть действие после `do` выполняется столько раз, сколько находится элементов в массиве, а переменная  $I$  принимает на каждой итерации(итерация — одно выполнение цикла) значения  $a[1], a[2], \dots, a[n]$ . Далее идет хитрая функция `ldisp`, которая позволяет нам увидеть, что она там такого насчитала. А параметром этой функции служит выражение. Если вы ничего не поняли, то не вдавайтесь в эти описания. А просто попробуйте сами дать эту команду, и поизменять ее параметры.

*Примечание: Имена переменных и других объектов в максима регистрозависимы. Это значит, что  $I$  и  $i$  это две разные переменные.*

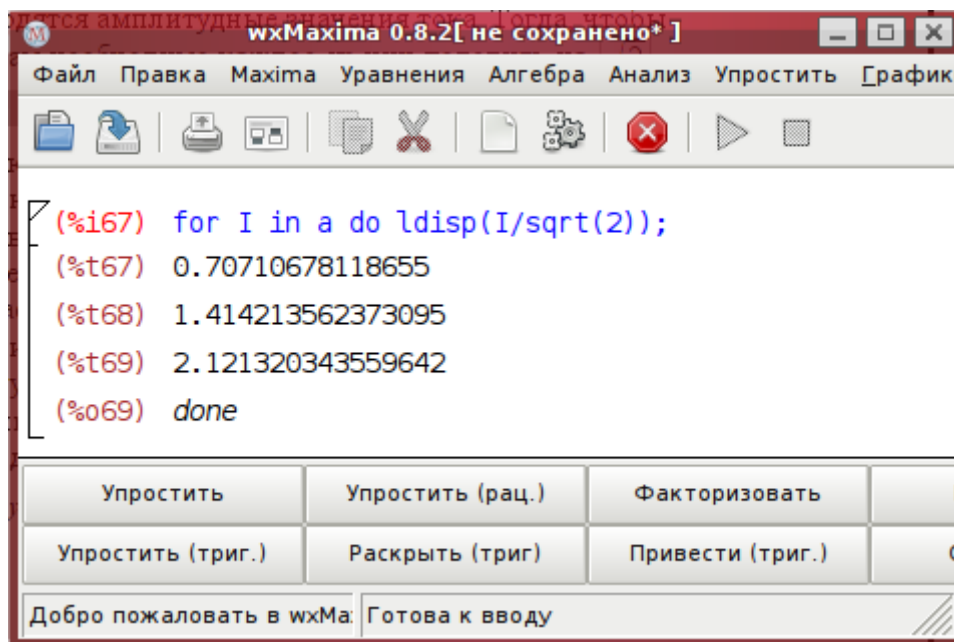


Рисунок 10: Команда циклической обработки.

Этот способ подходит вам, если вы хотите посмотреть расчетные значения и затем записать их куда-нибудь к себе в лабник. А если вам требуется произвести над ними какие-то расчеты, то очевидно, их надо представить в удобном для этого виде. Например можно занести данные в список, аналогичный исходному. Для начала, нужно создать пустой список, чтобы потом в него можно было добавлять данные циклично. Это достигается командой:

```
b: []
```

Теперь можно заполнять:

```
for i in a do b:append(b, [i/sqrt(2.0)])
```

Этот цикл похож на предыдущий, но в него потребовалось внести некоторые

изменения, адекватно задаче. Теперь каждую итерацию цикла, в нем происходит переприсвоение списка `b`. Его присваивается список, составленный функцией `append` из прошлого состава списка `b` и еще одного списка, в котором находится лишь одно значение, нового рассчитанного элемента. Результатом является заполненный список `b`, которым вы можете оперировать, как будто бы сами ввели его. Чтобы посмотреть, что в нем есть, просто введите команду

```
b
```

Вы увидите ваш список.

