

# ИСПОЛЬЗОВАНИЕ GNU OCTAVE ДЛЯ ИНЖЕНЕРНЫХ И МАТЕМАТИЧЕСКИХ РАСЧЁТОВ

*Алексеев Е.Р., Чеснокова О.В.*

Донецкий национальный  
технический университет,  
[EAlekseev@gmail.com](mailto:EAlekseev@gmail.com),

icq 327022282

+38(050)0511472, +38(067)6282413

Octave – высокоуровневый интерпретируемый язык программирования, предназначенный для решения задач вычислительной математики. Octave представляет собой интерактивный командный интерфейс (интерпретатор), реализованный в ОС Windows и Linux.

# Окно интерпретатора Octave

Файл П\_правка В\_ид Т\_ерминал С\_правка

```
octave-3.2.3:2> a=[1 2 3;4 5 6;8 7 9]
```

```
a =
```

```
 1  2  3
 4  5  6
 8  7  9
```

```
octave-3.2.3:3> inverse(a)
```

```
ans =
```

```
-0.33333 -0.33333  0.33333
-1.33333  1.66667 -0.66667
 1.33333 -1.00000  0.33333
```

```
octave-3.2.3:4> a*ans
```

```
ans =
```

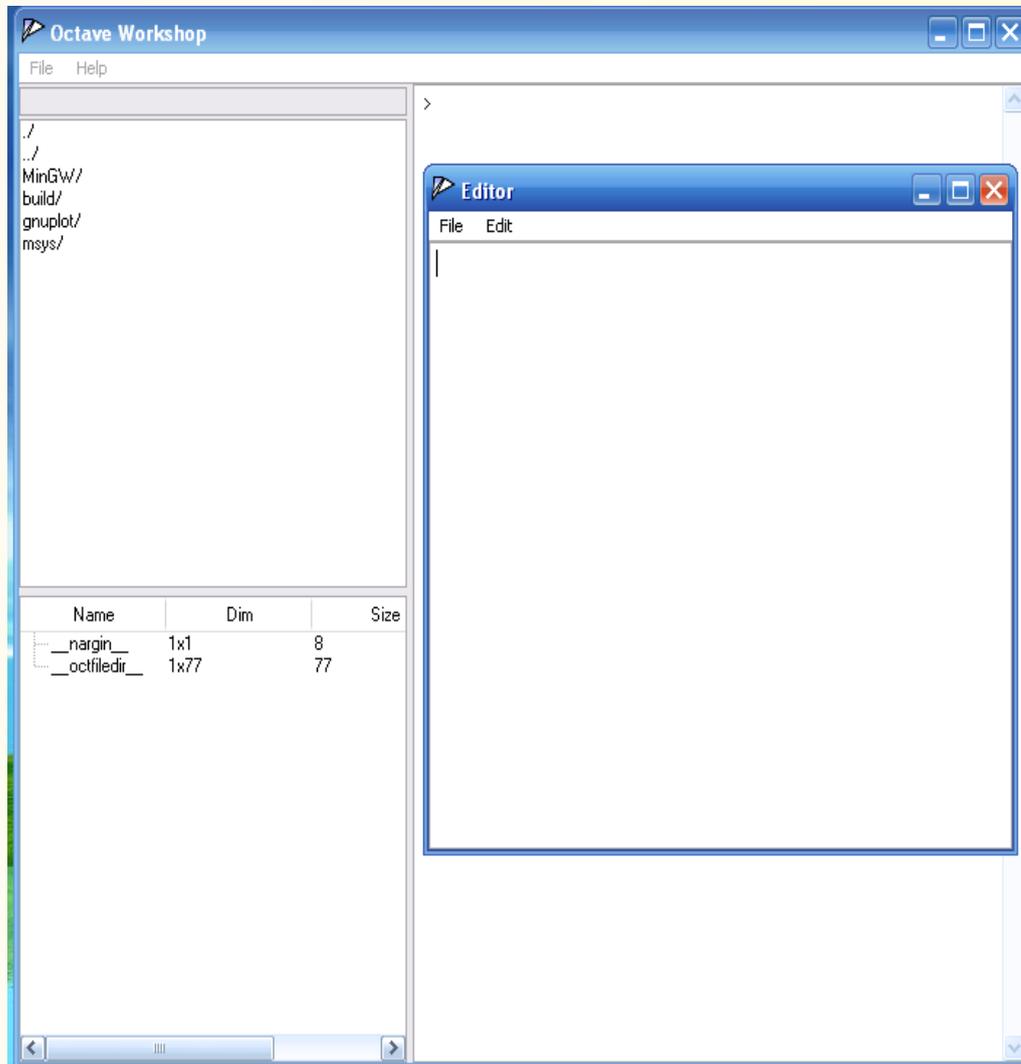
```
 1.0000e+00 -2.2204e-16  1.6653e-16
-2.2204e-16  1.0000e+00  3.3307e-16
-4.4409e-16 -4.4409e-16  1.0000e+00
```

```
octave-3.2.3:5> 
```

