Казанский (Приволжский) федеральный университет

М.С. МАЛАКАЕВ, Л.Р. СЕКАЕВА, О.Н. ТЮЛЕНЕВА

ОСНОВЫ РАБОТЫ С СИСТЕМОЙ КОМПЬЮТЕРНОЙ АЛГЕБРЫ MAXIMA

Учебно-методическое пособие

Казань

2012

УДК 510

Авторы-составители:

ст. преп. М.С. Малакаев,

канд. физ.- мат. наук, доц. Л.Р. Секаева,

канд. физ.- мат. наук, доц. О.Н. Тюленева

Научный редактор, рецензент

Доктор физ.- мат. наук, Е.А. Широкова

Основы работы с системой компьютерной алгебры Maxima: Учебно-методическое пособие / М.С. Малакаев, Л.Р. Секаева, О.Н. Тюленева. Казань: Казанский университет, 2012. – 57с.

Данное пособие предназначено для студентов 1-го курса Института геологии и нефтегазовых технологий, содержит описание основных приемов работы с компьютерной программой *Maxima* для выполнения алгебраических преобразований, символьных вычислений и построения разнообразных графиков.

Оглавление

§1. Введение. Система компьютерной алгебры *Maxima* 4

§2. Ввод простейших команд в *Maxima* 6

§3. Решение задач элементарной математики 11

§4. Решение задач линейной алгебры 15

§5. Построение графиков функций 24

§6. Программирование в *Maxima* на встроенном макроязыке 37

§7. Вычисление пределов функции 45

§8. Вычисление производных функции 47

§9. Задачи для самостоятельного решения 53

§1. ВВЕДЕНИЕ.
СИСТЕМА КОМПЬЮТЕРНОЙ АЛГЕБРЫ *MAXIMA*

Компьютерные программы по математике делят на:

1. Программы, предназначенные для вычислений, которые работают с числовыми выражениями (например, *Excel*). Их результаты, обычно, бывают приближенными, они оперируют с вещественными числами, представленными в виде бесконечных десятичных дробей, поэтому при вычислениях происходит их округление.
2. Программы, которые наряду с математическими вычислениями и построением графиков, проводят символьные преобразования, вычисления в символьном виде, например, производных или первообразных заданной функции, находят корни различных уравнений и систем уравнений и т.д. такие программы называют – ***«системы компьютерной алгебры»*** (например, *Mathematica, MatLab*, *Maxima* и т.п.).

Мы познакомимся ***системой компьютерной алгебры Maxima.***

Основными преимуществами программы являются:

1. Возможность свободного использования (*Maxima* относится к классу свободных программ и распространяется на основе лицензии GNU).

   GNU General Public License **–** (*Универсальная общедоступная лицензия GNU* или *Открытое лицензионное соглашение GNU*) **–** наиболее популярная лицензия на свободное программное обеспечение, созданная в 1988 г. Её также сокращённо называют ***GNU GPL*** или даже просто ***GPL***. Эта лицензия предоставляет пользователям компьютерных программ следующие права:

1) свободу запуска программы, с любой целью;

2) свободу изучения того, как программа работает, и её модификации;

3) свободу распространения копий;

4)свободу улучшения программы, и выпуска улучшений в публичный доступ;

2. Возможность функционирования под управлением различных ОС (в частности *Linux* и *Windows*);

3. Небольшой размер программы (дистрибутив занимает порядка 23 мегабайт, в установленном виде со всеми расширениями потребуется около 80 мегабайт);

4. *Maxima* имеет удобный графический интерфейс (*wxMaxima*) на русском языке, а также есть возможность работать в режиме командной строки.

5. *Maxima* дает возможность решать широкий класс задач.

Недостаток: отсутствие справки на русском языке, но в сети Интернет присутствует большое количество статей с примерами использования *Maxima*;

В *Maxima* сейчас принят такой же принцип нумерации версий, как и в ядре *Linux*: номер состоит из трёх чисел, разделённых точками, причём номера с нечётным средним числом соответствуют так называемым разрабатываемым версиям, с чётным – к стабильным версиям.

Установить последнюю версию программы на свой компьютер можно с ее сайта в сети Интернет: http://maxima.sourceforge.net/. Русская локализация сайта: http://maxima.sourceforge.net/ru/.

§2. ВВОД ПРОСТЕЙШИХ КОМАНД В *MAXIMA*

После запуска *Maxima* появляется окно программы, в верхней графической части окна интерфейса указано, какая загружена версия.

Попробуем набрать несколько команд.

Разделителем команд является символ “ ; ” (в ранних версиях *Maxima* и некоторых ее оболочках наличие точки с запятой после каждой команды строго обязательно, поэтому рекомендуется добавлять ; после каждой команды).

После ввода команды необходимо нажать клавиши *Shift* и *Enter* для ее обработки и вывода результата.

После ввода каждой команде присваивается порядковый номер (%i1), (%i2), (%i3) и т.д. Результаты вычислений имеют соответственно порядковый номер (%o1), (%o2) и т. д. Где "i" –сокращение от англ. input (ввод), а "o" – англ. output (вывод).

Это позволяет при дальнейшей записи команд сослаться на ранее записанные, например (%i1)+(%i2) будет означать добавление к выражению первой команды выражения второй с последующим вычислением результата. Также можно использовать и номера результатов вычислений, например, таким образом (%o1)*(%o2).

Используемые обозначения для ввода команд.

Ввод числовой информации

Правила ввода чисел в *Maxima* точно такие, как и для многих других подобных программ.

- Целая и дробная часть десятичных дробей разделяются символом точка.

- Перед отрицательными числами ставится знак минус.
- Числитель и знаменатель обыкновенных дробей разделяется при помощи символа / (прямой слеш).

```
(%i1) 2/3+5/6;
       3
(%o1)  -
       2
(%i2) 2/3+5/6,numer;
(%o2) 1.5
```

Обратите внимание, что если в результате выполнения операции получается некоторое символьное выражение, а необходимо получить конкретное числовое значение в виде десятичной дроби, то решить эту задачу позволит применение оператора numer. В частности он позволяет перейти от обыкновенных дробей к десятичным.

Константы

В *Maxima* для удобства вычислений есть ряд встроенных констант:

Название	Обозначение
π (число Пи)	%pi
e (экспонента)	%e
+∞ (плюс бесконечность)	inf
-∞ (минус бесконечность)	minf
Комплексная бесконечность	infinity
Мнимая единица ($\sqrt{-1}$)	%i
Истина	true
Ложь	false

Арифметические операции

Для обозначения арифметических операций в *Maxima* используются математические знаки: «+» - сложение, «–» - вычитание, «*» - умножение, «/»- деление.

Возведение в степень можно обозначать тремя способами: ^ , ^^ , **. Извлечение корня степени *n* записывают, как степень ^^(1/*n*).

Напомним еще одну встроенную в *Maxima* полезную операцию – нахождение факториала числа. Эта операция обозначается восклицательным знаком.

Например, 6!=1. 2. 3. 4. 5. 6=120.

Для увеличения приоритета операции, как и в математике, при записи команд для *Maxima* используют круглые ( ) скобки.

Переменные

Для хранения результатов промежуточных расчетов применяются переменные. Заметим, что при вводе названий переменных, функций и констант важен регистр букв, так переменные x и X – это две разные переменные. Присваивание значения переменной осуществляется с использованием символа

: (двоеточие),

Например, *x*: 5- «переменной x присвоено значение 5» или *b: a^2+3*- «переменная *b* будет иметь значение равное a^2+3».

Если необходимо удалить значение переменной (очистить ее), то применяется метод kill:

kill(x) – удалить значение переменной x;

kill(all) – удалить значения всех используемых ранее переменных.

Кроме того, kill начинает новую нумерацию для исполняемых команд.

Математические функции

В *Maxima* имеется достаточно большой набор встроенных математических функций. Для записи функции необходимо указать ее название, а затем, в круглых скобках записать через запятую значения аргументов.

Например, sin(x);

Следует иметь в виду, что некоторые названия функций отличаются от названий, используемых в отечественной литературе:

Функции	*Обозначение*
Тригонометрические	sin(x) (синус), cos(x) (косинус), tan(x) (тангенс), cot(x) (котангенс), sec(x) (секанс, $\frac{1}{\cos(x)}$), csc(x) (косеканс, $\frac{1}{\sin(x)}$).
Обратные тригонометрические	asin(x) (арксинус), acos(x) (арккосинус), atan(x) (арктангенс), acot(x) (арккотангенс).

Гиперболические	sinh(x) (гиперболический синус), cosh(x) (гиперболический косинус), tanh(x) (гиперболический тангенс), coth(x) (гиперболический котангенс), sech(x) (гиперболический секанс), csch(x) (гиперболический косеканс).
Натуральный логарифм, Остаток от деления Квадратный корень Модуль	log(x) mod(x) sqrt(x) abs(x)
Минимальный элемент из списка	min($x1$,$x2$,…,xn)
Максимальный элемент из списка	max($x1$, $x2$,…,xn)

Пользовательские функции

Пользователь может задать собственные функции. Для этого сначала указывается название функции, в скобках перечисляются названия аргументов, после знаков := (двоеточие и равно) следует описание функции. После задания пользовательская функция вызывается точно так, как и встроенные функции *Maxima*.

Пример.

```
(%i1) f(x):=x^2;
(%o1) f(x):=x^2
(%i2) f(3+sin(%pi/2));
(%o2) 16
```

Как записать число *π* и *e*

```
(%i1) ?pi;
(%o1) 3.141592653589793

(%i2) sin(pi);
(%o2) sin(π)

(%i3) sin(%pi);
(%o3) 0
```

```
(%i5) log(e);
(%o5) log(e)

(%i6) log(%e);
(%o6) 1
```

§3. РЕШЕНИЕ ЗАДАЧ ЭЛЕМЕНТАРНОЙ МАТЕМАТИКИ

К таким задачам можно отнести

- вычисление и преобразование арифметических выражений,
- построение графиков функций,
- решение уравнений и систем алгебраических уравнений.

Упрощение выражений

Рассмотрим возможности *Maxima* по упрощению и прочим преобразованиям выражений. В частности, речь пойдет

- об автоматическом раскрытии скобок и вынесении за скобки;
- об упрощении как арифметических действий над некоторыми элементами, так и выражений с участием степенных, показательных и логарифмических функций;
- об обработке тригонометрических выражений.

Все эти функции призваны облегчать читаемость математических формул и повышать простоту их восприятия.

rat – эта функция преобразовывает рациональное выражение к так называемой *канонической форме*, то есть раскрывает все скобки, затем приводит все к общему знаменателю, суммирует и сокращает; кроме того, приводит все числа в конечной десятичной записи к рациональным.

```
(%i1)    (x-1)^2/(x^2+x)+1/(x+1);
```

(%o1) $\frac{(x-1)^2}{x^2+x}+\frac{1}{x+1}$

```
(%i2)  rat(%o1);
```

(%o2)/R/ $\frac{x^2-x+1}{x^2+x}$

Если требуется вычислить числовое значение, полученного выражения, то можно применить функцию at, указав в скобках выражение или его адрес и значение переменной.

```
(%i3)  at(%o2,x=-2);
```

(%o3) $\frac{7}{2}$

divide – нахождение частного и остатка от деления одного многочлена на другой

```
(%i10)  divide(x^3-2,x-1);
```

(%o10) $[x^2+x+1, -1]$

Первый элемент полученного массива – частное, второй – остаток от деления.

factor – разложение на множители

```
(%i19)  factor(a*x^2+a*x+a);
```

(%o19) $a\left(x^2+x+1\right)$

```
(%i20)  factor(x^2+2*x+1);
```

(%o20) $(x+1)^2$

expand – раскрытие скобок

```
(%i9) expand((2+3*x)*(3*y+5*x));
```

(%o9) $9\,x\,y+6\,y+15\,x^{2}+10\,x$

gcd – наибольший общий делитель многочленов

```
(%i11) gcd(x^3-1, x^2-1, x-1);
```

(%o11) $x-1$

ratsimp – упрощение выражения

```
(%i12) a/(5*x)+b/x-c/x;
```

(%o12) $-\frac{c}{x}+\frac{b}{x}+\frac{a}{5\,x}$

```
(%i13) ratsimp(a/(5*x)+b/x-c/x);
```

(%o13) $-\frac{5\,c-5\,b-a}{5\,x}$

fullratsimp(выражение) – применяется для более серьезных упрощений. Она последовательно применяет к переданному выражению функцию ratsimp, а так же некоторые нерациональные преобразования, до тех пор, пока выражение не перестанет упрощаться.

```
(%i1) (x^(a/2)-1)^2*(x^(a/2)+1)^2/(x^a-1);
```

(%o1) $\frac{(x^{a/2}-1)^{2}\,(x^{a/2}+1)^{2}}{x^{a}-1}$

```
(%i2) ratsimp(%);
```

(%o2) $\frac{x^{2\,a}-2\,x^{a}+1}{x^{a}-1}$

```
(%i3) fullratsimp(%o1);
```

(%o3) $x^{a}-1$

partfrac – преобразует в простые дроби по заданной переменной

Пусть дано выражение $-\frac{x}{x^3+4x^2+5x+2}$

```
(%i31) partfrac(-x/(x^3+4*x^2+5*x+2),x);
```