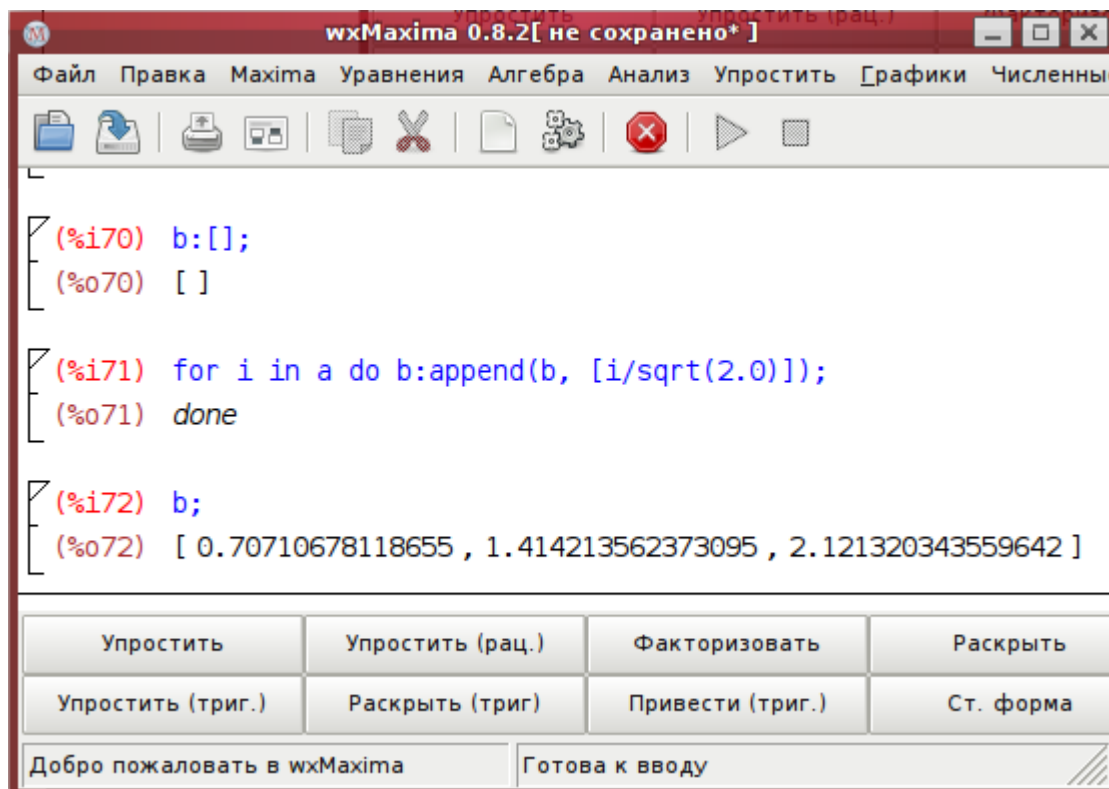


Рисунок 11: Расчет с сохранением результатов.

Всего сказанного уже достаточно, чтобы посчитать обычную лабораторную работу. Особенно, если вы все это прочитали. Последовательность действий ваша должна быть примерно такой:

1. Ввести список исходных экспериментальных значений.
2. Задать функции для расчета значений.
3. Дать команды на циклический обсчет списков.
4. Занести данные в ваш лабник.
5. Закрывать лабник, и идти пить пиво на паркет, или куда-нибудь еще.

А теперь я расскажу о некоторых дополнительных фишках, которые могут вам помочь при подготовке к сдаче работы.

## Сохранение и печать результатов.

Часто бывает так, что некоторые нехорошие преподаватели требуют наличия расчетов в вашем лабораторном журнале. Путь решения проблемы несколько. Самым наверно симпатичным путем будет пожалуй функция печати в wxMaxima. Она незамысловатым способом распечатает для вас содержимое всего листа с вашими расчетами.