# Графические оболочки для работы с Octave

Octave workshop

(<http://sourceforge.net/projects/octave-workshop/files/>)



# QtOctave

<http://qtoctave.wordpress.com/what-is-qtoctave>

The screenshot displays the QtOctave application window, titled "QtOctave [Empty]". The interface includes a menu bar (File, View, Analysis, Data, Equations, Matrix, Plot, Statistics, Config, Help) and a toolbar with various icons. The main workspace is divided into several panels:

- Variables' List:** Shows a table with columns "Name", "Size", and "Bytes". A folder icon labeled "Loca..." is visible.
- Octave Terminal:** Contains text about contributing to Octave, reporting bugs to [bug@octave.org](mailto:bug@octave.org), and a "Command line" prompt with a text input field containing "our commands here...".
- Help:** Displays the "View" section of the Octave manual, including copyright information for John W. Eaton (1996-2007) and a permission notice for verbatim copies.
- Commands' List:** Lists recent system commands such as "%% воскресенье августа 8 2010 -- 12:10:".
- Navigator:** Shows the current directory as "/home/evgeniy" and a file list with filters set to "\*.m". Files listed include "1.m", "MICROSOFT.OFFICE.2010.SELECT.ED", "mailnew", "mathcad 11", "mathcad 2001", "matlab", and "portable\_windows\_soft".
- Code Editor (prim1\_2.m):** Contains the following MATLAB code:

```
1 a=2;
2 b=5;
3 c=8;
4 d=b^2-4*a*c;
5 x1=(-b+sqrt(d))/2/a
6 x2=(-b-sqrt(d))/2/a
7
```
- Small Clipboard:** Lists copied text, including "sportacentrs.com" and a URL from altlinux.org.

# Элементарные вычисления.

```
>>> a=1;b=2;c=a*b;d=c^2  
d = 4
```

```
>>> x=pi/7  
x = 0.44880
```

```
>>> (1-cos(x)^2)^0.5  
ans = 0.43388
```

# Работа с комплексными числами

```
>>> a=-5+2i;b=3-5*i;
```

```
>>> a+b
```

```
ans = -2 - 3i
```

```
>>> a^2+b^2
```

```
ans = 5 - 50i
```

```
>>> a=-3;b=4;Z=a+b*i
```

```
Z = -3 + 4i
```

```
>>> real(Z)
```

```
ans = -3
```

```
>>> imag(Z)
```

```
ans = 4
```

```
>>> angle(Z)
```

```
ans = 2.2143
```

# ФУНКЦИИ ПОЛЬЗОВАТЕЛЯ.

Решение кубического уравнения

```
function [x1,x2,x3]=cub(a,b,c,d)
r=b/a; s=c/a; t=d/a;
p=(3*s-r^2)/3; q=2*r^3/27-r*s/3+t;
D=(p/3)^3+(q/2)^2;
u=(-q/2+sqrt(D))^(1/3);
v=(-q/2-sqrt(D))^(1/3);
y1=u+v;
y2=-(u+v)/2+(u-v)/2*i*sqrt(3);
y3=-(u+v)/2-(u-v)/2*i*sqrt(3);
x1=y1-r/3; x2=y2-r/3; x3=y3-r/3;
endfunction
```

```
>>> [x1,x2,x3]=cub(3,-2,-1,-4)
```

```
x1 = 1.4905
```

```
x2 = -0.41191 + 0.85141i
```

```
x3 = -0.41191 - 0.85141i
```

# СИМВОЛЬНЫЕ ВЫЧИСЛЕНИЯ

`% Найти значение выражения`

`% при заданном значении переменной`

```
>>> x = sym ("x"); y = sym ("y");
```

```
>>> y = Sin(x)^2 - Cos(x)^2
```

```
y = -cos(x)^(2.0) + sin(x)^(2.0)
```

```
>>> subs(y, x, Pi/3)
```

```
ans = 0.5
```