(%o31) $\frac{2}{x+2}-\frac{2}{x+1}+\frac{1}{(x+1)^2}$

radcan – преобразует выражения, содержащие логарифмические, показательные и степенные функции.

```
(%i1) (sqrt((x-a)^3)-(x+a)*sqrt(x-a))/sqrt((x-a)*(x+a));
```

(%o1) $\frac{(x-a)^{3/2}-\sqrt{x-a}\,(x+a)}{\sqrt{(x-a)(x+a)}}$

```
(%i2) radcan(%);
```

(%o2) $-\frac{2\,a}{\sqrt{x+a}}$

Иногда требуется последовательно несколько раз применить функцию radcan.

```
(%i1) (b^(log(a)/log(b))-1)/(a^(1/2)+1);
```

(%o1) $\frac{b^{\frac{\log(a)}{\log(b)}}-1}{\sqrt{a}+1}$

```
(%i2) radcan(%);
```

(%o2) $\frac{a-1}{\sqrt{a}+1}$

```
(%i3) radcan(%);
```

(%o3) $\sqrt{a}-1$

Для упрощения тригонометрических выражений применяется приставка trig. Для наилучшего результата ее можно комбинировать с функциями ratsimp, fullratsimp, radcan и другими.

trigsimp – тригонометрическое упрощение

```
(%i43) tan(x)*sec(x)^2+((1-sin(x)^2)*cos(x))/cos(x)^2;
```

$$(\%o43)\quad sec(x)^2\, tan(x)+\frac{1-sin(x)^2}{cos(x)}$$

```
(%i44) trigsimp(%);
```

$$(\%o44)\quad \frac{sin(x)+cos(x)^4}{cos(x)^3}$$

trigexpand – тригонометрическое раскрытие скобок. Использует формулы преобразования сумм двух углов для представления введенного выражения в как можно более простом виде – где в качестве аргумента только одна переменная

```
(%i25) trigexpand(sin(u+v)*cos(u)^3);
```

$$(\%o25)\quad cos(u)^3(cos(u)sin(v)+sin(u)cos(v))$$

§4. РЕШЕНИЕ ЗАДАЧ ЛИНЕЙНОЙ АЛГЕБРЫ

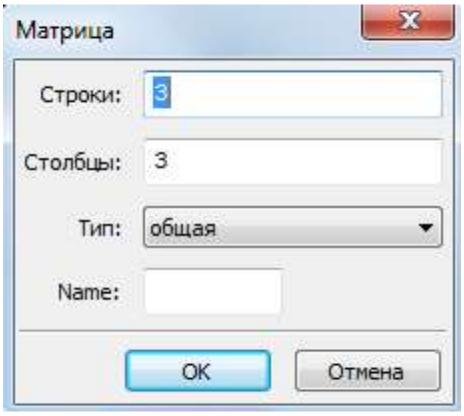
Для задания матрицы используется функция matrix:

```
(%i1) matrix([15,2],[-7,10]);
```

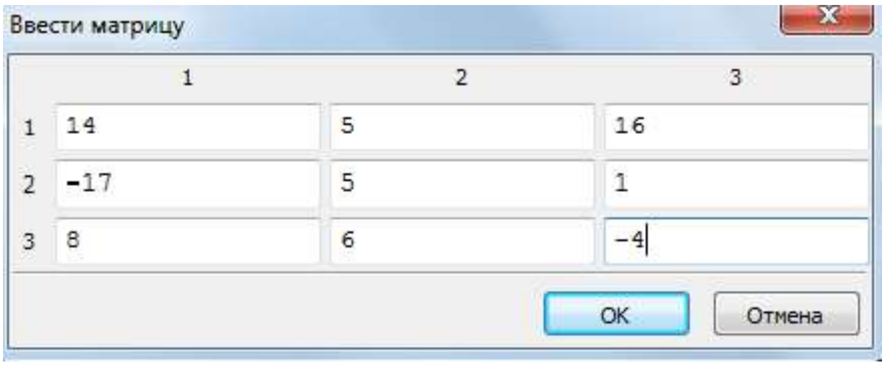
$$(\%o1)\quad \begin{bmatrix} 15 & 2 \\ -7 & 10 \end{bmatrix}$$

Командой (%i1) записывается матрица $\begin{bmatrix} 15 & 2 \\ -7 & 10 \end{bmatrix}$ и в строке (%o1) записан ответ $\begin{bmatrix} 15 & 2 \\ -7 & 10 \end{bmatrix}$.

Интерфейс *wxMaxima* достаточно удобен и не требует умения запоминать и безошибочно вводить длинный текст для вызова функции matrix. Достаточно лишь заполнить вспомогательные формы. Используя меню, щелкнуть по кнопкам «Алгебра → Enter Matrix…». При этом появится окно, которое необходимо заполнить, щелкнуть по команде «ОК»,



далее появится следующее окно, которое нужно заполнить и по команде «ОК» получим результат



(%i2) matrix([14,5,16],[-17,5,1],[8,6,-4]);

(%o2) $\begin{bmatrix} 14 & 5 & 16 \\ -17 & 5 & 1 \\ 8 & 6 & -4 \end{bmatrix}$

В этих примерах были определены две квадратные матрицы: второго и третьего порядка.

Действия с матрицами

Рассмотрим основные действия с матрицами на примере. Для этого зададим две матрицы: A и B.

```
(%i3) A: matrix([15,2],[-7,10]);
```

$$(\%o3)\quad \begin{bmatrix} 15 & 2 \\ -7 & 10 \end{bmatrix}$$

```
(%i4) B: matrix([-6,4],[5,13]);
```

$$(\%o4)\quad \begin{bmatrix} -6 & 4 \\ 5 & 13 \end{bmatrix}$$

Поэлементное сложение, вычитание, умножение матриц на число.

```
(%i5) A+B;
```

$$(\%o5)\quad \begin{bmatrix} 9 & 6 \\ -2 & 23 \end{bmatrix}$$

```
(%i6) A-B;
```

$$(\%o6)\quad \begin{bmatrix} 21 & -2 \\ -12 & -3 \end{bmatrix}$$

```
(%i7) k.A;
```

$$(\%o7)\quad k\ .\ \begin{bmatrix} 15 & 2 \\ -7 & 10 \end{bmatrix}$$

Умножение матриц

```
(%i8) A.B;
```

$$(\%o8)\quad \begin{bmatrix} -80 & 86 \\ 92 & 102 \end{bmatrix}$$

Вычисление матрицы, обратной данной

```
(%i9) A^^-1;
```

$$(\%o9)\quad \begin{bmatrix} \frac{5}{82} & -\frac{1}{82} \\ \frac{7}{164} & \frac{15}{164} \end{bmatrix}$$

Так же обратную матрицу можно получить с помощью функции invert

```
(%i10) invert(matrix([15,2],[-7,10]));
```

(%o10) $\begin{bmatrix} \frac{5}{82} & -\frac{1}{82} \\ \frac{7}{164} & \frac{15}{164} \end{bmatrix}$

Функции для работы с матрицами

determinant – нахождение определителя матрицы

```
(%i11) determinant(matrix([15,2],[-7,10]));
(%o11) 164
```

eigenvalues – нахождение собственных значений матрицы

```
(%i12) eigenvalues(A);
```

(%o12) $[[-\frac{\sqrt{31}\,\%i-25}{2},\frac{\sqrt{31}\,\%i+25}{2}],[1,1]]$

minor – определяет минор матрицы. Первый аргумент – матрица, второй и третий – индексы строки и столбца соответственно

```
(%i13) minor(matrix([15,2],[-7,10]),1,1);
```

(%o13) $\begin{bmatrix} 10 \end{bmatrix}$

rank – ранг матрицы

```
(%i14) rank(matrix([15,2],[-7,10]));
(%o14) 2
```

submatrix – возвращает матрицу, полученную из исходной удалением соответствующих строк и (или) столбцов. В качестве параметров следуют номера удаляемых строк, исходная матрица, номера удаляемых столбцов.

```
(%i15) M: matrix([2,6,7,9],[a,d,s,r],[4,8,3,5]);
```

$$(\%o15)\quad \begin{bmatrix} 2 & 6 & 7 & 9 \\ a & d & s & r \\ 4 & 8 & 3 & 5 \end{bmatrix}$$

```
(%i16) submatrix(1,2,M);
```

$$(\%o16)\quad \begin{bmatrix} 4 & 8 & 3 & 5 \end{bmatrix}$$

transpose – транспонирование матрицы

```
(%i17) transpose(matrix([15,2],[-7,10]));
```

$$(\%o17)\quad \begin{bmatrix} 15 & -7 \\ 2 & 10 \end{bmatrix}$$

Решение систем линейных алгебраических уравнений

Рассмотрим различные способы решения систем.

- **Метод Крамера**

Пример. Решить систему линейных алгебраических уравнений методом Крамера:

$$\begin{cases} x+2y+10z=-15, \\ 2x+4y-z=12, \\ x+y-3z=9. \end{cases}$$

Решение.

Вычислим основной определитель системы:

```
(%i1) D:matrix([1,2,10],[2,4,-1],[1,1,-3]);
      D:determinant(D);
```

$$(\%o1)\quad \begin{bmatrix} 1 & 2 & 10 \\ 2 & 4 & -1 \\ 1 & 1 & -3 \end{bmatrix}$$

$$(\%o2)\quad -21$$

Командой (%i1) записываются коэффициенты при неизвестных x, y, z и вычисляется этот определитель, в строке (%o1) записан определитель $\begin{bmatrix} 1 & 2 & 10 \\ 2 & 4 & -1 \\ 1 & 1 & -3 \end{bmatrix}$ и получен ответ в строке (%o2) $D=-21$.

Так как $D=-21\neq 0$, то система имеет единственное решение. Чтобы получить его, необходимо вычислить дополнительные определители D_x, D_y, D_z и используя формулы Крамера $x=\frac{D_x}{D}$, $y=\frac{D_y}{D}$, $z=\frac{D_z}{D}$ найти неизвестные x, y, z.

Вычислим дополнительные определители.

```
(%i3) Dx:matrix([-15,2,10],[12,4,-1],[9,1,-3]);
      Dx:determinant(Dx);
      x=Dx/D;
```

$$(\%o3) \begin{bmatrix} -15 & 2 & 10 \\ 12 & 4 & -1 \\ 9 & 1 & -3 \end{bmatrix}$$

```
(%o4) -21
(%o5) x=1
```

Командой (%i3) задается определитель D_x, составленный следующим образом:

- первая строка записывается на основе коэффициентов первого уравнения системы, взятых в следующем порядке: свободный член, коэффициент при неизвестном y, коэффициент при неизвестном z,

- вторая и третья строки записываются аналогично из коэффициентов второго и третьего уравнений соответственно.

В строке (%o4) вычисляется определитель $D_x=-21$, в строке (%o5) находим неизвестную $x=1$.

```
(%i6) Dy:matrix([1,-15,10],[2,12,-1],[1,9,-3]);
      Dy:determinant(Dy);
      y=Dy/D;
```

$$(\%o6)\quad \begin{bmatrix} 1 & -15 & 10 \\ 2 & 12 & -1 \\ 1 & 9 & -3 \end{bmatrix}$$

(%o7) -42

(%o8) $y=2$

Командой (%i6) записывается определитель D_y, составленный следующим образом:

- первая строка (из первого уравнения) – коэффициент при неизвестном x, свободный член, коэффициент при неизвестном z,

- вторая и третья строки записываются аналогично (из второго и третьего уравнения).

В строке (%o7) вычисляется дополнительный определитель $D_y = -42$, в строке (%o8) находим неизвестную $y = 2$.

```
(%i9) Dz:matrix([1,2,-15],[2,4,12],[1,1,9]);
      Dz:determinant(Dz);
      z=Dz/D;
```

$$(\%o9)\quad \begin{bmatrix} 1 & 2 & -15 \\ 2 & 4 & 12 \\ 1 & 1 & 9 \end{bmatrix}$$

(%o10) 42

(%o11) $z=-2$

Командой (%i9) записывается определитель D_z, составленный следующим образом:

- первая строка (из первого уравнения) – коэффициент при неизвестном x, коэффициент при неизвестном y, свободный член,

- вторая и третья строки записываются аналогично.

В строке (%i10) вычисляется дополнительный определитель $D_y = 42$, в строке (%i11) находим неизвестную $z = -2$.

Ответ: $(1;2;-2)$.

