Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования

Томский государственный университет
систем управления и радиоэлектроники (ТУСУР)

Г.Н. Нариманова, Р.К. Нариманов, Ю.Н. Рыжих

Основы работы в пакете Maxima

Учебное пособие

Томск
2024

УДК 519.6+004.9(075)

ББК 22.19

***Рецензенты*:**

Бикмуллин Э.А., технический директор ООО «Ди Эй Групп»;
Борзенко Е.И., докт. физ.-мат. наук, профессор НИ Томского государственного университета

Нариманова Г.Н.

Основы работы в пакете Maxima: учеб. пособие / Г.Н. Нариманова, Р.К. Нариманов, Ю.Н. Рыжих – Томск: ТУСУР, 2024. – с. 119

Учебное пособие предназначено для студентов инженерных направлений. Рассматриваются основы работы в свободно распространяемом пакете символьных и численных вычислений Maxima. Описаны возможности решения задач из широкого спектра разделов математики таких как математический анализ, дифференциальное и интегральное исчисление, линейная алгебра, решение систем нелинейных уравнений и неравенств, решение дифференциальных уравнений, преобразование символьных выражений, необходимых для решения прикладных инженерных задач. Представлены средства визуализации результатов в виде двух- и трехмерных статических и динамических изображений.

Одобрено на заседании кафедры Управления инновациями; протокол № 3 от 5 ноября 2024 года.

УДК 519.6+004.9(075)

ББК 22.19

1 Введение. Обзор основных возможностей.

Пакет для выполнения математических вычислений Maxima является интереснейшим свободно распространяемым программным продуктом. Основной особенностью этого пакета является то, что он в первую очередь работает с символьными вычислениями. История его появления и развития начинается в 60х годах прошлого столетия, но он до сих пор не утратил своей ценности.
Свободно распространяемая система символьных и числовых вычислений Maxima является развитием коммерческого продукта Macsyma, созданного по проекту MACSYMA. Аббревиатура MACSYMA получена из начальных букв слов MAC's SYmbolic Manipulation. Слово MAC само является аббревиатурой, правда расшифровываемой двумя разными способами –Man and Computer (человек и компьютер) или Machine Aided Cognition (познание с помощью компьютера).
Пакет Максима активно развивается и позволяет выполнять сложейшие вычисления при решении разнообразных научно-исследовательских, инженерных, экономических и других задач.
Версия wxMaxima является свободно распространяемой программой для выполнения математических вычислений, символьных преобразований и построения графиков.
И является по сути своей графической надстройкой над ядром Максима, помогая правильно выставлять команды.
Программа разработана профессором Техасского университета (г. Остин, США) Уильямом Шелтером (William F. Schelter). Получить полную информацию о wxMaxima и скачать ее последнюю версию можно с официального сайта http://maxima.sourceforge.net/ru/.
Помимо привычных пунктов главного меню программы {Файл, Правка, Помощь), здесь же находятся и функции для решения большого количества типовых математических задач, разделенные по группам: Уравнения, Алгебра, Анализ, Упростить, Графики, Численные расчеты.
В wxMaxima команды и результаты вычислений имеют определенное обозначение. Так, каждой команде сразу после ввода присваивается порядковый номер.

Введенная команда имеет номер 1 и обозначается соответственно (%il). Буква i является сокращением от английского слова input (ввод). Результат вычисления также имеет порядковый номер, соответственно в нашем случае (%ol). Здесь применяемая буква o выступает как сокращение от английского output (вывод). Предложенный разработчиками способ нумерации позволяет значительно упрощать вычисления. Например, вместо того чтобы снова повторять полную запись уже выполненных ранее команд, можно кратко записать: (%il)+(%i2). Это будет означать добавление к выражению первой команды выражения второй и последующего вычисления результата. Кроме того, можно использовать и номера результатов вычислений, например (%ol)*(%o2). Команда на исполнение операции сочетание Shift+Enter.
Служебные символы отмечаются знаком % впереди. Например %e - число основание натуральных логарифмов, %pi число пи.

Список основных математических функций, доступных в Maxima
sqrt квадратный корень
sin синус
cos косинус
tan тангенс
cot котангенс
sec секанс
csc косеканс
asin арксинус
acos арккосинус
atan арктангенс
acot арккотангенс
asec арксеканс
acsc арккосеканс
exp экспонента
log натуральный логарифм
sinh гиперболический синус
cosh гиперболический косинус
tanh гиперболический тангенс
asinh обратный гиперболический синус
acosh обратный гиперболический косинус
atanh обратный гиперболический тангенс
floor округление до целого с недостатком
ceiling округление до целого с избытком
fix целая часть
float преобразование к формвту с плавающей точкой
abs абсолютная величина

Список основных математических констант, доступных в Maxima
%e основание натуральных логарифмов
%i мнимая единица (√(− 1))
inf положительная бесконечность (на действитель-
ной оси)
minf отрицательная бесконечность (на действительной
оси)
infinite бесконечность (на комплексной плоскости)
% phi Золотое сечение (φ)
% pi Постоянная π - отношение длтны окружности к
её диаметру
%gamma Постоянная Эйлера (γ)
false, true логические (булевы) величины

Простые расчеты. Сложение, возведение в степень и разность

(%i1) (6+5^2−7)/(5^(1/3));

Warning: Can set maxima's working directory but cannot change it during the maxima session :

(%o1) $\frac{24}{5^{1/3}}$

Выведем результат первой команды в виде десятичной дроби

```
(%i2) %i1, numer;
(%o2) 14.03528514342176
```

Можно указать сколько цифр должно быть в числовом ответе:
Численные расчеты -> Установить отображаемую точность.
Изменение точности вывода результата на 5 цифр.

```
(%i3) fpprintprec : 5;
(%o3) 5

(%i4) %i1, numer;
(%o4) 14.035
```

Изменение точности вывода результата на 15 цифр

```
(%i5) fpprintprec : 15;
(%o5) 15

(%i6) %i1, numer;
(%o6) 14.0352851434218
```

присвоение символьной переменной a значения 4-5

```
(%i7) a:4-5;
(%o7) -1
```

проверка значения a

```
(%i8) a;
(%o8) -1

(%i9) a:66-sin(5)+log(12);
(%o9) log(12)-sin(5)+66
```

Вывод значения a в виде десятичной дроби

```
(%i10) a,numer;
(%o10) 69.4438309244511
```

Использование ранее определенных выводов

```
(%i11) (%o6)+(%o9) ;
(%o11) log(12)-sin(5)+80.0352851434218
```

```
(%i12) (%o6)+(%o9),numer;
(%o12) 83.4791160678729
```

```
(%i13) %o11,numer;
(%o13) 83.4791160678729
```

Получили одинаковое значение

```
(%i14) a+1,numer;
(%o14) 70.4438309244511
```

Сложение символьной и числовой констант

Проверка логарифма

```
(%i15) log(%e);
(%o15) 1
```

```
(%i16) a:%e^3;
(%o16) %e^3
```

```
(%i17) a,numer;
(%o17) 20.0855369231877
```

Теперь символьная переменная а имеет значение экспоненты
в кубе

Упрощение символьных выражений. Вначале смотрим вид самого выражения которое набрали.

```
(%i18) (a^2+2·a·b+b^2)/(a+b);
```

$$(\%o18)\quad \frac{b^2+2\,\%e^3\,b+\%e^6}{b+\%e^3}$$

Не получилось полностью символьное выражение, потому что а имеет числовое значение e^3.

очистка переменной а

```
(%i19) kill (a);
(%o19) done
```

Теперь при проверке набранного выражения мы получим правильное символьное представление

```
(%i20) ((a^2+2·a·b+b^2)/(a+b));
```

$$(\%o20)\ \frac{b^2+2\,a\,b+a^2}{b+a}$$

Набираем выражение, выделяем его и в главном меню выбираем:
Упростить -> Упростить выражение

```
(%i21) ratsimp(((a^2+2·a·b+b^2)/(a+b)));
(%o21) b+a
```

Появилась команда которая отвечает за упрощение выражения, а его аргументом является наше исходное выражение

Еще один пример:

```
(%i22) (c^2−d^2)/(c+d);
```

$$(\%o22)\ \frac{c^2-d^2}{d+c}$$

```
(%i23) ratsimp((c^2−d^2)/(c+d));
(%o23) c−d
```

Решение уравнений.
Для записи функции необходимо указать ее название, а затем в круглых скобках записать значение аргумента (или через запятую - значения аргументов). Если значением аргумента является список, то он заключается в квадратные скобки, а элементы списка также разделяются запятыми:

```
(%i24) x^2+4=0;
```

$$(\%o24)\ x^2+4=0$$

Применяем функцию solve. Для этого идем Уравнения -> Решить.
Появится команда solve. У нее два аргумента. Уравнение и переменная относительно которой нужно решить.

```
(%i25) solve([x^2+4=0], [x]);
(%o25) [x=-2 %i,x=2 %i]
```

Вместо самого выражения в аргументе функции может быть номер его ввода или вывода

(%i26) solve([%i24],[x]);

(%o26) [x=-2 %i,x=2 %i]

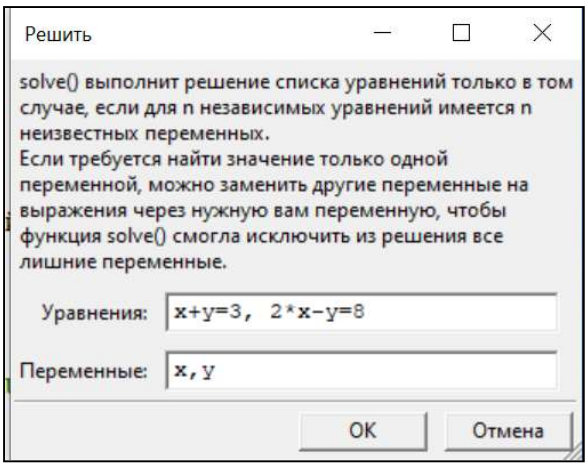
прямой ввод выражения в аргументы команды

(%i27) solve([x+y=3, 2·x−y=8], [x,y]);

(%o27) $\left[\left[x=\frac{11}{3},y=-\frac{2}{3}\right]\right]$

Использование графического интерфейса. Уравнение -Решить и в поле ввода набираем уравнения системы.

Фигура 1:



(%i28) solve([x+y=3, 2·x−y=8], [x,y]);

(%o28) $\left[\left[x=\frac{11}{3},y=-\frac{2}{3}\right]\right]$

(%i29) solve([x+y=3, 2·x−y=8], [x,y]), numer;

(%o29) [[x=3.66666666666667,y=-0.666666666666667]]

Фигура 2: Решить систему

$$x + z - 1 = 0$$
$$x^2 - (2 \cdot y)^2 - z = 5$$
$$2 \cdot x - y - z = 0$$

Когда решаем систему, то первый аргумент функции это список в квадратных скобках где указаны уравнения системы через запятую. Второй аргумент это список переменных

(%i30)
```
solve([x+z−1=0, x^2−(2·y)^2−z=5, 2·x−y−z=0], [x,y,z]);
```
(%o30)
$$\left[\left[x=\frac{\sqrt{31}\ \%i+5}{14},y=\frac{3\sqrt{31}\ \%i+1}{14},z=-\frac{\sqrt{31}\ \%i-9}{14}\right],\left[x=-\frac{\sqrt{31}\ \%i-5}{14},y=-\frac{3\sqrt{31}\ \%i-1}{14},z=\frac{\sqrt{31}\ \%i+9}{14}\right]\right]$$

(%i31)
```
solve([x+z−1=0, x^2−(2·y)^2−z=5,2·x−y−z=0], [x,y,z]), numer;
```
(%o31)
```
[[x=0.0714285714285714 (5.56776436283002 %i+5),y=
0.0714285714285714 (16.7032930884901 %i+1),z=-
0.0714285714285714 (5.56776436283002 %i-9)],[x=-
0.0714285714285714 (5.56776436283002 %i-5),y=-
0.0714285714285714 (16.7032930884901 %i-1),z=
0.0714285714285714 (5.56776436283002 %i+9)]]
```

При решении уравнения с бесконечным числом корней, выводится только один корень и пишется предупреждение.

(%i32)
```
solve([cos(x)=0.5], [x]);
```
rat: replaced -0.5 by -1/2 = -0.5
solve: using arc-trig functions to get a solution.
Some solutions will be lost.
(%o32)
$$\left[x=\frac{\%pi}{3}\right]$$

Создание и введение матриц

Создание матрицы Матрица - > Создать матрицу

(%i33)
```
A: genmatrix(a, 3, 3);
```
(%o33)
$$\begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{pmatrix}$$

Проверка значения A

(%i34) A;

(%o34) $$\begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{pmatrix}$$

Обращение и проверка элемента матрицы

(%i35) a[1,1];

(%o35) $a_{1,1}$

Ввод матрицы Матрица - > Ввести матрицу

(%i36)
```
C: matrix(
 [1,4,9,11],
 [3,8,4,9]
);
```

(%o36) $$\begin{pmatrix} 1 & 4 & 9 & 11 \\ 3 & 8 & 4 & 9 \end{pmatrix}$$

Проверка матрицы C

(%i37) C;

(%o37) $$\begin{pmatrix} 1 & 4 & 9 & 11 \\ 3 & 8 & 4 & 9 \end{pmatrix}$$

(%i38)
```
B: matrix(
 [1,3,6],
 [5,−9,4],
 [7,6,57]
);
```

(%o38) $$\begin{pmatrix} 1 & 3 & 6 \\ 5 & -9 & 4 \\ 7 & 6 & 57 \end{pmatrix}$$

(%i39) determinant(%);

(%o39) -750

Считаем определитель предыдущего результата (матрица B)

(%i40) determinant(C);

determinant: matrix must be square; found *2* ***rows,*** *4* ***columns.***
-- an error. To debug this try: debugmode(true);

Определитель не удалось посчитать так как матрица C не квадратная.

(%i41) determinant(A);

(%o41) $a_{1,1}\left(a_{2,2}a_{3,3}-a_{2,3}a_{3,2}\right)-a_{1,2}\left(a_{2,1}a_{3,3}-a_{2,3}a_{3,1}\right)+a_{1,3}\left(a_{2,1}a_{3,2}-a_{2,2}a_{3,1}\right)$

Получился определитель символьной матрицы

Производные

Прежде чем вычислить производную выражения, необходимо посмотреть на его внешний вид

(%i42) x^4−x·cos(x);

(%o42) $x^4-x\cos(x)$

Чтобы вычислить производную идем в Анализ - > Дифференцировать.
Там указываем выражение и переменную и порядок дифференцирования

(%i43) diff(x^4−x·cos(x),x,1);

(%o43) $x\sin(x)-\cos(x)+4x^3$

(%i44) diff(x^5−sin(x)^3+tan(x+4),x,1);

(%o44) $\sec(x+4)^2-3\cos(x)\sin(x)^2+5x^4$

(%i45) diff(x^4−x·cos(x),x,3);

(%o45) $-x\sin(x)+3\cos(x)+24x$

Вычисление производной в конкретной точке. Использовать возможно только результат в виде вывода. Исходная операция diff требует чтобы аргумент был переменной и поэтому туда числовое значение переменной вставлять нельзя

(%i46) (%o45) ,x=5.4;

(%o46) 135.67700686063

Вычислили значение третьей производной в точке x=5.4

Вычисление неопределенного интеграла.
Анализ - > Интегрировать

Вначале очистим все символьные переменные от числовых значений. Это необходимо делать если собираемся выполнять символьные преобразования.

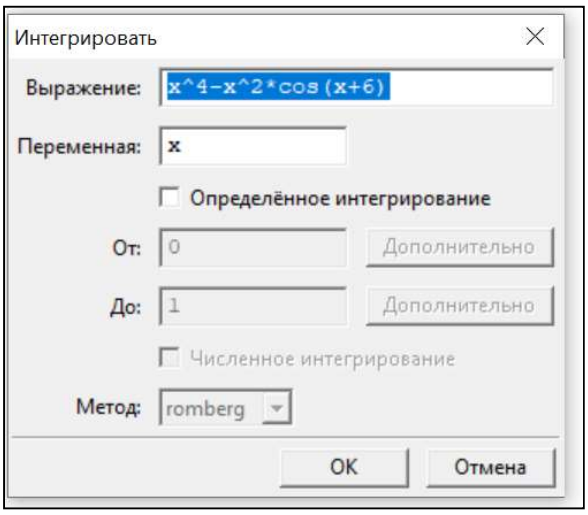
```
(%i47) kill(all);
(%o0)  done
```

После выполнения данной команды отсчет начинается заново.

Вначале проверка вида выражения.

```
(%i1) x^4-x^(2)·cos(x+6);
(%o1) x^4-x^2 cos(x+6)
```

Фигура 3:



В результате получаем команду и ответ

```
(%i2) integrate(x^4-x^(2)·cos(x+6), x);
(%o2) -((x+6)^2-12(x+6)+34)sin(x+6)-(2(x+6)-12)cos(x+6)+x^5/5
```

Использование знака % как результат предыдущего расчета

```
(%i3) cos(%pi/4);
(%o3) 1/sqrt(2)

(%i4) acos(%);
(%o4) %pi/4
```

Для того чтобы в символьном виде увидеть команду неоходимо поставить перед ней апостроф. Апостроф запрещает выполнение команды и тогда результат будет символьным представлением команды. Это выражение можно скопировать в различном формате во вкладке Правка и вставить в соответствующие приложения В Word, Latex, Power Point и т.д.

(%i5) 'diff(x^4−x·cos(x),x,1);

(%o5) $\frac{d}{dx}\left(x^4-x\cos(x)\right)$

(%i6) 'diff(x^4−x·cos(x),x,3);

(%o6) $\frac{d^3}{dx^3}\left(x^4-x\cos(x)\right)$

(%i7) 'integrate(x^4−cos(x)·tg(x), x);

(%o7) $\int x^4-tg(x)\cos(x)$

Примеры использования стандартных функций

(%i8) cos([%pi/3,%pi/2,%pi/6]);

(%o8) $\left[\frac{1}{2},0,\frac{\sqrt{3}}{2}\right]$

У функции косинус список аргументов и ответ получился списком

(%i9) mod(5,2);

(%o9) 1

Вычисляем остаток от деления 5 на 2

(%i10) a:abs(−9);

(%o10) 9

Переменной а присвоили модуль -9

(%i11) a;

(%o11) 9

Проверка значения а

(%i12) kill(all);

(%o0) done

2 Упрощение символьных выражений

wxMaxima может обрабатывать базовые алгебраические операции над выражениями переменных, а также числа: объединение одинаковых терминов, расширение и разложение, добавление / вычитание / уменьшение рациональные выражения и т. д. В некоторых случаях полное упрощение происходит автоматически, а в других случаях мы должны добиться упрощения с помощью ratsimp или fullratsimp (последняя команда просто многократно применяет ratsimp). Либо путем применения специализированных команд.

1. Упростить: (3a + b) + 2(2a-b). Сначала набираем это выражение.

```
(%i1) (3·a+b)+2·(2·a−b);
(%o1) b+2 (2 a−b)+3 a
```

Упростить - Упростить выражение. Получаем команду:

```
(%i2) ratsimp(%);
(%o2) 7 a−b
```

2. Вкладка Раскрыть выражение

```
(%i3) (a+b)^3;
```

(%o3) $(b+a)^3$

```
(%i4) expand(%);
```

(%o4) $b^3+3\,a\,b^2+3\,a^2\,b+a^3$

3. С целью разложения на множители некоторого выражения нужно обратиться к подпункту Факторизовать выражение. Однако если есть необходимость разложения вплоть до комплексных чисел, то выбрать следует подпункт Факторизовать комплексное число.

Факторизовать: x^2-8x + 12

```
(%i5) x^2−8·x + 12;
```

(%o5) $x^2-8\,x+12$

```
(%i6) factor(%);
(%o6) (x−6) (x−2)
```

Разложить на множители x^24 - 1.

(%i7) x^24 − 1;

(%o7) $x^{24}-1$

(%i8) factor(%);

(%o8) $(x-1)\,(x+1)\,(x^2+1)\,(x^2-x+1)\,(x^2+x+1)\,(x^4+1)$
$(x^4-x^2+1)\,(x^8-x^4+1)$

Применение более глубокого разложения на комплексные составляющие

(%i9) gfactor(%);

(%o9) $(x-1)\,(x+1)\,(x-\%i)\,(x+\%i)\,(x^2-\%i)\,(x^2+\%i)$
$(x^2-x+1)\,(x^2+x+1)\,(x^2-\%i\,x-1)\,(x^2+\%i\,x-1)\,(x^4-\%i\,x^2-1)$
$(x^4+\%i\,x^2-1)$

4. В том случае, если необходимо упростить выражения, содержащие логарифмические, экспоненциальные функции и степенные функции с нецелыми рациональными показателями, следует выбрать подпункт Упростить - радикалы.

(%i10) ((x−a)^(3/2)−((x−a)^(1/2))·(x+a))/(((x−a)·(x+a))^(1/2));

(%o10) $\dfrac{(x-a)^{3/2}-\sqrt{x-a}\,(x+a)}{\sqrt{(x-a)\,(x+a)}}$

(%i11) radcan(%);

(%o11) $-\dfrac{2\,a}{\sqrt{x+a}}$

(%i12) ratsimp(((x−a)^(3/2)−((x−a)^(1/2))·(x+a))/(((x−a)·(x+a))^(1/2)));

(%o12) $-\dfrac{2\,a\,\sqrt{x-a}}{\sqrt{x^2-a^2}}$

Применение функции ratsimp к исходному выражению не дает такого хорошего сокращения

(%i13) fullratsimp(%);

(%o13) $-\dfrac{2\,a\,\sqrt{x-a}}{\sqrt{x^2-a^2}}$

Даже многократное применение ratsimp не улучшает результат.

Кроме обращения к пунктам главного меню программы, при работе с символьными выражениями можно непосредственно записывать нужные команды в строку ввода. Наиболее часто применяемыми
командами wxMaxima кроме рассмотренных являются следующие.

5. partfrac - разложение на простые дроби по заданной переменной.

```
(%i14) (x-2)/(x^3+4*x^2+5*x+2);
```

$$(\%o14)\ \frac{x-2}{x^3+4x^2+5x+2}$$

```
(%i15) partfrac((x-2)/(x^3+4*x^2+5*x+2),x);
```

$$(\%o15)\ -\frac{4}{x+2}+\frac{4}{x+1}-\frac{3}{(x+1)^2}$$

6. gcd - наибольший общий делитель многочленов.

```
(%i16) gcd(x^2-7*x+12,x^2-16,x^3-64);
(%o16) x-4
```

7. divide - нахождение частного и остатка от деления одного многочлена на другой.

```
(%i17) divide(x^5-3*x+2,x^2+7);
```

$$(\%o17)\ [x^3-7x,46x+2]$$

первый элемент списка результат деления, второй остаток от деления

Упрощение тригонометрических выражений
wxMaxima применяет тригонометрические функции. Мы вводим углы в радианах, поэтому любой угол, измеренный в градусах, должен быть преобразован в радианы с коэффициентом 2.360. % pi это специальный символ для числа пи.Пример: Вычислить синус 80 градусов и тангенс 12*%pi/17

```
(%i21) 80*2*%pi/360;
       sin(%);
       float(%);
       float(tan(12*%pi/17));
```

$$(\%o18)\ \frac{4\ \%pi}{9}$$

$$(\%o19)\ \sin\left(\frac{4\ \%pi}{9}\right)$$

```
(%o20) 0.984807753012208
(%o21) -1.32421400813415
```

Для работы с выражениями, содержащими тригонометрические функции, предназначены следующие функции:
1) trigsimp - упрощение тригонометрического выражения;
2) trigreduce - преобразование тригонометрических выражений к сумме элементов, каждый из которых содержит только sin или cos.
3) trigexpand - раскрытие скобок в выражениях, содержащих тригонометрические функции, с применением формул преобразования сумм двух углов (основная цель - представление введенного выражения в как можно более простом виде);
4) trigrat - приведение к каноническому виду. Наиболее упрощающая форма.
Все эти вкладки расположены Упростить - Тригонометрическое упрощение

Пример 1. Применим trigreduce к выражению sin(x)^2+cos(x)^2

```
(%i22) sin(x)^2+cos(x)^2;
```

$$(\%o22)\quad \sin(x)^2+\cos(x)^2$$

```
(%i23) trigreduce(%);
```

$$(\%o23)\quad \frac{\cos(2x)+1}{2}+\frac{1-\cos(2x)}{2}$$

Получилось только хуже. Применим trigsimp

```
(%i24) trigsimp(%);
```

$$(\%o24)\quad 1$$

Замечательно. Что и следовало ждать.

Пример 2. Выразить sin(x)^2 через "двойной угол ".
На этот раз trigreduce - это подходящая команда:

```
(%i25) trigreduce((sin(x))^2);
```

$$(\%o25)\quad \frac{1-\cos(2x)}{2}$$

Пример 3. Получить формулу для cos (x + y) через sin x и cos x.
trigexpand упростит аргумент косинуса:

```
(%i26) trigexpand(cos(x+y));
```

$$(\%o26)\quad \cos(x)\cos(y)-\sin(x)\sin(y)$$

Пример 4. Раскрыть скобки: sin (2x + y) + cos (2y + x).

```
(%i27) trigexpand(sin(2·x+y)+cos(2·y+x)) ;
(%o27) -sin(x) sin(2 y)+cos(x) cos(2 y)+cos(2 x) sin(y)+
       sin(2 x) cos(y)

(%i28) kill(all);
(%o0)  done
```

3 Функции. Определение и использование.

Определение функции происходит с применением оператора присвоения

Задание функции

```
(%i1) f(r):=%e^(r-2)+5/(r+5);
```

$$(\%o1)\quad f(r):=\%e^{r-2}+\frac{5}{r+5}$$

Проверка величины f(r)

```
(%i2) f(r);
```

$$(\%o2)\quad \%e^{r-2}+\frac{5}{r+5}$$

Вычисление значения функции двумя способами;

```
(%i3) f(3),numer;  /* непосредственная подстановка в функцию*/
(%o3) 3.34328182845905

(%i4) u:3;  /* задание символьной переменной и использование ее в качестве аргумента*/
(%o4) 3

(%i5) f(u), numer;
(%o5) 3.34328182845905
```

Задание функции двух переменных;

```
(%i6) g(x,y):=x^2+sin(1/(x^2+y));
```

$$(\%o6)\quad g(x,y):=x^2+\sin\left(\frac{1}{x^2+y}\right)$$

```
(%i7) g(u,6);
```

$$(\%o7)\quad \sin\left(\frac{1}{15}\right)+9$$

```
(%i8) g(u,6), numer;
(%o8) 9.06661729492339

(%i9) kill(u);
(%o9) done
```

```
(%i10) u;  /* теперь это символьная переменная без присвоенного значения*/
(%o10) u
```

```
(%i11) g(u,6);/* решение только в символьном виде*/
```

$$(\%o11)\quad \sin\left(\frac{1}{u^2+6}\right)+u^2$$

Для задания функции можно использовать функцию define. Возможности ее очень широки. Нас устроит пример

```
(%i14) expr : cos(y) − sin(x);
       define (F (x, y), expr);
        factor(F(a,b));
(%o12) cos(y)-sin(x)
(%o13) F(x,y):=cos(y)-sin(x)
(%o14) cos(b)-sin(a)
```

```
(%i15) kill(all);
(%o0)  done
```

4 Решение алгебраических уравнений

Пакет решает уравнения в символьном виде, поэтому и в задании уравнений может присутствовать символическая переменная, которая будет параметром

(%i1)
```
solve([x−a=0], [x]);
```
(%o1) $[x=a]$

(%i2)
```
solve([a·x^2+b·x+c=0], [x]);
```
(%o2) $\left[x=-\frac{\sqrt{b^2-4\,a\,c}+b}{2\,a},x=\frac{\sqrt{b^2-4\,a\,c}-b}{2\,a}\right]$

Получили стандартное выражение для корней квадратного уравнения

(%i3)
```
solve([ x−y=a, x+y=7], [x,y]);
```
(%o3) $\left[\left[x=\frac{a+7}{2},y=-\frac{a-7}{2}\right]\right]$

Решение системы, ответ зависит от параметра a.

Ответ представлен в виде списка и переменным никаких значений не присваивается. Чтобы в дальнейшем использовать результат нужно вытащим корни и присвоить их необходимым переменным.

(%i4)
```
x:ev(x,%o3[1]);
```
(%o4) $\frac{a+7}{2}$

(%i5)
```
y:ev(y,%o3[1]);
```
(%o5) $-\frac{a-7}{2}$

проверим

(%i6)
```
x;
```
(%o6) $\frac{a+7}{2}$

(%i7)
```
y;
```
(%o7) $-\frac{a-7}{2}$

```
(%i8)  kill(x,y); /* очищаем переменные x и y */
(%o8)  done
```

Maxima (как и любой другой пакет символьной математики) не всегда способен получить окончательное решение. Однако полученный результат может оказаться всё же проще, чем исходная задача.
Например решаем уравнение:

```
(%i9)  sqrt(x−2)=x−4;
(%o9)  sqrt(x-2) = x-4
```

```
(%i10) solve(sqrt(x−2)=x−4,x);
(%o10) [x = sqrt(x-2)+4]
```

Получили ответ не лучше вопроса. Возведем обе части уравнения в квадрат чтобы получить выражение полиномиальное. То есть избавляемся от иррациональности

```
(%i11) (sqrt(x−2))^2=(x−4)^2;
(%o11) x-2 = (x-4)^2
```

и применим снова функцию solve

```
(%i12) sol:solve([x−2=(x−4)^2], [x]);
(%o12) [x = 6 , x = 3]
```

Получили два корня, но один посторонний. Его нужно убрать. Сначала проверим корни, а потом подстановкой в исходное уравнение определим посторонний корень.

```
(%i13) ev(x,sol[1]);
(%o13) 6
```

```
(%i14) ev(x,sol[2]);
(%o14) 3
```

Подставляем в уравнение используя соответствующее значение аргумента

```
(%i15) sqrt(x−2)=x−4 , x=6;
(%o15) 2 = 2
```

```
(%i16) sqrt(x−2)=x−4 , x=3;
(%o16) 1 = -1
```

Корень x=3 посторонний

Для преобразования уравнений используются функции lhs и rhs, позволяющие выделить левую и правую часть уравнения соответственно.

```
(%i17) eqn:x^2+x+1=(x-1)^3;
```

$$x^2+x+1=(x-1)^3$$

```
(%i18) lhs(eqn);
```

$$x^2+x+1$$

```
(%i19) rhs(eqn);
```

$$(x-1)^3$$

Воспользуемся этими функциями и возведем в квадрат части нижеприведенного уравнения

```
(%i20) eq:sqrt(x+1)+sqrt(4·x+13)=sqrt(3·x+12);
```

$$\sqrt{4\,x+13}+\sqrt{x+1}=\sqrt{3\,x+12}$$

```
(%i21) eq1:lhs(eq)^2=rhs(eq)^2;
```

$$\left(\sqrt{4\,x+13}+\sqrt{x+1}\right)^2=3\,x+12$$

```
(%i22) radcan(2·sqrt(x+1)·sqrt(4·x+13)+5·x+14=3·x+12);
```

$$2\,\sqrt{x+1}\,\sqrt{4\,x+13}+5\,x+14=3\,x+12$$

```
(%i23) ratsimp(%);
```

$$2\,\sqrt{x+1}\,\sqrt{4\,x+13}+5\,x+14=3\,x+12$$

Попытка упростить выражение ничего не дала. Применим функцию solve и получим вид для x. Затем преобразуем выражение собрав все иррациональности в одну сторону, а рациональные выражения в другую

```
(%i24) solve([(sqrt(4·x+13)+sqrt(x+1))^2=3·x+12], [x]);
```

$$\left[x=-\sqrt{x+1}\,\sqrt{4\,x+13}-1\right]$$

```
(%i25) eq2:x+1=rhs(%[1])+1;
```

$$x+1=-\sqrt{x+1}\,\sqrt{4\,x+13}$$

Опять возводим обе части уравнения в квадрат и применяем solve

(%i26) eq3:lhs(eq2)^2=rhs(eq2)^2;

(%o26) $(x+1)^2=(x+1)\ (4\,x+13)$

(%i27) solve([(x+1)^2=(x+1)·(4·x+13)], [x]);

(%o27) $[x=-4\,,x=-1]$

Получили ответы. Теперь подставляем их в исходное выражение.