`% Раскрыть скобки в выражении`

```
>>> y = (Sqrt(x)+1) * (Sqrt(x)-1) + (x-1) * (x-1) * (x-1)
```

```
y = (-1.0+x)^3 + (-1.0+sqrt(x)) * (1.0+sqrt(x))
```

```
>>> expand(y)
```

```
ans = -2.0 + (4.0) * x + x^3 - (3.0) * x^2
```

# Язык программирования

%Вычислить n-е число Фибоначчи

```
function F=fibonachi(N)
```

```
    if (N==0) | (N==1)
```

```
        F=N;
```

```
    else
```

```
        F=fibonachi(N-1)+fibonachi(N-2);
```

```
    end
```

```
end
```

```
>>>% Вызов функции
```

```
>>> fibonachi(2)
```

```
ans = 1
```

```
>>> fibonachi(6)
```

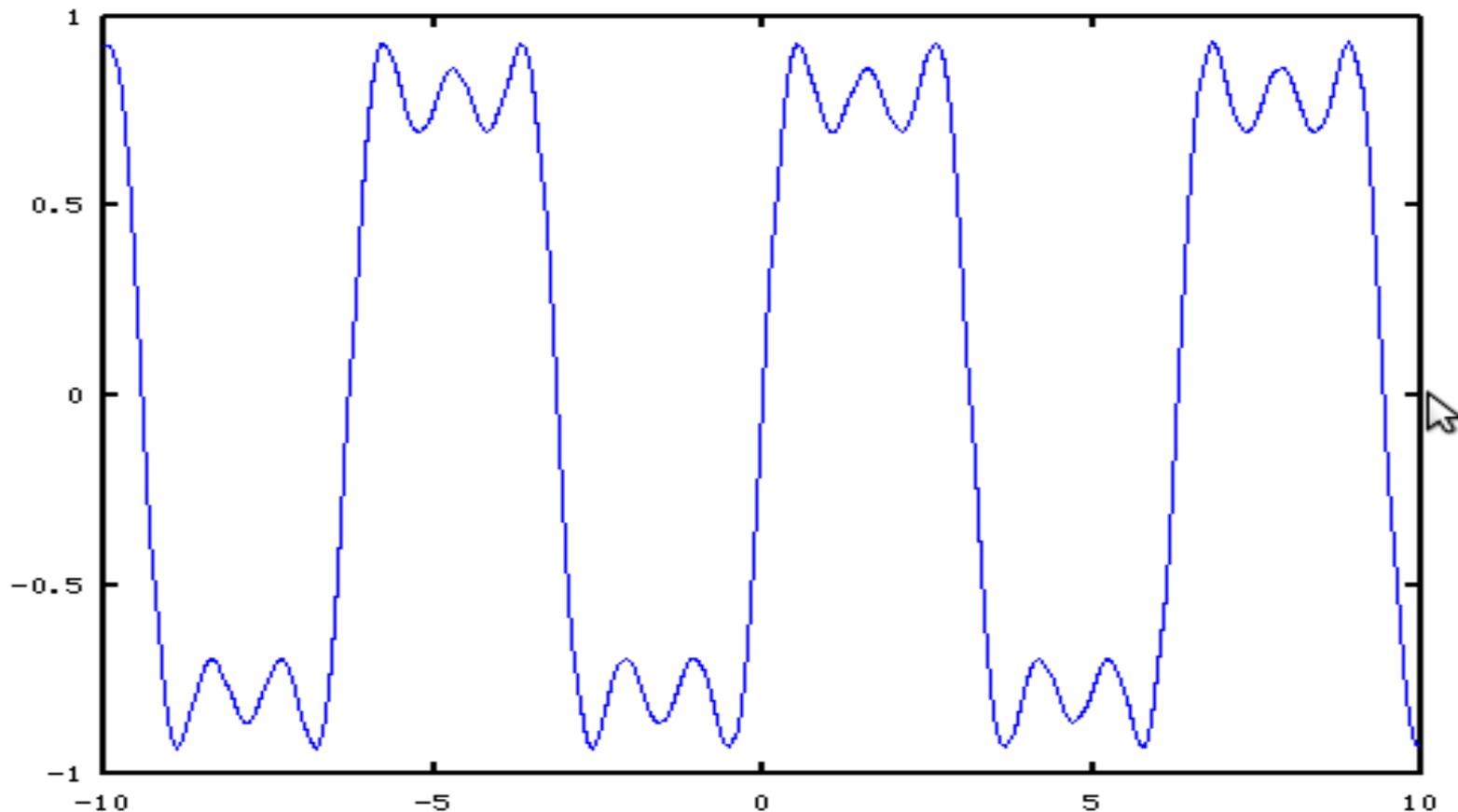
```
ans = 8
```

# Построение графиков в Octave

%График в прямоугольной системе координат.

```
x=-10:0.1:10; y=cos(x/2)+cos(5*x)/5;
```