Чтобы не набирать команды вручную можно используя меню провести действия, описанные выше «щелкнуть по кнопкам «Алгебра → Enter Matrix…». При этом появится окно, которое необходимо заполнить, щелкнуть по команде «ОК»…»

- **Решение системы линейных алгебраических уравнений с помощью обратной матрицы**

Пример. Решить систему линейных алгебраических уравнений с помощью обратной матрицы:

$$\begin{cases} 2x-3y+4z=21, \\ 4x-2y+3z=42, \\ 3x+4y-2z=21. \end{cases}$$

Решение.

```
(%i12) A:matrix([2,-3,4],[4,-2,3],[3,4,-2]);
```

$$(\%o12) \quad \begin{bmatrix} 2 & -3 & 4 \\ 4 & -2 & 3 \\ 3 & 4 & -2 \end{bmatrix}$$

Командой (%i12) записывается матрица A – матрица, составленная из коэффициентов при незвестных x, y, z.

```
(%i13) B:matrix([21],[42],[21]);
```

$$(\%o13) \quad \begin{bmatrix} 21 \\ 42 \\ 21 \end{bmatrix}$$

Командой (%i13) записывается матрица B – матрица, составленная из свободных членов.

```
(%i14) A^^(-1);
```

$$(\%o14)\quad \begin{bmatrix} -\frac{8}{21} & \frac{10}{21} & -\frac{1}{21} \\ \frac{17}{21} & -\frac{16}{21} & \frac{10}{21} \\ \frac{22}{21} & -\frac{17}{21} & \frac{8}{21} \end{bmatrix}$$

Командой (%i14) записывается матрица, обратная данной матрице A.

```
(%i15) A^^(-1).B;
```

$$(\%o15)\quad \begin{bmatrix} 11 \\ -5 \\ -4 \end{bmatrix}$$

Командой (%i15) обратная матрица умножается на матрицу B.

Ответ: $(11; -5; -4)$.

- **Методом Гаусса**

Пример. Решить систему линейных алгебраических уравнений методом Гаусса

$$\begin{cases} x + 2y - 8z = 14, \\ 2x + 4y - 7z = 13, \\ 5x + 3y + 4z = -2. \end{cases}$$

Решение.

Введем расширенную матрицу A

```
(%i16) A:matrix([1,2,-8,14],[2,4,-7,13],[5,3,4,-2]);
```

$$(\%o16)\quad \begin{bmatrix} 1 & 2 & -8 & 14 \\ 2 & 4 & -7 & 13 \\ 5 & 3 & 4 & -2 \end{bmatrix}$$

и приведем ее к треугольному виду

```
(%i17) echelon(A);
```

$$(\%o17)\quad \begin{bmatrix} 1 & 2 & -8 & 14 \\ 0 & 1 & -\frac{44}{7} & \frac{72}{7} \\ 0 & 0 & 1 & -\frac{5}{3} \end{bmatrix}$$

решим, получим искомые значения

```
(%i18) solve([x+2*y-8*z=14, y-(44/7)*z=72/7, z=-5/3], [x,y,z]);
```

$$(\%o18)\quad [[x=\frac{22}{21}, y=-\frac{4}{21}, z=-\frac{5}{3}]]$$

Ответ: $\left(\frac{22}{21};-\frac{4}{21};-\frac{5}{3}\right)$.

- Любую из трех выше приведенных систем можно с помощью команды solve

solve([*уравнение*1, *уравнение*2, …],[*переменная*1, *переменная*2, …]).

Пример:

```
(%i19) solve([x+2*y-8*z=14, 2*x+4*y-7*z=13, 5*x+3*y+4*z=-2],
       [x,y,z]);
```

$$(\%o19)\quad [[x=\frac{22}{21}, y=-\frac{4}{21}, z=-\frac{5}{3}]]$$

Ответ: $\left(\frac{22}{21};-\frac{4}{21};-\frac{5}{3}\right)$.

§5. ПОСТРОЕНИЕ ГРАФИКОВ ФУНКЦИЙ

В математике удобно полученное решение выводить в графическом виде. Система компьютерной математики *Maxima* может строить графики двумерных и трехмерных функций, заданных в явном виде, в параметрическом виде, в виде таблицы.

Рассмотрим основные графические функции. В *Maxima* имеется несколько альтернативных библиотек для отображения графиков функций. По умолчанию используется библиотека Plot. Также для решения некоторых задач используются библиотеки Draw.

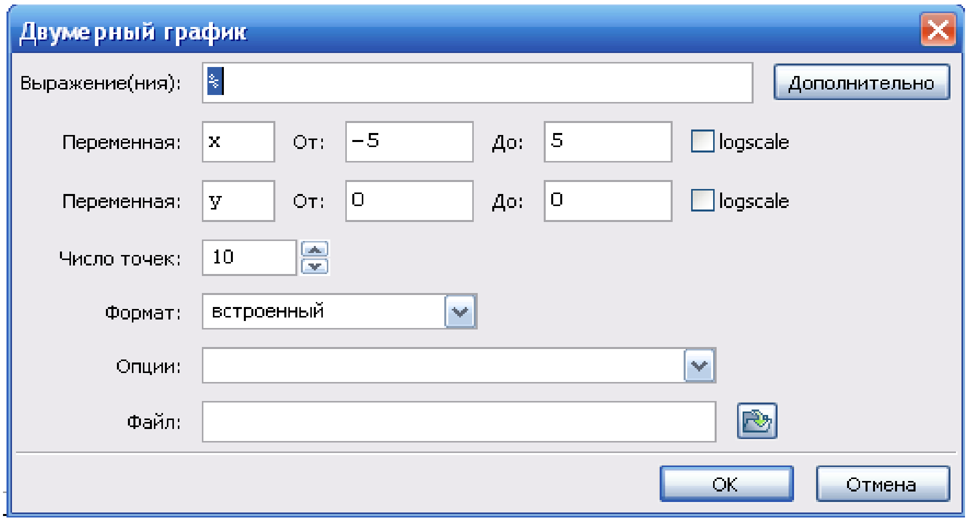
Построение графиков функций, заданных в явном виде

Для построения двумерных графиков используются функция

plot2d([*y*1, *y*2], [*x*,*a*,*b*],[*y*,*c*,*d*])

Первый аргумент – список функций, второй и третий – ограничения по осям координат. Третий аргумент является необязательным. Если его не указать – он будет подобран автоматически.

Чтобы не вводить длинный вызов функции plot2d со всеми её параметрами, заполним вспомогательные формы для построения графика. Для этого в меню выбираем команду «Графики → Plot2d ...» После выполнения данной команды появляется окно с формой, которую необходимо заполнить.



В первой строке необходимо ввести уравнение функции или название функций, если функция была задана ранее. Если функций несколько, то они отделяются запятыми. Графики в этом случае автоматически нарисуются разными цветами.

При помощи кнопки Дополнительно можно выбрать либо параметрический (функция задана параметрическом виде), либо дискретный график (функция задана по точкам). Во второй строчке формы задается диапазон изменения переменной x (можно поменять на другое название, например t). В строке Формат можно выбрать один из методов построения графиков функций. В поле Опции можно выбрать некоторые параметры графика.

Возможные Форматы:

I) встроенный – график нарисуется в том же окне, что и командная строка.

II) gnuplot – график нарисуется в отдельном окне, и его можно масштабировать (изменять размеры за счет изменения размеров окна). При движении мышки внизу слева отображаются координаты положения указателя.

III) openmath – в этом формате график может видоизменяться в интерактивном режиме, в частности его можно масштабировать не только за счет изменения размеров окна, но и с помощью кнопок.

IV) по-умолчанию – построением графиков занимается gnuplot.

Возможные Опции:

I) set zeroaxis; – проводит оси через начало координат,

II) set grid; – прорисовывает сетку,

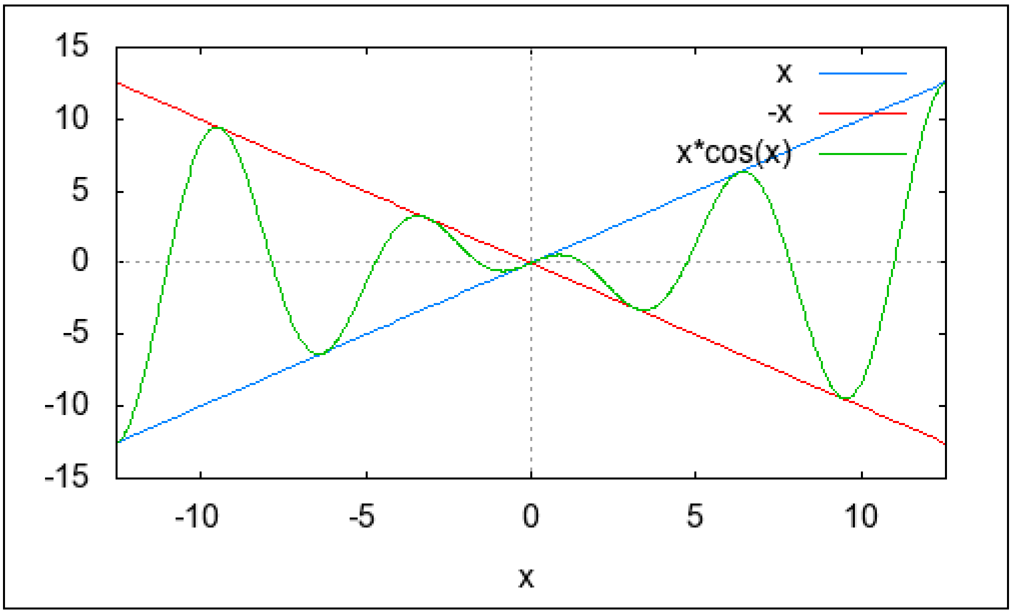
III) set size ratio 1; – выравнивает масштабы по осям координат, чтобы круг на мониторе выглядел круглым, а не в виде овала. Отметим, что последнее обстоятельство связано с тем, что разрешение монитора по горизонтали и по вертикали разное (пиксель не является «круглым»).

Пример. Построить графики функций

$y = x,\ \ y = -x,\ \ y = x\cos x$ на отрезке [-4 π;4π].

```
(%i2) wxplot2d([x,-x,x*cos(x)], [x,-4*%pi,4*%pi])$
```

(%t2)



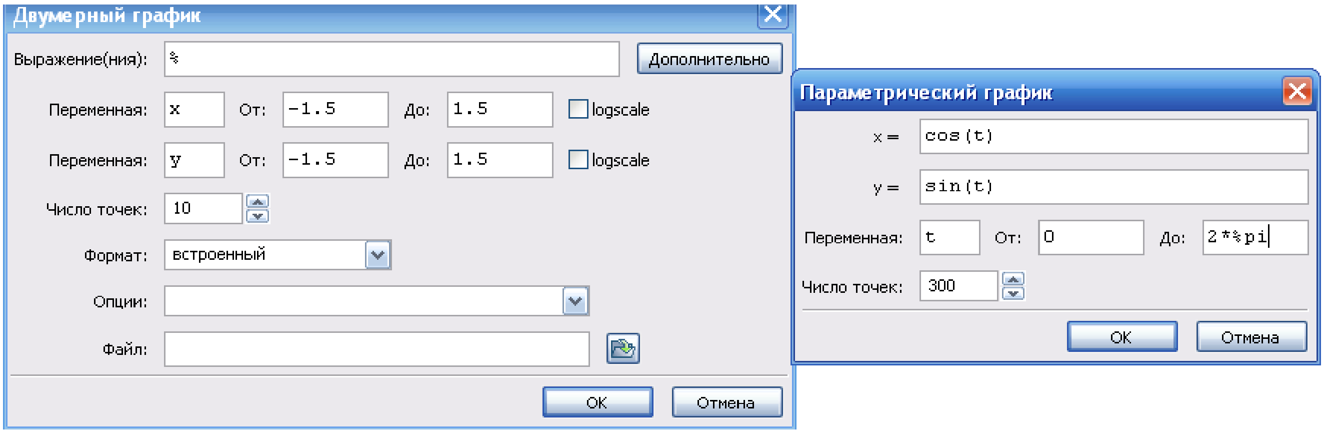
Построение графиков функций, заданных в параметрическом виде

Для построения графика параметрически заданной функции используется команда

plot2d([parametric, *x-выражение*, y-выражение, [*t*, *t*1, *t*2], [nticks, количество]]);

где *x-выражение* и *y-выражение* задают зависимость вида $x=x(t)$, $y=y(t)$, где t – переменная параметризации; [t, $t1$, $t2$] задает отрезок, в пределах которого параметр t будет изменяться; nticks задает количество кусочков, на которые будет разбит интервал изменения параметра при построении графика.

Для удобства набора команды достаточно заполнить две формы: после щелчка по кнопке «Графики → Plot2d …» появляется окно диалога *Двумерный график*, затем нажимаем на кнопку Дополнительно, на этой форме появляется второе окно Параметрический график.