(%i28) ev(eq,%[1]);

(%o28) $2\,\sqrt{3}\,\%i=0$

(%i29) ev(eq,%o27[2]);

(%o29) $3=3$

Корень x=-4 посторонний.

Замена и подстановка при решении алгебр уравнений

(%i30) eq:sqrt(x+3−4·sqrt(x−1))+sqrt(x+8−6·sqrt(x−1))=1;

(%o30) $\sqrt{x-4\,\sqrt{x-1}+3}+\sqrt{x-6\,\sqrt{x-1}+8}=1$

Заменим sqrt(x-1) на z, получим x=z^2+1. Произведем замену в выражении.

(%i31) eq1:subst(z, sqrt(x−1), eq);

(%o31) $\sqrt{-4\,z+x+3}+\sqrt{-6\,z+x+8}=1$

Избавимся окончательно от x

(%i32) subst(z^2+1, x, eq1);

(%o32) $\sqrt{z^2-4\,z+4}+\sqrt{z^2-6\,z+9}=1$

Упростим и решим.

(%i33) radcan(%);

(%o33) $2\,z-5=1$

(%i34) solve([%], [z]);

(%o34) $[z=3]$

Возвращаясь к замене, решаем уравнение относительно x

(%i35) solve([sqrt(x−1)=3], [x]);

(%o35) $[x=10]$

```
(%i36) ev(eq,%[1]);
```
$$1=1$$

Еще пример

```
(%i37) eq:sqrt(x+sqrt(6·x−9))+sqrt(x−sqrt(6·x−9))=sqrt(6);
```
$$\sqrt{\sqrt{6\,x-9}+x}+\sqrt{x-\sqrt{6\,x-9}}=\sqrt{6}$$

заменим sqrt(6*x-9)=z, получим x=(z^2+9)/6

```
(%i38) eq1:subst(z,sqrt(6·x−9) , eq);
```
$$\sqrt{z+x}+\sqrt{x-z}=\sqrt{6}$$

```
(%i39) subst((z^2+9)/6, x, eq1);
```
$$\sqrt{\frac{z^2+9}{6}+z}+\sqrt{\frac{z^2+9}{6}-z}=\sqrt{6}$$

```
(%i40) radcan(%);
```
$$\frac{\sqrt{2}\,z}{\sqrt{3}}=\sqrt{2}\,\sqrt{3}$$

```
(%i41) solve([%], [z]);
```
$$[\,z=3\,]$$

```
(%i42) solve([sqrt(6·x−9)=3], [x]);
```
$$[\,x=3\,]$$

```
(%i43) ev(eq,%[1]); /* проверка подстановкой в исходное уравнение */
```
$$\sqrt{6}=\sqrt{6}$$

Все корни полинома (действительные и комплексные) можно найти при помощи функции allroots.Способ представления решения определяется переменной polyfactor (по умолчанию false; если установить в true, то функция возвращает результат факторизации). Алгоритм поиска корней получисленный.
Для вычисления корней единичных полиномиальных уравнений используется функция realroots. Варианты синтаксиса: realroots (expr, bound); realroots (eqn, bound); realroots (expr); realroots (eqn). Функция находит все корни выражения expr=0 или уравнения eqn. Функция строит последовательность Штурма для изоляции каждого корня и использует алгоритм деления пополам для уточнения корня с точностью bound или с точностью, заданной по умолчанию.

```
(%i44) 2 − x + x^5;
```
$$x^5-x+2$$

```
(%i45) allroots(%);
(%o45) [x=0.534148546174733 %i+0.894548032657517,x=
       0.894548032657517-0.534148546174733 %i,x=
       1.17722615339419 %i-0.260963880386455,x=-
       1.17722615339419 %i-0.260963880386455,x=-
       1.26716830454212]
```

```
(%i46) realroots (2 − x + x^5, 5e-06);
(%o46) [x=-664361/524288]
```

```
(%i47) (%) , numer;
(%o47) [x=-1.26716804504395]
```

Решение систем уравнений

```
(%i48) solve([x+y−t=0, x−2·t=7], [x,y,t]);
(%o48) [[x=2 %r1+7,y=-%r1-7,t=%r1]]
```

```
(%i49) ev(%), %r1=4;
(%o49) [[x=15,y=-11,t=4]]
```

У нас количество неизвестных больше чем число уравнений. Переменная %r1 выступает в роли неопределенной константы. Таким образом общее решение имеет вид: x=2*%r1+7,y=-%r1-7,t=%r1], где %r1 – произвольная постоянная. Ей можно задавать произвольные действительные значения. При каждом значении получается частное решение.
Например, при %r1 =4 получается частное решение [x=15,y=-11,t=4]

Еще пример

```
(%i50) solve([x^2+y−t=0, x−2·t=7], [x,y,t]);
(%o50) [[x=%r2,y=-(2 %r2^2-%r2+7)/2,t=(%r2-7)/2]]
```

```
(%i51) ev(%),%r2=8;
(%o51) [[x=8,y=-127/2,t=1/2]]
```

Функция linsolve([expr_1, expr_2, ..., expr_m], [x_1, x_2,...,x_n]) решает список одновременных линейных уравнений [expr_1, expr_2, ..., expr_m] относительно списка переменных [x_1,...,x_n]. Выражения [expr1,...,exprn] могут быть полиномами указанных переменных и представляться в виде уравнений.
Путь: Уравнения - > Решить линейную систему

(%i54) ex1:x+y+z+t=6; ex2:2·x−2·y+z+3·t=2; ex3:3·x−y+2·z−t=8;

(%o52) $z+y+x+t=6$

(%o53) $z-2\,y+2\,x+3\,t=2$

(%o54) $2\,z-y+3\,x-t=8$

(%i55) linsolve([z+y+x+t=6, z−2·y+2·x+3·t=2, 2·z−y+3·x−t=8], [x,y,z,t]);

(%o55) $\left[x=-\frac{3\,\%r3-14}{4},y=-\frac{\%r3-10}{4},z=\%r3,t=0\right]$

Таким образом общее решение имеет вид: x=(–14+3%r3)/4, y= (%r3 - 10)/4, z = %r3 t = 0, где %r3 – произвольная постоянная. Ей можно задавать произвольные действительные значения. При каждом значении с получается частное решение. Например, при %r3= 1 получается частное решение

(%i56) ev(%), %r3=1;

(%o56) $\left[x=\frac{11}{4},y=\frac{9}{4},z=1,t=0\right]$

Во многом аналогичный результат позволяет получить функция algsys (фактически, это надстройка над solve).
Функция algsys([expr_1, expr_2, ..., expr_m], [x_1, x_2,...,x_n]) решает список одновременных полиномиальных уравнений [expr_1=0, expr_2=0, ..., expr_m=0] относительно списка переменных [x_1,...,x_n]. Выражения [expr1, . . . , exprn] могут быть представлены и в виде уравнений. Количество уравнений может превышать количество неизвестных, или наоборот.
Путь: Уравнения - > Решить алгебраическую систему.

(%i60) e1: 2·x·(1 − a1) − 2·(x − 1)·a2; e2: a2 − a1;
e3: a1·(−y − x^2 + 1); e4: a2·(y − (x − 1)^2);

(%o57) $2\,(1-a1)\,x-2\,a2\,(x-1)$

(%o58) $a2-a1$

(%o59) $a1\,(-y-x^2+1)$

(%o60) $a2\,(y-(x-1)^2)$

(%i61) algsys([2·(1−a1)·x−2·a2·(x−1), a2−a1, a1·(−y−x^2+1), a2·(y−(x−1)^2)], [x,y,a1,a2]);

(%o61) $\left[\left[x=0,y=\%r4,a1=0,a2=0\right],\left[x=1,y=0,a1=1,a2=1\right]\right]$

(%i62) ev(%), %r4=1;

(%o62) $\left[\left[x=0,y=1,a1=0,a2=0\right],\left[x=1,y=0,a1=1,a2=1\right]\right]$

Упрощение систем уравнений достигается функцией eliminate, позволяющей исключить те или иные переменные.
Вызов eliminate ([eqn_1, ..., eqn_n], [x_1, ..., x_k]) исключает переменные [x_1,..., x_k] из указанных выражений. А затем мы уже можем решить уравнение.
Об этом кстати говориться в тексте панельки при вызове функции solve.
Когда она предлагает для упрощения решения исключать некоторые переменные.
При исключении переменных получаем решения в виде выражений без правой части. По умолчанию это уравнения, где правая часть равна нулю.

```
(%i64) ex1:x+y=5;
       ex2:x-y=3;
(%o63) y+x=5
(%o64) x-y=3

(%i65) eliminate ([ex2, ex1], [y]);
(%o65) [-2 (x-4)]

(%i66) ratsimp(%);
(%o66) [8-2 x]

(%i67) solve([8-2·x], [x]);
(%o67) [x=4]
```

еще пример

```
(%i70) expr1: 2·x^2 + y·x + z;
       expr2: 3·x + 5·y - z - 1;
       expr3: z^2 + x - y^2 + 5;
(%o68) z+x y+2 x^2
(%o69) -z+5 y+3 x-1
(%o70) z^2-y^2+x+5
```

Исключим y

```
(%i71) eliminate ([expr3, expr2, expr1], [y]);
(%o71) [(x^2-1) z^2-4 x^2 z-4 x^4+x^3+5 x^2,24 z^2+(6 x-2) z-9 x^2
       +31 x+124]

(%i73) er5:(x^2-1)·z^2-4·x^2·z-4·x^4+x^3+5·x^2;
       er6:24·z^2+(6·x-2)·z-9·x^2+31·x+124;
(%o72) (x^2-1) z^2-4 x^2 z-4 x^4+x^3+5 x^2
(%o73) 24 z^2+(6 x-2) z-9 x^2+31 x+124
```

Исключим z

```
(%i74) eliminate([er5,er6],[z]);
```

$$(\%o74)\ \left[7425\,x^8-1170\,x^7+1299\,x^6+12076\,x^5+22887\,x^4-5154\,x^3-1291\,x^2+7688\,x+15376\right]$$

А могли сразу исключить обе переменные и получить тот же результат.

```
(%i75) eliminate ([expr3, expr2, expr1], [y,z]);
```

$$(\%o75)\ \left[7425\,x^8-1170\,x^7+1299\,x^6+12076\,x^5+22887\,x^4-5154\,x^3-1291\,x^2+7688\,x+15376\right]$$

```
(%i76) kill(all);
(%o0)  done
```

5 Решение неравенств.

Для решения неравенств требуется пакет fourier_elim . Функция fourier_elim требует двух аргументов: списка неравенств и списка неизвестных.

```
(%i3) load ("fourier_elim") $
ine: (5 · x−16) / 6 + (x + 8) / 12 <(x + 1) / 3;
fourier_elim ([ine], [x]);
```

```
WARNING: redefining MAXIMA::OPAPPLY in DEFMACRO
WARNING: redefining MAXIMA::OPCONS in DEFMACRO
```

$$(\%o2)\quad \frac{5\,x-16}{6}+\frac{x+8}{12}<\frac{x+1}{3}$$

$$(\%o3)\quad [\,x<4\,]$$

Неравенство четвертой степени

```
(%i5) load ("fourier_elim") $
fourier_elim ([x ^ 4 + 5 · x ^ 3 + 5 · x ^ 2−5 · x−6> 0], [x]);
```

$$(\%o5)\quad [\,1<x\,]\ \ \mathrm{or}\ \ [\,-2<x\,,x<-1\,]\ \ \mathrm{or}\ \ [\,x<-3\,]$$

Система неравенств с одним неизвестным.

```
(%i7) load ("fourier_elim") $
fourier_elim (
  [(x + 2) / 4 <= x / 2−3,
   (8−x) / 3 <(1 + x) / 2−1],
  [x]);
```

$$(\%o7)\quad [\,x=14\,]\ \ \mathrm{or}\ \ [\,14<x\,]$$

Система неравенств с двумя неизвестными.

```
(%i12) load ("fourier_elim") $
des1: 3 · x − 5 · y <2;
des2: x + y> (1 + x) / 3;
des3: y <2;
fourier_elim ([des1, des2, des3], [x, y]);
```

$$(\%o9)\quad 3\,x-5\,y<2$$

$$(\%o10)\quad y+x>\frac{x+1}{3}$$

$$(\%o11)\quad y<2$$

$$(\%o12)\quad \left[\frac{1}{2}-\frac{3\,y}{2}<x\,,x<\frac{5\,y}{3}+\frac{2}{3}\,,-\frac{1}{19}<y\,,y<2\right]$$

(Примеры с сайта Yamwi help system)

Попробуем решить иррациональное неравенство

```
(%i15) load ("fourier_elim") $
       des1: sqrt(x−1)>x+1;
       fourier_elim ([des1], [x]);
```

(%o14) $\sqrt{x-1}>x+1$

(%o15) $\left[-x+\sqrt{x-1}-1>0\right]$

Напрямую не вышло. Работаем также как при решении иррациональных неравенств. И сразу укажем область определения y.

```
(%i20) load ("fourier_elim") $
       des1: sqrt(y+10)>10−y;
       des2: (sqrt(y+10))^2>(10−y)^2;
       des3: y+10 >0;
       fourier_elim ([ des1,des2, des3], [y]);
```

(%o17) $\sqrt{y+10}>10-y$

(%o18) $y+10>(10-y)^2$

(%o19) $y+10>0$

(%o20) $\left[6<y,y<15,\sqrt{y+10}+y-10>0\right]$

Условие y меньше 15 лишнее. Возникло из-за возведения в квадрат.
Итак ответ y>6.

```
(%i21) kill(all);
```

(%o0) done

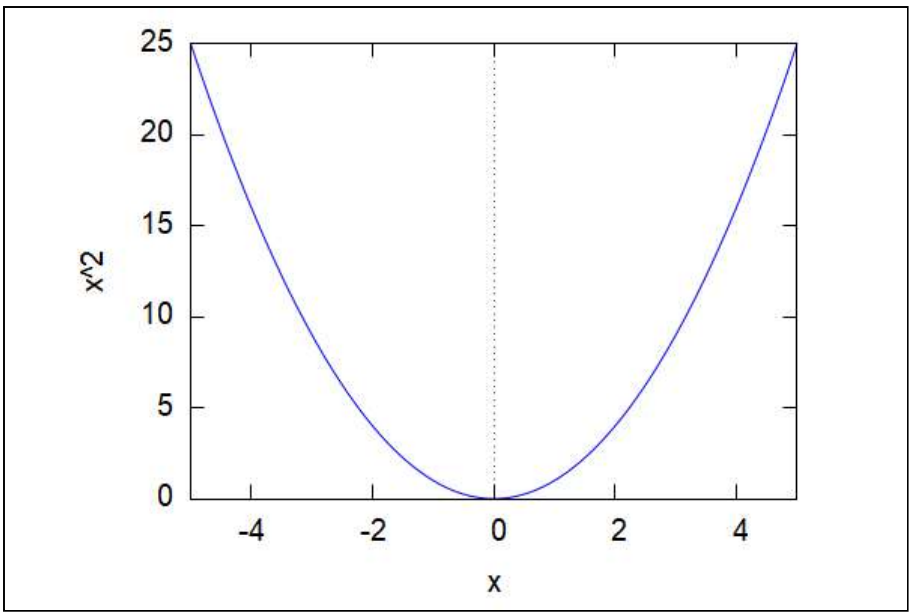
6 Построение графиков

Программа wxMaxima позволяет строить графики функций как на плоскости, так и в пространстве. Для этого предназначен пункт Графики главного меню.

Для построения графиков на плоскости предназначен подпункт Двумерный график. При его выборе появляется диалоговое окно, в котором необходимо указать саму функцию, граничные значения x и y, количество опорных точек (параметр nticks), используемых для построения графика, а также выбрать Формат и Опции

(%i1)
```
wxplot2d([x^2], [x,−5,5])$
```

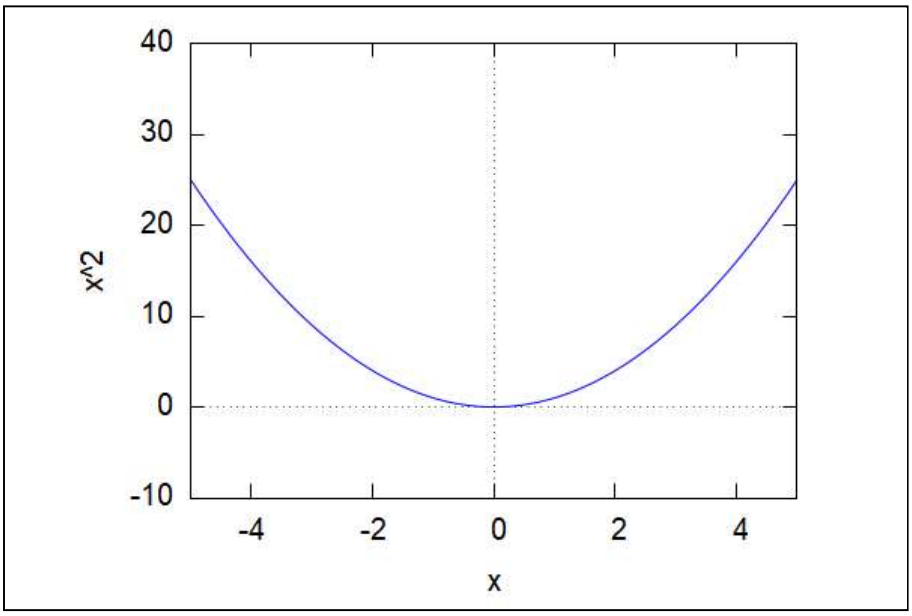
(%t1)



Пределы изменения x определяют шкалу графика по x. Если мы не указываем в поле для y никаких величин, то шкала по y определяется автоматически. Если же укажем, то появится еще один аргумент в командной строке.

(%i2) wxplot2d([x^2], [x,−5,5], [y,−10,40])$

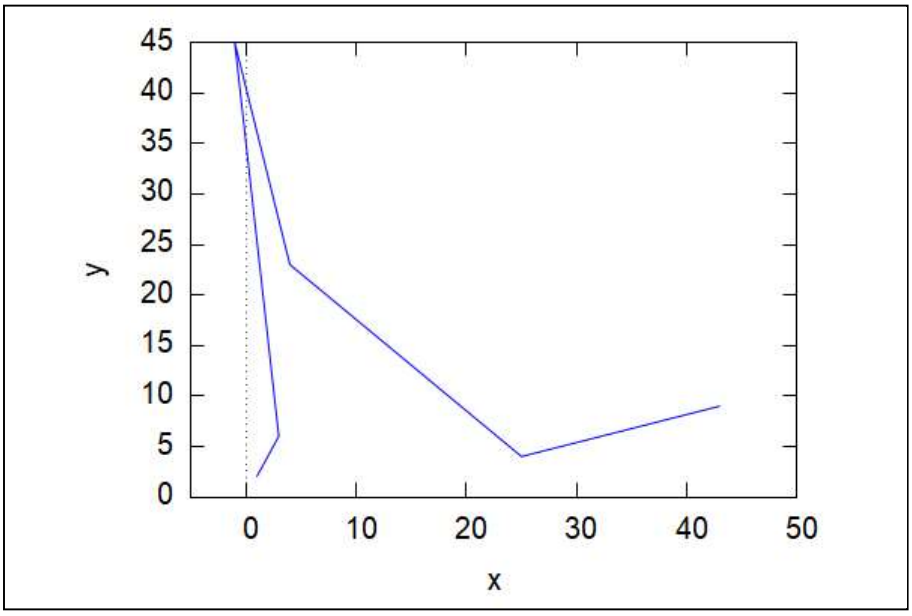
(%t2)



Пример дискретного задания функции. Нам заданы значения x и соответствующие им значения y. на графике эти точки соединяются прямой и получается ломаная линия.

(%i3) wxplot2d([['discrete, [1,3,−1,4,25,43], [2,6,45,23,4,9]]], [x,−5,50])$

(%t3)



Важно что пакет соединяет точки линиями в соответствии с порядком аргумента x! И поэтому для того чтобы увидеть хотя бы приблизительно функциональную зависимость необходимо размещать данные в порядке возрастания первого аргумента!

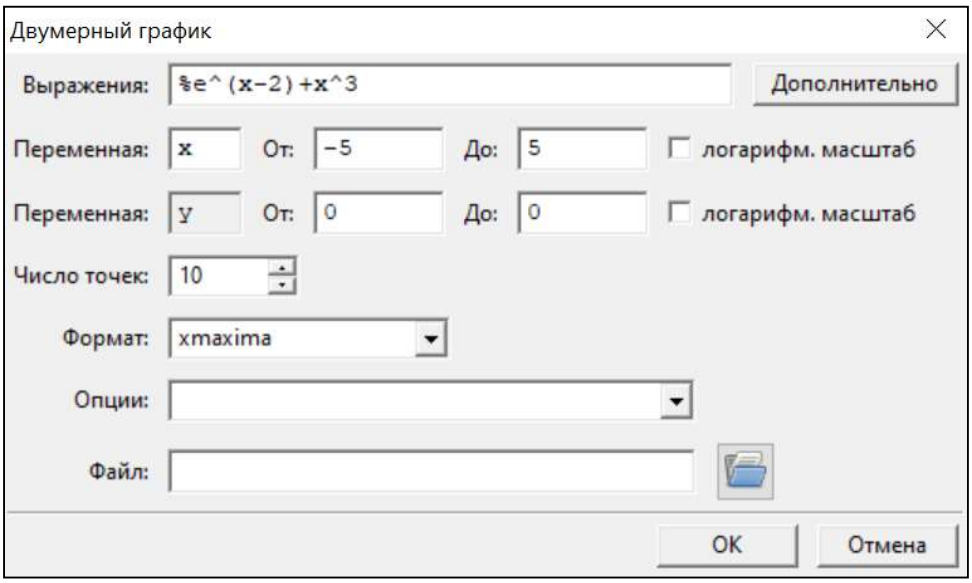
Пример.
Построить график функции на отрезке x=[-5; 5]. Для построения графика функции можно выделить функцию и вызвать из меню Графики диалоговое окно построения графиков. В качестве выражения укажем данную функцию, граничные значения для x от -5 до 5, число опорных точек выберем равным 10, формат - xmaxima , опции -пусто. Я рекомендую выписывать выражения отдельно на рабочем столе, чтобы быть уверенным, что правильно написали. Затем при их выделении они автоматически появляются в строке выражение панели графика.

```
(%i4)  y=x^3+%e^(x-2); /* переменной y никакого значения не присваивается! */
                       /* Эта запись нужна для того чтобы увидеть функцию */
```

$$(\%o4)\quad y = \%e^{x-2}+x^3$$

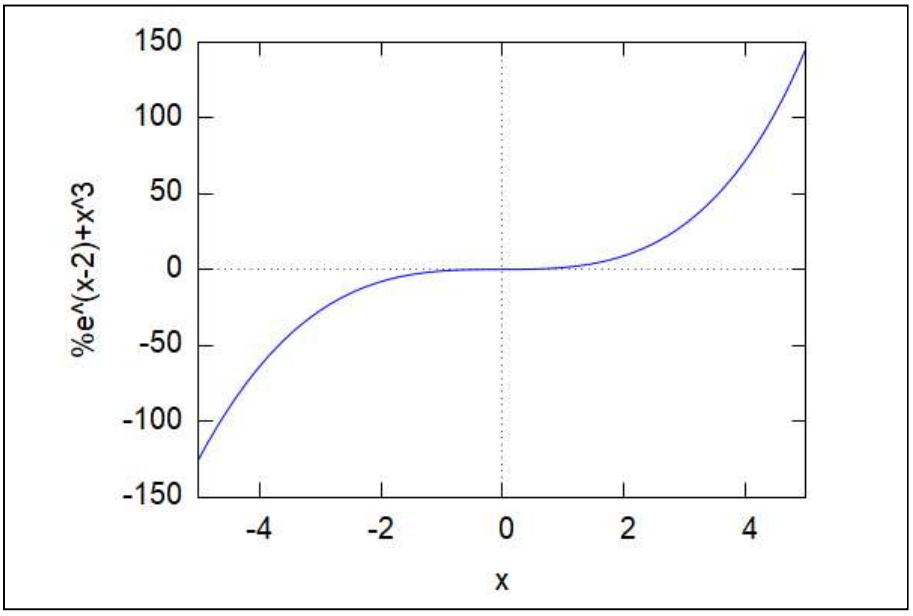
Набираем команду с помощью графического интерфейса

Фигура 4:




```
(%i5)  wxplot2d([%e^(x-2)+x^3], [x,-5,5])$
```

(%t5)



Так же можно присвоить выражение какой-либо переменной или определить функцию. Тогда в окошке для выражения можно указывать эту переменную или функцию

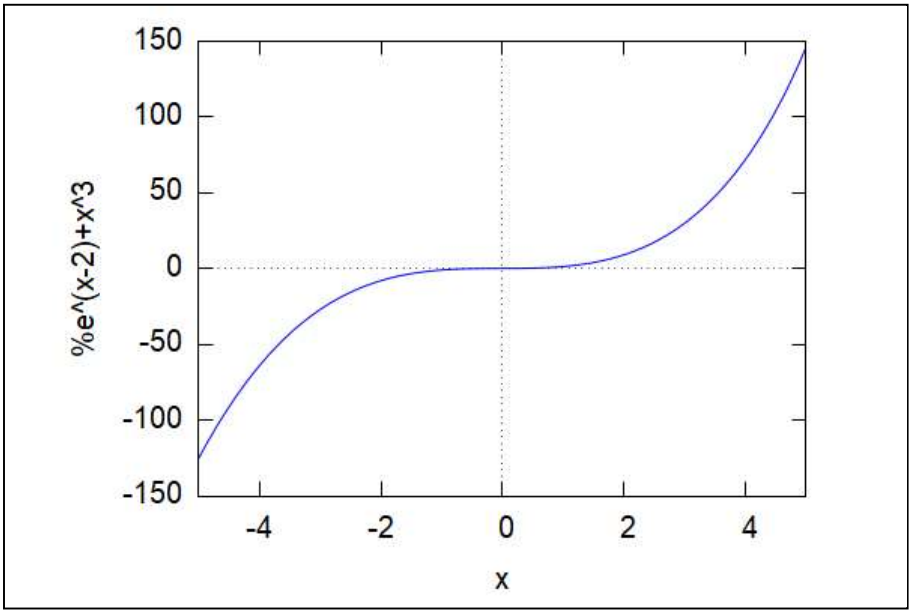
При построении графика, как уже отмечалось, требуется указать Формат.
Возможно выбрать встроенный формат -график получается хороший и правильный, но его невозможно видоизменять. По умолчанию применяется формат gnuplot и возможен формат xmaxima.
При выборе этих форматов рисунок рисуется в отдельном окне. Возможно преобразовывать, масштабировать и вращать (трехмерные графики).

Выбор опций позволяет выбирать внешний вид графиков.
1) set zeroaxis; - проводит оси через начало координат;
2) set grid; - прорисовывает сетку;
3) set size ratio 1; - выравнивает масштабы по осям координат, чтобы круг на мониторе выглядел круглым, а не в виде овала;
4) set polar; set zeroaxis; - построение в полярной системе координат так, что оси проходят через начало координат.

Проверим различные виды выбора опций

(%i6)
```
wxplot2d([%e^(x−2)+x^3], [x,−5,5],
 [gnuplot_postamble, "set zeroaxis;"])$
```

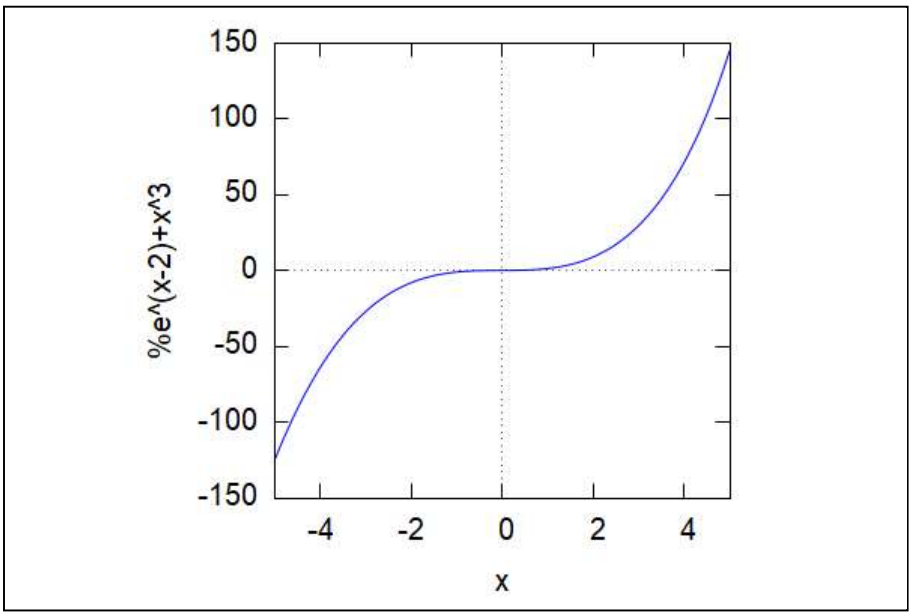
(%t6)



Ничем не отличается от первоначального вида без указания опций.