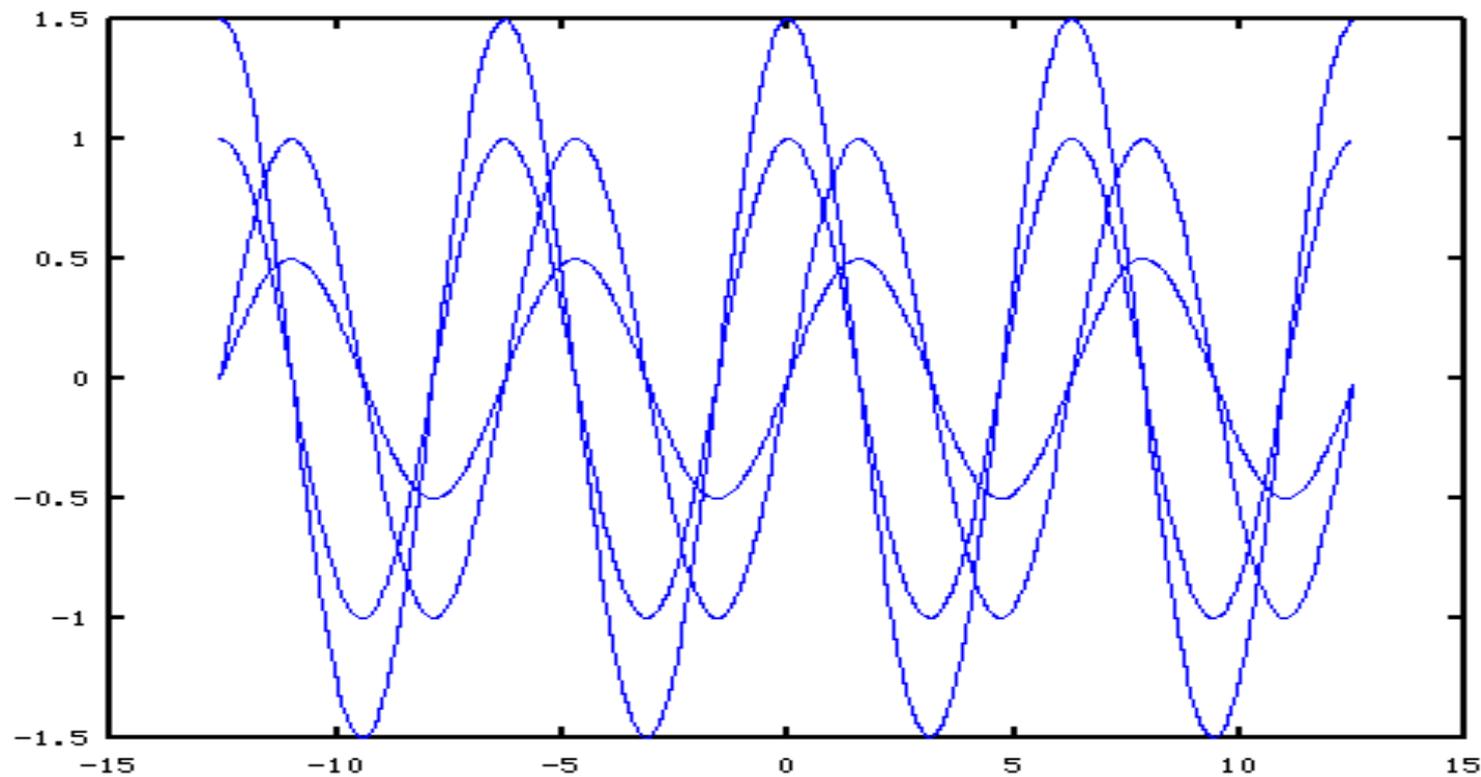
```
plot(x,y) %Построение графика функции.
```



10.0819, 0.000299670

**%Изображение графиков нескольких функций**

```
x=-4*pi:0.1:4*pi;  
v=sin(x);w=cos(x);  
r=sin(x)/2;p=1.5*cos(x);  
plot(x,v,x,w,x,r,x,p);
```