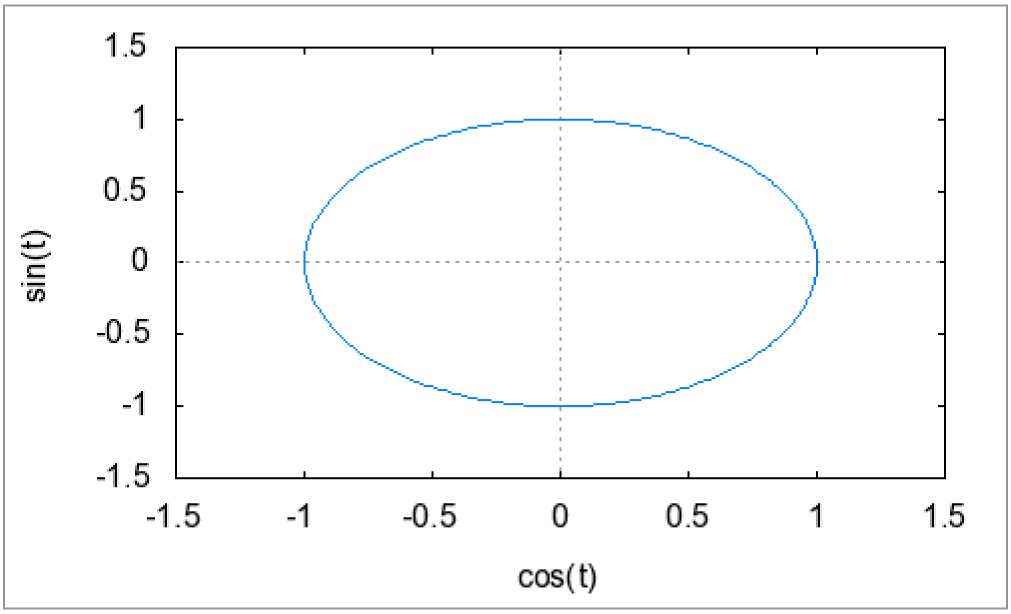


Пример. Построить окружность единичного радиуса.

$$\begin{cases} x = \cos t, \\ y = \sin t \end{cases} \qquad t \in [0, 2\pi)$$

```
--> wxplot2d([['parametric, cos(t), sin(t), [t, 0, 2*%pi],
    [nticks, 300]]], [x,-1.5,1.5], [y,-1.5,1.5])$
```

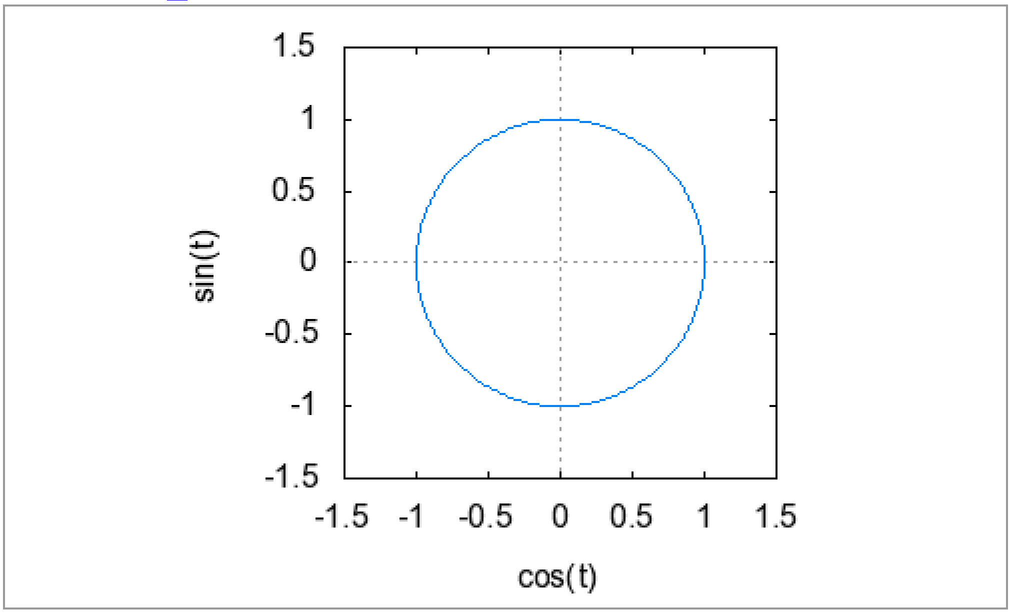
(%t1)



После выполнения этой команды на мониторе появляется эллипс, а не окружность. Чтобы этого не происходило необходимо использовать опцию set size ratio 1; которая выравнивает масштабы по осям координат. Тогда в результате получим

```
--> wxplot2d([['parametric, cos(t), sin(t), [t, 0, 2*%pi],
    [nticks, 300]]],[x,-1.5,1.5], [y,-1.5,1.5],
    [gnuplot_preamble, "set size ratio 1;set zeroaxis;"])$
```

(%t2)

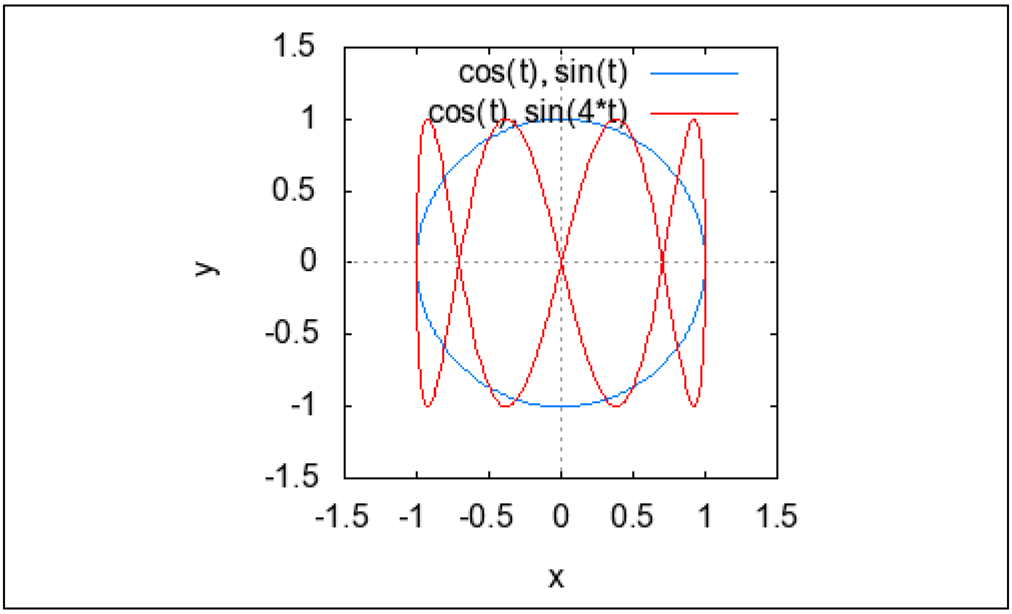


Для построения на одном рисунке двух параметрических графиков, нужно сначала построить первый график. Затем в графической части окна надо щелкнуть на тексте–вызове plot2d(['param... первого графика, а потом щелкнуть по кнопкам «Графики → Plot2d ...» и Дополнительно. После заполнения сведений о втором графике на одном рисунке появятся два графика. Графики автоматически будут нарисованы разными цветами.

<u>Пример.</u> Построить два графика на одном рисунк

$$\begin{cases} x = \cos t, \\ y = \sin t \end{cases} \text{ и } \begin{cases} x = \cos t, \\ y = \sin 4t \end{cases} \quad t \in [0, 2\pi)$$

```
(%i1) wxplot2d([['parametric,cos(t),sin(t),
      [t,0,2*%pi],[nticks,300]],['parametric,
      cos(t),sin(4*t),[t,0,2*%pi],[nticks,300]]],
      [x,-1.5,1.5],[y,-1.5,1.5],[gnuplot_preamble,
      "set size ratio 1; set zeroaxis;"]);
```

(%t1) 

(%o1)

Построение дискретных функций

Maxima может рисовать графики функций, заданных таблично. Для этого ей нужны два списка: один – для значений абсцисс дискретных точек, второй – для значений ординат этих точек. Командная строка в этом случае выглядит так

wxplot2d([['discrete, [*x*1,*x*2,*x*3,*x*4], [*y*1,*y*2,*y*3,*y*4]]], [*x*,*a*,*d*], [style,[points,3,2,6]]);

Стили бывают: точечный график (points), сплошная линия (lines) и линии с точками (linespoints).

[points,3,2,6] означает следующее

3 – толщина маркеров;

2 – номер цвета;

6 – тип маркера.

[lines,2,1] означает следующее

2 – толщина линии;

1 – цвет линии.

[linespoints,1,2,3,4] означает следующее

1 – толщина линии;

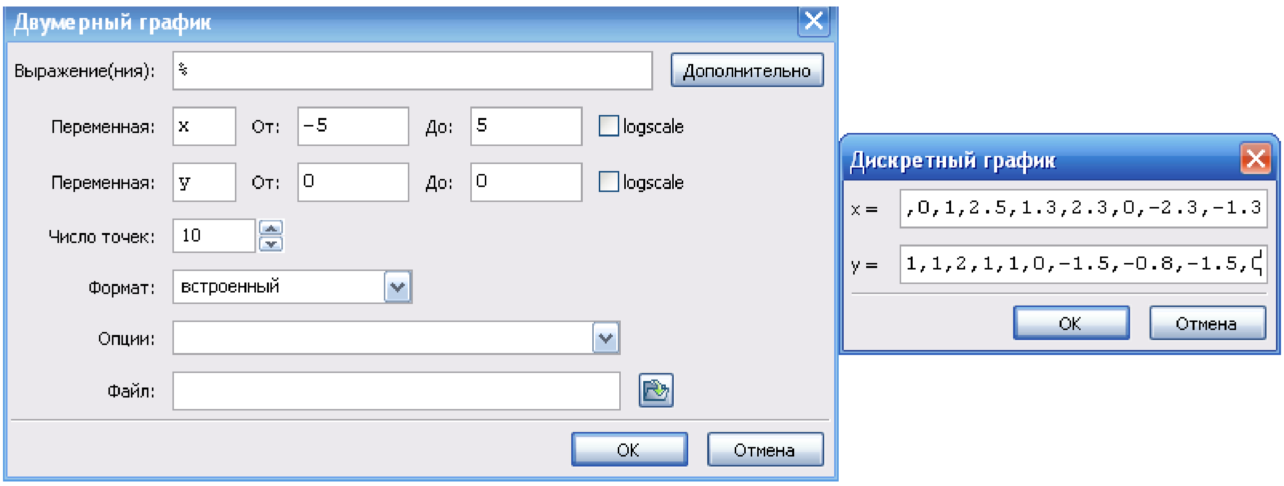
2 – толщина маркеров;

3 – номер цвет;

4 – тип маркера.

Предусмотрено 13 типов маркеров: 1– заполненные кружочки; 2 – незаполненные кружочки; 3 – знак +; 4 – крестик; 5 – звездочка; 6, 7– заполненный и незаполненный квадратик; 8, 9 – заполненный и незаполненный треугольник; 10,11 – повернутые заполненные и незаполненные треугольник; 12,13 – заполненные и незаполненные ромбик.

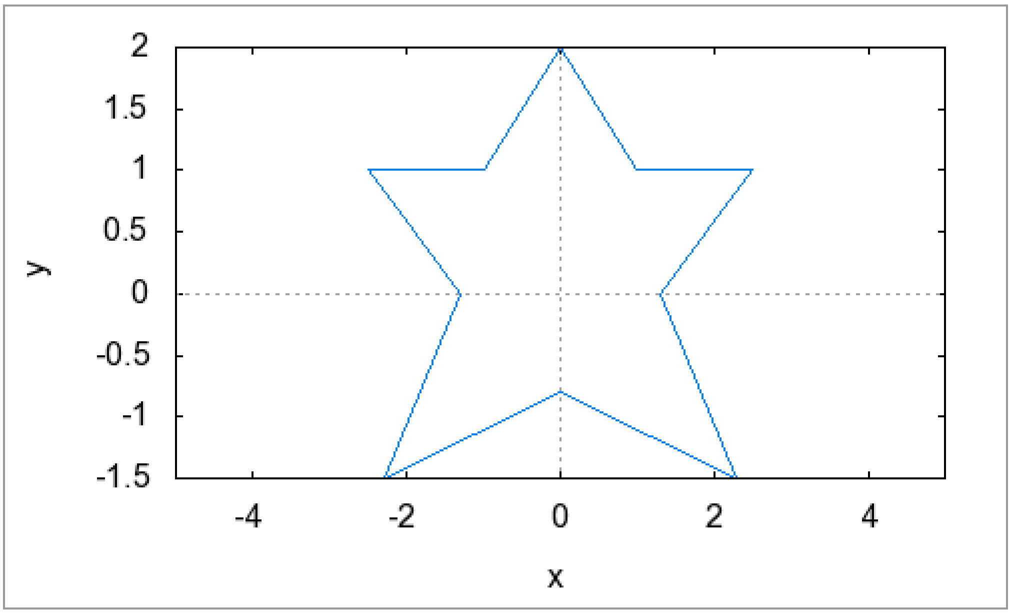
Для удобства набора команды можно заполнить две формы: после щелчка по кнопке «Графики → Plot2d …» появляется окно диалога *Двумерный график*, затем нажимаем на кнопку Дополнительно, на этой форме появляется второе окно Дискретный график.



Пример. Построить пятиконечную звезду

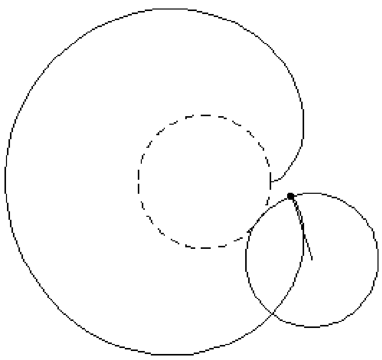
```
--> wxplot2d([['discrete, [-1.3,-2.5,-1,0,1,2.5,1.3,2.3,0,
-2.3,-1.3], [0,1,1,2,1,1,0,-1.5,-0.8,-1.5,0]]], [x,-5,5])$
```

(%t2)



Графики в полярной системе координат

Если использовать две окружности с одинаковыми радиусами и вращать одну вокруг другой, то получится *кардиоида* (греч. кардиа – сердце). По мнению математиков, получаемая кривая напоминающая сердце.