(%i7)
```
wxplot2d([%e^(x−2)+x^3], [x,−5,5],
 [gnuplot_postamble, "set size ratio 1; set zeroaxis;"])$
```

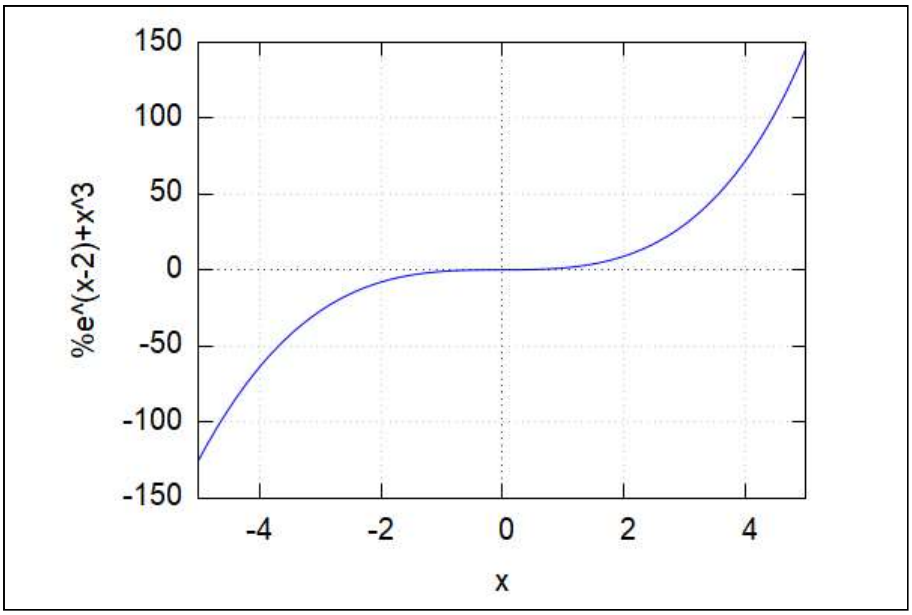
(%t7)



Размеры рисунка по оси x и оси y одинаковы. Удобно для размещения в презентации и печатные работы.

(%i8)
```
wxplot2d([%e^(x−2)+x^3], [x,−5,5],
 [gnuplot_postamble, "set grid;"])$
```

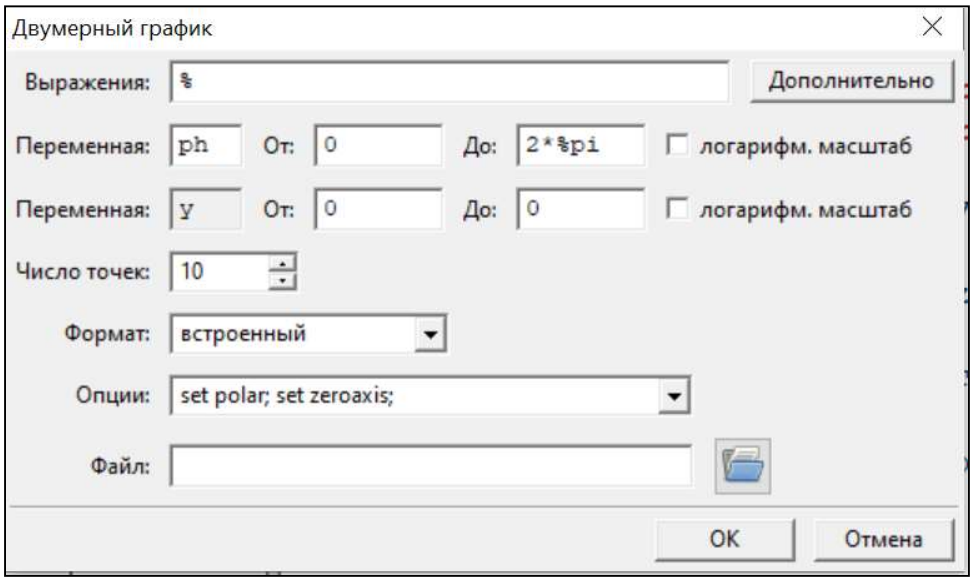
(%t8)



наличие сетки позволяет ориентировочно оценить значения функции в различных точках графика.

Выбор опции set polar; set zeroaxis; позволяет строить графики в полярной системе координат. При выборе данной опции появляется следующее окошко.

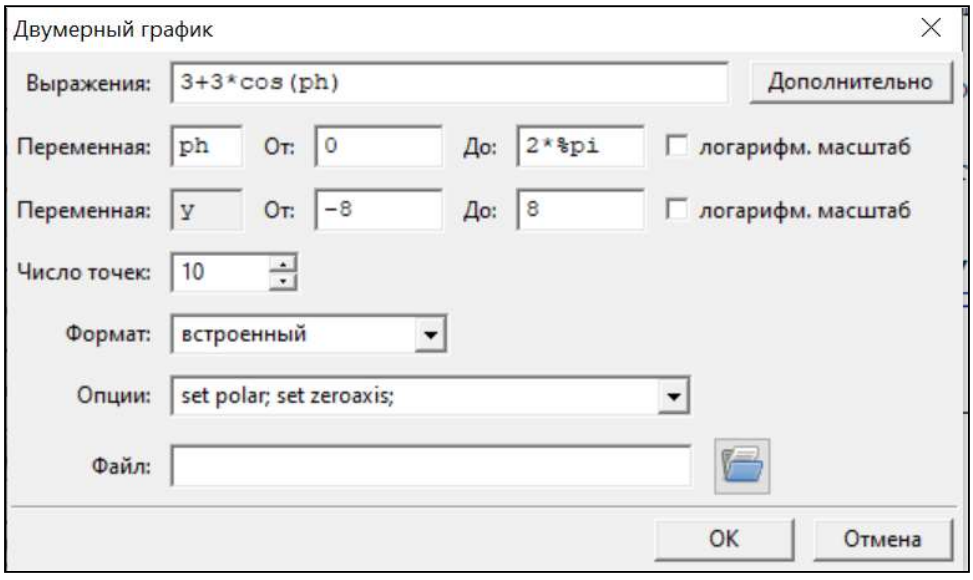
Фигура 5:



Особенность введения выражения состоит в том, что в качестве аргумента необходимо указывать не переменную x, а переменную ph, значения которой по умолчанию меняются от 0 до 2л.
Данные пределы можно изменить. Рассмотрим построение графика кардиоиды в полярной системе координат.

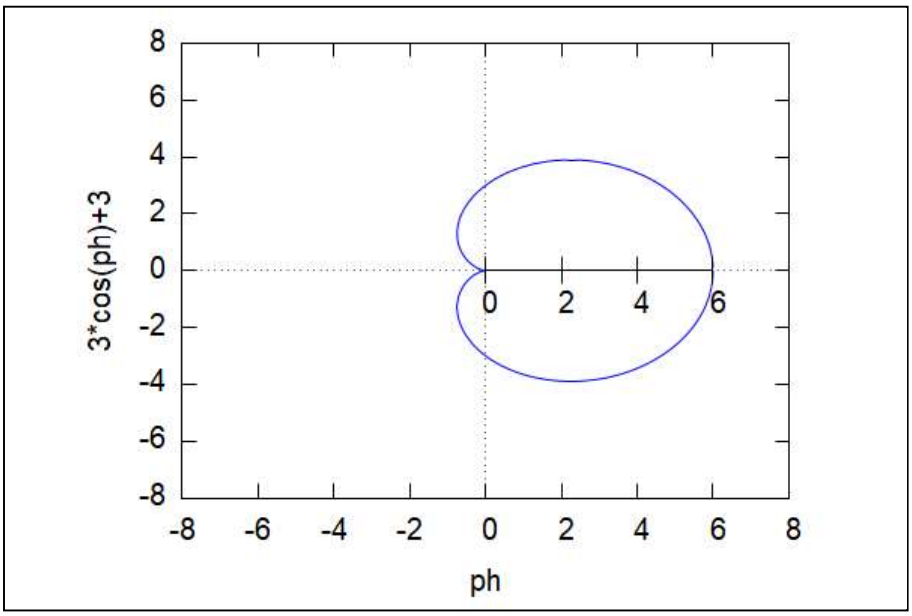
Пример 1. Построить график кардиоиды r(φ) = 3 + 3 cos(φ) в полярной системе координат.
Решение. Выберем подпункт Двумерный график пункта Графики главного меню. В качестве выражения укажем данную функцию, но с аргументом ph.
Граничные значения для ph - от 0 до 2π, для y - от -8 до 8, число опорных точек выберем равным 10, формат - поумолчанию, опции - set polar; set zeroaxis:

Фигура 6:



(%i9) wxplot2d([3+3·cos(ph)], [ph,0,2·%pi], [y,−8,8],
[gnuplot_postamble, "set polar; set zeroaxis;"])$

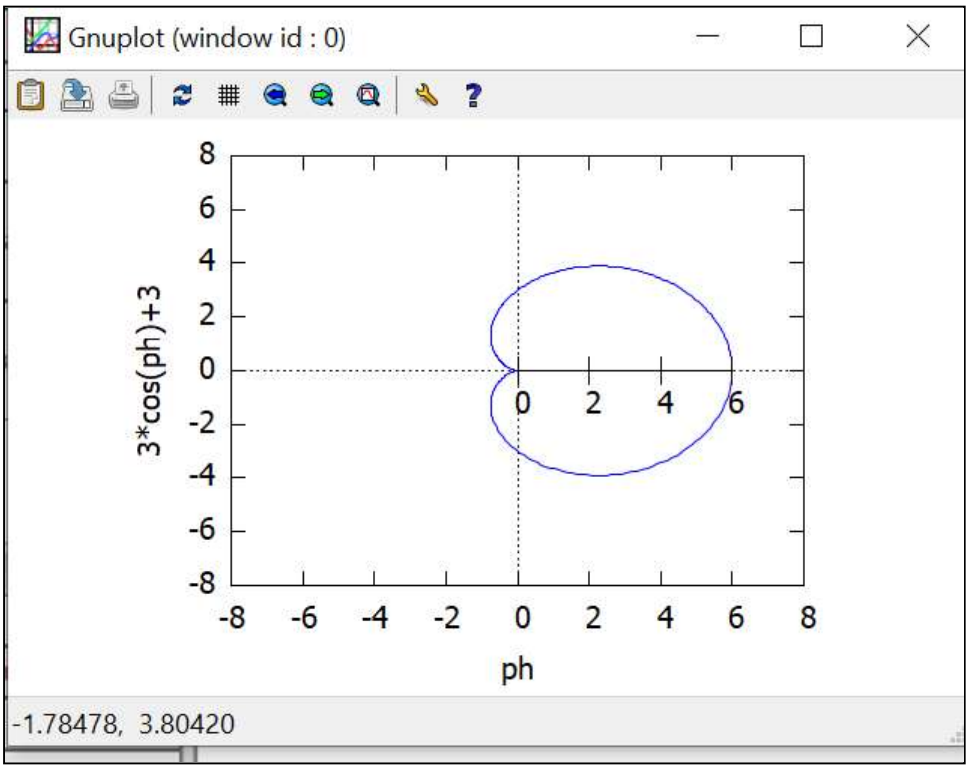
(%t9)



Другой формат

(%i10) plot2d([3+3·cos(ph)], [ph,0,2·%pi], [y,−8,8],
[plot_format, gnuplot],
[gnuplot_postamble, "set polar; set zeroaxis;"])$

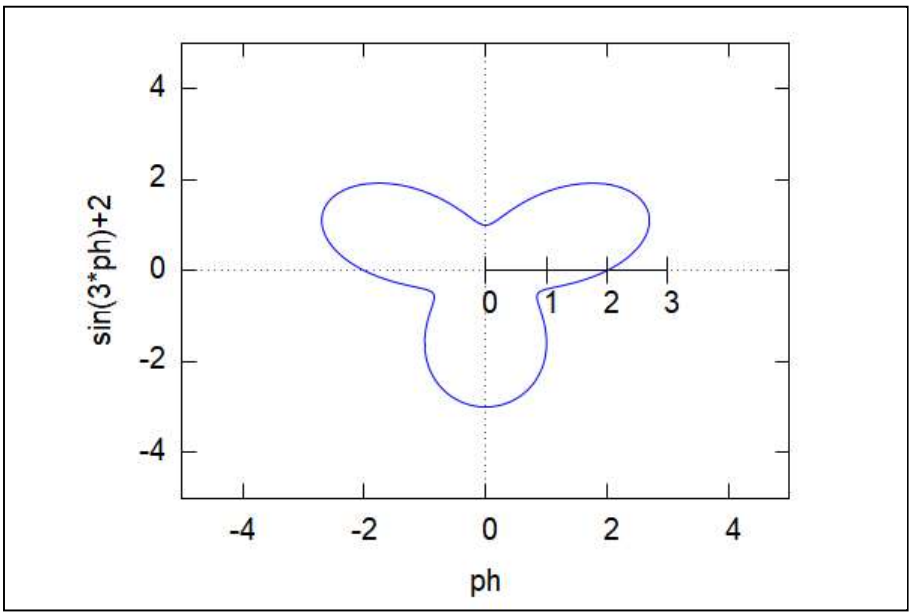
Фигура 7: Кардиоида



Пример 2. Построить график r(phi)=2+sin(3*phi) заданный в полярных координатах.

(%i11) wxplot2d([2+sin(3·ph)], [ph,0,2·%pi], [y,−5,5], [gnuplot_postamble, "set polar; set zeroaxis;"])$

(%t11)

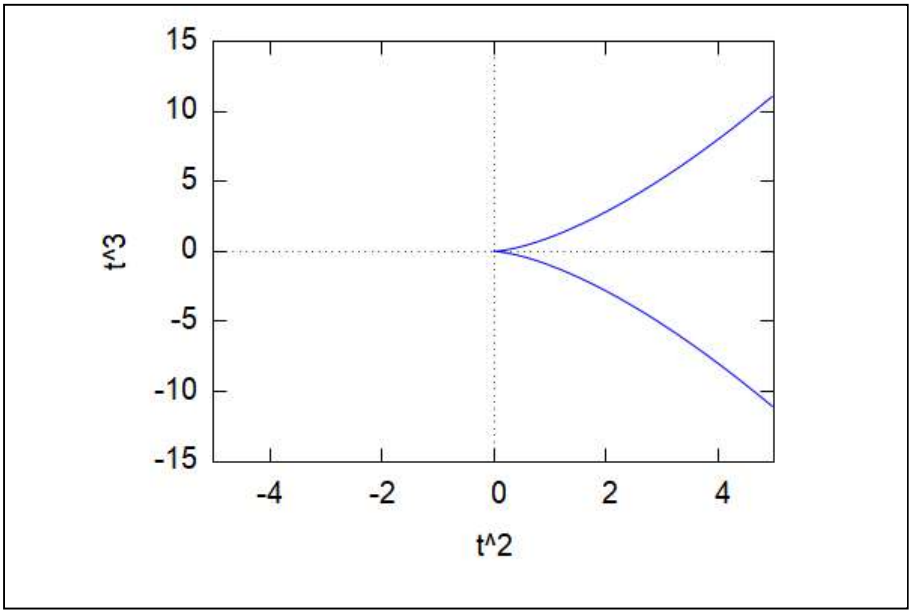


Программа wxMaxima позволяет строить и графики параметрически заданных функций. Для этого после выбора подпункта Двумерный график пункта Графики необходимо нажать на кнопку Дополнительно и далее указать Параметрический график. Появляется диалоговое окно в котором записывают формулы для x и y, границы изменения t и количество опорных точек. Выберем x=t^2, y=t^3. Пределы изменения [x,-5,5].

(%i12) wxplot2d(['parametric, t^2, t^3, [t, −6, 6]], [nticks, 300], [x,−5,5])$

plot2d: some values were clipped.

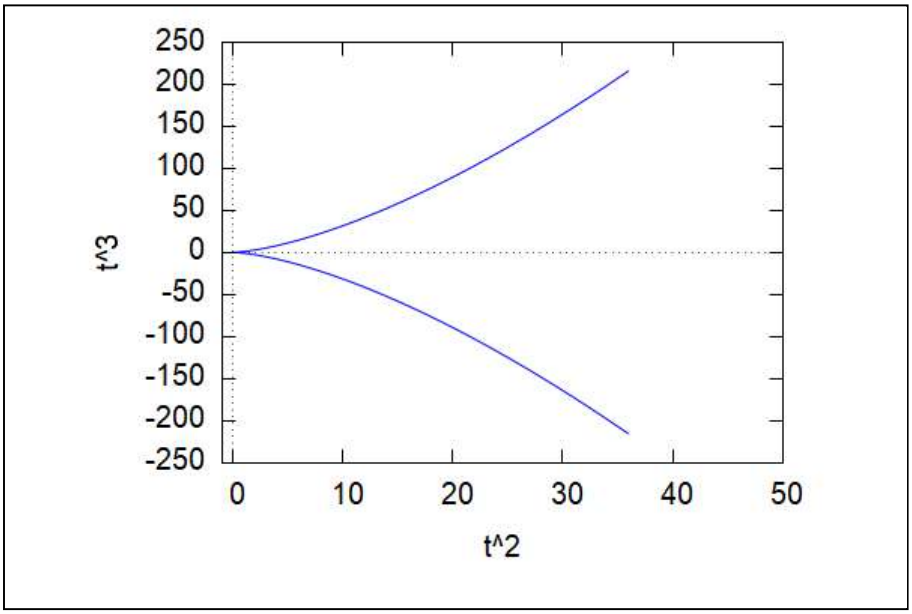
(%t12)



Получили график многозначной функции, однако пределы изменения x на рисунке от -5 до 5 меньше чем реальный диапазон изменения x от 0 до 36 при изменении параметра от -6 до 6. Часть данных была потеряна. Об этом пакет написал plot2d: some values were clipped.
Если увеличить шкалу изменения x, то рисунок будет полным.

(%i13) wxplot2d(['parametric, t^2, t^3, [t, −6, 6]], [nticks, 300], [x,−1,50])$

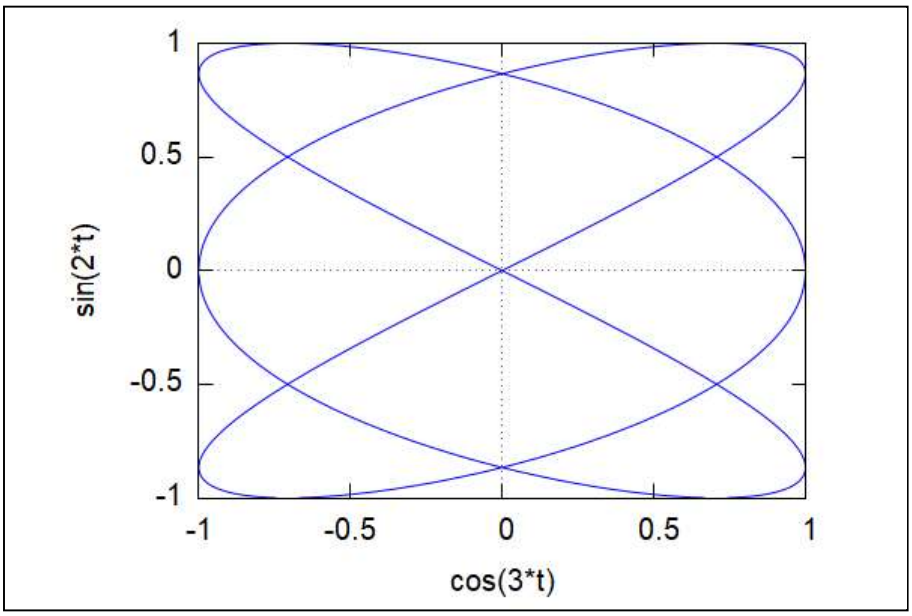
(%t13)



Теперь выберем функции x=cos(3t) и y=sin(2t).
Пусть t меняется от 0 до 2%pi. Это уже похоже на изменения полярной координаты φ.
Еще нужно отследить, чтобы границы изменеия x были соответствующие

(%i14) wxplot2d(['parametric, cos(3·t), sin(2·t), [t, 0, 2·%pi]], [nticks, 300], [x,−1,1])$

(%t14)



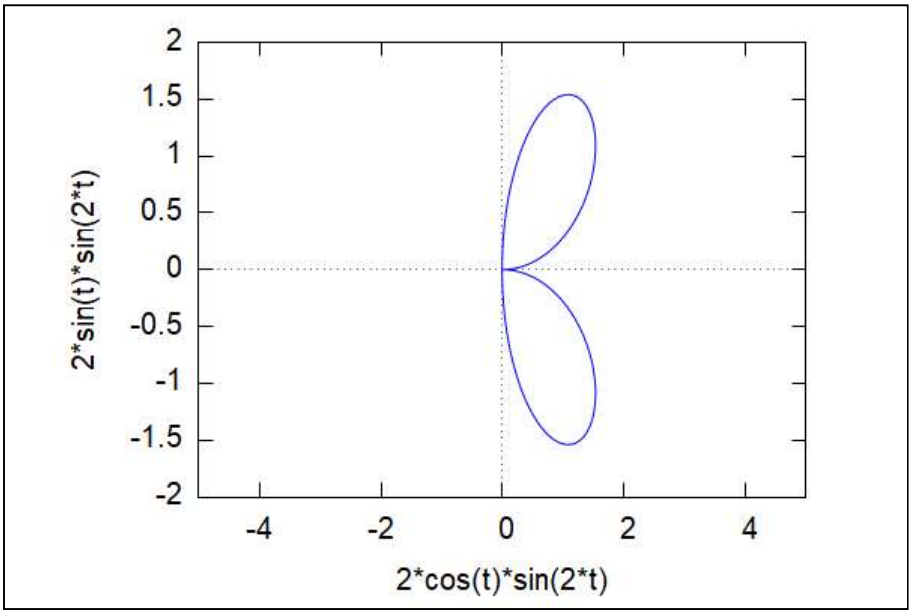
В случае необходимости построить график заданный в полярных координатах, мы можем воспользоваться опцией параметрически заданной функции. При этом учитываем, что x=r(φ)*cos(φ), y=r(φ)*sin(φ). Где r и φ радиус и угол полярной системы координат соответственно. Переменная параметра t будет выступать в качестве угла. А функцию r мы зададим через нее.

Пример: построить график функции r(φ)=2sin(2*φ)

(%i17)
```
r(t):=2·sin(2·t);
plot2d(['parametric, r(t)·cos(t), r(t)·sin(t), [t, 0,%pi]], [nticks, 300], [x,−2,2],
 [plot_format, xmaxima])$;
wxplot2d(['parametric, r(t)·cos(t), r(t)·sin(t), [t, 0, %pi]], [nticks, 300], [x,−5,5])$
```

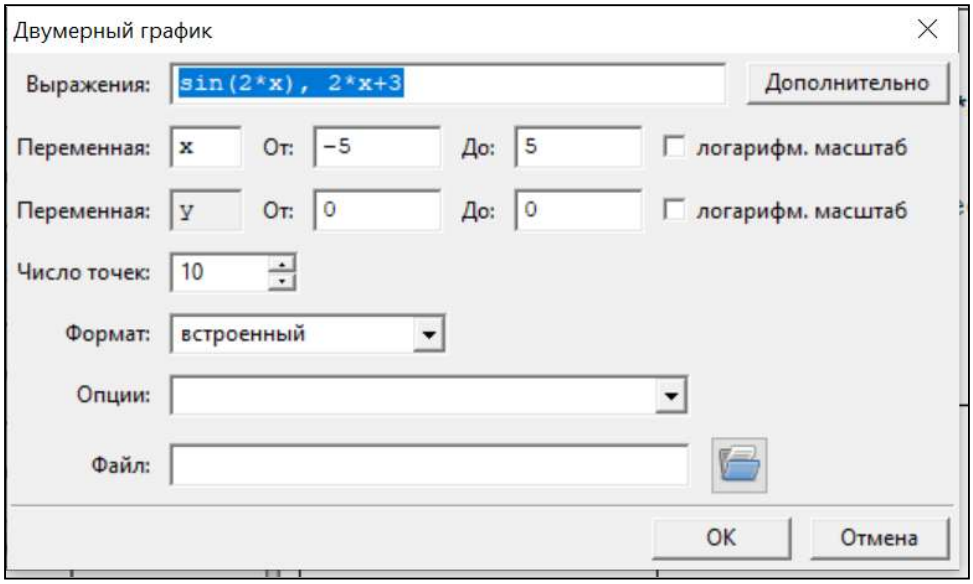
(%o15)
```
r(t):=2 sin(2 t)
```

(%t17)



Если необходимо построить несколько графиков на одном рисунке, то мы их просто перечисляем в окошке выражения.

Фигура 8:



(%i18) wxplot2d([sin(2·x), 2·x+3], [x,−5,5])$

(%t18)

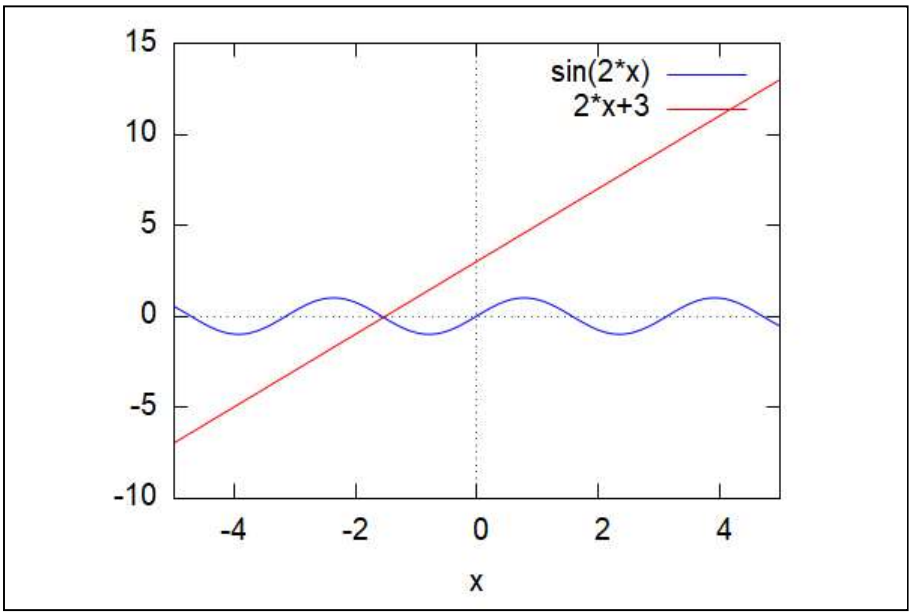
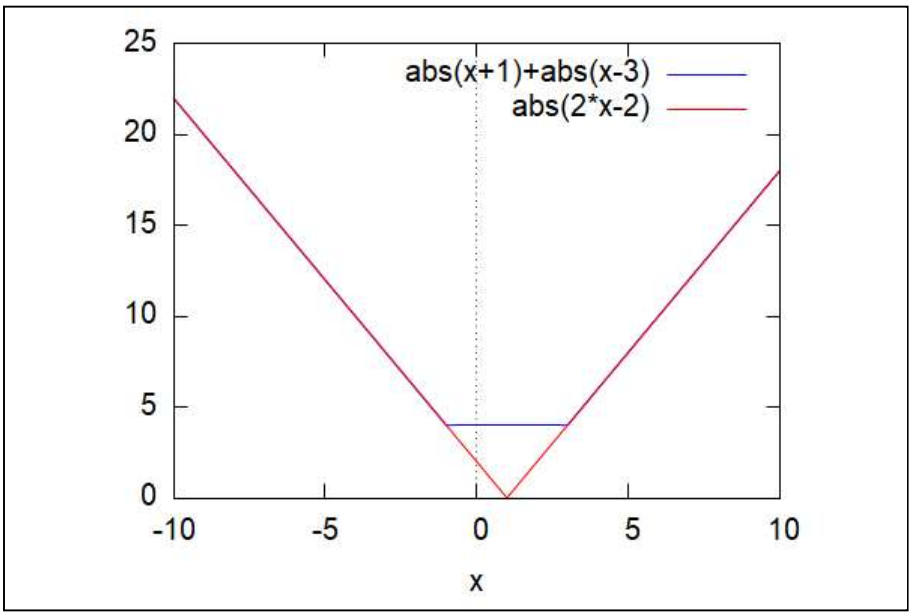


График помогает определить, что у уравнения sin(2*x)=2*x+3 есть корень, и что он находится около точки x=-1.5

Одновременное построение графиков двух функций может помочь найти корни уравнения, в том случае когда аналитическое решение может привести к тождеству и вызвать затруднения.
Пример: Решить |x-3|+|x+1|=|2x-2|. Построим графики правой и левой части.
По крайней мере можно определить сколько корней будет

(%i19) wxplot2d([abs(x−3)+abs(x+1), abs(2·x−2)], [x,−10,10]
)$

(%t19)



Мы видим, что графики наложились и на самом деле корней бесконечноо много.