Другим способом является экспорт в HTML или LaTeX. Выбор конкретного формата зависит от ваших предпочтений-извращений, но для большинства пользователей подойдет HTML. Получившийся файл можно распечатать из браузера.

Третьим способом, который в ранних версиях wxMaxima работал хорошо, а сейчас работает так себе, это экспорт выделения в изображение. Выделите несколько расчетных полей и в меню «Правка» выберите пункт «Save selection as image». В результате вы получите изображение с дурацким форматированием, но легко встраиваемое в другие документы. Вроде этого:

```
(%i7) sqrt(3.9)+1;  
(%o7)  
2.97484176581315  
(%i6) pi:3.14;  
(%o6)  
3.14  
(%i1) a:1;  
(%o1)  
1
```

*Рисунок 12: Экспорт из wxMaxima.*

Этот способ хорош тем, что сразу позволяет пользователю выбрать, что печатать, а что нет. Но как вы могли заметить, если аккуратно вести расчеты и удалять ошибочные поля из листа, то выбирать совсем ничего не нужно — вы напечатаете все.

Программа wxMaxima предусматривает возможность сохранения листа расчетов в файл. Этот файл имеет расширение .wxm и представляет собой список всех команд, находившихся в листе на момент сохранения. В более ранних версиях эти команды автоматически выполнялись, а в текущей на момент написания версии 0.8.2 не выполняются, вероятно в целях экономии процессорного времени, потому что файлы уже даже среднего размера открывались весьма долго. Чтобы выполнить нужную вам группу команд, выделите их(выделяются их «корешки») и нажмите ctrl-enter (или enter). Сохранение и открытие таких документов осуществляется стандартными способами через меню — Файл → Save As и Файл → Open.

## Построение дискретных графиков.

Другой возможностью wxMaxima, которая вам может пригодиться, является возможность построения дискретных графиков. Она не избавит вас от муторной процедуры построения графиков вручную, но возможно поможет проверить себя в этой или в других задачах. Командой, которая строит дискретный график является:

```
wxplot2d([[ 'discrete, X, Y]], [x,-5,5])$
```

Эта команда нарисует для вас график, построенный по точкам из списков X и Y для осей абсцисс и ординат соответственно. Если вы сейчас работаете в том же листе, в котором работали при расчете в прошлых разделах, то у вас сохранились значения списков a и b, которые вы могли бы подставить. Вторым аргументом этой команды является диапазон значений x. Для дискретных графиков это всего лишь минимальное и максимальное значения X, которые могли бы быть нанесены на график. В конце команды находится знак '\$'. Он отвечает за настройку способа вывода в maxima, но если вы его забудете, то конец света не случится, и даже график нарисует.