В прямоугольной декартовой системе координат уравнение кардиоиды имеет сложный вид

$$(x^2 + y^2 – ax)^2 = a^2 (x^2 + y^2).$$

Но в полярной системе координат уравнение кардиоиды имеет простой вид

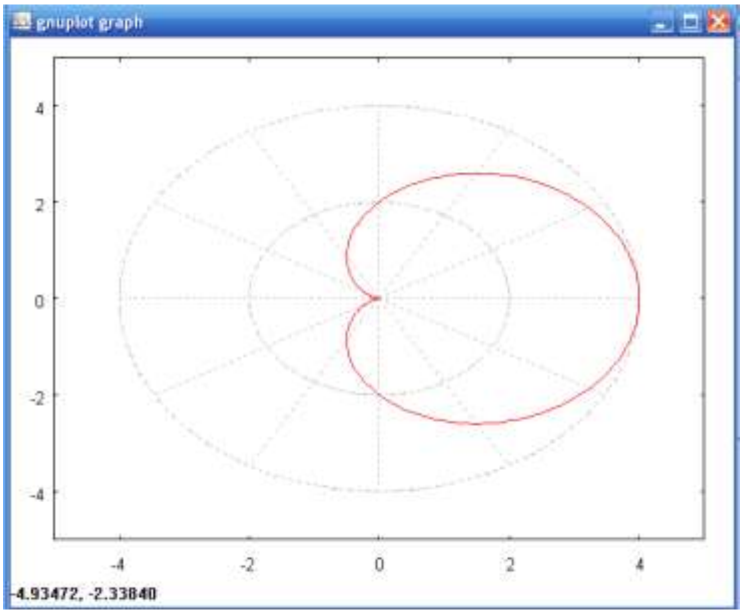
$$\rho = a + a \cos t,$$

где ρ – расстояние от точки кривой до начала координат, t– полярный угол, a – диаметр окружности.

В *Maxima* графики в полярной системе координат рисует функция draw2d(); но, прежде чем пользоваться этой функцией, нужно дополнительно загрузить этот модуль оператором load(draw).

Пример. Построить кардиоиду

```
(%i1) load(draw)$
(%i2) draw2d(user_preamble="set grid polar",nticks=300,
      color=red,xrange=[-5,5],yrange=[-5,5],
      polar(2*(1+cos(theta)),theta,0,2*%pi));
(%o2) [gr2d(polar)]
```



Графики функций, заданных неявно

В системе *Maxima* есть специальная команда, которая позволяет строить графики функций, заданных неявно. Ее синтаксис:

implicit_plot (*выражение*, *x_range*, *y_range*)

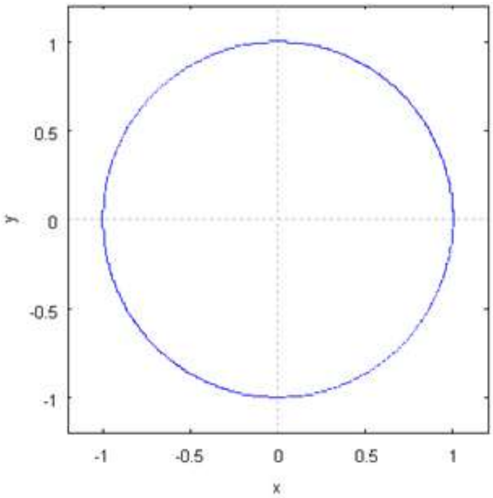
implicit_plot ([*выражение*1, *выражение*2, ...], *x_range*, *y_range*)

где *выражение* – это уравнение, задающее неявную функцию, *x_range* и y_range – промежутки изменения переменных *x* и *y*.

Для того, чтобы можно было использовать функцию implicit_plot, необходимо подключить пакет, содержащий эту функцию, с помощью команды load(implicit_plot).

Пример. Построить окружность единичного радиуса

```
(%i6) load(implicit_plot)$
(%i7) implicit_plot(x^2+y^2-1,[x,-1.2,1.2],
      [y,-1.2,1.2],[gnuplot_preamble,
      "set size ratio 1; set zeroaxis"]);
(%o7) done
```



Точки пересечения кривых

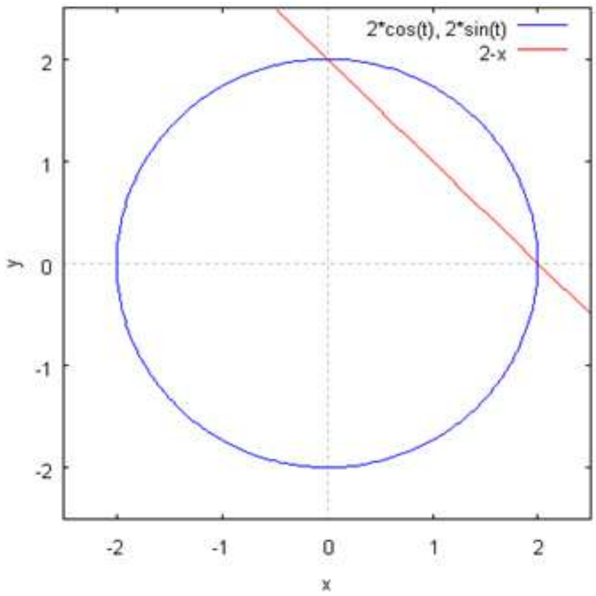
Чтобы найти точки пересечения кривых необходимо решить систему уравнений, составленную из уравнений этих кривых. В *Maxima* для этого можно использовать команду

solve([*уравнение*1,*уравнение*2, …],[*переменная*1, *переменная*2, …]).

Пример. Найти точки пересечения окружности $x^2 + y^2 = 4$ и прямой $x + y = 2$.

Построим графики окружности и прямой, чтобы убедиться в том, что решение существует. Окружность удобно задать в параметрическом виде

```
--> plot2d([['parametric, 2*cos(t), 2*sin(t),
     [t, 0, 2*%pi], [nticks, 300]],2-x],
     [x,-2.5,2.5],[y,-2.5,2.5], [plot_format,
     gnuplot], [gnuplot_preamble,
     "set size ratio 1; set zeroaxis;"])$
plot2d: some values were clipped.
```



Из графика видим, что решением исследуемой системы уравнений являются координаты 2 точек. Теперь решаем систему

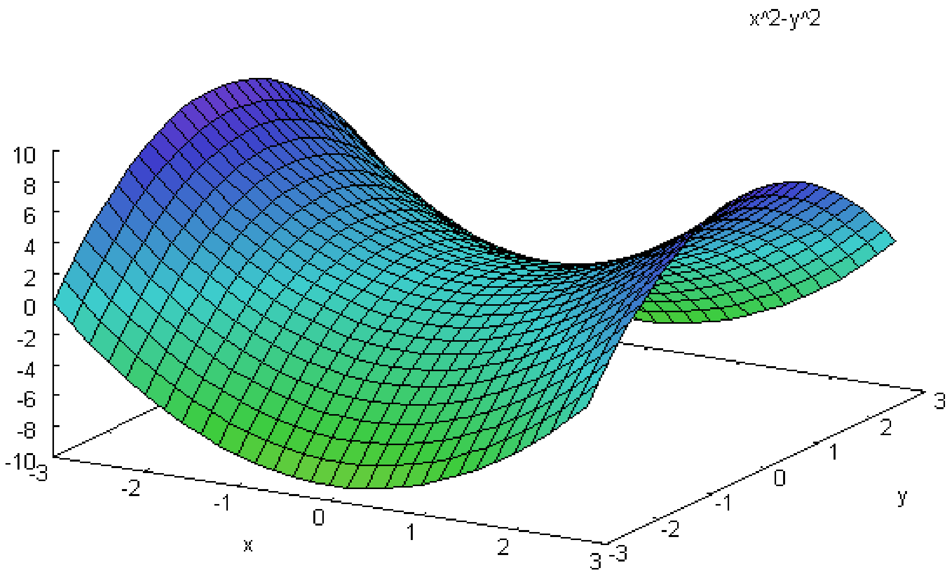
```
(%i2) solve([x^2+y^2=4,x+y=2],[x,y]);
(%o2) [[x=2,y=0],[x=0,y=2]]
```

Здесь в качестве решения возвратились координаты точек пересечения.

Построение трёхмерных графиков

Основная команда для построения трёхмерных графиков – plot3d.

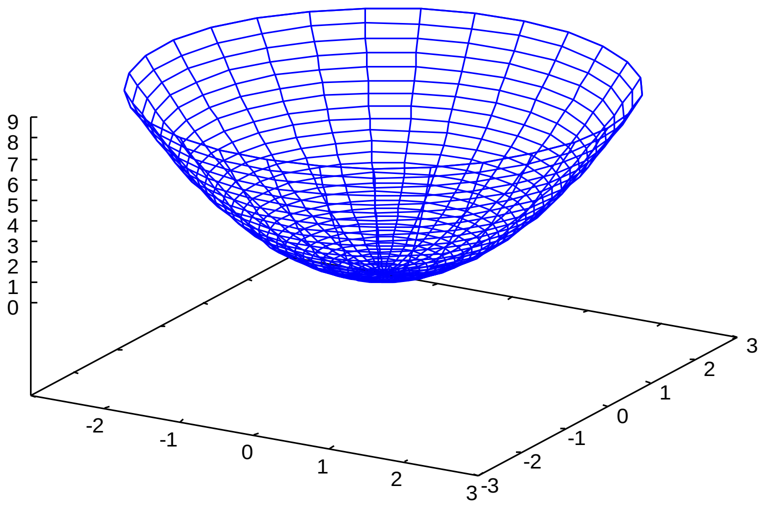
```
--> plot3d(x^2-y^2, [x,-3,3], [y,-3,3],
    [plot_format,gnuplot])$
```



Трехмерные графики удобно строить в параметрическом виде. Для этого используется библиотека Draw:

<u>Пример:</u> Построим параболоид вращения $z = x^2 + y^2$. В параметрическом виде уравнение параболоида имеет вид: $\begin{cases} x = s\ \cos t\,, \\ y = s \sin t\,, \\ z = s^2, \end{cases} \quad t \in [0,\ 2\pi]$.

```
(%i1) load(draw)$
(%i2) draw3d(parametric_surface(s*cos(t),s*sin(t),
      s^2,s,0,3,t,0,2*%pi))$
```



§6. ПРОГРАММИРОВАНИЕ В *MAXIMA*

НА ВСТРОЕННОМ МАКРОЯЗЫКЕ

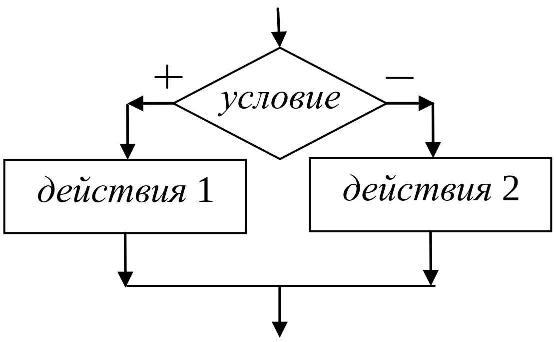
В *Maxima* можно программировать. Рассмотрим основные операторы.

Ветвление

Обеспечивает в зависимости от результата проверки условия (**да** или **нет**) выбор одного из альтернативных путей работы алгоритма. Каждый из путей ведет к **общему выходу**, так что работа алгоритма будет продолжаться независимо от того, какой путь будет выбран.