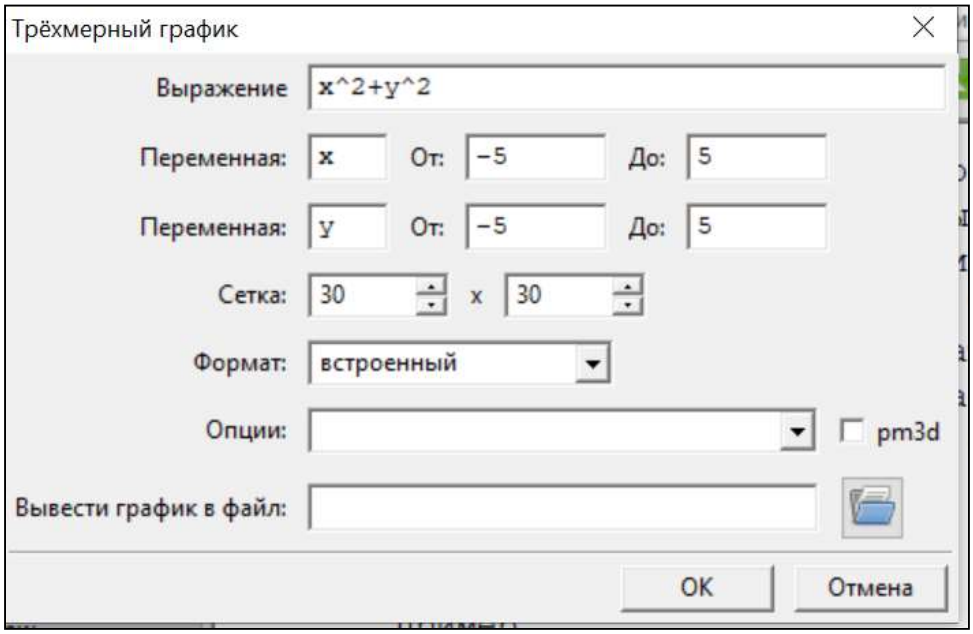
```
(%i20) solve([abs(x−3)+abs(x+1), abs(2·x−2)], [x]);
(%o20) []
```

Функция solve не может дать нам решения этого уравнения.

Многие трехмерные графики могут быть построены естественным образом в привычной прямоугольной системе координат как поверхность вида z =f(x,y).
Для этого необходимо вызвать подпункт Трехмерный график. При заполнении поля
Выражение важно, что функция должна быть однозначной, поэтому в случае, например, эллипсоида построить можно только одну его часть: верхнюю z > 0 или нижнюю z < 0.

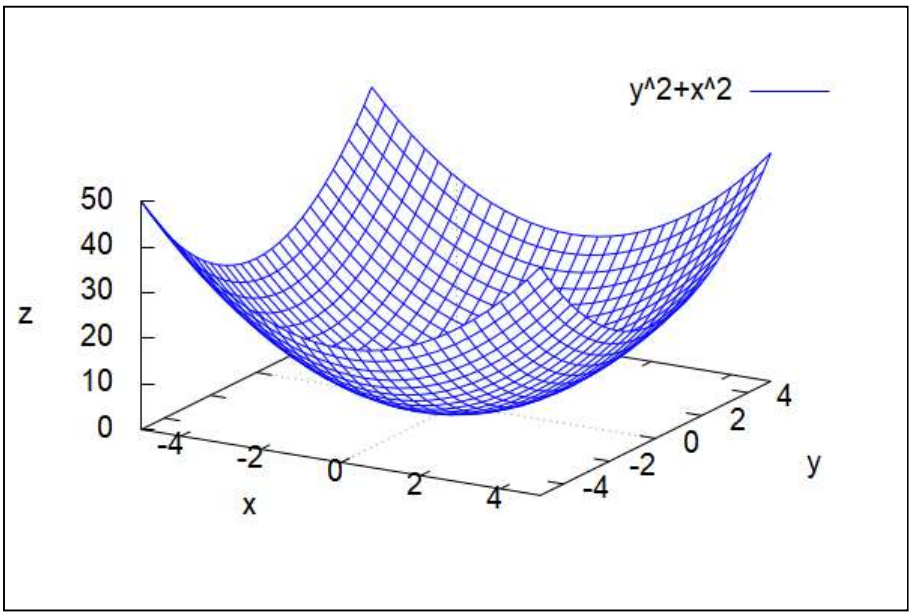
Пример
Построить график z=x^2+y^2,

Фигура 9:



(%i21) wxplot3d(x^2+y^2, [x,−5,5], [y,−5,5],
[gnuplot_pm3d,false])$

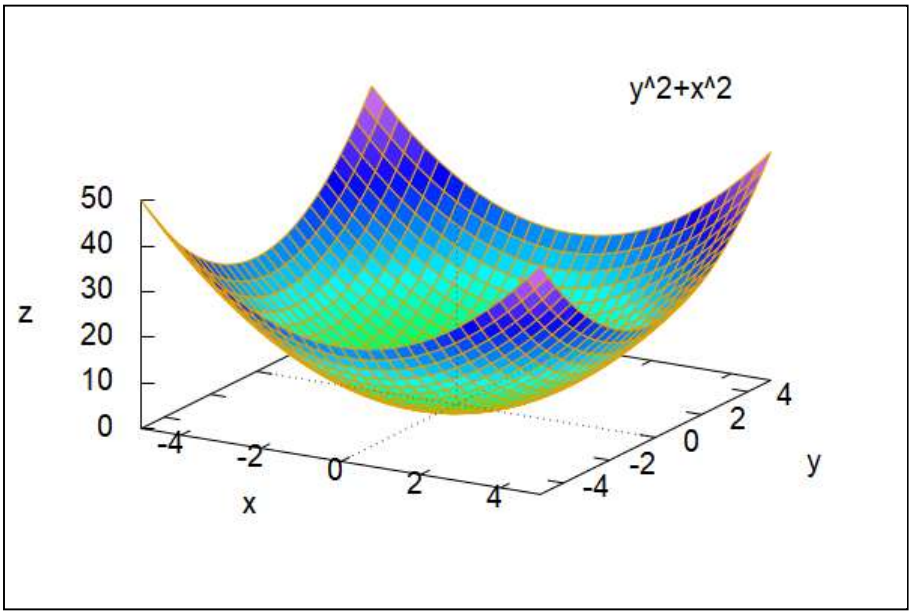
(%t21)



Использование галочки pm3d позволяет применить заливку поверхности соответственно уровню

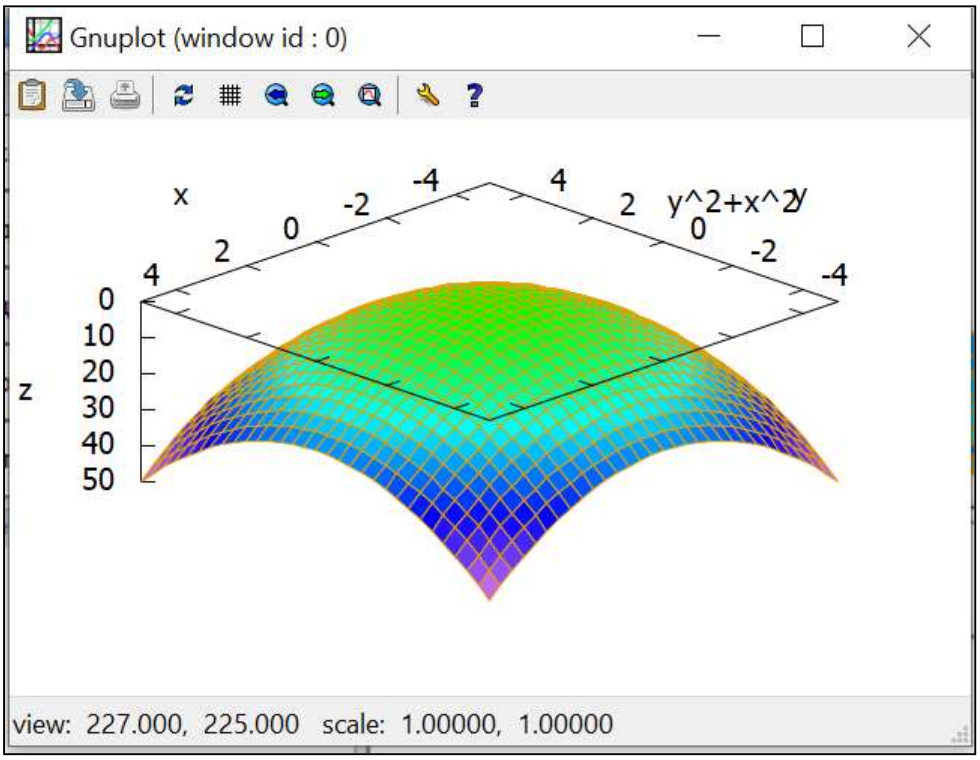
(%i22) wxplot3d(x^2+y^2, [x,−5,5], [y,−5,5])$

(%t22)



(%i23) plot3d(x^2+y^2, [x,−5,5], [y,−5,5], /* другой формат результатт в отдельном окне*/
[plot_format,gnuplot])$

Фигура 10:



Та же поверхность, но перевернутая

Трехмерные рисунки в отдельных окнах мы можем вращать, масштабировать преобразовывать. И главное вы можете показть рисунок именно в том ракурсе, который вам необходим, чтобы подчеркнуть особенности полученного вами результата.

При построении трехмерных графиков появляется возможность использовать сферические и цилиндрические координаты. Принцип работы с ними такой же как с полярными для 2х мерных случаев. Открываются они в панели на вкладке опции.

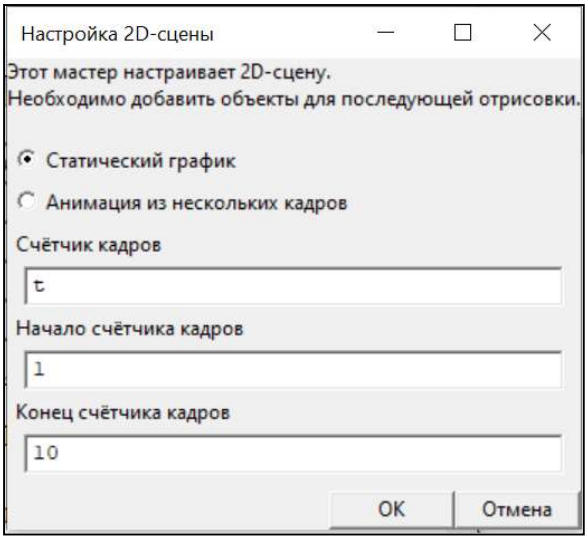
```
(%i24) kill(all);
(%o0)  done
```

7 Построение графиков при помощи панели Draw.

Фигура 11:

На рисунке показана панель Draw. При создании графика сначала нужно определиться с его размерностью нажав 2D или 3D.
При нажатии возникнет панель

Фигура 12:



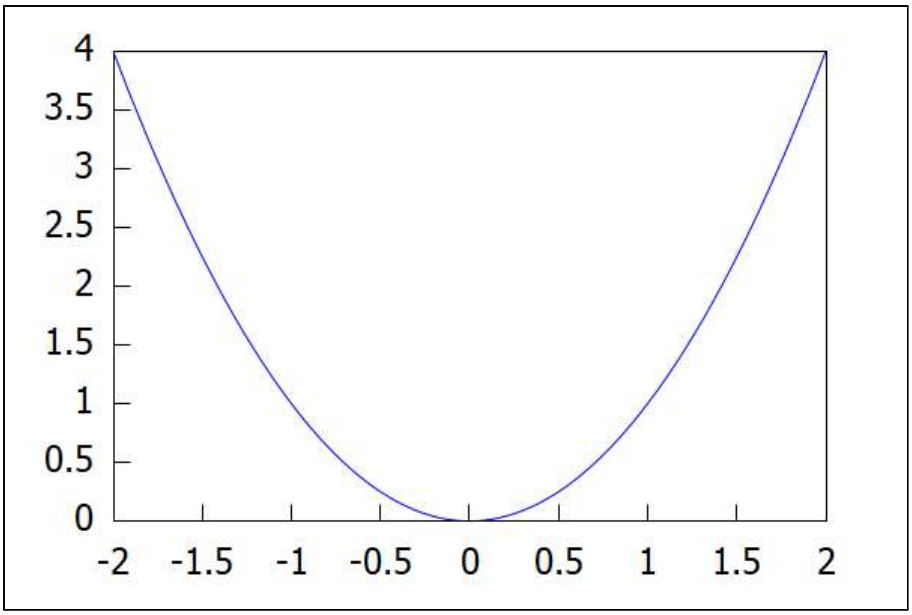
В которой вы проводите выбор типа графика - статический либо анимация.

Рассмотрим статические графики

Построение графика явной функции осуществляется нажатием кнопки Выражение.
В соответствующем поле вводим выражение и пределы изменения переменной.
Получим результат

(%i1)
```
wxdraw2d(
  explicit(
    x^2,
    x,−2,2
    )
)$
```

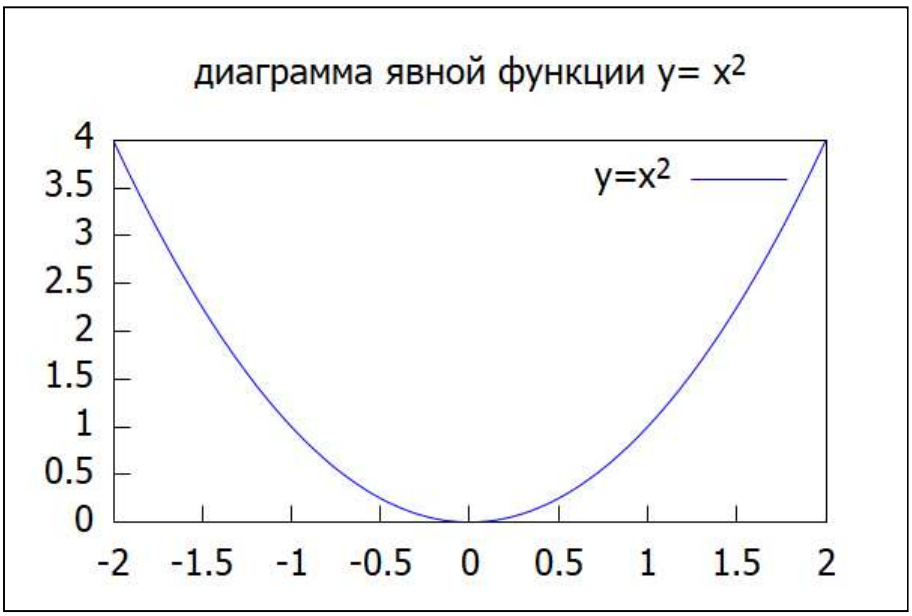
(%t1)



Добавим название графика и название всей диаграммы

(%i2)
```
wxdraw2d(
  title="диаграмма явной функции y= x^2",
   key="y=x^2",
  explicit(
    x^2,
    x,−2,2
    )

)$
```

(%t2)



Добавим цвет линии

(%i3)
```
wxdraw2d(
  title="диаграмма явной функции y=  x^2",
   key="y=x^2",
  color="#ff0000",
   explicit(
     x^2,
     x,−2,2
     )
)$
```

(%t3)

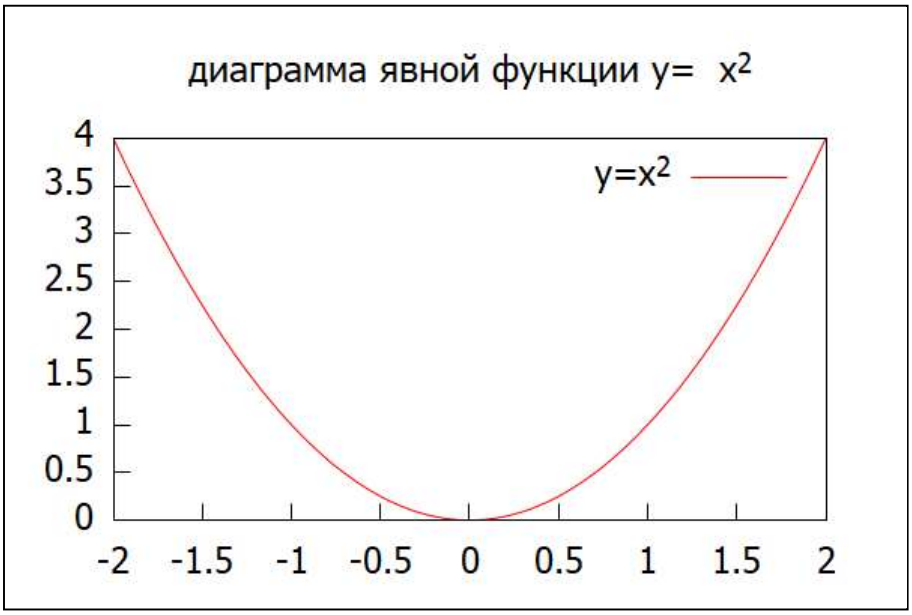


График заданный дискретно. Кнопка "Точки". Задаем формат данных - два списка: значения x и соответствующие им значения y. Форматов данных может быть много, в частности через задание матриц.
Также задаем "Тип точек"" выбирая один из 14 видов

(%i4)
```
wxdraw2d(
  title="дискретное задание",
 color="#ff0000",
 key="points",
 point_type=6,
 points_joined=true,
 apply('points,[[2,4,8,9],[5,12,3,8]])
 );
```

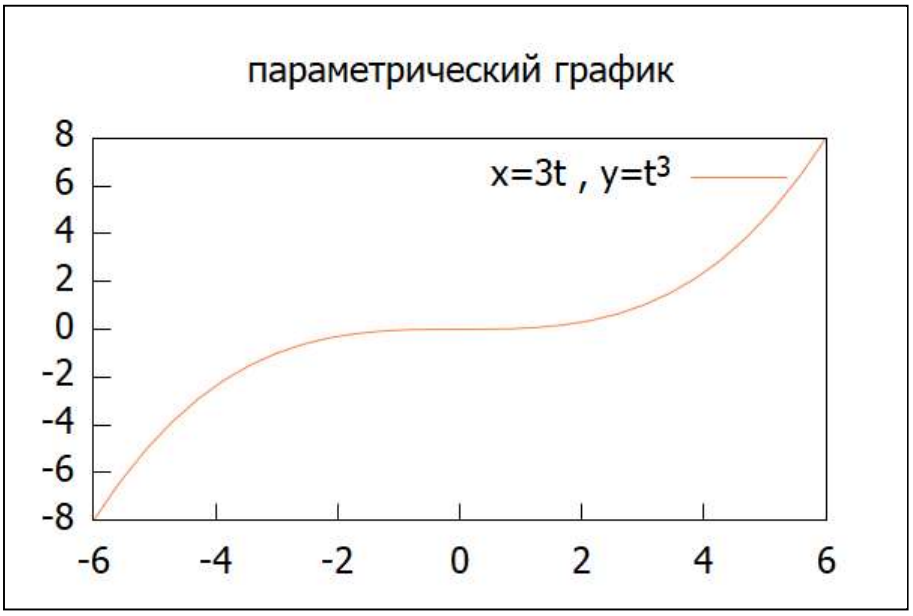
(%t4)

(%o4)

График параметрически заданной функции требует указать зависимость координат от параметра и пределы его изменения

(%i5)
```
wxdraw2d(
  title="параметрический график",
  color="#ff8040",
  key=" x=3t , y=t^3",
  parametric(
    3·t,
    t^3,
    t,−2,2
    )
);
```

(%t5)



(%o5)

Построение графика неявной функции требует задания уравнения и пределов изменения переменных

Фигура 13:

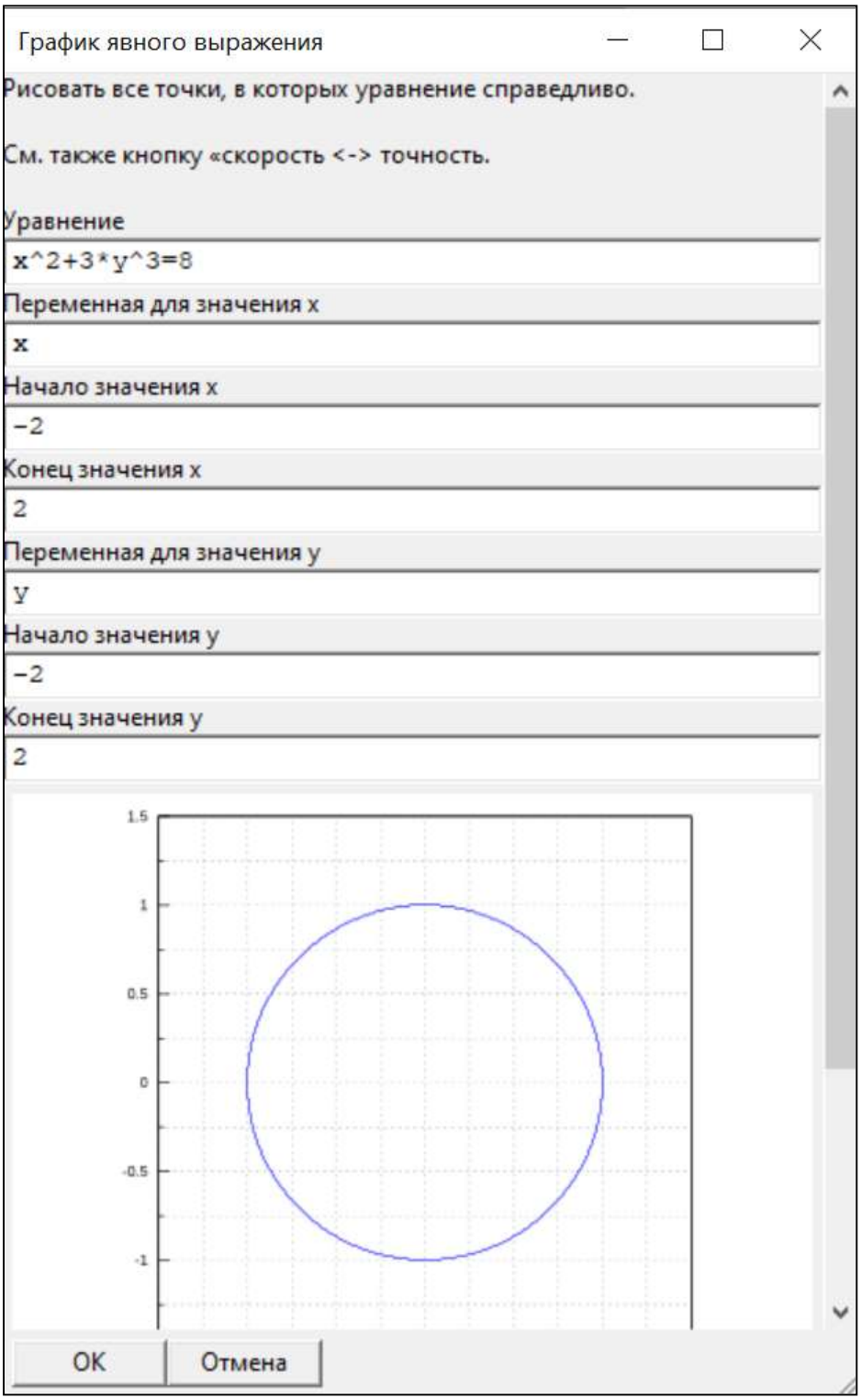
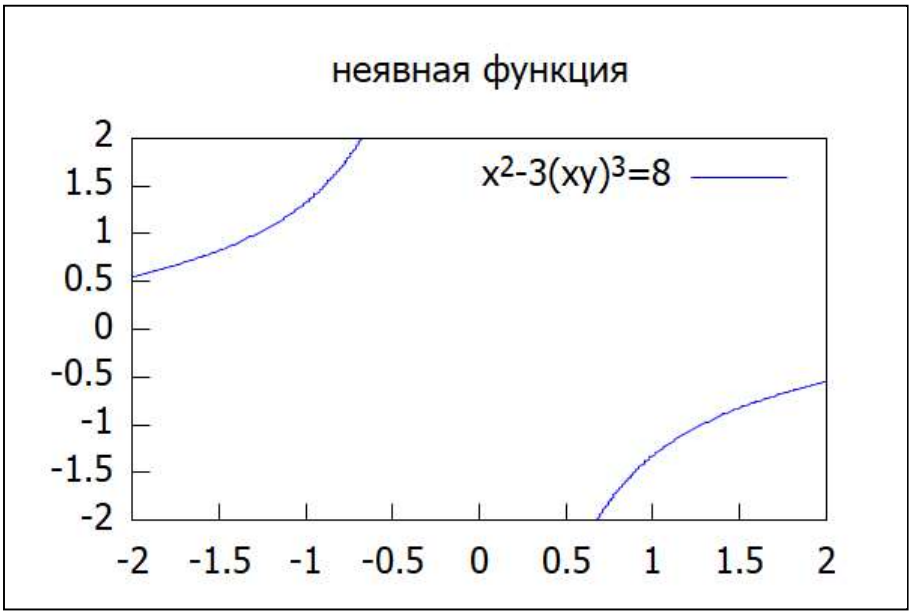


График явного выражения
Рисовать все точки, в которых уравнение справедливо.
См. также кнопку «скорость <-> точность.
Уравнение
x^2+3*y^3=8
Переменная для значения x
x
Начало значения x
-2
Конец значения x
2
Переменная для значения y
y
Начало значения y
-2
Конец значения y
2
1.5
1
0.5
0
-0.5
-1
OK
Отмена

(%i6)
```
wxdraw2d(
   title="неявная функция",
   key="x^2-3(xy)^3=8",
    implicit(
      x^2−3·(x·y)^3=8,
      x,−2,2,
      y,−2,2    )
)$
```

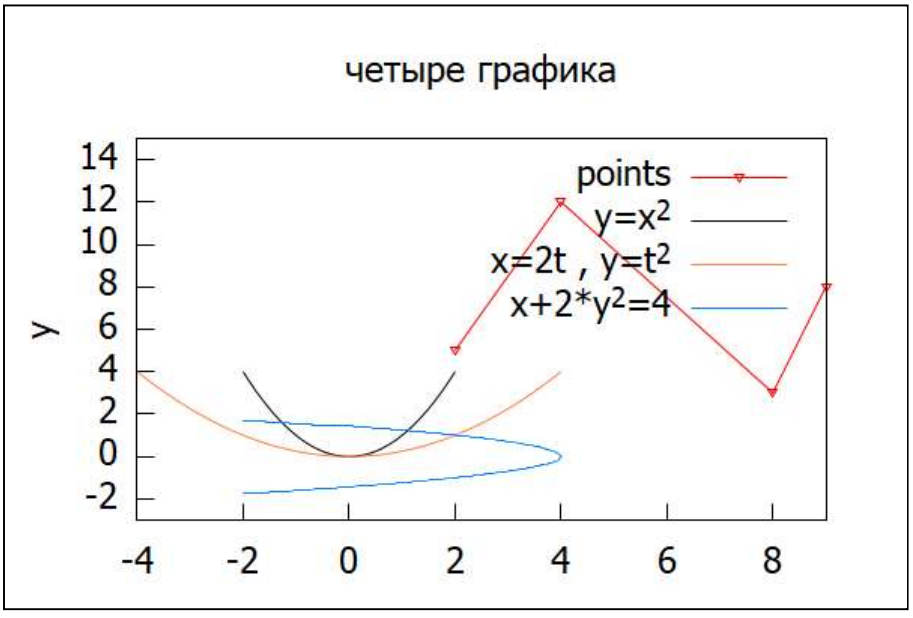
(%t6)



Ну и четыре графика на одном рисунке. Для правильного позиционирования необходимо задать границы изменения оси У. Кнопкой "Ось".

(%i7)
```
wxdraw2d(
 title="четыре графика",
ylabel="y",
yrange=[−3,15],

color="#ff0000",
key="points",
point_type=10,
points_joined=true,
apply('points,[[2,4,8,9],[5,12,3,8]]),
color="#000000",
key="y=x^2",
 explicit(
   x^2,
   x,−2,2
   ),

color="#ff8040",
key=" x=2t , y=t^2",
parametric(
   2·t,
   t^2,
   t,−2,2
   ),
color="#0080ff",
key="x+2*y^2=4",
implicit(
   x+2·y^2=4,
   x,−2,9,
   y,−2,15
   )


)$
```

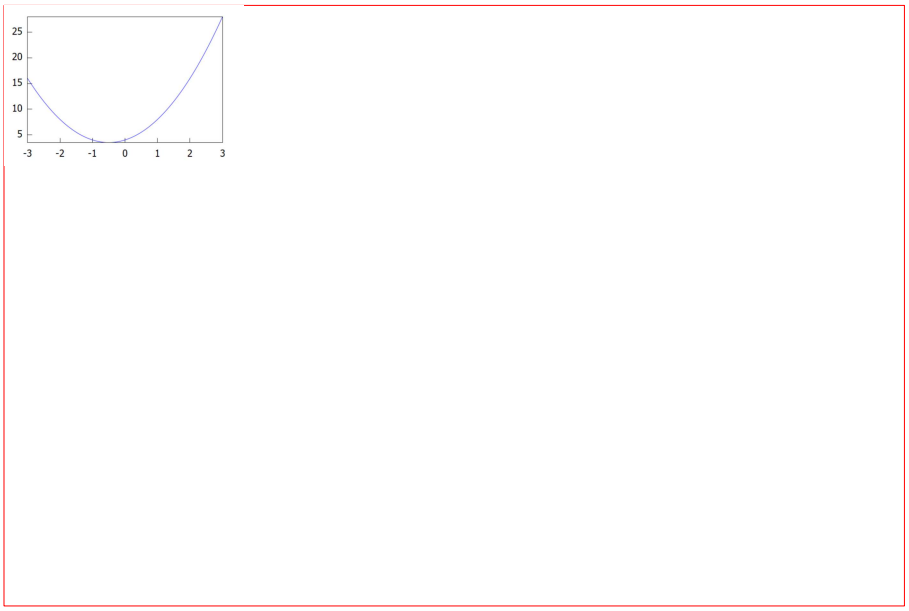
(%t7)



Анимация получается путем выбора "анимация из нескольких кадров". Появляется параметр счетчика кадров и если этот параметр будет входить в выражения для графиков функций, то и графики будут соответственно менятся.