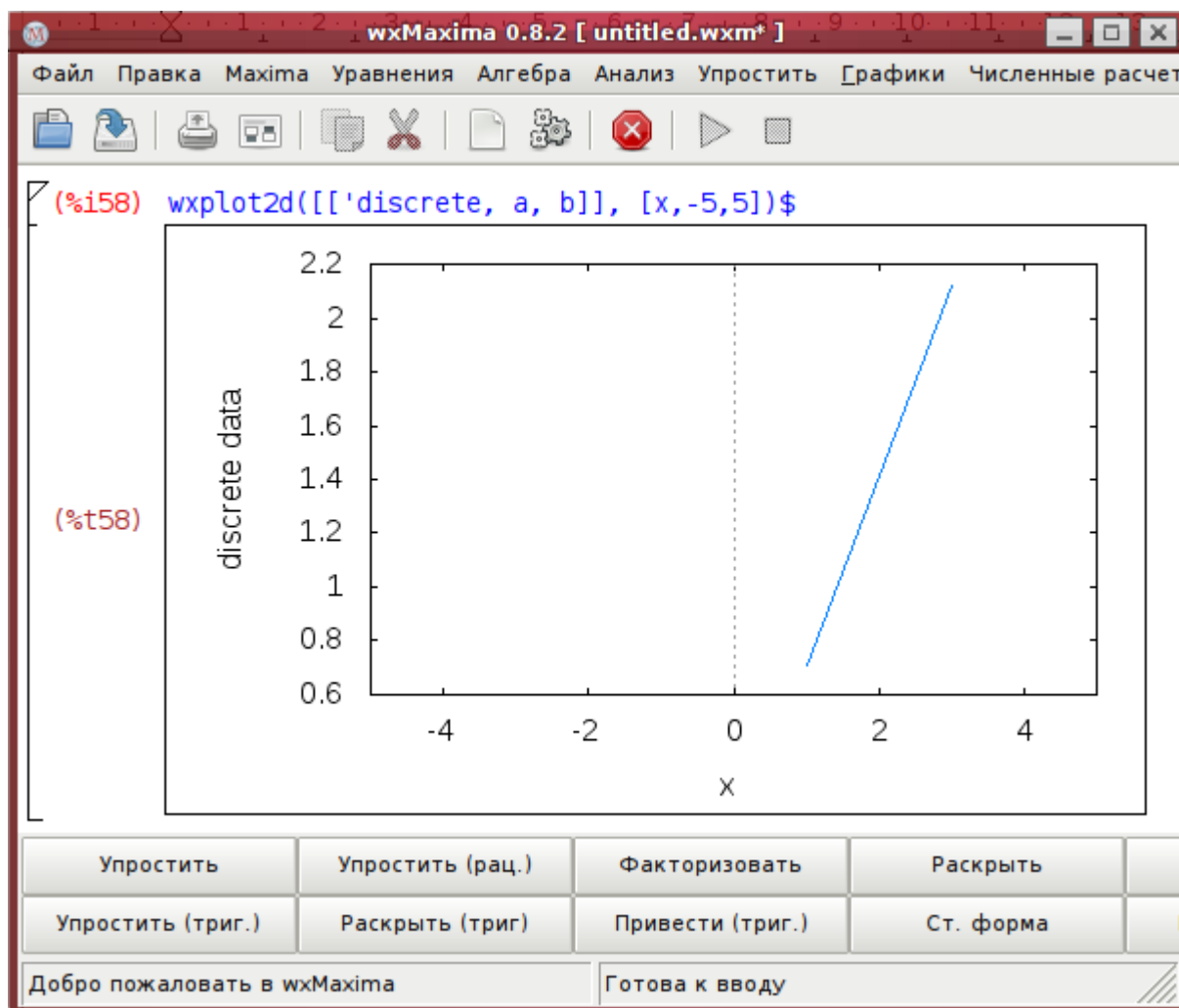


Рисунок 13: Дискретный график в wxMaxima.

## Метод наименьших квадратов.

Для расчета коэффициента наклона прямой по графикам в курсе физпрактикума нас учат пользоваться методом наименьших квадратов. Метод простой, но он требует множества расчетов. Естественно, что такая задача должна быть автоматизирована. Для этого я написал расширение Maxima, которое вы найдете где-нибудь поблизости с этим текстом. Файл называется `mnk.mac`. Чтобы подключить его к Maxima, нужно в меню Файл выбрать пункт «Load package» и указать имя файла. После этого вы сможете вызывать функцию MNK, примерно следующим образом:

```
MNK(x,y)$
```

Параметры  $x$  и  $y$  это списки, в которых находятся ваши экспериментальные данные графиков.