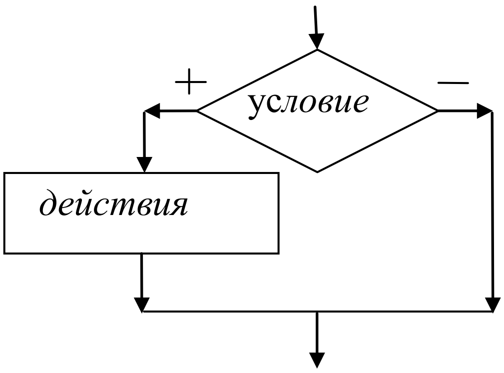
<u>Виды ветвлений:</u>

Условный оператор в *Maxima* имеет вид



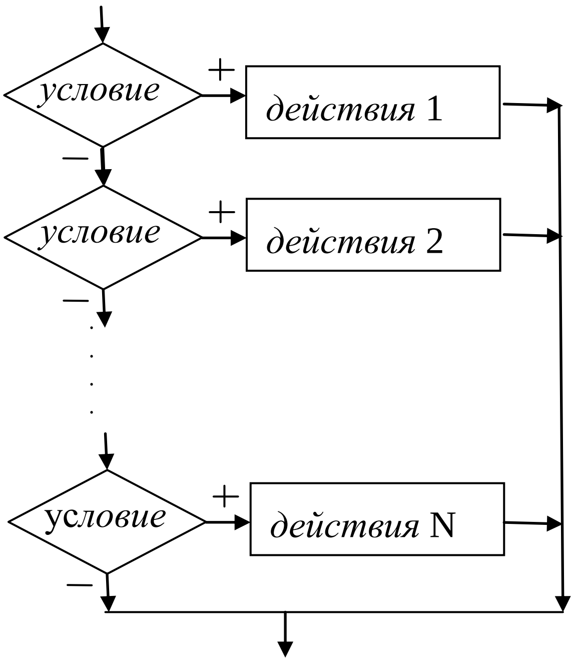
if *условие* then *действие*1 else *действие*2;

Если *условие* истинно, то выполняется выражение *действие*1, иначе – выполняется выражение *действие*2.



if *условие* then *действие*;

Если *условие* истинно, то выполняется выражение *действие*.



if *условие*1 then *действие*1 else

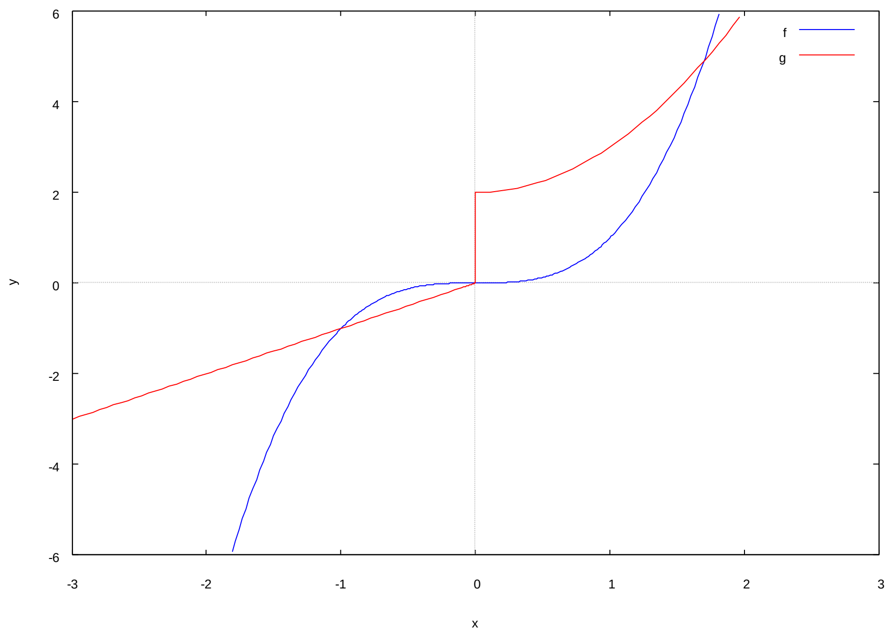
if *условие*2 then *действие*2 else

if...else *действие*N

Если выполняется *условие 1*, то выполняется выражение *действие1*, иначе – проверяется *условие2*, и если оно истинно – выполняется выражение *действие 2*, и т.д. Если ни одно из условий не является истинным – выполняется выражение *действиеN*.

<u>Пример:</u> Построить графики функций $y = x^3$ и $y = \begin{cases} x, & x < 0, \\ 2 + x^2, & x > 0. \end{cases}$

```
(%i1) f(x):=x^3;
(%o1) f(x):=x^3
(%i2) g(x):=if x<0 then x else 2+x^2;
(%o2) g(x):=if x<0 then x else 2+x^2
(%i3) plot2d([f,g],[x,-3,3],[y,-6,6]);
plot2d: some values were clipped.
 plot2d: some values were clipped.
  (%o3)
```

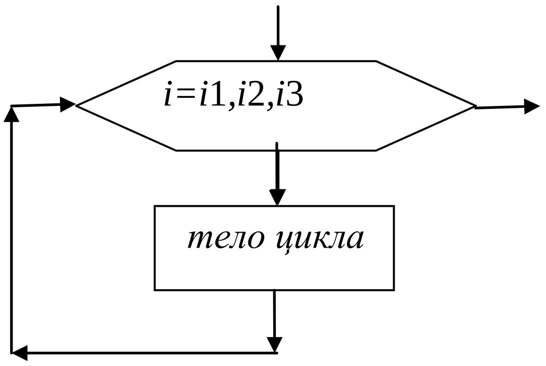


Цикл

Обеспечивает многократное выполнение некоторой совокупности действий, которая называется **телом** цикла.

<u>Виды циклов.</u>

✓ Цикл «**для**»



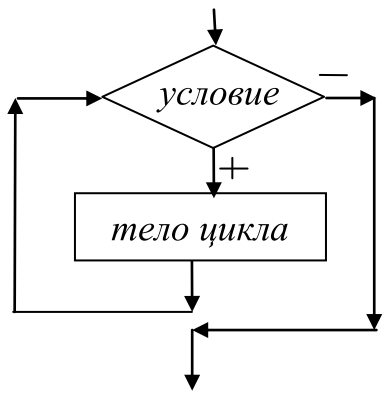
Оператор цикла в *Maxima*

Для выполнения итераций используется оператор do.

for *i*: *i*1 step *i*2 thru *i*3 do *тело цикла*;

Здесь *i* – переменная цикла; *i*1 – начальное значение; *i*2 – шаг (по умолчанию равен 1); *i*3 – конечное значение переменной цикла; *тело цикла* - операторы тела цикла. *i*1, *i*2, *i*3 и *тело цикла* могут быть произвольными выражениями. Ключевое слово thru указывает, что завершение цикла происходит при достижении переменной цикла значения *i*3;

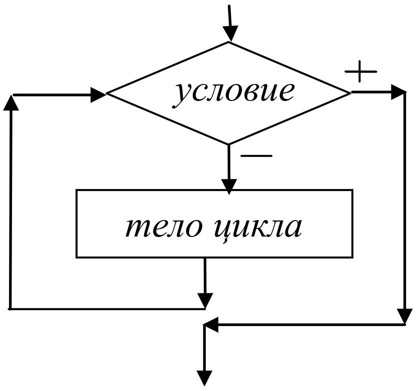
✓ Цикл «**пока**» (предписывает выполнять тело цикла до тех пор, пока выполняется условие)



for i: i1 step i2 while *условие* do *тело цикла*

Ключевое слово while указывает, что цикл выполняется, пока выполняется *условие*.

✓ Цикл «до» (предписывает выполнять тело цикла до тех пор, пока не будет выполняться некоторое условие)



for i: i1 step i2 unless *условие* do *тело цикла*

Ключевое слово unless указывает на то, что цикл выполняется, пока не будет достигнуто *условие*.

При нормальном завершении цикла на экран возвращается *done*. Принудительный выход из цикла осуществляется при помощи оператора return, который может возвращать произвольное значение.

Пример: Задать первые 5 членов арифметической прогрессии, где a_1=*-4* с шагом 5.

```
(%i1) a:-4$
      display(a);
      for i:2 thru 5 do [a:a+5,display(a)];
a=-4
(%o2) done
a=1
a=6
a=11
a=16
(%o3) done
```

Пример: Вычислить сумму всех чисел от 1 до 10

```
(%i4) s:0;
(%o4) 0
(%i5) for i:1 while i<=10 do s:s+i;
      s;
(%o5) done
(%o6) 55
```

Чтобы полноценно использовать в *Maxima* циклы необходимо ввести следующие понятия: список и массивы.

Списки

Списки – базовые строительные блоки для *Maxima*. Чтобы создать список необходимо в квадратных скобках записать все его элементы через запятую. Список может быть пустым или состоять из одного элемента.

Пример:

```
(%i1) a1:[1,6,-3,s,2*t];
(%o1) [1,6,-3,s,2 t]
(%i2) a2:[4];
(%o2) [4]
(%i3) b:[];
(%o3) []
```

Элементом списка может и другой список

```
(%i4) d:[1,2,[6,-7],[s,t,q]];
(%o4) [1,2,[6,-7],[s,t,q]]
```


```
(%i4) d:[1,2,[6,-7],[s,t,q]];
(%o4) [1,2,[6,-7],[s,t,q]]
```


```
(%i5) r:[a1,a2,3,8];
(%o5) [[1,6,-3,s,2 t],[4],3,8]
```

Чтобы вывести на экран один из элементов списка нужно записать имя списка, а затем в квадратных скобках указать номер интересующего элемента:

```
(%i6) d[1];
(%o6) 1
(%i7) r[2];
(%o7) [4]
(%i8) r[1];
(%o8) [1,6,-3,s,2 t]
```

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.

<u>Пример:</u>

```
(%i1) array(a,2,2);
(%o1) a

(%i2) a[0,0]:0;a[0,1]:2;a[1,0]:3;
      a[1,1]:1;a[1,2]:-1;a[2,1]:-2;
      a[2,2]:3;a[0,2]:4;a[2,0]:3;
(%o2) 0
 (%o3) 2
 (%o4) 3
 (%o5) 1
 (%o6) -1
 (%o7) -2
 (%o8) 3
 (%o9) 4
 (%o10) 3

(%i11) a[1,2];
(%o11) -1
```

В некоторых задачах бывает необходимо массив преобразовать в список, для этого используется функция listarray :

```
(%i12) listarray(a);
(%o12) [0,2,4,3,1,-1,3,-2,3]
```

Пример:

Найти максимальный элемент в каждом столбце матрицы $a(3,3)$

Решение:

Для решения данной задачи в программе необходимо описать два массива:

- двумерный массив $a(3,3)$ – это заданный массив,
- одномерный массив $b(3)$ – в него будем записывать максимальные элементы каждого столбца.

```
(%i1) array(a,3,3);
(%o1) a
(%i1) array(b,3);
(%o1) b
```

Затем нужно ввести элементы массива $a(3,3)$

```
(%i2) a[i,j]:=i-j*3;
(%o2) a   :=i-j 3
       i,j
```

Для нахождения максимального элемента по столбцам необходим двойной цикл (вложенный). Внешний цикл с переменной k будет перебирать столбцы, а во внутреннем цикле с переменной i будут сравнивать элементы столбца между собой для поиска наибольшего.

```
(%i4) for k:0 thru 3 do (b[k]:a[1,k], for i:1 thru 3 do
      if b[k]<a[i,k] then b[k]:a[i,k]);
(%o4) done
```

```
(%i5) display(b[0],b[1],b[2],b[3]);
```

$b_0=3$

$b_1=0$

$b_2=-3$

$b_3=-6$

```
(%o5) done
```

Пример: Протабулировать функцию:

$$f(x) := x\sin(2x);\ x\in[0;\pi\ ]$$

с шагом h=0.1π и по точкам построить график функции.

Решение: Сначала нужно описать массивы *x* и *y* при помощи функции array.

Если отрезок $x\in[0;\pi\ ]$ пробегать с шагом h=0.1π, то получиться 11 точек. Определяем диапазон изменения индекса *i* узлов сетки:

$$i=0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ 8{,}9,\ 10.$$

```
(%i1) array(x,10);array(y,10);
(%o1) x
(%o2) y
```

Значения этих массивов вычисляются в цикле for:

```
(%i3) for i:0 thru 10 do (x[i]:i*%pi*0.1,
      y[i]:x[i]*sin(2*x[i]));
(%o3) done
```

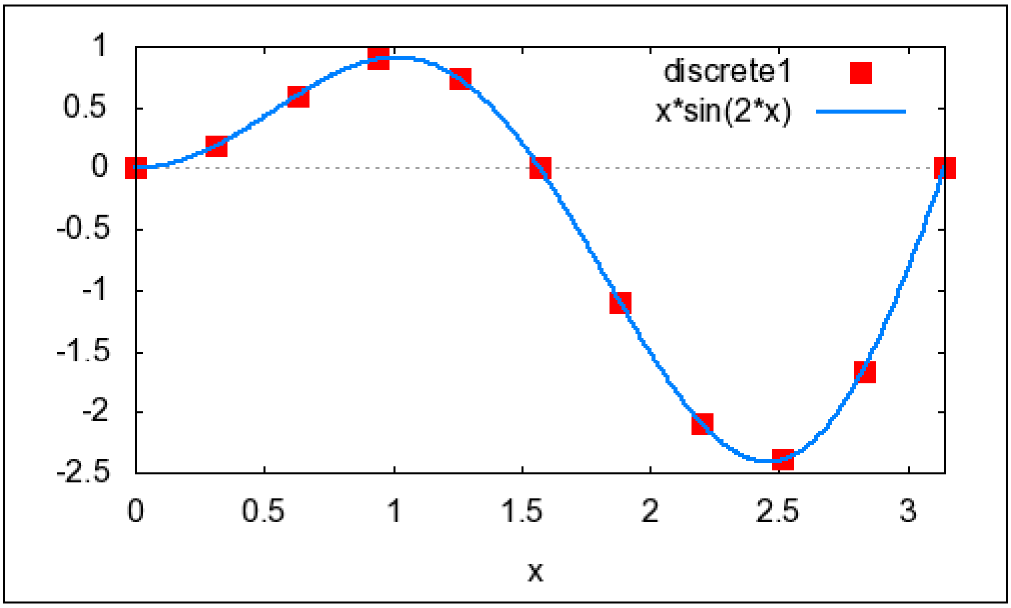
Для просмотра значений массивов используется функция listarray(x).

```
(%i4) x1:listarray(x);
(%o4) [0,0.1π,0.2π,0.3π,0.4π,0.5π,0.6π,0.7π,0.8π,0.9π,1.0π]
(%i5) y1:listarray(y);
(%o5) [0,0.1π sin(0.2π),0.2π sin(0.4π),0.3π sin(0.6π),0.4π sin(0.8π),0.5π
sin(1.0π),0.6π sin(1.2π),0.7π sin(1.4π),0.8π sin(1.6π),0.9π sin(1.8π),1.0π
sin(2.0π)]
```