График параболы с изменяющимся коэффициентом при старшем слагаемом

(%i8)
```
with_slider_draw(
t,makelist(i,i,1,10),
   explicit(
      t·x^2+2·x+4,
      x,−3,3
      )
)$
```

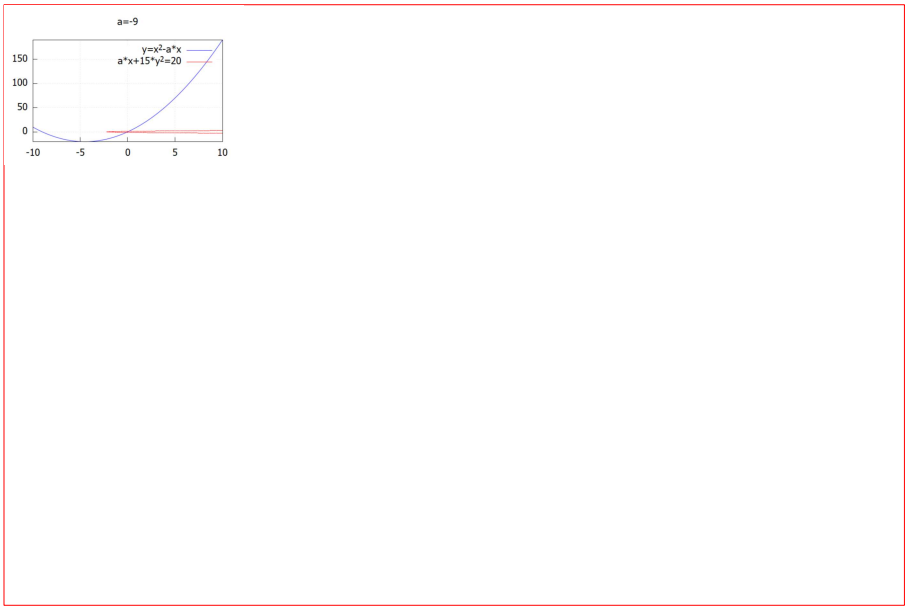


(%t8)

(%t9)

Счетчик кадров a задан списком со значениями. в названии использован прием указания кадра. Приведены два графика явной и неявной функций.

```
(%i10) with_slider_draw(
         a,[−16,−9,−4,−2,0,1,2,3,4,5,6],
         title=concat("a=",a),
         grid=true,
       key="y=x^2-a*x",
        explicit(
            x^2−a·x,
            x,−10,10
         ),
       key="a*x+15*y^2=20",
       color="#ff0000",
        implicit(
         a·x+15·y^2=20,
       x,−10,10,
       y,−10,10
       )
       );
```



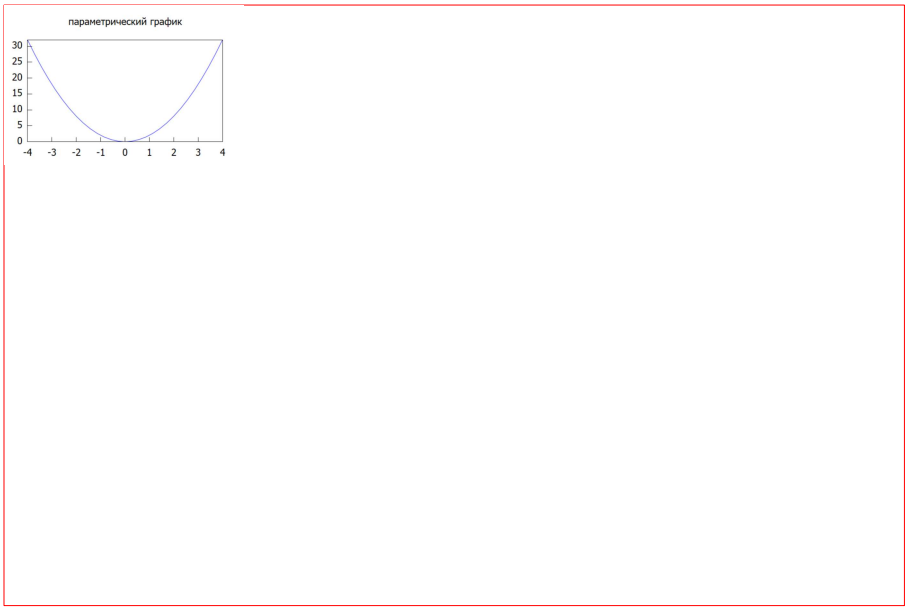
(%t10)

(%t11)

(%o11)

(%i12)
```
with_slider_draw(
t,makelist(i,i,1,10),
   title="параметрический график",
    parametric(
      t·k,
      2·(t·k)^2,
      k,−2,2
      )
)$
```

(%t12)



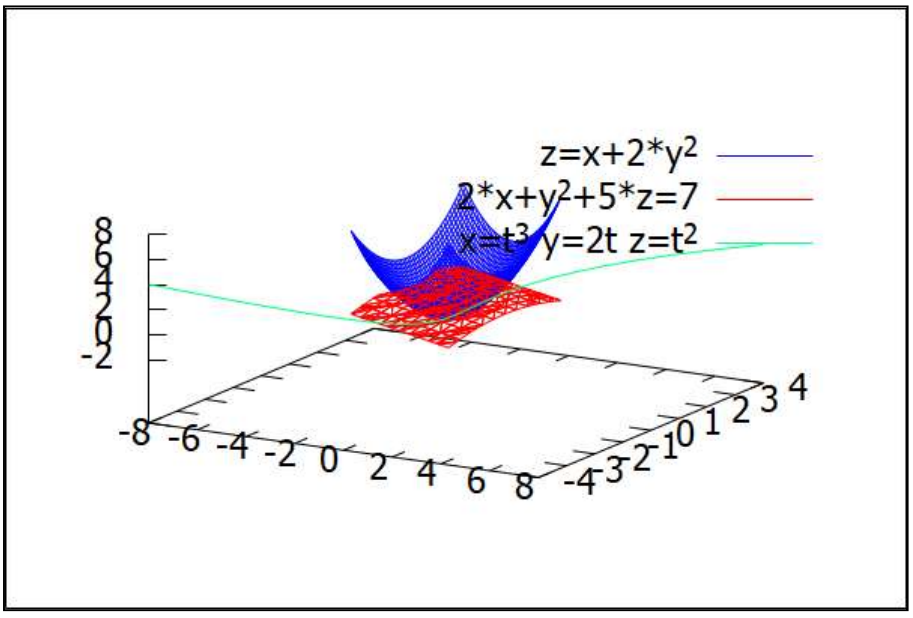
(%t13)

Построение 3х мерных конструкций ничем принципиально не отличается

Статический рисунок с тремя графиками. явное и неявное представление дает поверхности, а параметрическое -линию в пространстве.

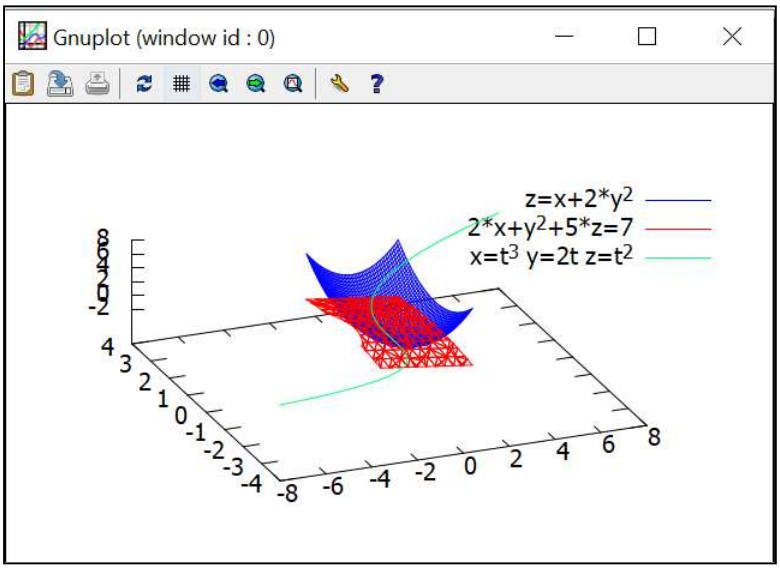
(%i14)
```
wxdraw3d(
  key="z=x+2*y^2",
   explicit(
     x^2+y^2,
     x,−2,2,
     y,−2,2
     ),
   color="#ff0000",
   key="2*x+y^2+5*z=7",
   implicit(
     2·x+y^2+5·z=7,
     x,−2,2,
     y,−2,2,
     z,−2,2
     ),
   color="#00ff80",
   key="x=t^3 y=2t z=t^2",
   parametric(
     t^3,
     2·t,
     t^2,
     t,−2,2
     )

)$
```

(%t14)



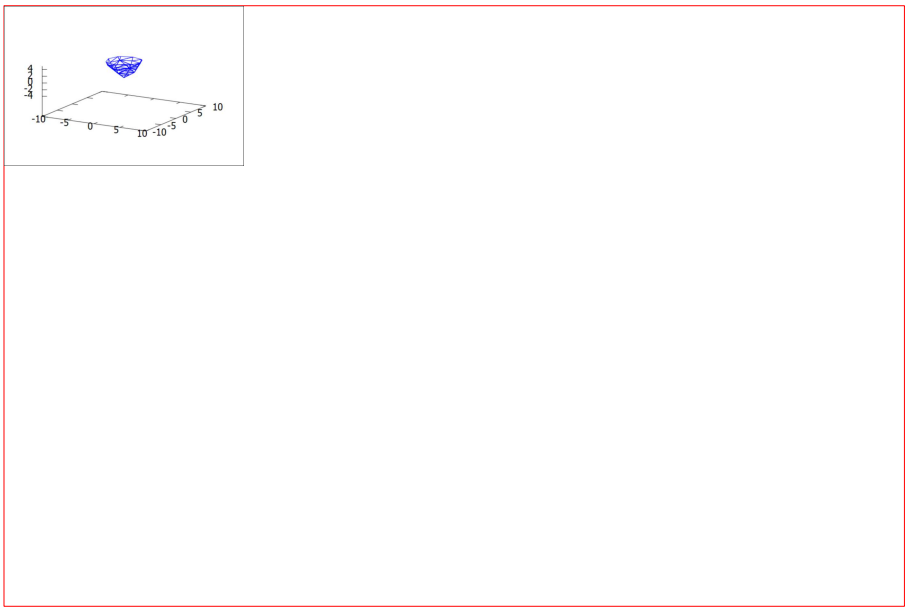
Нажатие на рисунке правой кнопкой мыши дает интерактивное представление в отдельном окне. в котором возможно вращать и перемещать рисунок для лучшего обзора.

Фигура 14:



Примеры использования 3х мерной анимации

(%i15)
```
with_slider_draw3d(
t,makelist(i,i,1,10),
   implicit(
      x^2+y^2=t·z,
      x,−10,10,
      y,−10,10,
      z,−5,5
      )
)$
```



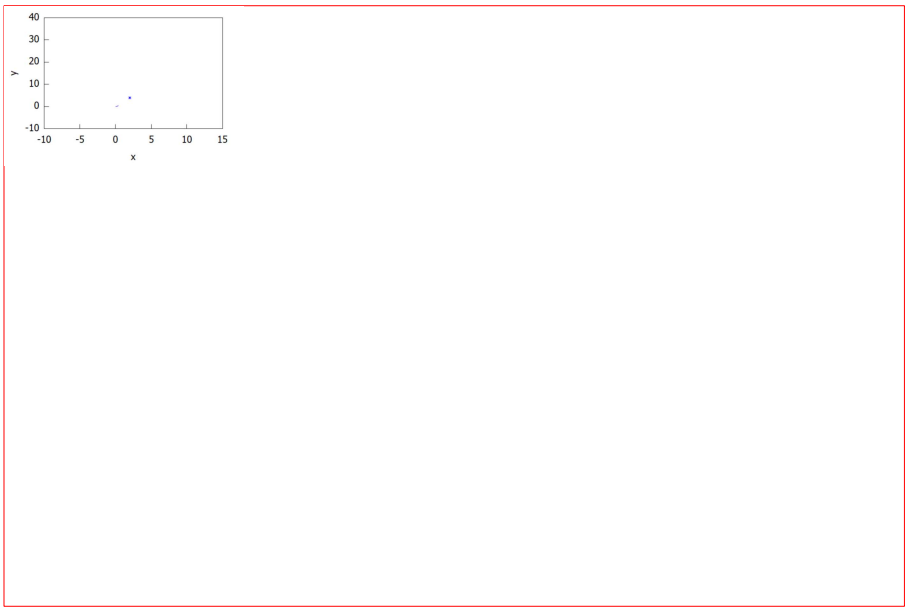
(%t15)

(%t16)

Рост параболы и перемещение точки в зависимости от изменения параметра t, который может играть роль переменной времени для иллюстрации динамически развивающегося процесса.

(%i17)
```
with_slider_draw(
t,makelist(i,i,1,10),
   xlabel="x",
   ylabel="y",
   xrange=[−10,15],
   yrange=[−10,40],
   point_type=3,
   points(transpose(apply('matrix,[[1·t],[2·t]]))),
   parametric(
      r,
      3·r^2,
      r,0,0.2·t
      )
)$
```

(%t17)



(%t18)

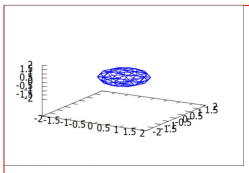
(%i19)
```
with_slider_draw(
t,makelist(i,i,1,50),
   xlabel="t",
   ylabel="y",
   xrange=[−1,12],
   yrange=[−1,220],
   parametric(
      r,
      r^2,
      r,0,0.1·t
      )
)$
```

(%t19)

(%t20)

Расширение сферы. x_voxel=10, y_voxel=10, z_voxel=10,
параметры отвечающие за точность отрисовки. чем они больше тем четче рисунок,
но выше затраты ресурсов.

(%i21)
```
with_slider_draw3d(
t,makelist(i,i,1,10),
   nticks=20,
   x_voxel=10,
   y_voxel=10,
   z_voxel=10,
   implicit(
      x^2+y^2+z^2=4·0.1·t,
      x,−2,2,
      y,−2,2,
      z,−2,2
      )
)$
```



(%t21)

(%t22)

8 Дифференцирование функций.

Мы уже знаем, что определение производной функции начинается с выбора подпункта дифференцировать пункта Анализ главного меню. Указав в Выражение функцию, производную которой требуется найти, переменную, по которой эта производная должна быть найдена и в окошке Умножить порядок производной мы получим результат. Если мы исследуем функцию многих переменных, то мы должны указывать по какой переменной берем производную.

```
(%i23) kill(all);
(%o0)  done
```

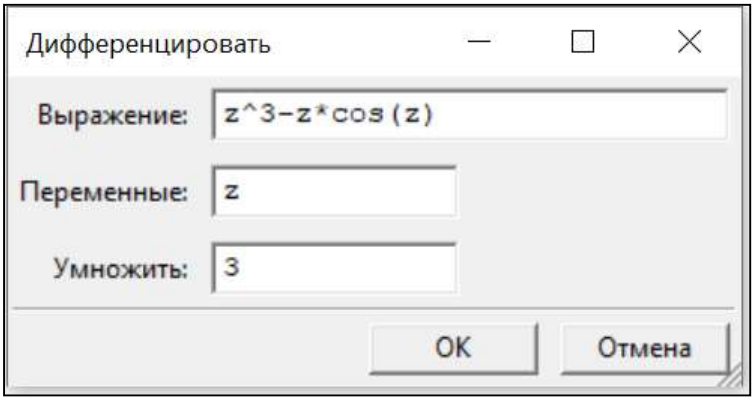
Задаем функции

```
(%i2) f(x,y):=x^2·y+sin(2·x+y);
      r(z):=z^3−z·cos(z);
(%o1) f(x,y):=x^2 y+sin(2 x+y)
(%o2) r(z):=z^3−z cos(z)
```

Идем Анализ-Дифференцировать. Возьмем производную третьего порядка от функции

```
(%i3) 'diff(r(z),z,3);
(%o3) d^3/dz^3 (z^3−z cos(z))
```

Фигура 15:




```
(%i4) diff(z^3−z·cos(z),z,3);
(%o4) −z sin(z)+3 cos(z)+6
```

Можно использовать непосредственно имя функции

```
(%i5) diff(f(x,y),x,1);
(%o5) 2 cos(y+2 x)+2 x y

(%i6) diff(f(x,y),y,1);
(%o6) cos(y+2 x)+x^2

(%i7) diff(f(x,y),y); /* первый порядок можно не указывать*/
(%o7) cos(y+2 x)+x^2
```

При вычислении частных производных n-го порядка функции нескольких переменных в поле Переменные также записывается требуемая переменная, а в поле Умножить - искомый порядок. Если возникает необходимость определения смешанной частной производной, то в поле Переменные записывают через запятую в требуемом порядке переменные, по которым функцию будут дифференцировать, а в поле Умножить соответствующие кратности дифференцирования. В каком порядке переменные в том порядке и кратности дифференцирования.

Пример: найти смешанную производную от функции f(x,y) второго порядка по x и первого по y

```
(%i8) 'diff(f(x,y),y,1,x,2);
(%o8) d^3/(d x^2 d y) (sin(y+2 x)+x^2 y)

(%i9) diff(f(x,y),x,2,y,1);
(%o9) 2-4 cos(y+2 x)

(%i10) diff(f(x,y),y,1,x,2);
(%o10) 2-4 cos(y+2 x)

(%i11) diff(f(x,y),x,1,y,1,x,1);
(%o11) 2-4 cos(y+2 x)
```

У непрерывных функций результат смешанного дифференцирования не зависит от порядка.

Если не указана переменная, то мы получаем дифференциал.
В максима del(f) обозначает дифференциал. Как его пишут в математике df.

```
(%i12) diff(f);
(%o12) del(f)
```

```
(%i13) diff(f(x,y));
(%o13) (cos(y+2 x)+x^2) del(y)+(2 cos(y+2 x)+2 x y)
       del(x)
```

Производную у по х в случае неявно заданной функции можно искать двумя способами.
Допустим стоит задача найти производную у по х в случае заданной зависимости $y=\sin(2x+y)$.

Первый. Будем использовать тот вид выражения который задан.
Однако в максима считается что все переменные независимы друг от друга. Если мы прямо возьмем производную у по х, то получим 0.

```
(%i14) diff(y,x,1);
(%o14) 0
```

Для того чтобы пакет мог найти эту производную, необходимо чтобы она существовала. для этого мы указываем, что у зависит от х.

```
(%i15) depends(y,x);    /*Указываем что переменная у зависит от переменной х.*/
(%o15) [y(x)]

(%i16) a:diff(y,x,1); /* присваиваем символьной переменной а смысл */
                      /* первой производной у по х */
(%o16) d/dx y

(%i17) a;   /* проверка а*/
(%o17) d/dx y
```

Теперь у нас признано, что у функция от х. Присваиваем символьной переменной f значение нашего выражения и берем производную по х от обеих частей равенства. Присваиваем результат дифференцирования переменной g.

```
(%i18) f:y=sin(2·x+y);
(%o18) y=sin(y+2 x)

(%i19) g:diff(f,x);
(%o19) d/dx y=(d/dx y+2) cos(y+2 x)
```

И теперь остается только решить уравнение g относительно искомой производной. Можно использовать символьную переменную a. Но можно и прямо в качестве переменной относительно которой работает функция solve указать производную

```
(%i20) solve(g,a);
```

$$(\%o20)\quad \left[\frac{d}{dx}\,y=-\frac{2\cos(y+2x)}{\cos(y+2x)-1}\right]$$

```
(%i21) solve(g,diff(y,x));
```

$$(\%o21)\quad \left[\frac{d}{dx}\,y=-\frac{2\cos(y+2x)}{\cos(y+2x)-1}\right]$$

И так и так результат одинаков

Второй способ заключается в представлении выражения в классическом виде для неявной функции F(x,y)=0 и применении соотношения dy/dx=- (∂F/∂x)/(∂F/∂y). Но вначале мы должны отменить зависимость y от x, иначе при выполнении дифференцирований у нас будут лишние слагаемые.

```
(%i22) kill(all);
(%o0)  done

(%i1)  F(x,y):=sin(2·x+y)−y;
(%o1)  F(x,y):=sin(2x+y)−y

(%i2)  a:diff(F(x,y),x,1);  /* dF_dx  */
(%o2)  2cos(y+2x)

(%i3)  b:diff(F(x,y),y,1);   /* dF_dy*/
(%o3)  cos(y+2x)−1

(%i4)  dy_dx:−a/b;
```

$$(\%o4)\quad -\frac{2\cos(y+2x)}{\cos(y+2x)-1}$$

```
(%i5)  dy_dx;
```

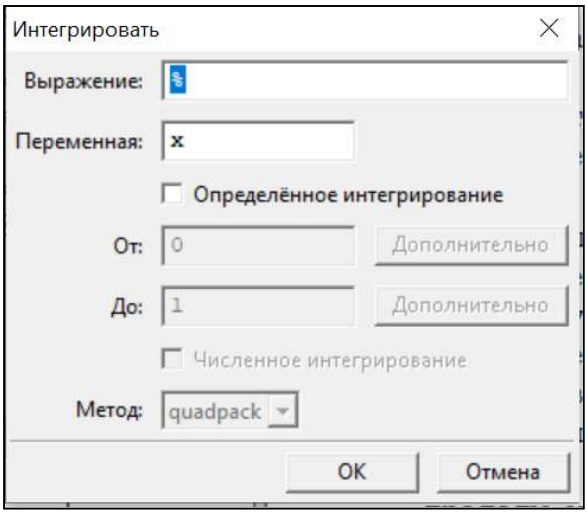
$$(\%o5)\quad -\frac{2\cos(y+2x)}{\cos(y+2x)-1}$$

Получили знакомый результат.

9 Интегрирование выражений.

Пакет позволяет вычислять символьно неопределенный интеграл и символьно и численно определенные интегралы.
Путь Анализ - Интегрирование. Откроется панель

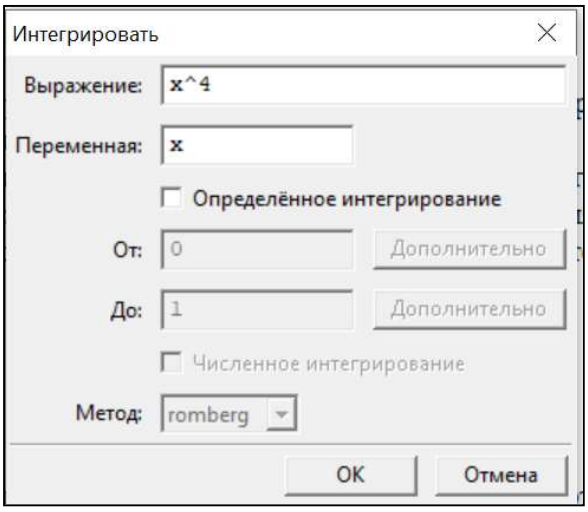
Фигура 16: панель интегрирования



В появившемся диалоговом окне в поле Выражение мы записываем подынтегральное выражение, в поле Переменная - переменную, по которой происходит интегрирование. В том случае, если ищем определенный интеграл, то ставим галочку в поле Определенное интегрирование.
Как результат, появляется возможность указать нижний и верхний пределы интегрирования (поля От и До). Если ставим галочку численное инегрирование, то появляется возможность выбрать метод интегрирования

Неопределенный интеграл от функции x^4

Фигура 17:




```
(%i6)  kill(all);
(%o0)  done
```

(%i1) integrate(x^4, x);

(%o1) $\frac{x^5}{5}$

Определенный интеграл, пределы интегрирования по x от 0 до 4. Сначала изображение, потом само вычисление

(%i2) 'integrate(x, x, 0, 4);

(%o2) $\int_0^4 x$

(%i3) integrate(x, x, 0, 4);

(%o3) 8

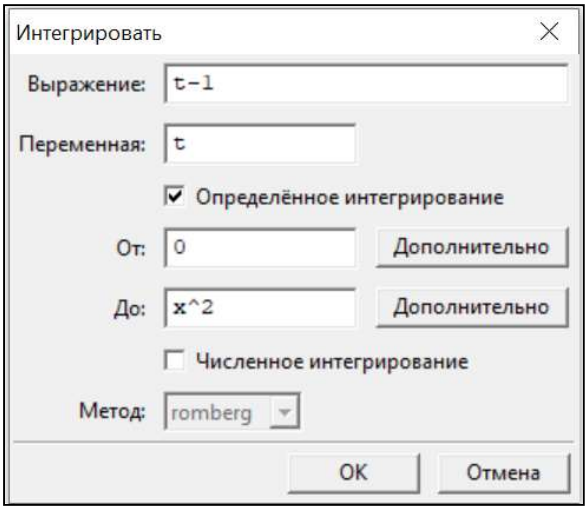
Интеграл с переменным верхним пределом

(%i4) 'integrate(t−1, t, 0, x^2);

(%o4) $\int_0^{x^2} t-1$

При помощи графического интерфейса задаем переменный верхний предел интегрирования как функцию от x

Фигура 18:



(%i5) integrate(t−1, t, 0, x^2);

(%o5) $\frac{x^4-2x^2}{2}$

К определенному интегралу приводит решение самых разнообразных задач.Например.
Найти площадь криволинейной трапеции, если она ограничена:
а) графиком функции y = x^3 - 9*x, прямыми x = 13, x = 24 и
осью абсцисс;
б) параболой y = (x - 1)^2 и гиперболой x^2 - 0,5*y^2 = 1.

Задаем выражение функции.

(%i6) a: x^3 − 9·x;

(%o6) $x^3 - 9\,x$

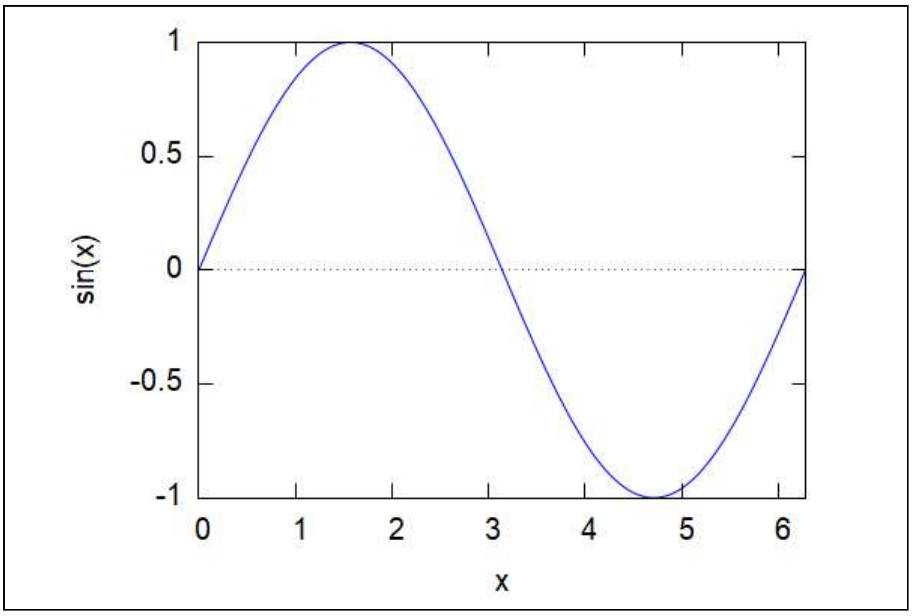
Для того, чтобы получить площадь между кривой и осью абсцисс, нужно быть уверенным, что наша функция на заданном промежутке не пересекает эту ось. В случае когда значения функции отрицательны, то и площадь будет отрицательной. Например поиск площади от функции синус на промежутке от 0 до 2π дает 0! Потому что синус на одной части промежутка положителен, а на другой отрицателен.

(%i7) integrate(sin(x), x, 0, 2·%pi);

(%o7) 0

(%i8) wxplot2d([sin(x)], [x,0,2·%pi])$

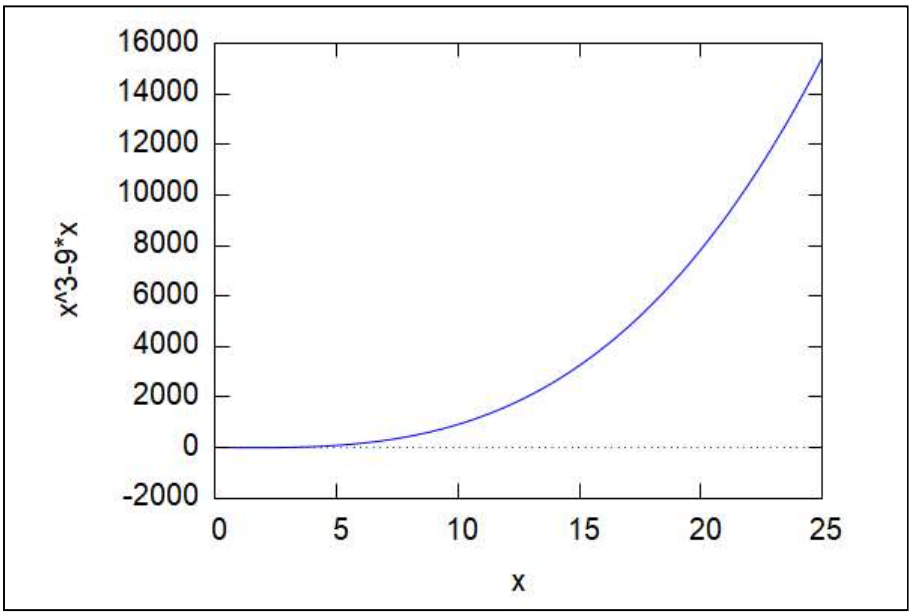
(%t8)



Нарисуем нашу функцию

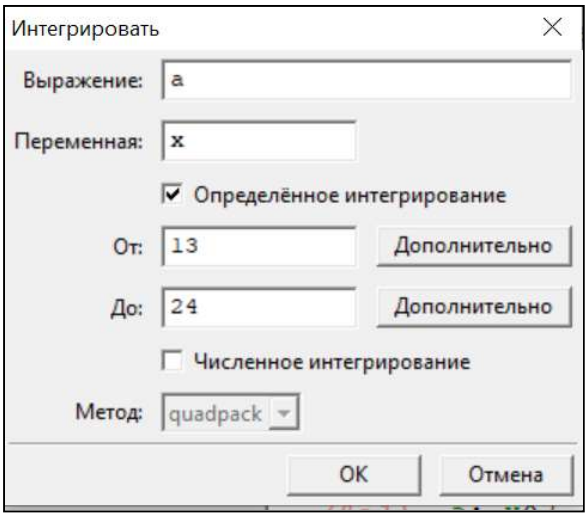
(%i9) wxplot2d([x^3−9·x], [x,0,25])$

(%t9)



Она положительна на всем участке интегрирования. Смело вычисляем интеграл

Фигура 19: Вместо выражения подставляем символьную переменную a.