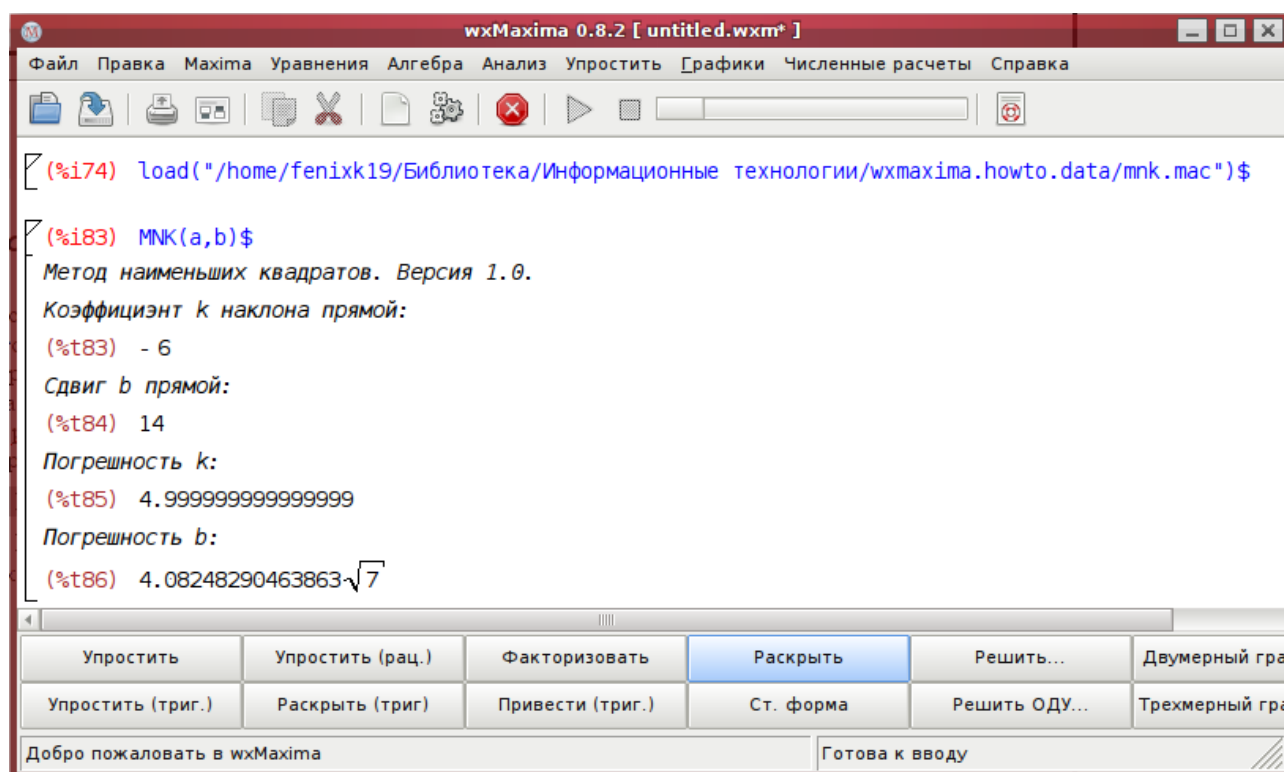
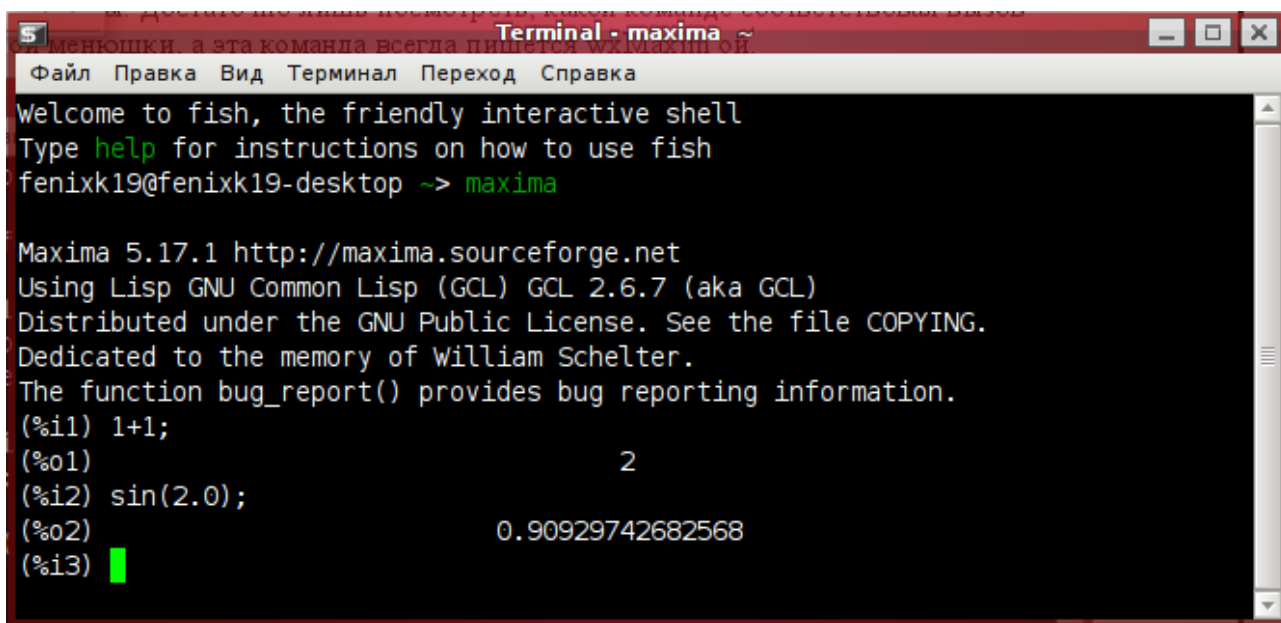