На одном рисунке построим два графика: один точечный – результат табуляции, а второй непрерывный – график явно заданной функции.

```
(%i6) wxplot2d([[discrete,x1,y1],x*sin(2*x)],[x,0,%pi],
      [style,[points,3,2,6],[lines,2,1]]);
```

(%t6)



(%o6)

§7. ВЫЧИСЛЕНИЕ ПРЕДЕЛОВ ФУНКЦИИ

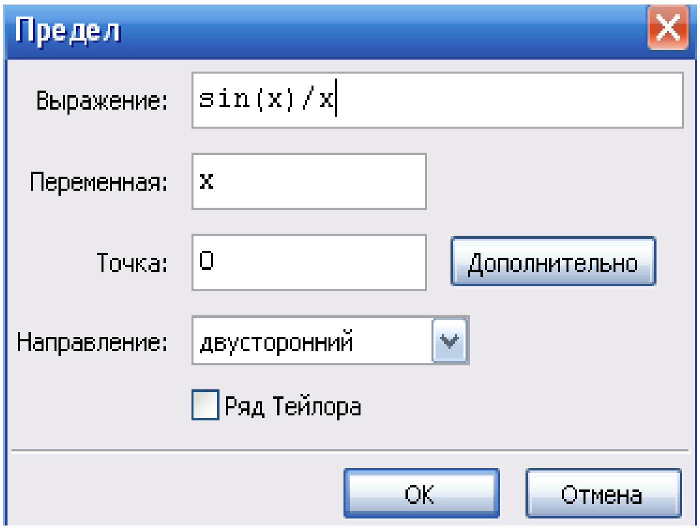
Предел функции $f(x)$ при $x \to a$ вычисляется в *Maxima* с помощью оператора limit($f(x)$, x, a);

<u>Пример:</u> Вычислить $\lim\limits_{x\to 0}\dfrac{\sin x}{x}$.

```
(%i1) limit(sin(x)/x, x, 0);
(%o1) 1
```

Командой (%i1) находится предел от функции $\dfrac{\sin x}{x}$ при $x \to 0$ и в строке (%o1) записан ответ 1.

Возможен другой вариант ввода команды интегрирования. Используя меню, щелкнуть по кнопкам «Анализ→Find Limit». Появится окно, которое нужно заполнить и по команде «OK» получить результат.



Примеры: Вычислить пределы функций

$$\lim_{x\to\infty} e^x,\ \lim_{x\to-\infty} e^x,\ \lim_{x\to-\infty} e^x,\ \lim_{x\to-1} \frac{x^3-3x-2}{x^2-x-2}.$$

```
(%i1) 'limit(exp(x), x, inf);
```

(%o1) $\lim\limits_{x\to\infty} \%e^x$

```
(%i2) limit(exp(x), x, inf);
```

(%o2) ∞

```
(%i3) 'limit(exp(x), x, minf);
```

(%o3) $\lim\limits_{x\to-\infty} \%e^x$

```
(%i4) limit(exp(x), x, minf);
```

(%o4) 0

```
(%i5) y(x):=(x^3-3*x-2)/(x^2-x-2)^2; 'limit(y(x), x,-1);
```

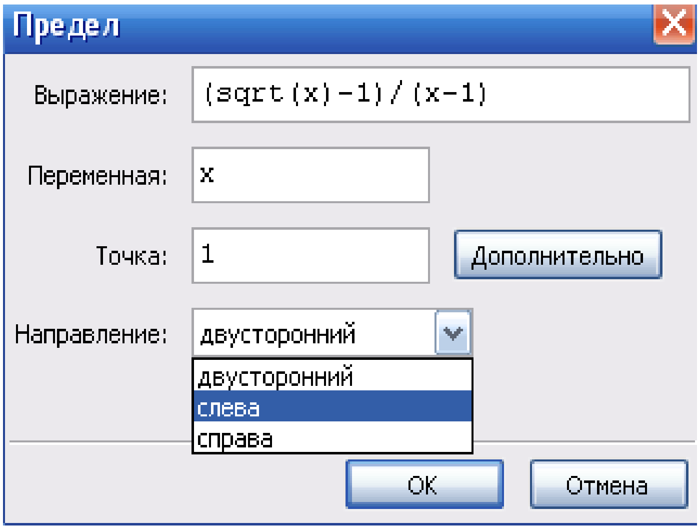
(%o5) $y(x):=\frac{x^3-3x-2}{(x^2-x-2)^2}$

(%o6) $\lim\limits_{x\to -1}\frac{x^3-3x-2}{(x^2-x-2)^2}$

```
(%i7) limit(y(x), x,-1);
```

(%o7) $-\frac{1}{3}$

С помощью *Maxima* можно вычислять односторонние пределы, для этого опять зайдем в меню в «Анализ→Find Limit». В появившемся окне в строке направление нужно выбрать




```
(%i2) 'limit((sqrt(x)-1)/(x-1), x, 1, minus);
```

(%o2) $\lim\limits_{x\to 1-}\frac{\sqrt{x}-1}{x-1}$

```
(%i3) limit((sqrt(x)-1)/(x-1), x, 1, minus);
```

(%o3) $\frac{1}{2}$

§8. ВЫЧИСЛЕНИЕ ПРОИЗВОДНЫХ ФУНКЦИИ

Производная функции *f(x)* вычисляется с помощью оператора diff(*f*(*x*), *x*, *n*);. Здесь *n* – это порядок производной.

<u>Пример:</u> Вычислить производную функции $(\sin x)^{\arcsin x}$.

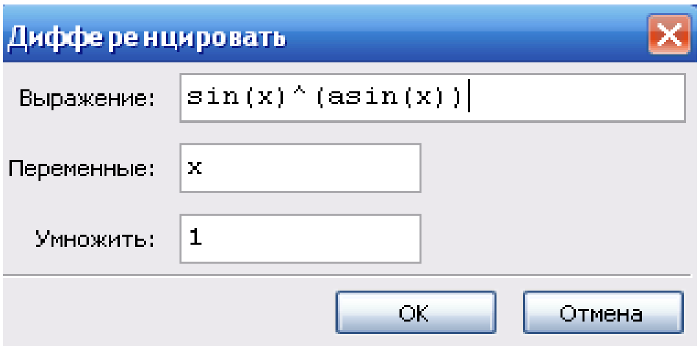
```
(%i1) 'diff(sin(x)^(asin(x)),x,1);
```

$$(\%o1)\quad \frac{d}{d\,x}\,\sin(x)^{\mathrm{asin}(x)}$$

```
(%i2) diff(sin(x)^(asin(x)),x,1);
```

$$(\%o2)\quad \sin(x)^{\mathrm{asin}(x)}\left(\frac{\log(\sin(x))}{\sqrt{1-x^2}}+\frac{\mathrm{asin}(x)\cos(x)}{\sin(x)}\right)$$

Возможен другой вариант ввода команды интегрирования. Используя меню, щелкнуть по кнопкам «*Анализ→Differentiate*». Появится окно, которое нужно заполнить и по команде «ОК» получить результат.



С помощью функции diff можно вычислять частные производные функции многих переменных:

$$\mathrm{diff}(f(x,y),\, x,\, n,\, y,\, m);$$

Здесь n – это порядок производной по переменной x, m– это порядок производной по переменной y.

<u>Примеры:</u>

Вычислить следующие производные

$$\frac{d^5}{dx^2\,dy^3}(x^7y^8)\ \text{и}\ \frac{d^2}{dx^2}(\sqrt{3\,xy^2+x^2y})$$

```
(%i3) 'diff(x^7*y^8,x,2,y,3);
```

(%o3) $\frac{d^5}{d x^2 \, d y^3}(x^7 y^8)$

```
(%i4) diff(x^7*y^8,x,2,y,3);
```

(%o4) $14112\, x^5 y^5$

```
(%i5) 'diff(sqrt(x^2*y+3*x*y^2),x,2);
```

(%o5) $\frac{d^2}{d x^2}\sqrt{3\, x\, y^2 + x^2\, y}$

```
(%i6) diff(sqrt(x^2*y+3*x*y^2),x,2);
```

(%o6) $\frac{y}{\sqrt{3\, x\, y^2 + x^2\, y}} - \frac{(3\, y^2 + 2\, x\, y)^2}{4\,(3\, x\, y^2 + x^2\, y)^{3/2}}$

Приложение производных

I. *Нахождение точек экстремума функции.*

При решении подобных задач следует придерживаться предлагаемой схемы:

1. Изобразить рассматриваемый объект.
2. Вычислить первую производную функции, приравнять ее к нулю, решить полученное уравнение, т.е. найти критические точки функции.
3. Вычислить вторую производную функции в каждой из критических точек. По знаку второй производной определить, какая из точек является максимум, а какая минимумом функции.
4. Записать координаты этих точек.

<u>Примеры:</u> 1.Найти экстремумы функции $x(x-1)^3$.

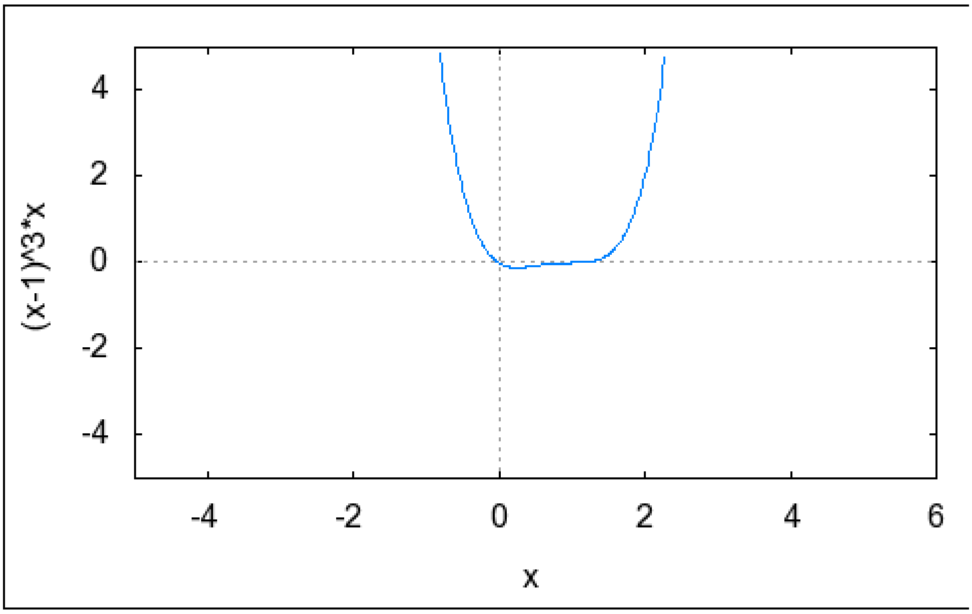
Решение: Зададим функцию и построим график

```
(%i1) f(x):=x*(x-1)^3;
```

(%o1) $f(x):=x(x-1)^3$

```
(%i2) wxplot2d([f(x)], [x,-5,6],[y,-5,5])$
plot2d: some values were clipped.
```

(%t2)



Найдем первую производную данной функции, приравняем к нулю и решим уравнение:

```
(%i3) solve(diff(f(x),x,1)=0);
```

(%o3) $[x=\frac{1}{4},x=1]$

В результате решения уравнения получились две точки, которые являются критическими.

Вычислим вторую производную

```
(%i4) g(x):=diff(f(x),x,2);
(%o4) g(x):=diff(f(x),x,2)
```

```
(%i5) g(x);
```

(%o5) $6(x-1)x+6(x-1)^2$

и посчитаем ее значение в критических точках

```
(%i6) at(%o5,x=1/4);
```

$$(\%o6)\ \frac{9}{4}$$

```
(%i7) at(%o5,x=1);
(%o7) 0
```

В точке $x=\frac{1}{4}$ вторая производная больше нуля, значит это точка минимума, в точке $x=1$ вторая производная равна нулю, значит, это точка перегиба.

Сравним с графиком, построенным ранее. Действительно на графике есть точка минимума и точка перегиба.

Найдем значение функции в этих точках:

```
(%i8) f(1/4);
```

$$(\%o8)\ -\frac{27}{256}$$

```
(%i9) f(1);
(%o9) 0
```

Ответ: Точка $\left(\frac{1}{4},-\frac{27}{256}\right)$является точкой минимума.