(%i10) integrate(a, x, 13, 24);

(%o10) $\frac{295889}{4}$

(%i11) float(%); /* Просим выдать результат в виде десятичной дроби */

(%o11) 73972.25

Неопределенный от заданной функции интеграл будет

(%i12) integrate(a, x);

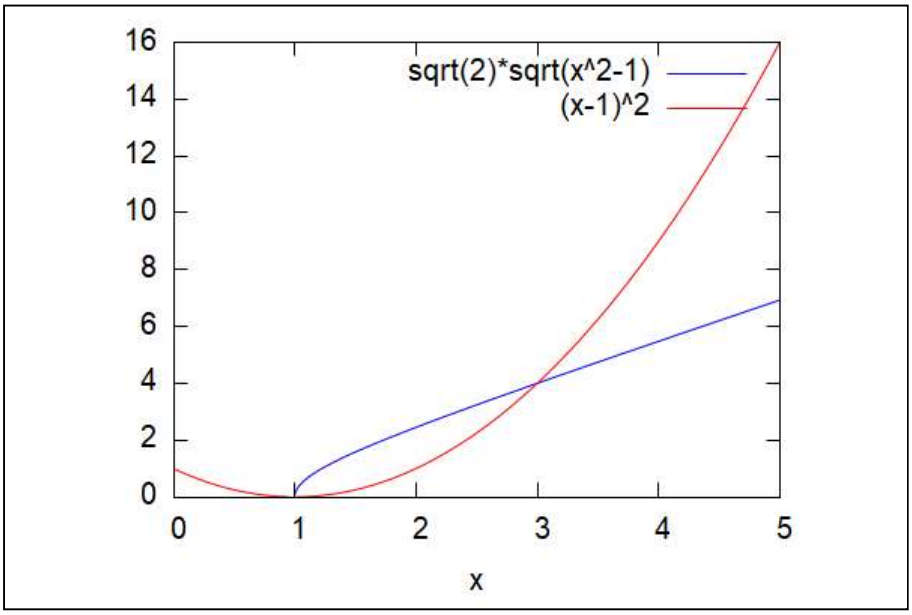
(%o12) $\frac{x^4}{4} - \frac{9\,x^2}{2}$

Вторая задача.
Для ее решения второй задачи нам необходимо удостоверится есть ли вообще область площадь которой мы ищем. Для этого нужно нарисовать эти функции.

Посмотрим на графике картинку пересечения параболы y = (x - 1)^2 и гиперболы x^2 - 0,5*y^2 = 1. Нам необходимо задать только положительную ветку гиперболы для этого явно выражаем y через x.
x^2 - 0,5*y^2 = 1 -> x^2-1=y^2/2 -> y^2=2*(x^2-1) -> y=(+/-)sqrt(2*(x^2-1))

(%i13)
```
wxplot2d([sqrt(2·(x^2−1)),(x−1)^2], [x,0,5],
 [nticks,18])$
```
plot2d: expression evaluates to non-numeric value somewhere in plotting range.

(%t13)



Ищем точки пересечения наших кривых, решая систему уравнений

(%i14)
```
solve([y=(x−1)^2, x^2−y^2/2=1], [x,y]);
```
(%o14) `[[x=-%i,y=2 %i],[x=%i,y=-2 %i],[x=1,y=0],[x=3,y=4]]`

Пределы будут от x=1 до x=3

Второй способ. Сначала исключаем переменную y из системы уравнений, таким образом получаем одно уравнение для переменной x. А затем его решаем. Этот способ надежней, потому что не всегда функция solve дает сразу решение системы уравнений. Тем более, что нам значения y по большому счету и не нужны. нас интересуют только значения x как пределы интегрирования.

```
(%i17) a:y=(x−1)^2;
       b:x^2−y^2/2=1;
       eliminate([a,b],[y]);
```

$$(\%o15)\quad y = (x-1)^2$$

$$(\%o16)\quad x^2 - \frac{y^2}{2} = 1$$

$$(\%o17)\quad [-x^4+4x^3-4x^2+4x-3]$$

```
(%i18) solve([−x^4+4·x^3−4·x^2+4·x−3], [x]);
```

$$(\%o18)\quad [x=3, x=1, x=-\%i, x=\%i]$$

Пределы по x получили те же самые.

Первый вариант определения площади: отнимем друг друга площади под кривыми

```
(%i19) c:integrate(sqrt(2·(x^2−1)), x, 1, 3);
```

$$(\%o19)\quad \sqrt{2}\left(\frac{\log(2)}{2} - \frac{\log(2^{5/2}+6)-3\,2^{3/2}}{2}\right)$$

```
(%i20) c;
```

$$(\%o20)\quad \sqrt{2}\left(\frac{\log(2)}{2} - \frac{\log(2^{5/2}+6)-3\,2^{3/2}}{2}\right)$$

```
(%i21) d:integrate((x−1)^2, x, 1, 3);
```

$$(\%o21)\quad \frac{8}{3}$$

```
(%i22) float(c−d);
```

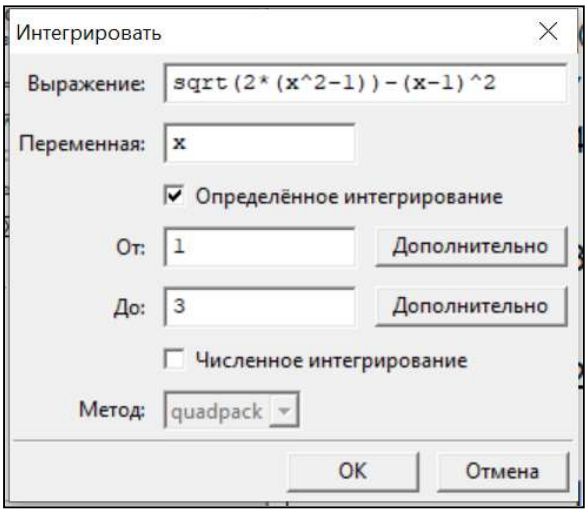
$$(\%o22)\quad 2.08688285305287$$

Второй способ. Интегрируем разность функций. Для этого вводим символьную переменную (от верхней функции отнимаеим нижнюю)

```
(%i23) f:sqrt(2·(x^2−1))−(x−1)^2;
```

$$(\%o23)\quad \sqrt{2}\sqrt{x^2-1}-(x-1)^2$$

Фигура 20:



(%i24) integrate(sqrt(2·(x^2−1))−(x−1)^2, x, 1, 3), numer;

rat: replaced 0.5 by 1/2 = 0.5
rat: replaced 1.4142135623731 by 22619537/15994428 = 1.4142135623731
rat: replaced 0.5 by 1/2 = 0.5
rat: replaced 2.82842712474619 by 22619537/7997214 = 2.82842712474619
rat: replaced -1.0 by -1/1 = -1.0
rat: replaced 0.333333333333333 by 1/3 = 0.333333333333333
rat: replaced 0.333333333333333 by 1/3 = 0.333333333333333
rat: replaced 3.0 by 3/1 = 3.0
rat: replaced 0.5 by 1/2 = 0.5
rat: replaced 0.5 by 1/2 = 0.5
rat: replaced 0.5 by 1/2 = 0.5
rat: replaced 0.5 by 1/2 = 0.5
rat: replaced -0.5 by -1/2 = -0.5
rat: replaced 0.5 by 1/2 = 0.5
rat: replaced 0.5 by 1/2 = 0.5
rat: replaced 0.5 by 1/2 = 0.5
rat: replaced -0.5 by -1/2 = -0.5
rat: replaced 0.5 by 1/2 = 0.5
rat: replaced 0.5 by 1/2 = 0.5
rat: replaced 0.5 by 1/2 = 0.5
rat: replaced 1.5 by 3/2 = 1.5
rat: replaced 0.5 by 1/2 = 0.5
rat: replaced 1.5 by 3/2 = 1.5
rat: replaced 1.5 by 3/2 = 1.5
rat: replaced 1.5 by 3/2 = 1.5
rat: replaced 1.5 by 3/2 = 1.5
rat: replaced 0.5 by 1/2 = 0.5
rat: replaced 0.5 by 1/2 = 0.5
rat: replaced 2.08688285305287 by 13495189/6466673 = 2.08688285305288

(%o24) 2.08688285305288

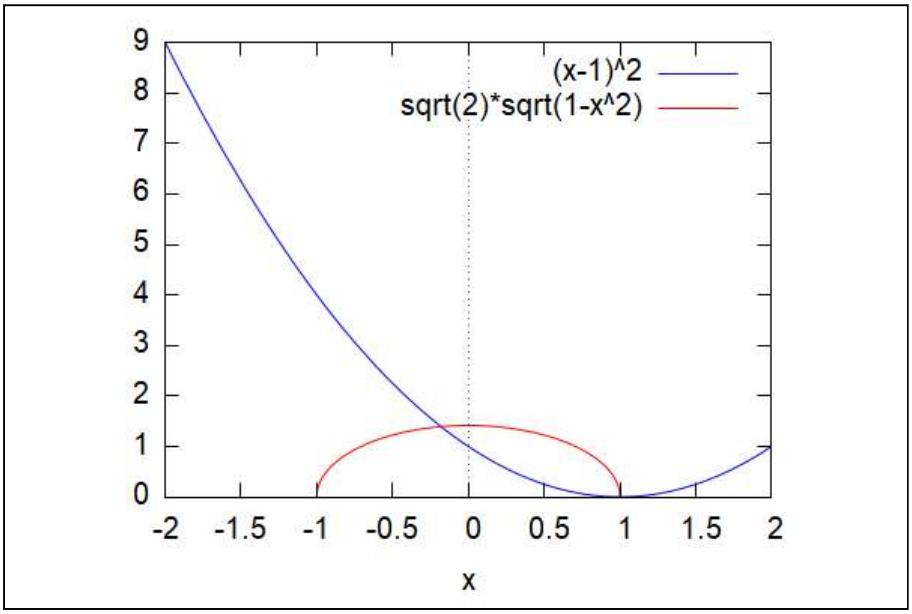
Как и в случае вычисления пределов, при вычислении интегралов полезно применять упрощения подинтегрального выражения.
Это можно делать различными способами.
1. Прямым упрощением подинтегрального выражения.
2. Разложением подинтегрального выражения на простые дроби.
3. Заменой переменного.
4. Интегрированием по частям.
5. Разложение подинтегрального выражения в ряд Тейлора.
Эти техники доступны при углубленном изучении пакета

Задание
а)Найти площадь криволинейной трапеции, если она ограничена:
параболой y = (x - 1)^2 и эллипсом x^2 + 0,5*y^2 = 1.

(%i25)
```
wxplot2d([(x−1)^2, sqrt(2·(1−x^2))], [x,−2,2],
 [nticks,18])$
```

plot2d: expression evaluates to non-numeric value somewhere in plotting range.

(%t25)



(%i28)
```
a:y=(x−1)^2;
b:x^2+y^2/2=1;
eliminate([a,b],[y]);
```

(%o26) $y=(x-1)^2$

(%o27) $\frac{y^2}{2}+x^2=1$

(%o28) $[x^4-4x^3+8x^2-4x-1]$

(%i29) float(solve(x^4−4·x^3+8·x^2−4·x−1=0,x));

(%o29) [x=−1.5969670278346 (0.866025403784439 %i−0.5)+ 0.417458003231683 (−0.866025403784439 %i−0.5)+1.0,x= 0.417458003231683 (0.866025403784439 %i−0.5)− 1.5969670278346 (−0.866025403784439 %i−0.5)+1.0,x=− 0.179509024602913,x=1.0]

(%i30) integrate(sqrt(2·(1−x^2))−(x−1)^2, x,−0.1795090246029128, 1);

rat: replaced 1.17950902460291 by 26141488/22163025 = 1.17950902460291
rat: replaced -0.179509024602913 by -8003491/44585452 = -0.179509024602913
rat: replaced 1.17950902460291 by 26141488/22163025 = 1.17950902460291
rat: replaced 1.17950902460291 by 26141488/22163025 = 1.17950902460291
rat: replaced -0.179509024602913 by -8003491/44585452 = -0.179509024602913
rat: replaced 1.17950902460291 by 26141488/22163025 = 1.17950902460291
rat: replaced -0.213660650264717 by -13347027/62468344 = -0.213660650264716
rat: replaced 1.17950902460291 by 26141488/22163025 = 1.17950902460291
rat: replaced -0.179509024602913 by -8003491/44585452 = -0.179509024602913
rat: replaced 1.17950902460291 by 26141488/22163025 = 1.17950902460291
rat: replaced 1.17950902460291 by 26141488/22163025 = 1.17950902460291
rat: replaced -0.179509024602913 by -8003491/44585452 = -0.179509024602913
rat: replaced 1.17950902460291 by 26141488/22163025 = 1.17950902460291
rat: replaced 1.17950902460291 by 26141488/22163025 = 1.17950902460291
rat: replaced -0.179509024602913 by -8003491/44585452 = -0.179509024602913
rat: replaced 1.17950902460291 by 26141488/22163025 = 1.17950902460291
rat: replaced 0.178540242401081 by 3796276/21262859 = 0.178540242401081

(%o30)

$$(4980958464133112^{3/2}\,\%pi+22232538289401\,2^{11/2}-2179643449946075)/3984766771306488$$

(%i31) float(%);

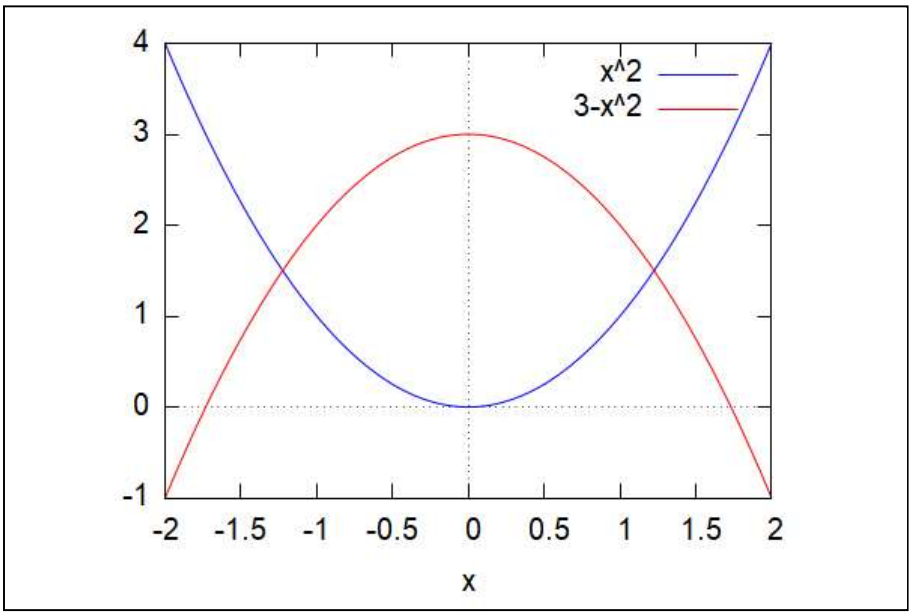
(%o31) 0.816220783174531

б) Найти объем тела, образованного вращением фигуры, ограниченной линиями y =x^2 и y=3-x^2 вокруг оси абсцисс.

Посмотрим на рисунке как они пересекаются.

(%i32) wxplot2d([x^2,3−x^2], [x,−2,2])$

(%t32)



Очистка всех переменных.

(%i33) kill(all);

(%o0) done

Формула определения объема фигуры образованной вращением вокруг OX кривой с образующей равной f(x) в пределах от a до b.

(%i1) V='integrate(%pi·f(x)^2,x,a,b);

(%o1) $$V = \%pi \int_a^b f(x)^2$$

(%i2) d:(3−x^2)^2−(x^2)^2; /* смотрим сразу разность двух функций */

(%o2) $$(3-x^2)^2-x^4$$

(%i3) f:x^2=3−x^2; /* приравниваем функции друг другу и находим корни*/
/* уравнения, которые будут точками пересечения */

(%o3) $$x^2=3-x^2$$

(%i4) allroots(%);

(%o4) [x=1.22474487139159,x=-1.22474487139159]

(%i5) integrate(d, x, −1.224744871391589, 1.224744871391589);

rat: replaced 2.44948974278318 by 46099201/18819920 = 2.44948974278318
rat: replaced -1.22474487139159 by -46099201/37639840 = -1.22474487139159
rat: replaced 1.22474487139159 by 46099201/37639840 = 1.22474487139159
rat: replaced 2.44948974278318 by 46099201/18819920 = 2.44948974278318
rat: replaced -7.34846922834953 by -138297603/18819920 = -7.34846922834954
rat: replaced 7.34846922834953 by 138297603/18819920 = 7.34846922834954

(%o5) $\frac{138297603}{9409960}$

Применение функции solve дало бы координаты точек пересечения в другом виде:

(%i6) solve([x^2=3−x^2], [x]);

(%o6) $\left[x=-\frac{\sqrt{3}}{\sqrt{2}},x=\frac{\sqrt{3}}{\sqrt{2}}\right]$

Пример с использованием подстановки.
Использовать подстановку u = x^2, чтобы вычислить интеграл от 5*x*sin(x^2) по x

(%i7) a:5·x·sin(x^2);

(%o7) $5\,x\,\sin\left(x^2\right)$

(%i9) INTEGRAND:(5·x)·sin(x^2)·del(x); /* Запись подинтегрального выражения*/
solve(diff(u)=diff(x^2),del(x)); /*берем производную от обеих частей и выражаем dx*/

(%o8) $5\,x\,\sin\left(x^2\right)\,\mathrm{del}\left(x\right)$

(%o9) $\left[\mathrm{del}\left(x\right)=\frac{\mathrm{del}\left(u\right)}{2\,x}\right]$

(%i10) %[1]; /*проверка*/

(%o10) $\mathrm{del}\left(x\right)=\frac{\mathrm{del}\left(u\right)}{2\,x}$

(%i11) subst(rhs(%),del(x),INTEGRAND); /* замена в интеграле dx через du*/

(%o11) $\frac{5\,\sin\left(x^2\right)\,\mathrm{del}\left(u\right)}{2}$

(%i12) subst(u,x^2,%); /* замена переменной x на переменную u*/

(%o12) $\frac{5\,\sin\left(u\right)\,\mathrm{del}\left(u\right)}{2}$

Теперь интеграл полностью выражается через u , что достаточно легко использовать.
Для полноты картины мы используем wxMaxima для вычисления интеграла, затем мы преобразуем результат в функцию исходной переменной x:

```
(%i13) integrate(coeff(%,del(u)),u);
```

$$(\%o13) \quad -\frac{5\cos(u)}{2}$$

```
(%i14) subst(x^2,u,%);
```

$$(\%o14) \quad -\frac{5\cos(x^2)}{2}$$

Мы получили результат для исходного интеграла. Можем проверить его дифференцированием.

```
(%i15) diff(%,x,1);
```

$$(\%o15) \quad 5\,x\sin(x^2)$$

Что и требовалось доказать. Интеграл сам по себе простой, но здесь приведен прием замены. переменной

10 Линейная алгебра. Работа с матрицами и векторами.

Во введении мы уже рассматривали способы создания матриц и введения матриц. Напомню, что это две разные операции. Когда вы создаете матрицу , то вы создаете конструкцию состоящую из символьных элементов. Когда же вы вводите матрицу, то вы задаете конкретное числовое значение элементам матрицы. Вспомним, что у нас было:

```
(%i16) kill(all);
(%o0)  done
```

Создание и введение матриц

Создание матрицы Матрица - > Создать матрицу

```
(%i1)  A_simvol: genmatrix(a, 3, 3);
```

$$(\%o1)\quad \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{pmatrix}$$

Это простейшая версия команды genmatrix(имя, d1, d2).
В этом случае индексы по умолчанию будут от единицы до соответствующего значения рамерности. Однако вариант команды genmatrix(имя,d1, d2, c1, c2) создаст матрицу с размерностью имя[c1:d1,c2:d2].

Проверка значения A_simvol

```
(%i2)  A_simvol;
```

$$(\%o2)\quad \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{pmatrix}$$

Обращение и проверка элемента матрицы

```
(%i3)  a[1,1];
(%o3)  a1,1

(%i4)  A_simvol[1,1];
(%o4)  a1,1
```

Ввод матрицы Матрица - > Ввести матрицу

```
(%i5) C: matrix(
 [1,4,h,11],
 [3,8,4,9]
);
```

$$(\%o5)\quad \begin{pmatrix} 1 & 4 & h & 11 \\ 3 & 8 & 4 & 9 \end{pmatrix}$$

Проверка матрицы C

```
(%i6) C;
```

$$(\%o6)\quad \begin{pmatrix} 1 & 4 & h & 11 \\ 3 & 8 & 4 & 9 \end{pmatrix}$$

Итак вы способны создавать символьные и числовые матрицы. И те и другие подчиняются общим правилам работы с матрицами. Единственное отличие в том, что у символьных матриц в результате выполнения каких либо операция результат будет символьным, а у числовых - числом.

К числу основных операций, которые выполняются над матрицами, относятся операции сложения, вычитания, умножения, деления и возведения в степень.
При работе с матрицами в wxMaxima следует учитывать следующие особенности:
1) возведение матрицы в степень, обозначенное как ^ , сводится к поэлементному возведению в степень компонент матрицы;
2) возведение в степень вида ^^ выполняется как умножение матрицы саму на себя требуемой число раз;
3) возведение в степень ^^(- 1) означает поиск обратной матрицы;
4) произведение матриц с использованием символа * рассматривается как поэлементное умножение соответствующих элементов;
5) произведение матриц, обозначаемое как точка, производится по правилу умножения матриц {«строка умножается на столбец»).
Рассмотрим примеры действия с матрицами.
Введем две матрицы A и B: Найти сумму, разность, произведение, частное матриц, куб матрицы A, матрицу, обратную к матрице A.

Решение. Используя главное меню, введем данные матрицы:

```
(%i7) A: matrix(
 [2,−1,3],
 [6,7,−2],
 [5,4,9]
);
```

$$(\%o7)\quad \begin{pmatrix} 2 & -1 & 3 \\ 6 & 7 & -2 \\ 5 & 4 & 9 \end{pmatrix}$$

(%i8)
```
B: matrix(
 [1,3,6],
 [5,−9,4],
 [7,6,57]
);
```

(%o8)
$$\begin{pmatrix} 1 & 3 & 6 \\ 5 & -9 & 4 \\ 7 & 6 & 57 \end{pmatrix}$$

Сумма и разность матриц

(%i9)
```
R:A+B;
```

(%o9)
$$\begin{pmatrix} 3 & 2 & 9 \\ 11 & -2 & 2 \\ 12 & 10 & 66 \end{pmatrix}$$

(%i10)
```
R;
```

(%o10)
$$\begin{pmatrix} 3 & 2 & 9 \\ 11 & -2 & 2 \\ 12 & 10 & 66 \end{pmatrix}$$

(%i11)
```
A−B;
```

(%o11)
$$\begin{pmatrix} 1 & -4 & -3 \\ 1 & 16 & -6 \\ -2 & -2 & -48 \end{pmatrix}$$

Поэлементное умножение
и деление матриц:

(%i12)
```
P:A·B;
```

(%o12)
$$\begin{pmatrix} 2 & -3 & 18 \\ 30 & -63 & -8 \\ 35 & 24 & 513 \end{pmatrix}$$

(%i13)
```
D:A/B;
```

(%o13)
$$\begin{pmatrix} 2 & -\frac{1}{3} & \frac{1}{2} \\ \frac{6}{5} & -\frac{7}{9} & -\frac{1}{2} \\ \frac{5}{7} & \frac{2}{3} & \frac{3}{19} \end{pmatrix}$$

(%i14) A·C;

fullmap: arguments must have same formal structure.
-- an error. To debug this try: debugmode(true);

При поэлементном действии матрицы должны быть одинаковой размерности.
A и C различаются.

Произведение матриц:

(%i15) A.B;

(%o15)
$$\begin{pmatrix} 18 & 33 & 179 \\ 27 & -57 & -50 \\ 88 & 33 & 559 \end{pmatrix}$$

Куб матрицы A, это матрица сама на себя 3 умножается:

(%i16) A^^3;

(%o16)
$$\begin{pmatrix} 219 & 148 & 348 \\ 228 & 145 & -64 \\ 952 & 686 & 911 \end{pmatrix}$$

Поэлементное умножение элементов матрицы на себя

(%i17) A^3;

(%o17)
$$\begin{pmatrix} 8 & -1 & 27 \\ 216 & 343 & -8 \\ 125 & 64 & 729 \end{pmatrix}$$

Поиск матрицы, обратной к матрице A, кроме записи A^^(-1),
можно осуществить и с помощью главного меню. Для этого выделим матрицу A
и вызовем функцию обратить матрицу в пункте Алгебра.
Применяется команда invert()

(%i18) A;

(%o18)
$$\begin{pmatrix} 2 & -1 & 3 \\ 6 & 7 & -2 \\ 5 & 4 & 9 \end{pmatrix}$$

(%i19) A_1:invert(A);

(%o19)
$$\begin{pmatrix} \frac{71}{173} & \frac{21}{173} & -\frac{19}{173} \\ -\frac{64}{173} & \frac{3}{173} & \frac{22}{173} \\ -\frac{11}{173} & -\frac{13}{173} & \frac{20}{173} \end{pmatrix}$$

Проверим, умножив ее на A

(%i20) A.A_1;

(%o20)
$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Получили единичную матрицу

Сложение матрицы с числом и умножение на число приводит как всегда
к увеличению элементов матрицы на это число(символ) или умножению
на это число(символ) соответственно.
Вычитание и деление есть подобные операции и мы их приводить не будем

(%i24) A+5; A·3;
A+x; A·k;

(%o21)
$$\begin{pmatrix} 7 & 4 & 8 \\ 11 & 12 & 3 \\ 10 & 9 & 14 \end{pmatrix}$$

(%o22)
$$\begin{pmatrix} 6 & -3 & 9 \\ 18 & 21 & -6 \\ 15 & 12 & 27 \end{pmatrix}$$

(%o23)
$$\begin{pmatrix} x+2 & x-1 & x+3 \\ x+6 & x+7 & x-2 \\ x+5 & x+4 & x+9 \end{pmatrix}$$

(%o24)
$$\begin{pmatrix} 2\,k & -k & 3\,k \\ 6\,k & 7\,k & -2\,k \\ 5\,k & 4\,k & 9\,k \end{pmatrix}$$

В wxMaxima есть ряд встроенных функций, применение которых возможно напрямую путем их непосредственного вызова или с помощью подпунктов пункта Матрица главного меню:
1) submatrix - возвращает матрицу, полученную из исходной,
удалением соответствующих строк и (или) столбцов.
В качестве параметров передаются сначала номера удаляемых строк,
а затем - исходная матрица и в самом конце - номера удаляемых столбцов.

```
(%i25) M: matrix(       /*   Введем матрицу */
        [1,3,7,9],
        [4,f,8,k],
        [5,7,9,4],
        [7,11,6,2]
       );
```

$$(\%o25)\begin{pmatrix} 1 & 3 & 7 & 9 \\ 4 & f & 8 & k \\ 5 & 7 & 9 & 4 \\ 7 & 11 & 6 & 2 \end{pmatrix}$$

```
(%i26) M[3,1];       /* Проверим ее элементы*/
(%o26) 5

(%i27) M[2,2];
(%o27) f

(%i28) submatrix(1,2,M);   /* Удаляем первую и вторую строки */
```

$$(\%o28)\begin{pmatrix} 5 & 7 & 9 & 4 \\ 7 & 11 & 6 & 2 \end{pmatrix}$$

```
(%i29) G:submatrix(1,M,1,2);  /*Удаляем первую строку и первый и второй столбцы*/
```

$$(\%o29)\begin{pmatrix} 8 & k \\ 9 & 4 \\ 6 & 2 \end{pmatrix}$$

```
(%i30) G;
```

$$(\%o30)\begin{pmatrix} 8 & k \\ 9 & 4 \\ 6 & 2 \end{pmatrix}$$

2)transpose() - выполняет транспонирование матрицы.
Можно вызвать из главного меню

(%i31) transpose(%);

(%o31) $\begin{pmatrix} 8 & 9 & 6 \\ k & 4 & 2 \end{pmatrix}$

3) rank() - возвращает ранг матрицы.

(%i32) rank(%);

(%o32) 2

4) minor() - определяет минор матрицы. Первый аргумент -матрица, второй и третий - индексы строки и столбца соответственно.

(%i33) minor(M,2,3);

(%o33) $\begin{pmatrix} 1 & 3 & 9 \\ 5 & 7 & 4 \\ 7 & 11 & 2 \end{pmatrix}$

(%i34) M[2,3];

(%o34) 8

5) Характеристический полином матрицы вычислячеься функцией charpoly(M,x) (M - матрица , x- переменная, относительно которой строится полином). Корни характеристического полинома являются собственными числами матрицы.