Рисунок 14: Использование расширения MNK

## Maxima без интерфейса.

Если у вас вдруг сломался X сервер, или еще какая-то проблема с графическим интерфейсом, то с Maxima всегда можно работать из командной строки, она умеет все практически то же самое, разве что кроме отрисовки графиков. Запустите в терминале команду `maxima`, и вы увидите уже знакомое вам приглашение ввода. В отличие от `wxMaxima`, концом команды здесь считаются символы ';' или '\$', а не факт нажатия на [ctrl-j]enter.

Некоторые действия, которые вам было привычно запускать из менюшек, здесь тоже наверняка доступны. Достаточно лишь посмотреть, какой команде соответствовал вызов определенной менюшки, а эта команда всегда пишется `wxMaxim`'ой.



```
Terminal - maxima
Файл Правка Вид Терминал Переход Справка
Welcome to fish, the friendly interactive shell
Type help for instructions on how to use fish
fenixk19@fenixk19-desktop -> maxima

Maxima 5.17.1 http://maxima.sourceforge.net
Using Lisp GNU Common Lisp (GCL) GCL 2.6.7 (aka GCL)
Distributed under the GNU Public License. See the file COPYING.
Dedicated to the memory of William Schelter.
The function bug_report() provides bug reporting information.
(%i1) 1+1;
(%o1)
      2
(%i2) sin(2.0);
(%o2)
      0.90929742682568
(%i3) █
```

Рисунок 15: Maxima. Вид из терминала.

*Конец.*

Вот я и написал все, что вам могло бы пригодиться при расчете лаб. Всей этой технологии есть куда расти, также как и этому руководству. И здесь есть огромный плюс — все описанное программное обеспечение — свободное, а лицензия этого руководства — Creative Commons. Это значит, что его можно изменять, дополнять, а затем распространять далее на тех же условиях. Если вы как-то решите изменить то, что я написал, дайте мне знать — мой JID: [senya@jabber.org](mailto:senya@jabber.org). Если у вас возникнут какие-то вопросы по wxMaxima, у меня могут случайно оказаться на них ответы, по этому тоже не стесняйтесь задавать.

*Свободу образованию!*

*С вами был Александр Варнин, учебная группа А3-06. До встречи.*

*08.12.09. Пятый час утра.*