2. Найти экстремумы функции двух переменных $z=e^{-x^2-y^2}(3x^2+y^2)$

Решение: Зададим функцию

```
(%i1) z:%e^(-x^2-y^2)*(3*x^2+y^2);
```

$$(\%o1)\ (y^2+3\ x^2)\%e^{-y^2-x^2}$$

Найдем стационарные точки, в которых выполняется необходимое условие экстремума функции:

```
(%i2) solve([diff(z,x)=0,diff(z,y)=0],[x,y]);
(%o2) [[x=0,y=0],[x=-1,y=0],[x=1,y=0],[x=0,y=-1],[x=0,y=1]]
```

В результате получено пять точек. Для каждой из них проверим выполнение достаточного условия экстремума. Проделаем это только для точки (1,0).

```
(%i3) A:diff(z,x,2);
```
(%o3) $4x^2(y^2+3x^2)\%e^{-y^2-x^2}-2(y^2+3x^2)\%e^{-y^2-x^2}-24x^2\%e^{-y^2-x^2}+6\%e^{-y^2-x^2}$

```
(%i4) d:determinant(matrix([diff(z,x,2),diff(z,x,1,y,1)],
      [diff(z,x,1,y,1),diff(z,y,2)]));
```
(%o4) $\left(4x^2(y^2+3x^2)\%e^{-y^2-x^2}-2(y^2+3x^2)\%e^{-y^2-x^2}-24x^2\%e^{-y^2-x^2}+6\%e^{-y^2-x^2}\right)$
$\left(4y^2(y^2+3x^2)\%e^{-y^2-x^2}-2(y^2+3x^2)\%e^{-y^2-x^2}-8y^2\%e^{-y^2-x^2}+2\%e^{-y^2-x^2}\right)-$
$\left(4xy(y^2+3x^2)\%e^{-y^2-x^2}-16xy\%e^{-y^2-x^2}\right)^2$

```
(%i5) at(d,[x=1,y=0]);
```
(%o5) $48\%e^{-2}$

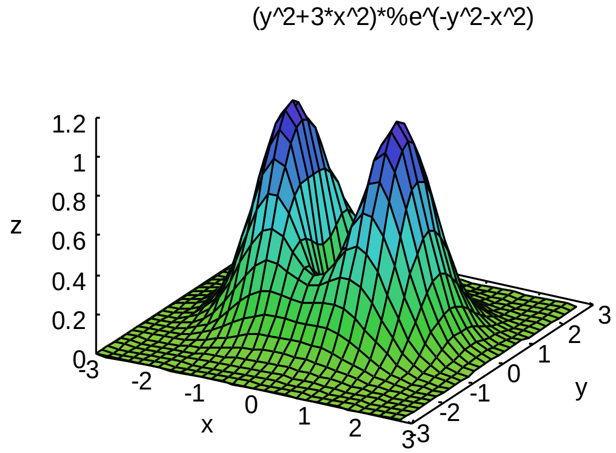
```
(%i6) at(A,[x=1,y=0]);
```
(%o6) $-12\%e^{-1}$

```
(%i7) at(z,[x=1,y=0]);
```
(%o7) $3\%e^{-1}$

Так как значение определителя в этой точке $48/e^2$ положительно, а значение $\left.\frac{\partial^2 z}{\partial x^2}\right|_{(1,0)} < 0$, то точка (1,0) является точкой максимума и $z_{\max} = \frac{3}{e}$.

Сравним с графиком, построенным ранее.

```
(%i9) plot3d(%e^(-x^2-y^2)*(3*x^2+y^2),[x,-3,3],[y,-3,3]);
(%o9)
```



3. Ответ: Точка $\left(\frac{1}{4}, -\frac{27}{256}\right)$ является точкой минимума .

§9. ЗАДАЧИ ДЛЯ САМОСТОЯТЕЛЬНОГО РЕШЕНИЯ

1.Вычислить произведение матриц AB

а) $A=\begin{pmatrix} 2 & 2 & 1 & 5 \\ 1 & 3 & 0 & 1 \\ 2 & 1 & 3 & 11 \end{pmatrix}$, $B=\begin{pmatrix} -1 & 3 \\ 7 & -1 \\ 6 & 5 \\ 4 & 0 \end{pmatrix}$, б) $A=\begin{pmatrix} 4 & 2 & -1 & 6 \\ 2 & 3 & 5 & 11 \\ 3 & 5 & 4 & 8 \end{pmatrix}$, $B=\begin{pmatrix} -1 & 2 \\ 0 & -1 \\ 9 & 4 \\ 2 & 0 \end{pmatrix}$,

в) $A=\begin{pmatrix} 3 & 6 & 2 & 13 \\ 2 & 6 & 5 & 10 \\ 3 & 6 & 4 & 9 \end{pmatrix}$, $B=\begin{pmatrix} 3 & -2 \\ -1 & -1 \\ 2 & 0 \\ 3 & 3 \end{pmatrix}$, г) $A=\begin{pmatrix} 5 & 2 & -1 \\ 2 & 3 & 2 \\ 3 & 5 & 4 \end{pmatrix}$, $B=\begin{pmatrix} -2 & 2 & 0 \\ 4 & -1 & 4 \\ 3 & 1 & 0 \end{pmatrix}$,

2. Вычислить обратную матрицу

а) $A=\begin{pmatrix} 3 & 2 & -1 \\ 1 & -3 & 2 \\ 7 & 2 & 0 \end{pmatrix}$, б) $A=\begin{pmatrix} 1 & 1 & -1 \\ 8 & 3 & -6 \\ -4 & -1 & 3 \end{pmatrix}$,

в) $A=\begin{pmatrix} 3 & -1 & 2 \\ 2 & -1 & -1 \\ 4 & -2 & -2 \end{pmatrix}$, г) $A=\begin{pmatrix} 3 & 4 & 2 \\ 2 & -1 & -3 \\ 1 & 5 & 1 \end{pmatrix}$.

3. Найти решения методом Крамера или с помощью обратной матрицы

а) $\begin{cases} x+2y+10z=-15, \\ 2x+4y-z=12, \\ x+y-3z=9, \end{cases}$ б) $\begin{cases} x+2y+z=4, \\ 4x-3y+4z=5, \\ 2x+7y-z=8, \end{cases}$

в) $\begin{cases} 2x-3y+z=2, \\ 3x+2y-3z=-3, \\ 4x+y-3z=-4, \end{cases}$ г) $\begin{cases} 2x+3y+2z=9, \\ x+y+5z=-5, \\ 3x+4y+z=16. \end{cases}$

4. Методом Гаусса решить следующие системы уравнений

а) $\begin{cases} x+3y-2z=8, \\ 2x-y+4z=2, \\ 5x+y+6z=12, \end{cases}$ б) $\begin{cases} 4x+y-z=6, \\ x-3y+2z=1, \\ 6x-5y+3z=8, \end{cases}$

в) $\begin{cases} 3x+2y-z=6, \\ x+y+3z=2, \\ 2x+y-4z=4, \end{cases}$ г) $\begin{cases} 5x+8y-3z=8, \\ x+3y+z=5, \\ 4x+5y-4z=3. \end{cases}$

5. Вычислить определители

а) $\begin{vmatrix} 2 & -1 & 0 & 1 \\ 3 & -2 & 1 & 4 \\ 1 & 0 & 0 & 2 \\ -2 & 5 & 2 & 1 \end{vmatrix}$, б) $\begin{vmatrix} 1 & -3 & 2 & 0 \\ 0 & -2 & 4 & -1 \\ 1 & 0 & 0 & 2 \\ -4 & 5 & 2 & 1 \end{vmatrix}$,

в) $\begin{vmatrix} 0 & -1 & 0 & 1 \\ 4 & -2 & 1 & 4 \\ 3 & 0 & 0 & 2 \\ -2 & 5 & 2 & 1 \end{vmatrix}$, г) $\begin{vmatrix} 4 & -3 & 2 & 0 \\ 0 & -1 & 4 & -1 \\ 1 & 0 & 0 & 2 \\ -4 & 3 & 2 & 1 \end{vmatrix}$.

6. Решить системы

а) $\begin{cases} x_1+x_2+x_3+x_4=0, \\ 2x_1-2x_2+3x_3-3x_4=15, \\ 9x_1+9x_2+4x_3+4x_4=5, \\ 3x_1-3x_2+2x_3-2x_4=15. \end{cases}$ б) $\begin{cases} x_1+x_2+x_3+x_4=1, \\ x_1-x_2+4x_3-4x_4=-4, \\ 16x_1+16x_2+x_3+x_4=-6, \\ 16x_1-16x_2+4x_3-4x_4=7. \end{cases}$

в) $\begin{cases} 2x_1-x_2+x_3+2x_4+3x_5=2, \\ 6x_1-3x_2+2x_3+4x_4+5x_5=3, \\ 6x_1-3x_2+4x_3+8x_4+13x_5=9, \\ 4x_1-2x_2+x_3+x_4+2x_5=4. \end{cases}$ г) $\begin{cases} x_1-2x_2+x_3-x_4+3x_5=2, \\ 2x_1-4x_2+3x_3-2x_4+6x_5=5, \\ 3x_1-6+4x_3-3x_4+9x_5=7. \end{cases}$

7. Построить графики функций

а) $y=\sqrt[3]{(2-x)(x^2+4x+1)}$, б) $y=\dfrac{x^3+8}{x^2}$,

в) $y=\sqrt[3]{(3+x)(x^2+4x+1)}$, г) $y=\dfrac{x^2-x+1}{x-2}$,

д) $y=\dfrac{x^2-x-6}{|x-3|}$, е) $y=|x^2+2x-6|$,

ж) $y=|\cos x|\cdot \operatorname{tg} x$, з) $y=|\operatorname{ctg} x|\cdot \sin x$.

8. Построить графики функций, заданных в параметрическом виде

а) циклоиду

$$\begin{cases} x=2(t-\sin t), \\ y=2(1-\cos t), \end{cases}$$

б) окружность

$$\begin{cases} x=3\cos t, \\ y=3\sin t, \end{cases}$$

в) параболу

$$\begin{cases} x=\sqrt{t}, \\ y=\dfrac{1}{2}t, \end{cases}$$

г) прямую

$$\begin{cases} x=2-t, \\ y=1+2t. \end{cases}$$

9. Построить графики функций в полярной системе координат

а) окружность

$r=3,$

б) улитку Паскаля

$r=a\cos\varphi+l$,

в) логарифмическую спираль

$r=\rho_0 q^{\frac{\varphi}{2\pi}}$,

где q - коэффициент роста,

г) архимедову спираль

$r=k\varphi$, при k=1, 3, 5;

д) полярную розу

$r=2\sin 3\varphi,$

е) кардиоиду

$r=2a\,(\cos\varphi+1)$, при a=1, 3.

7. По точкам с шагом $\varphi = \frac{\pi}{8}$ построить кривую, заданную в полярной системе координат в промежутке $0 \le \varphi < 2\pi$.

а) $r = \frac{6}{2+\cos\varphi}$, б) $r = 2\sin^2 2\varphi$, в) $r = \frac{1}{2+\cos\varphi}$, г) $r = 4\cos^2 2\varphi$.

8. Построить трехмерный график функции

а) конус

$$\begin{cases} x = 3s\cos t, \\ y = 3s\sin t, \\ z = s. \end{cases}$$

б) цилиндр

$$\begin{cases} x = 3\cos t, \\ y = 2\sin t, \\ z = s. \end{cases}$$

в) эллипсоид

$$\begin{cases} x = 3\cos t\sin s, \\ y = 3\sin t\sin s, \\ z = 2\cos s. \end{cases}$$

г) гиперболоид

$$\begin{cases} x = 3\,\mathrm{ch}\, t\cos s, \\ y = 2\,\mathrm{sh}\, t, \\ z = 3\,\mathrm{ch}\, s\sin s. \end{cases}$$

9. Построить

а) $y = \begin{cases} -x, & x \le 0 \\ x^2, & 0 < x \le 2, \\ x+1, & x > 2 \end{cases}$ б) $y = \begin{cases} x-3, & x < 0 \\ x+1, & 0 \le x \le 4, \\ \sqrt{x}+3, & x > 4 \end{cases}$

в) $y = \begin{cases} \sqrt{1-x}, & x \le 0 \\ 0, & 0 < x \le 2, \\ x-2, & x > 2 \end{cases}$ г) $y = \begin{cases} \sin x, & x < 0 \\ x, & 0 \le x \le 2. \\ 0, & x > 2 \end{cases}$

10. Найти максимальный элемент, стоящий на главной диагонали матрицы A(5,5).

11. Найти сумму элементов матрицы A(5,5).

12. Найти произведение минимальных элементов каждого столбца матрицы А(5,5).

13. Найти максимальный элемент, находящийся выше главной диагонали матрицы А(5,5).

14. Вычислить пределы

а) $\lim\limits_{x\to\infty}\dfrac{10x^2-x+1}{5x^2+6x-2}$, б) $\lim\limits_{x\to-2}\dfrac{x^2-x-6}{x^2+7x+10}$,

в) $\lim\limits_{x\to7}\dfrac{\sqrt{2+x}-3}{x-7}$, г) $\lim\limits_{x\to0}\dfrac{\sin\frac{x^2}{5}}{3x^2}$,

д) $\lim\limits_{x\to\infty}\left(1+\dfrac{5}{x}\right)^{3x}$, е) $\lim\limits_{x\to7}\left(\dfrac{2}{x-7}-\dfrac{2x+1}{x^2-49}\right)$.

15. Вычислить все производные с первого до десятого порядка

а) $y=\dfrac{x}{\sqrt{1-x^2}}$, б) $y=2\,tg^2(1-3x)$,

в) $y=\ln\left(x+\sqrt{1+x^2}\right)$, г) $y=3^{\sin^2 5x}$,

д) $y=(\sin\,x)^{\arcsin x}$.

16. Найти экстремумы функций и записать уравнения асимптот

а) $y=\dfrac{12-3x^2}{x^2+12}$, б) $y=\dfrac{4}{(x^2+2x-3)}$,

в) $y=\dfrac{2x^3+1}{x^2}$, д) $y=\dfrac{x^2}{(x-1)^2}$.

17. Найти экстремумы функции двух переменных

а) $z=x^2-2y^2+4xy-6x-1$, б) $z=x^2+6xy-x+3y$.