(%i35) charpoly(A,x);

(%o35) $6\,(9-x)+((7-x)\,(9-x)+8)\,(2-x)+3\,(24-5\,(7-x))+10$

(%i36) ratsimp(%);

(%o36) $-x^3+18\,x^2-94\,x+173$

(%i37) solve([%], [x]);

(%o37) $$\left[x=\frac{14\left(\frac{\sqrt{3}\,\%i}{2}+\frac{-1}{2}\right)}{3\left(\frac{\sqrt{34411}}{2\,3^{3/2}}+\frac{41}{2}\right)^{1/3}}+\left(\frac{\sqrt{34411}}{2\,3^{3/2}}+\frac{41}{2}\right)^{1/3}\left(\frac{-1}{2}-\frac{\sqrt{3}\,\%i}{2}\right)+6,\ x=\left(\frac{\sqrt{34411}}{2\,3^{3/2}}+\frac{41}{2}\right)^{1/3}\left(\frac{\sqrt{3}\,\%i}{2}+\frac{-1}{2}\right)+\frac{14\left(\frac{-1}{2}-\frac{\sqrt{3}\,\%i}{2}\right)}{3\left(\frac{\sqrt{34411}}{2\,3^{3/2}}+\frac{41}{2}\right)^{1/3}}+6,\ x=\left(\frac{\sqrt{34411}}{2\,3^{3/2}}+\frac{41}{2}\right)^{1/3}+\frac{14}{3\left(\frac{\sqrt{34411}}{2\,3^{3/2}}+\frac{41}{2}\right)^{1/3}}+6\right]$$

(%i38) solve([−x^3+18·x^2−94·x+173=0],[x]),numer;

rat: replaced 318.62037037037 by 34411/108 = 318.62037037037

(%o38) $$\left[x=1.3838378872495\left(0.866025403784439\,\%i+\frac{-1}{2}\right)+3.37226398385585\left(\frac{-1}{2}-0.866025403784439\,\%i\right)+6,\ x=3.37226398385585\left(0.866025403784439\,\%i+\frac{-1}{2}\right)+1.3838378872495\left(\frac{-1}{2}-0.866025403784439\,\%i\right)+6,\ x=10.7561018711054\right]$$

(%i39) float(%);

(%o39) $$\left[x=1.3838378872495\ (0.866025403784439\,\%i-0.5)+3.37226398385585\ (-0.866025403784439\,\%i-0.5)+6.0,\ x=3.37226398385585\ (0.866025403784439\,\%i-0.5)+1.3838378872495\ (-0.866025403784439\,\%i-0.5)+6.0,\ x=10.7561018711054\right]$$

(%i40) allroots(−x^3+18·x^2−94·x+173);

(%o40) $$\left[x=1.72202751320903\,\%i+3.62194906444732,\ x=3.62194906444732-1.72202751320903\,\%i,\ x=10.7561018711054\right]$$

6) Однако для вычисления собственных чисел и собственных векторов матрицы обычно используют специальные функции: eigenvalues и eigenvectors. Функция eigenvalues вычисляет собственные значения матрицы аналитически, если это возможно. Для нахождения корней характеристического полинома используется функция solve. Результат,возвращаемый eigenvalues - список, содержащий два подсписка. Первый содержит собственные значения, а второй их кратности.
Функция eigenvectors аналитически вычисляет собственные значения и собственные вектора матрицы, если это возможно. Она возвращает список, первый элемент которого - список собсвенных чисел (аналогично "eigenvalues"), а далее идут собственные вектора, каждый из которых представлен как список своих проекций.
Определение: ненулевой вектор , который при умножении на некоторую квадратную матрицу превращается в самого же себя с числовым коэффициентом , называется собственным вектором матрицы .
Число называют собственным значением или собственным числом данной матрицы.
A*u=λu

(%i41)
```
eigenvalues(A),numer;
```

rat: replaced 318.62037037037 by 34411/108 = 318.62037037037

(%o41)
$$[[1.3838378872495\left(0.866025403784439\,\%i+\frac{-1}{2}\right)+3.37226398385585\left(\frac{-1}{2}-0.866025403784439\,\%i\right)+6,3.37226398385585\left(0.866025403784439\,\%i+\frac{-1}{2}\right)+1.3838378872495\left(\frac{-1}{2}-0.866025403784439\,\%i\right)+6,10.7561018711054],[1,1,1]]$$

(%i42)
```
a: matrix(
 [1,1,1],
 [2,2,2],
 [3,3,3]
);
```

(%o42)
$$\begin{pmatrix}1 & 1 & 1\\ 2 & 2 & 2\\ 3 & 3 & 3\end{pmatrix}$$

(%i43)
```
eigenvalues(a);
```

(%o43) $[[0,6],[2,1]]$

(%i44)
```
charpoly(a,x);
```

(%o44) $-2\,(3-x)-3\,(2-x)+((2-x)\,(3-x)-6)\,(1-x)+12$

(%i45) ratsimp(%);

(%o45) $6x^2 - x^3$

(%i46) factor(%);

(%o46) $-(x-6)x^2$

(%i47) eigenvectors(a);

(%o47) [[[0,6],[2,1]],[[1,0,-1],[0,1,-1]],[[1,2,3]]]

(%i48) u:[1,0,−1];

(%o48) [1,0,-1]

(%i49) a.u;

(%o49) $\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$

(%i50) v:[1,2,3];

(%o50) [1,2,3]

(%i51) a.v;

(%o51) $\begin{pmatrix} 6 \\ 12 \\ 18 \end{pmatrix}$

7) Преобразование матрицы к треугольной форме осуществляется методом исключения Гаусса посредством функции echelon(M) (аналогичный результат дает функция triangularize(M)):

(%i52) echelon(A);

(%o52) $\begin{pmatrix} 1 & -\frac{1}{2} & \frac{3}{2} \\ 0 & 1 & -\frac{11}{10} \\ 0 & 0 & 1 \end{pmatrix}$

Определитель матрицы вычисляется командой determinant()

(%i53) determinant(%);

(%o53) 1

(%i54) determinant(a);

(%o54) 0

Проверка операции возведения матрицы в степень

(%i55) S:A^^(−2);

$$(\%o55)\begin{pmatrix} \frac{3906}{29929} & \frac{1801}{29929} & -\frac{1267}{29929} \\ -\frac{4978}{29929} & -\frac{1621}{29929} & \frac{1722}{29929} \\ -\frac{169}{29929} & -\frac{530}{29929} & \frac{323}{29929} \end{pmatrix}$$

(%i56) F:A^^(−1);

$$(\%o56)\begin{pmatrix} \frac{71}{173} & \frac{21}{173} & -\frac{19}{173} \\ -\frac{64}{173} & \frac{3}{173} & \frac{22}{173} \\ -\frac{11}{173} & -\frac{13}{173} & \frac{20}{173} \end{pmatrix}$$

(%i57) F;

$$(\%o57)\begin{pmatrix} \frac{71}{173} & \frac{21}{173} & -\frac{19}{173} \\ -\frac{64}{173} & \frac{3}{173} & \frac{22}{173} \\ -\frac{11}{173} & -\frac{13}{173} & \frac{20}{173} \end{pmatrix}$$

(%i58) F^^2;

$$(\%o58)\begin{pmatrix} \frac{3906}{29929} & \frac{1801}{29929} & -\frac{1267}{29929} \\ -\frac{4978}{29929} & -\frac{1621}{29929} & \frac{1722}{29929} \\ -\frac{169}{29929} & -\frac{530}{29929} & \frac{323}{29929} \end{pmatrix}$$

(%i59) A^^(−2)=(A^^(−1))^^2;

(%o59) $$\begin{pmatrix} \frac{3906}{29929} & \frac{1801}{29929} & -\frac{1267}{29929} \\ -\frac{4978}{29929} & -\frac{1621}{29929} & \frac{1722}{29929} \\ -\frac{169}{29929} & -\frac{530}{29929} & \frac{323}{29929} \end{pmatrix} = \begin{pmatrix} \frac{3906}{29929} & \frac{1801}{29929} & -\frac{1267}{29929} \\ -\frac{4978}{29929} & -\frac{1621}{29929} & \frac{1722}{29929} \\ -\frac{169}{29929} & -\frac{530}{29929} & \frac{323}{29929} \end{pmatrix}$$

(%i60) S−F^^2;

(%o60) $$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

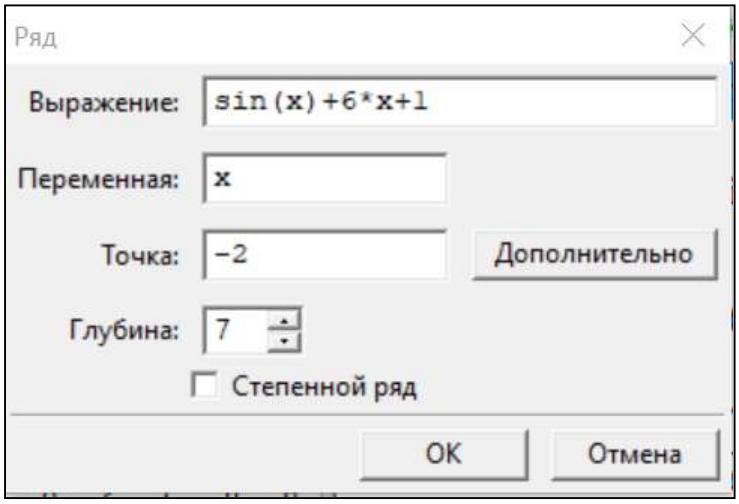
11 Разложение в ряд Тейлора и суммирование.

Решение ряда задач математического анализа значительно упрощается за счет разложения функций в ряд Тейлора. Программа wxMaxima поддерживает такую возможность. Для этого предназначен подпункт Разложить в ряд пункта Анализ

Пример
Разложить в ряд Тейлора в окрестности точки (-2) функцию y = sin x + 6x +1.
Решение. Сначала запишем выражение. Укажем в диалоговом окне в качестве выражения данную функцию, в качестве переменной - x, точки - даннуюточку (-2), зададим глубину, равную 7:

```
(%i61) kill(all);
(%o0)  done

(%i1)  y = sin(x) + 6·x +1;
(%o1)  y=sin(x)+6 x+1
```

Фигура 21:




```
(%i2)  taylor(sin(x)+6·x+1, x, −2, 7);
```

$$(\%o2)/T/\quad -11-\sin(2)+(\cos(2)+6)\,(x+2)+\frac{\sin(2)\,(x+2)^2}{2}-\frac{\cos(2)\,(x+2)^3}{6}-\frac{\sin(2)\,(x+2)^4}{24}+\frac{\cos(2)\,(x+2)^5}{120}+\frac{\sin(2)\,(x+2)^6}{720}-\frac{\cos(2)\,(x+2)^7}{5040}+\ldots$$

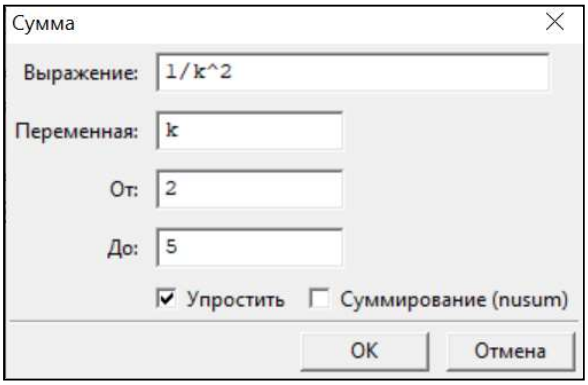
В том случае, если необходимо представить функцию в общем виде (с применением математического знака суммы , следует поставить галочку в поле Степенной ряд

(%i3) niceindices(powerseries(cos(x)+5·x+1, x, −1));

(%o3) $$\sin(1)\left(\sum_{i=0}^{inf}\left(\frac{(-1)^{i}\,(x+1)^{2i+1}}{(2i+1)!}\right)\right)+\cos(1)\left(\sum_{i=0}^{inf}\left(\frac{(-1)^{i}\,(x+1)^{2i}}{(2i)!}\right)\right)+5\,(x+1)-4$$

Нахождение суммы ряда в wxMaxima происходит с помощью подпункта Вычислить сумму. В поле Выражение необходимо ввести выражение для общего члена ряда, в поле Переменная - ту переменную, по которой ряд составлен, в полях От и До требуется указать индексы начального и конечного членов ряда. Например, если стоит задача вычислить сумму 1/4 + 1/9 + 1/16 + 1/25, то диалоговое окно примет вид

Фигура 22:



(%i4) sum(1/k^2, k, 2, 5), simpsum;

(%o4) $\frac{1669}{3600}$

В приведенном выше листинге в строке ввода появился флаг simpsum. Он может принимать два значения: true и false. Если в приведенном выше примере убрать его из строки ввода, то ничего не изменится:

(%i5) sum(1/k^2, k, 2, 5);

(%o5) $\frac{1669}{3600}$

Проверим влияние данного флага на нахождение суммы ряда с членами 1/5^n. Зададим необходимые параметры

(%i6) sum(1/5^n, n, 1, inf), simpsum;

(%o6) $\frac{1}{4}$

Теперь в строке ввода удалим simpsum и, нажав Shift + Enter, получим:

```
(%i7) sum(1/5^n, n, 1, inf);
```

(%o7) $$\sum_{n=1}^{inf}\left(\frac{1}{5^{n}}\right)$$

Таким образом, wxMaxima просто записала заданный ряд с использованием знака суммы, но задачу с вычислением суммы бесконечно убывающей геометрической прогрессии без указания флага simpsum не решила. Если вручную его дописать то получим, то что нужно

```
(%i8) sum(1/5^n, n, 1, inf), simpsum;
```

(%o8) $$\frac{1}{4}$$

Для расходящегося ряда получаем сообщение об ошибке.

```
(%i9) sum(1/k, k, 1, inf);
```

(%o9) $$\sum_{k=1}^{inf}\left(\frac{1}{k}\right)$$

```
(%i10) sum(1/k, k, 1, inf), simpsum;
sum: sum is divergent.
 -- an error. To debug this try: debugmode(true);
```

Примеры сходящихся рядов

```
(%i11) sum(1/k^2, k, 1, inf);
```

(%o11) $$\sum_{k=1}^{inf}\left(\frac{1}{k^{2}}\right)$$

```
(%i12) %,simpsum;
```

(%o12) $$\frac{\%pi^{2}}{6}$$

```
(%i13) sum(1/k^4, k, 1, inf);
```

(%o13) $$\sum_{k=1}^{inf}\left(\frac{1}{k^{4}}\right)$$

(%i14) sum(1/k^4, k, 1, inf), simpsum;

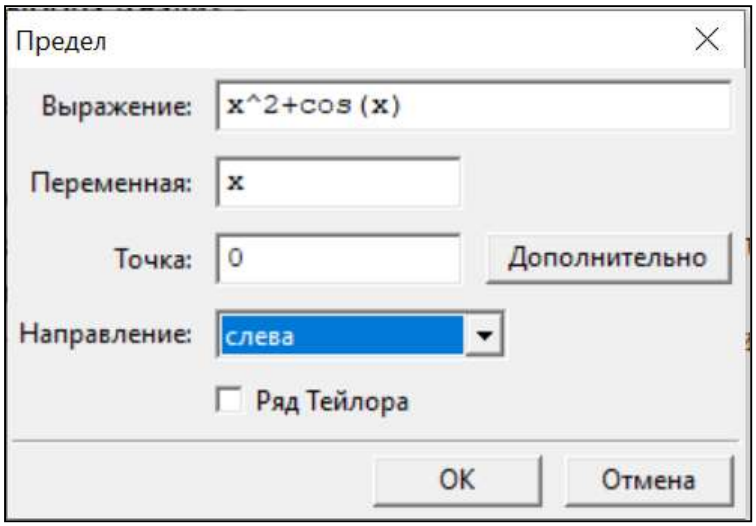
(%o14) $\frac{\%pi^4}{90}$

12 Вычисление пределов.

Нахождение предела функции начинается с выбора Анализ -"Найти предел".
В появившемся диалоговом окне необходимо указать выражение,
предел которого требуется найти, переменную, по которой этот предел
будет найден, точку, к которой стремится значение переменной, и, наконец,
направление предела -двусторонний, предел слева или предел справа .
В wxMaxima предел слева обозначается minus, справа - plus,
плюс бесконечность - inf, минус бесконечность - minf.
Для того, чтобы указать в качестве точки бесконечность и минус бесконечность
необходимо нажать кнопку "Дополнительно".

```
(%i15) kill(all);
(%o0)  done
```

Фигура 23:




```
(%i1) limit(x^2+cos(x), x, 0, minus);
(%o1) 1

(%i2) limit(cos(x)-tan(x^2), x, 0);
(%o2) 1
```

Рассмотрим более сложные выражения. Сначала как и всегда
набираем исходное выражение

(%i4)
```
a:(x^3−2·x+3)/(4·x^3+5·x^2−9); /* задаем символьную переменную для */
                                /* обозначения выражения*/
f(x):=(x^3−2·x+3)/(4·x^3+5·x^2−9);  /* задаем функцию*/
```

(%o3)
$$\frac{x^3-2x+3}{4x^3+5x^2-9}$$

(%o4)
$$f(x):=\frac{x^3-2x+3}{4x^3+5x^2-9}$$

(%i5)
```
a;
```

(%o5)
$$\frac{x^3-2x+3}{4x^3+5x^2-9}$$

Сначала вид предела

(%i6)
```
'limit((x^3−2·x+3)/(4·x^3+5·x^2−9), x, minf);
```

(%o6)
$$\lim_{x\to -\inf}\frac{x^3-2x+3}{4x^3+5x^2-9}$$

И сам предел

(%i7)
```
limit((x^3−2·x+3)/(4·x^3+5·x^2−9), x, minf);
```

(%o7)
$$\frac{1}{4}$$

Поменяем на плюс бесконечность

(%i8)
```
'limit((x^3−2·x+3)/(4·x^3+5·x^2−9), x, inf);
```

(%o8)
$$\lim_{x\to \inf}\frac{x^3-2x+3}{4x^3+5x^2-9}$$

(%i9)
```
limit((x^3−2·x+3)/(4·x^3+5·x^2−9), x, inf);
```

(%o9)
$$\frac{1}{4}$$

Можем использовать как символьную переменную,
так и обозначение функции

(%i10)
```
limit(a, x, inf);
```

(%o10)
$$\frac{1}{4}$$

(%i11) limit(f(x), x, inf);

(%o11) $\frac{1}{4}$

Второй пример

(%i12) b:7·sin(x−%pi/2)/(x+5);

(%o12) $-\frac{7\cos(x)}{x+5}$

(%i13) 'limit(b, x, %pi/2); /* вид предела*/

(%o13) $-7\left(\lim_{x\to\frac{\%pi}{2}}\frac{\cos(x)}{x+5}\right)$

(%i14) limit(b, x, %pi/2); /* значение*/

(%o14) 0

(%i15) c:7·sin(x−%pi/2)/(x−%pi/2);

(%o15) $-\frac{7\cos(x)}{x-\frac{\%pi}{2}}$

(%i16) 'limit(7·sin(x−%pi/2)/(x−%pi/2), x, %pi/2);

(%o16) $-7\left(\lim_{x\to\frac{\%pi}{2}}\frac{\cos(x)}{x-\frac{\%pi}{2}}\right)$

(%i17) limit(7·sin(x−%pi/2)/(x−%pi/2), x, %pi/2);

(%o17) 7

При вычислении предела функции при условии, что ее аргумент стремится к некоторому значению слева или справа, следует выбрать требуемое 'Направление' указывая предел двусторонний или подходим слева или справа

(%i18) limit(c, x, %pi/2, minus);

(%o18) 7

(%i19) limit(c, x, %pi/2, plus);

(%o19) 7

В данном случае разницы нет как приближаться к точке

Найдем

```
(%i20) 'limit(atan(1/(x−5)), x, 5, minus);  /* приближаемся слева запись*/
                                             /* 5- обозначает приближение слева*/
```

$$(\%o20)\quad \lim_{x\to 5-} \operatorname{atan}\left(\frac{1}{x-5}\right)$$

```
(%i21) limit(atan(1/(x−5)), x, 5, minus);
```

$$(\%o21)\quad -\frac{\%pi}{2}$$

```
(%i22) 'limit(atan(1/(x−5)), x, 5, plus);/* приближаемся справа*/
```

$$(\%o22)\quad \lim_{x\to 5+} \operatorname{atan}\left(\frac{1}{x-5}\right)$$

```
(%i23) limit(atan(1/(x−5)), x, 5, plus);
```

$$(\%o23)\quad \frac{\%pi}{2}$$

При приближении с разных сторон значение предела арктангенса меняется

```
(%i24) 'limit(1/x, x, 0, minus);
```

$$(\%o24)\quad \lim_{x\to 0-} \frac{1}{x}$$

```
(%i25) limit(1/x, x, 0, minus);
(%o25) -inf
(%i26) 'limit(1/x, x, 0, plus);
```

$$(\%o26)\quad \lim_{x\to 0+} \frac{1}{x}$$

```
(%i27) limit(1/x, x, 0, plus);
(%o27) inf
```

При вычислениях с использованием Maxima более естественно использовать при вычислении сложных пределов и сравнении бесконечно малых разложение числителя и знаменателя в ряд Тейлора При вычислении с использованием меню во вкладке меню Анализ → Найти предел, установить пункт "Использовать ряд Тейлора".
При вычислении пределов рациональных функций целесообразно сначала факторизовать их с помощью функции factor. Если же исследуются выражения содержащие иррациональности и показательные и логарифмические функции, то полезна будет функция radcan.

Особенности при вычислении пределов.

(%i28) 'limit(sin(1/x), x, 0);

(%o28) $\lim_{x \to 0} \sin\left(\frac{1}{x}\right)$

(%i29) limit(sin(1/x), x, 0);

(%o29) ind

(%i30) 'limit(tan(1/x), x, 0);

(%o30) $\lim_{x \to 0} \tan\left(\frac{1}{x}\right)$

(%i31) limit(tan(1/x), x, 0);

(%o31) und

Здесь имеют место два ответа, которые означают, что искомый предел не существует: ind (от indefinite - неопределенный) и
und (от undefined - неопределенный). Первый из этих ответов в руководствах по работе с wxMaxima описан как indefinite but bounded (не определен, но ограничен), то есть речь идет о функции,не имеющей предела, но при этом ограниченной сверху либо в окрестности предельной точки, либо на всей прямой.
Второй ответ предполагает возможно неограниченную функцию.

13 Решение дифференциальных уравнений и систем дифференциальных уравнений.

При работе с дифференциальными уравнениями нам нужно сначала указать Maxima, какая переменная зависит от независимой переменной.

Используем функцию ode2, чтобы решить обыкновенные дифференциальные уравнения.
Посредством функции ode2 могут быть решены следующие типы ОДУ первого порядка:
линейные, ОДУ с разделяющимися переменными, однородные ОДУ, уравнения в полных дифференциалах, уравнения Бернулли, обобщённые однородные уравнения.
Кроме того, при помощи функции ode2 могут быть решены следующие типы уравнений второго порядка:
с постоянными коэффициентами; в полных дифференциалах; линейные однородные с переменными коэффициентами, которые могут быть сведены к уравнениям с постоянными коэффициентами; уравнения Эйлера; уравнения,разрешимые методом вариации постоянных; уравнения, свободные от независимой переменной, допускающие понижение порядка.
В последних версиях добавились еще виды дифуравнений, которые можно решить с помощью пакета. Для уточнения следует обратиться к справке.

Воспользуемся графическим интерфейсом. Уравнения - Решить ОДУ

```
(%i32) kill (all);
(%o0)  done

(%i1)  ode2(diff(y,x)=0, y, x);
(%t1)  0=0
       not a proper differential equation
(%o1)  false
```

Не вышло потому что необходимо сначала задать зависимость y от x.

```
(%i3)  depends(y, x);
       ode2(diff(y,x)=0, y, x);   /* теперь мы получили правильный результат,*/
                                  /* функция равна константе */
(%o2)  [y(x)]
(%o3)  y=%c

(%i4)  ode2(diff(y(x),x)=0, y(x), x);
(%o4)  y(x)=%c
```

```
(%i8) depends(y, x) $
      /* уравнение с раздельными переменными*/
      e: (x−1)·y^3+(y−1)·x^3·diff(y,x)=0;
      r: ode2(e,y,x);
      /* применяем начальные условия */
      ic1(r,x=2,y=−3);
```

$$(\%o6)\quad x^3\,(y-1)\left(\frac{d}{dx}\,y\right)+(x-1)\,y^3=0$$

$$(\%o7)\quad \frac{2\,y-1}{2\,y^2}=\%c-\frac{2\,x-1}{2\,x^2}$$

$$(\%o8)\quad \frac{2\,y-1}{2\,y^2}=-\frac{x^2+72\,x-36}{72\,x^2}$$

Решение дифференциального уравнения типа Бернулли.

```
(%i9) 'diff(y,x)−y+sqrt(y);
```

$$(\%o9)\quad \frac{d}{dx}\,y-y+\sqrt{y}$$

```
(%i12) depends(y, x) $
       r: ode2(diff(y,x)−y+sqrt(y),y,x);
       /* выражаем зависимую переменную с помощью решения solve */
       solve(r, y);
```

$$(\%o11)\quad 2\,\log\left(\sqrt{y}-1\right)=x+\%c$$

$$(\%o12)\quad \left[y=\%e^{x+\%c}+2\,\%e^{x/2+\frac{\%c}{2}}+1\right]$$

Решение дифференциального уравнения второго порядка, а именно простого гармонического уравнения движения.

```
(%i13) 'diff(x,t,2) + 3 · x = 0;
```

$$(\%o13)\quad \frac{d^2}{dt^2}\,x+3\,x=0$$

```
(%i19) depends(x, t) $
       e: diff(x,t,2) + 3 · x = 0;
       r: ode2(e, x, t);
       /* Применяем начальные условия */
       ic2(r, t=0, x=1, diff(x,t)=1/2);
       /* или применяем граничные условия */
       s: bc2(r, t=0, x=1, x=3, t=−1);
       /* Рисуем */
       wxplot2d(rhs(s), [t, 0, 10])$
```

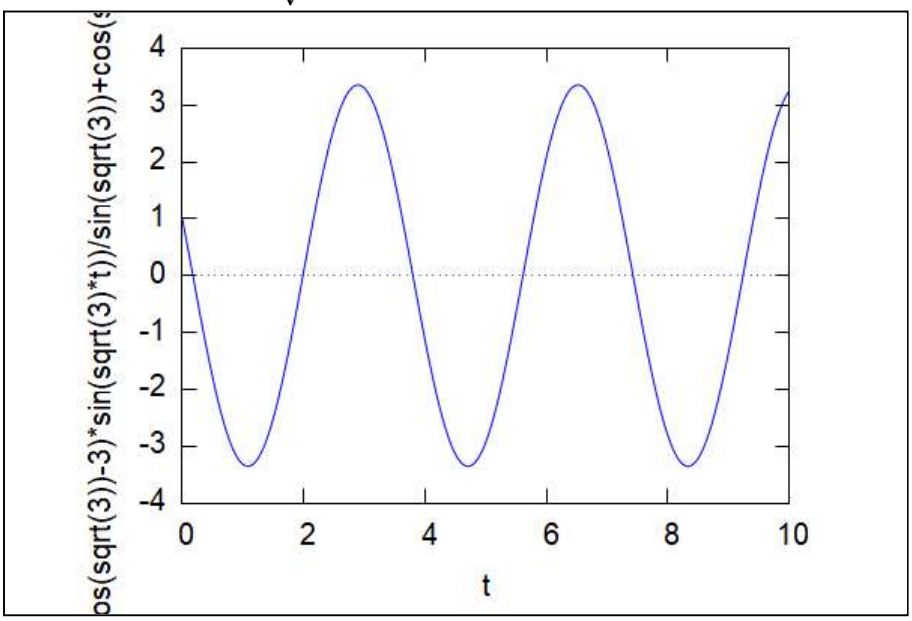
$$(\%o15)\quad \frac{d^2}{dt^2}x+3x=0$$

$$(\%o16)\quad x=\%k1\,\sin\left(\sqrt{3}\,t\right)+\%k2\,\cos\left(\sqrt{3}\,t\right)$$

$$(\%o17)\quad x=\frac{\sin\left(\sqrt{3}\,t\right)}{2\sqrt{3}}+\cos\left(\sqrt{3}\,t\right)$$

$$(\%o18)\quad x=\frac{\left(\cos\left(\sqrt{3}\right)-3\right)\sin\left(\sqrt{3}\,t\right)}{\sin\left(\sqrt{3}\right)}+\cos\left(\sqrt{3}\,t\right)$$

(%t19)



С функцией desolve мы можем решать системы дифференциальных уравнений, но в этом случае мы должны явно указать зависимости функций.

Система уравнений

```
(%i21) 'diff(f(x),x)=3·f(x)−2·g(x);
       'diff(g(x),x)=2·f(x)−2·g(x);
```

$$(\%o20)\quad \frac{d}{dx}f(x)=3f(x)-2g(x)$$

$$(\%o21)\quad \frac{d}{dx}g(x)=2f(x)-2g(x)$$

Решаем

```
(%i22) desolve(
         [diff(f(x),x)=3·f(x)−2·g(x),
          diff(g(x),x)=2·f(x)−2·g(x)],
         [f(x),g(x)]);
```

$$(\%o22)\ \left[f(x) = \frac{(2\,g(0)-f(0))\ \%e^{-x}}{3} - \frac{(2\,g(0)-4\,f(0))\ \%e^{2x}}{3},\ g(x) = \frac{(4\,g(0)-2\,f(0))\ \%e^{-x}}{3} - \frac{(g(0)-2\,f(0))\ \%e^{2x}}{3} \right]$$

В ответе неопределенными остаются начальные значения функций в точке x=0

Если мы хотим применить начальные условия, мы должны объявить их с помощью atvalue перед вызовом desolve.

Пример системы из трех уравнений

```
(%i26) atvalue(f(x),x=0,−1)$
       atvalue(g(x),x=0,3)$
       atvalue(h(x),x=0,1)$
       desolve(
        [diff(f(x),x)=f(x)+g(x)+3·h(x),
         diff(g(x),x)=g(x)−2·h(x),
         diff(h(x),x)=f(x)+h(x)],
        [f(x),g(x),h(x)]);
```

$$(\%o26)\ \left[f(x) = x\ \%e^{2x} + \%e^{2x} - 2\ \%e^{-x},\ g(x) = -2\,x\ \%e^{2x} + 2\ \%e^{2x} + \%e^{-x},\ h(x) = x\ \%e^{2x} + \%e^{-x} \right]$$

Еще пример решения системы дифуравнений.

f(x)' = 4f(x) – 3g(x) + sin(x); f(0)=5;
g(x)' = 2f(x) – g(x) – 2 cos(x) ; g(0)= 5;

```
(%i27) eq1:'diff(f(x),x)=4·f(x)−3·g(x)+sin(x);/* задаем дифур-ние */
```

$$(\%o27)\ \frac{d}{dx} f(x) = \sin(x) - 3\,g(x) + 4\,f(x)$$

```
(%i28) eq2:'diff(g(x),x,1)=2·f(x)−g(x)−2·cos(x);  /* задаем дифур-ние */
```

$$(\%o28)\ \frac{d}{dx} g(x) = -2\cos(x) - g(x) + 2\,f(x)$$

```
(%i29) atvalue(g(x),x=0,5);  /* задаем начальные условия */
(%o29) 5
```

```
(%i30) atvalue(f(x),x=0,5);  /* задаем начальные условия */
(%o30) 5

(%i31) desolve([eq1,eq2],[f(x),g(x)]);  /* вызываем функцию решения системы */
(%o31) [f(x)=-2*sin(x)+cos(x)+3*%e^(2*x)+%e^x,g(x)=-2*
       sin(x)+2*cos(x)+2*%e^(2*x)+%e^x]
```

Задаем функции для проверки решения, потому что если их не задать , то они будут неопределены так как решение функции desolve пока только выражение. Функции подставим в исходную систему дифуров, отнимая от левой части правую часть равенств.

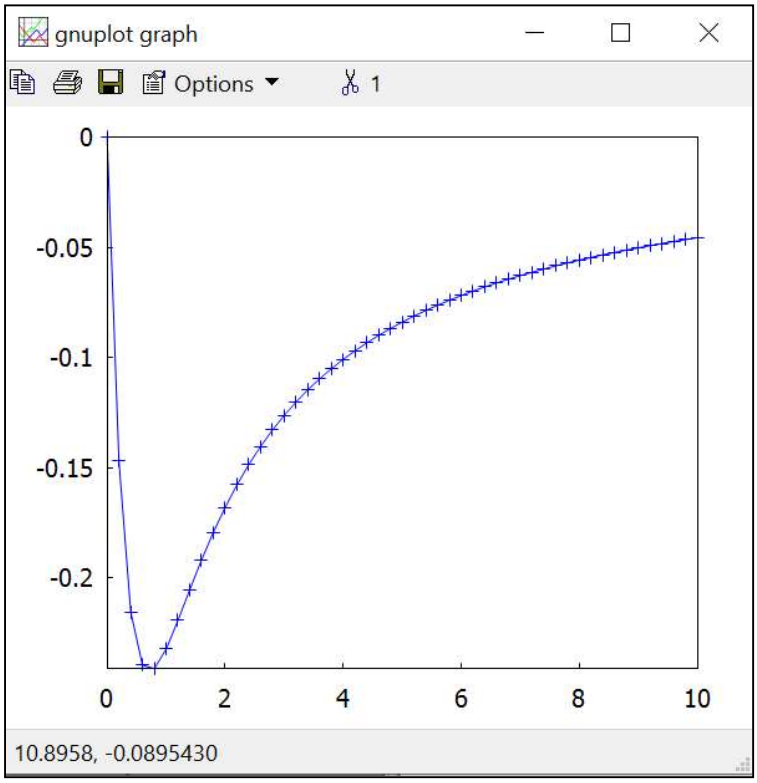
```
(%i33) f(x):=-2·sin(x)+cos(x)+3·%e^(2·x)+%e^x;
       g(x):=-2·sin(x)+2·cos(x)+2·%e^(2·x)+%e^x;
(%o32) f(x):=(-2)*sin(x)+cos(x)+3*%e^(2*x)+%e^x
(%o33) g(x):=(-2)*sin(x)+2*cos(x)+2*%e^(2*x)+%e^x

(%i34) diff(f(x),x)-(4·f(x)-3·g(x)+sin(x));
(%o34) -2*sin(x)+3*(-2*sin(x)+2*cos(x)+2*%e^(2*x)+%e^x)-4
       *(-2*sin(x)+cos(x)+3*%e^(2*x)+%e^x)-2*cos(x)+6*%e^(2*x)+%e^x

(%i35) radcan(%);
(%o35) 0

(%i36) diff(g(x),x,1)-(2·f(x)-g(x)-2·cos(x));
(%o36) -4*sin(x)-2*(-2*sin(x)+cos(x)+3*%e^(2*x)+%e^x)+2*
       cos(x)+6*%e^(2*x)+2*%e^x

(%i37) radcan(%);
(%o37) 0
```

Получили в обоих случаях 0. Что подтверждает правильность решения

Когда символьные методы недопустимы, мы можем использовать метод Рунге-Кутта (rk). Здесь мы также строим получившуюся функцию.

```
(%i39) /* решаем dy/dt = 2 y^2-exp(-3t) */
       sol: rk(2·y^2−exp(−3·t), y, 0, [t, 0, 10, 0.2]) $
       draw2d(
        points_joined = true,
        points(sol))$   /* Нажать исполнение для получения графика в отдельном окне*/
```

Фигура 24:



Попробуем решить символьно

```
(%i40) 'diff(y,t)−(2·y^2−exp(−3·t))=0;
```

$$\frac{d}{d\,t}\,y-2\,y^{2}+\%e^{-3\,t}=0$$

```
(%i43) depends(y, t) ;
       r: ode2(diff(y,t)−(2·y^2−exp(−3·t)),y,t);
       solve(r, y);
(%o41) [y(t)]
(%o42) false
(%o43) []
```

Не нашлось символьного решения

(%i46)
```
depends(y, t) ; /* уберем экспоненту и получим легкое дифуравнение*/
r: ode2(diff(y,t)=(2·y^2),y,t);
solve(r, y);
```

(%o44) $$[y(t)]$$

(%o45) $$-\frac{1}{2y}=t+\%c$$

(%o46) $$\left[y=-\frac{1}{2t+2\,\%c}\right]$$

14 Аппроксимация табличных данных трехпараметрическим уравнением

Задаем список значений x и список соответствующих y.
И выбираем вид аппроксимирущей функции. Успех приближения зависит от выбора вида функции.

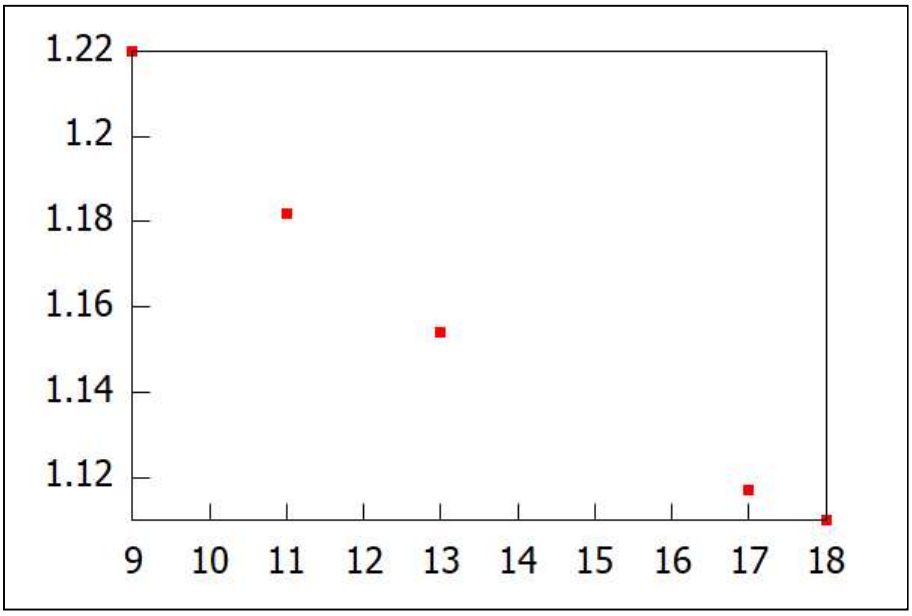
```
(%i47) kill(all);
(%o0)  done
```

Пусть заданы табличные значения x и y.

```
(%i2)  xdata:[9,11,13,17,18];
       ydata:[1.22,1.182,1.154,1.117,1.11];
(%o1)  [9,11,13,17,18]
(%o2)  [1.22,1.182,1.154,1.117,1.11]
```

Построим график этих точек чтобы определиться с видом функции.

```
(%i3)  wxdraw2d(
         color="#ff0000",
         point_type=5,
       points(matrix(xdata,ydata))
       )$
```



(%t3)

В качестве аппроксимирующей функции можем взять F(x)=a+b/x. Построим на графике аппроксимирующую функцию и функцию, заданную таблично.

(%i4)
```
data1:transpose(
  matrix(
    xdata,
    ydata
    )
  );
```

(%o4)
$$\begin{pmatrix} 9 & 1.22 \\ 11 & 1.182 \\ 13 & 1.154 \\ 17 & 1.117 \\ 18 & 1.11 \end{pmatrix}$$

данные задаются в матрице data1 первый столбец x второй столбец соответсвующие y.

Ниже мы записывам предполагаемый вид зависимости y от x максимум с
тремя параметрами : a,b,c. Если какой то параметр не использовать,
то он останется равным нулю.

(%i5)
```
approach1:y=a+b/x;  /* это пример, но стуктуру и название переменных необходимо */
                    /* сохранить, чтобы не менять обозначения в нижележащих*/
                    /* выражениях                                           */
```

(%o5)
$$y = \frac{b}{x} + a$$

Производим аппроксимацию данных, определяя параметры a, b и c:

(%i6)
```
load("lsquares")$
```

```
(%i8) lsquares_estimates_approximate(
        lsquares_mse(
          data1,[x,y],approach1
        ),
        [a,b,c],
        initial=[0,0,0]
      );
      params1:%[1];    /* выводит первый из возможных ответов*/
      ************************************************
       N=    3   NUMBER OF CORRECTIONS=25
          INITIAL VALUES
       F=  1.339405800000000D+00   GNORM=  2.320479586624841D+00
      ************************************************
        I  NFN     FUNC                    GNORM                 STEPLENGTH
        1    2     2.511645258381046D-02   3.081022767452563D-01   4.309453984271024D-01
        2    3     1.529896128569744D-03   1.610554253954131D-03   1.000000000000000D+00
        3    8     1.123971046397626D-03   3.185795640709781D-03   3.410000000000000D+02
        4    9     1.045438800752243D-06   1.135212923586848D-03   1.000000000000000D+00
       THE MINIMIZATION TERMINATED WITHOUT DETECTING ERRORS.
       IFLAG = 0
(%o7)  [[a=1.00154857773925,b=1.97824378240822,c=
       0.0]]
(%o8)  [a=1.00154857773925,b=1.97824378240822,c=
       0.0]
```

В этом примере нет реальной необходимости в «initial=».
Однако использование хорошей стартовой точки позволяет,
что lsquares быстрее найти оптимум задачи.

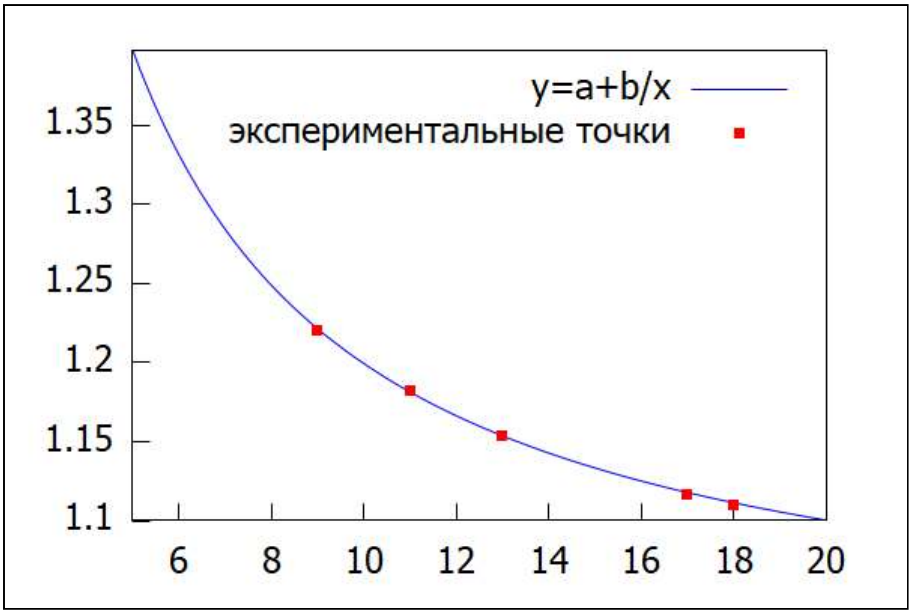
Выводим значения найденных параметров

```
(%i10) a:1.001548577739248;
       b:1.978243782408221;
(%o9)  1.00154857773925
(%o10) 1.97824378240822
```

Рисуем графики

```
(%i11) wxdraw2d(
         key="y=a+b/x",
         explicit(
           b/x+a,
           x,5,20
         ),
         color="#ff0000",
         key="экспериментальные точки",
         point_type=5,
       points(matrix(xdata,ydata))

       )$
```

(%t11)



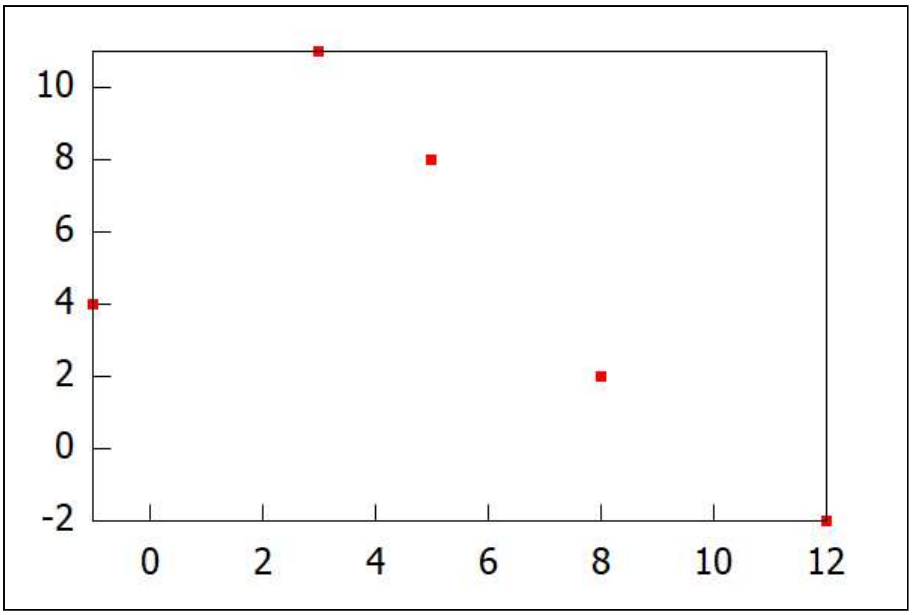
Очень хорошее совпадение.

Еще пример

```
(%i12) kill(all);
(%o0)  done

(%i2)  xdata:[3,8,12,5,-1];
       ydata:[11,2,-2,8,4];
(%o1)  [3,8,12,5,-1]
(%o2)  [11,2,-2,8,4]
```

(%i3)
```
wxdraw2d(

   color="#ff0000",
   point_type=5,
points(matrix(xdata,ydata))
)$
```

(%t3)



Похоже на параболу. Попробуем ее

(%i4)
```
data1:transpose(
   matrix(
      xdata,
      ydata
      )
   );
```

(%o4)
$$\begin{pmatrix} 3 & 11 \\ 8 & 2 \\ 12 & -2 \\ 5 & 8 \\ -1 & 4 \end{pmatrix}$$

(%i5)
```
approach1:y=a·x^2+b·x+c;  /* это пример, но стуктуру и */
                          /* название переменных необходимо */
                          /* сохранить, чтобы не менять обозначения */
                          /* в нижележащих выражениях */
```

(%o5)
$$y = a\,x^2 + b\,x + c$$

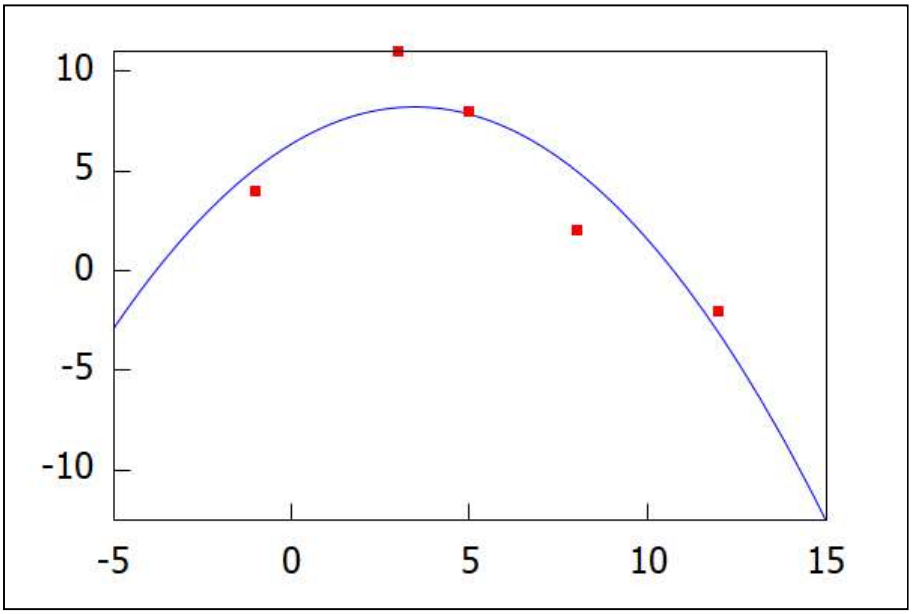
(%i6)
```
load("lsquares")$
```

```
(%i8) lsquares_estimates_approximate(
        lsquares_mse(
          data1,[x,y],approach1
        ),
        [a,b,c],
        initial=[0,0,0]
      );
      params1:%[1];    /* выводит первый из возможных ответов*/
      ************************************************
       N=    3   NUMBER OF CORRECTIONS=25
          INITIAL VALUES
       F=  4.180000000000000D+01   GNORM=  6.286366199960037D+01
      ************************************************
        I  NFN     FUNC                    GNORM                 STEPLENGTH
        1    3     4.158464396941993D+01   2.206184738810429D+01   1.089902579963167D-04
        2    8     2.779687254672353D+01   1.574627855762974D+01   3.410000000000000D+02
        3    9     1.218027240445397D+01   3.364253058027232D+00   1.000000000000000D+00
        4   10     1.086715949733484D+01   3.269026754575138D+00   1.000000000000000D+00
        5   11     4.131783138572919D+00   1.721519474953296D+00   1.000000000000000D+00
        6   12     3.909905762990540D+00   2.163494020864138D-03   1.000000000000000D+00
       THE MINIMIZATION TERMINATED WITHOUT DETECTING ERRORS.
       IFLAG = 0
(%o7)  [[a=-0.155659745826636,b=1.07743904960657,c
       =6.3466654711465]]
(%o8)  [a=-0.155659745826636,b=1.07743904960657,c=
       6.3466654711465]

(%i11) a:-0.1556597458266359;
       b:1.077439049606574;
       c:6.346665471146501;
(%o9)  -0.155659745826636
(%o10) 1.07743904960657
(%o11) 6.3466654711465
```

Рисуем

```
(%i12) wxdraw2d(
         explicit(
           a·x^2+b·x+c,
           x,−5,15
         ),
         color="#ff0000",
         point_type=5,
       points(matrix(xdata,ydata))

       )$
```

(%t12)



Не очень хорошее совпадение. Об этом нам говорит и первый параметр в диагностике вывода спецфункции Func= 3.909905762990540D+00. В первом примере он падал до D-06.

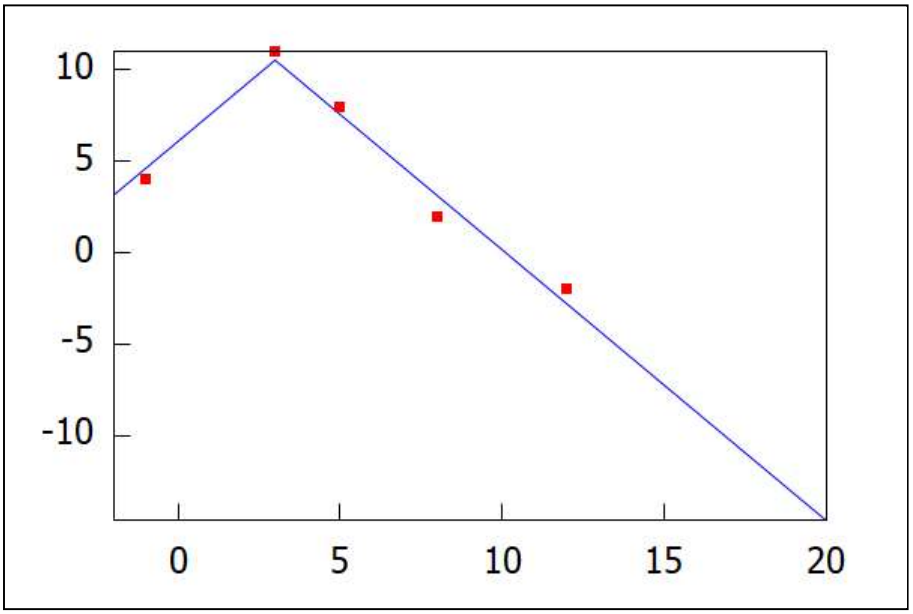
Попробуем другое. Но вначале нужно очистить параметры a,b,c

```
(%i13) kill(a,b,c);
(%o13) done

(%i14) approach1:y=a·abs(x−3)+b;  /* это пример, но стуктуру и */
                                  /* название переменных необходимо */
                                  /* сохранить, чтобы не менять обозначения */
                                  /* в нижележащих выражениях */
(%o14) y = a |x − 3| + b

(%i15) load("lsquares")$
```

```
(%i17) lsquares_estimates_approximate(
         lsquares_mse(
           data1,[x,y],approach1
         ),
         [a,b,c],
         initial=[0,0,0]
       );
       params1:%[1];    /* выводит первый из возможных ответов*/
       ************************************************
        N=    3   NUMBER OF CORRECTIONS=25
            INITIAL VALUES
       F=  4.180000000000000D+01   GNORM=  1.329661611087573D+01
       ************************************************
        I  NFN     FUNC                    GNORM                 STEPLENGTH
        1    3     3.929021078920052D+01   8.998969837241025D+00   2.839128066515247D-02
        2    5     3.180677913575112D+01   8.083569285438326D+00   5.000000000000000D+00
        3    6     5.356521739130432D-01   1.303415070299546D-14   1.000000000000000D+00
       THE MINIMIZATION TERMINATED WITHOUT DETECTING ERRORS.
       IFLAG = 0
(%o16) [[a=-1.47826086956522,b=10.5130434782609,c=
       0.0]]
(%o17) [a=-1.47826086956522,b=10.5130434782609,c=
       0.0]

(%i20) a:−1.478260869565216;
       b:10.51304347826086;
       c:0.0;
(%o18) -1.47826086956522
(%o19) 10.5130434782609
(%o20) 0.0
```

```
(%i21) wxdraw2d(
         explicit(
           a·abs(x−3)+b,
           x,−2,20
           ),
         color="#ff0000",
         point_type=5,
       points(matrix(xdata,ydata))

       )$
```

(%t21)



Результат вышел лучше.

15 Программирование в системе Maxima.

Язык Maxima содержит все необходимые средства для разработки программ: ввод, вывод, условные инструкции, циклы, массивы.
Основная функция для считывания вводимых пользователем данных: read(expr_1,... ,expr_n).
Вводимые выражения expr_1,expr_2,... при вводе интерпретируются.
Поля ввода разделяются точками с запятой или знаком $.
Аргументы функции read могут включать подсказку.

```
(%i22) kill(all);
(%o0)  done
```

```
(%i2)  a:42$
       a:read("Значение a = ",a," введите новую величину");
       Значение a =  42  введите новую величину 17;
(%o2)  17
```

Аналогичная функция readonly осуществляет только ввод данных (без их интерпретации).
Сравнение использования функций read и readonly:

```
(%i4)  a:7$
        readonly("Введите выражение:");
       Введите выражение:  a+5;
(%o4)  a+5
```

```
(%i5)  read("Введите выражение:");
       Введите выражение:  a+5;
(%o5)  12
```

Вывод данных на экран.
Как правило все действия(ответы) выводяться на экран по мере их выполнения.Чтобы на экран выводились не все результат работы программы, а только некоторые значения необходимо заменить завершающий команду символ ; на символ $.

Для вывода на экран значения переменной или какой-нибудь другого выражения используется функция print(exp1,exp2).
Выражения exp1,... ,expn интерпретируются и выводятся последовательно в строчку (в отличие от вывода, генерируемого функцией display).
Функция print возвращает значение последнего интерпретированного выражения.
Например, print(i) для вывода значения переменной i.

```
(%i6)  print(a,a+7);
       7  14
(%o6)  14
```

Вывод на экран осуществляется функцией display.
Синтаксис её вызова: display(expr_1,expr_2,...). Выражения из списка аргументов выводятся слева направо (сначала само выражение,
а затем после знака равенства — его значение).
Аналогичная функция disp (синтаксис вызова: disp(expr_1, expr_2,...))
выводит на экран только значение выражения после его интерпретации.

```
(%i10) a:1$ b:2$ c:3$
       display(a,b,c);
       a = 1
       b = 2
       c = 3
(%o10) done

(%i11) disp(a,b,c);
       1
       2
       3
(%o11) done
```

Блоки команд.
Для того, чтобы в теле команды выполнить несколько действий,
необходимо их записать в круглых скобках через запятую,
например, блок команд для обмена местами переменных аи b может
выглядеть так: (t:a, a:b, b:t).

Условная команда if
Условная инструкция if имеет следующий синтаксис:
if условие then выражение1 else выражение2
В этом случае if возвращает значение одного из двух выражений.
Например,присвоить переменной m максимума из двух переменных
a и b можно двумя способами:
if a>b then m:a else m:b или m: if a>b then a else b.
Вместо одной инструкции можно использовать блок инструкций,
как указано выше. Слово else и инструкцию после него можно опустить.
В качестве условий можно использовать операторы сравнения
<, >, <=, >=, =, # и логические операторы and, or, not

Примеры

```
(%i13) a:5; b:8;
(%o12) 5
(%o13) 8

(%i14) if a>b then m:a else m:b;
(%o14) 8

(%i15) m;
(%o15) 8

(%i16) if a<b then(g:9, i:7) else g:15;
(%o16) 7

(%i17) g;
(%o17) 9

(%i18) i;
(%o18) 7
```

Цикл for
Синтаксис цикла for следующий:
for переменная: начальное_значение step шаг thru конечное значение
do выражение
Любые части этой конструкции (step, thru) можно опускать.
При необходимости выполнить несколько инструкций необходимо объединить их в блок.

```
(%i19) for i:2 step 2 thru 10 do(k:2·i, print(i,k)) $/* присвоение k и печать i и k объединены в блок*/
2  4
4  8
6  12
8  16
10  20
```

Цикл while
Синтаксис цикла while следующий:
while условие do выражение. Цикл работает пока справедливо укзанное условие.

```
(%i21) i:1;
       while i< 5 do (i:i+1, print("i=",i)); /* Текстовая величина приводится
                                                в двойных кавычках*/
(%o20) 1
       i=  2
       i=  3
       i=  4
       i=  5
(%o21) done
```

Перечисляемые упорядоченные элементы. Списки и массивы.

Списки.
В программе Maxima существует понятие списка. Это упорядоченная совокупность объектов (могут быть разных типов: выражения, формулы, числа, списки...). Для задания списка объекты перечисляются через запятую в квадратных скобках, например: sp1:[a,b,c,45,k+m]. Пустые квадратные скобки означают пустой список. Создать список позволяет функция makelist: makelist(Выражение, переменная, начальное значения, конечное значение).
В графическом меню во вкладке Список указаны варианты создания списков. Для списков существуют функции присоединения, извлечения, модификации и сортировки элементов

```
(%i22) lst:makelist(i^2,i,1,10);/* получаем список квадратов целых чисел от 1 до 10.*/
(%o22) [1,4,9,16,25,36,49,64,81,100]

(%i25) /*Элемент списка извлекается по имени списка и своему номеру. */
       lst[3];
       a:makelist([i,i+1],i,1,5);
       a[3][2];
(%o23) 9
(%o24) [[1,2],[2,3],[3,4],[4,5],[5,6]]
(%o25) 4
```

Массивы
Массивы - это упорядоченная последовательность величин (элементов массива), обозначаемая одним именем. Прежде чем создать массив его нужно описать с помощью функции array
array (name, dim_1, ..., dim_n)
array (name, type, dim_1, ..., dim_n)
array ([name_1, ..., name_m], dim_1, ..., dim_n)
Здесь name - это имя массива, dim_1, ..., dim_n - размерность массива. Порядковый номер элемента называется индексом. Индексы элементов обычного массива - целые числа, изменяющиеся от 0 до dim_i.
Число индексов не должно превышать пяти. Чтобы получит доступ к нужному элементу, необходимо указать имя массива и порядковый номер элемента, называемый индексом.
Именованный список может выступать в роли массива. В максима существует определенная неоднозначность с массивами, списками и элементами матриц. В принципе они взаимозаменяемы

```
(%i26) array(massiv, 5); /* создаем массив*/
(%o26) massiv
```

```
(%i27) for i:0 step 1 thru 5 do( massiv[i]:i^2) ; /* присваиваем значения массиву*/
(%o27) done
```

```
(%i28) massiv[4]; /* проверяем четвертый элемент массива*/
(%o28) 16
```

Команда listarray(array) превращает массив в список с именем sp

```
(%i29) sp:listarray(massiv);
(%o29) [0,1,4,9,16,25]
```

```
(%i30) sp;
(%o30) [0,1,4,9,16,25]
```

16 Рекомендуемая литература и использованные источники

1. Документация по текущей версии пакета: http://maxima.sourceforge.net/docs/manual/en/maxima.html
2. Программирование в Maxima: Учеб. пособие / А. Р. Есаян, В. Н. Чубариков, Н. М. Добровольский, А. В. Якушин.– Тула: Изд-во Тул. гос. пед. ун-та им. Л. Н. Толстого, 2012.– 351 с.
3. Т. Н. Губина, Е. В. Андропова Решение дифференциальных уравнений в системе компьютерной математики Maxima: учебное пособие. – Елец: ЕГУ им. И.А. Бунина, 2009. – 99 с.
4. Zachary Hannan wxMaxima for Calculus I //Solano Community College, https://wxmaximafor.wordpress.com/
5. Zachary Hannan wxMaxima for Calculus II //Solano Community College, https://wxmaximafor.wordpress.com/
6. Стахин Н.А. Основы работы с системой аналитических (символьных) вычислений Maxima. (ПО для решения задач аналитических (символьных)вычислений): Учебное пособие. – Москва: 2008. — 86 с.
7. Основы работы с системой компьютерной алгебры Maxima: Учебно-методическое пособие / М.С. Малакаев, Л.Р. Секаева , О.Н. Тюленева. Казань: Казанский университет, 2012. – 57с.
8. Компьютерная математика с Maxima: Руководство для школьников и студентов / Е.А.Чичкарёв М. : ALT Linux,2009. 233с.
9. В.А. Ильина, П.К. Силаев. система аналитических вычислений Maxima для физиков-теоретиков. М.:МГУ им. М.В. Ломоносова, 2007. – 113 с. http://tex.bog.msu.ru/numtask/max07.ps
10. Статьи Тихона Тарнавского http://maxima.sourceforge.net/ru/maxima-tarnavsky-1.html
11. Gilberto E. Urroz http://www.neng.usu.edu/cee/faculty/gurro/Maxima.html
12. Решение математических задач с помощью пакета Maxima / Черкесова Л.В., Акишин Б.А., Галабурдин А.В., и др. Ростов–на–Дону : издательство ДГТУ, 2015. – 99 с.
13. Абзалилов Д.Ф. Практические задания по высшей математике с применением программы Maxima для студентов, обучающихся по специальности «социология». Учебно-методическое пособие, Казань : КФУ, 2012 г. – 80 с.
14. Арманьшин Д.Ф. Эффективность применения систем компьютерной математики в учебном процессе: материалы III Международной научно-практической конференции «Информационные системы и коммуникативные технологии в современном образовательном процессе» / науч. редкол.: Т.С. Волкова [и др.]. – Пермь : ИПЦ «Прокростъ», 2016. – С. 7-11.
15. Применение пакетов прикладных программ в математи-ке: учеб. пособие / О.Н. Троицкая, Н.Н. Конечная; Сев. (Арк-тич.) федер. ун-т им. М.В. Ломоносова. - Архангельск: САФУ, 2015.- 100 с.

17 Оглавление