18.0246, 1.26266

**%График функции, терпящей разрыв**

**%Аргументы функции**

**x1=-5:0.01:-1.1;**

**x2=-0.9:0.01:2.9;**

**x3=3.1:0.01:7;**

**%Значение функции**

**y1=1./ (x1.\*x1-2\*x1-3) ;**

**y2=1./ (x2.\*x2-2\*x2-3) ;**

**y3=1./ (x3.\*x3-2\*x3-3) ;**

**%Построение графика функции**

**plot(x1,y1,'k',x2,y2,'k',x3,y3,'k');**

**title('y=f(x)'); % Подпись над графиком.**

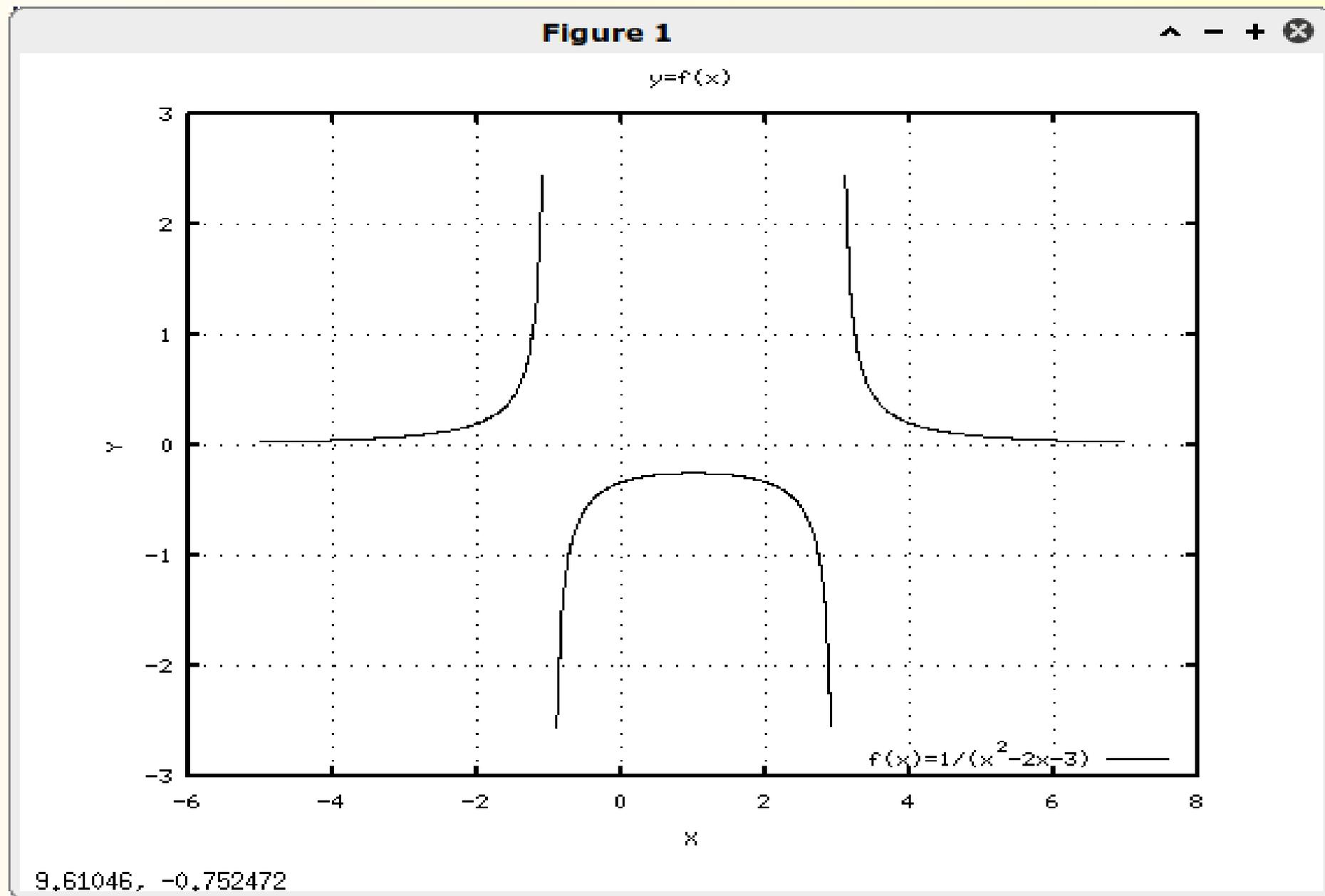
**xlabel('X'); % Подпись оси X.**

**ylabel('Y'); % Подпись оси Y.**

**legend('f(x)=1/(x^2-2x-3)',4); % Легенда.**

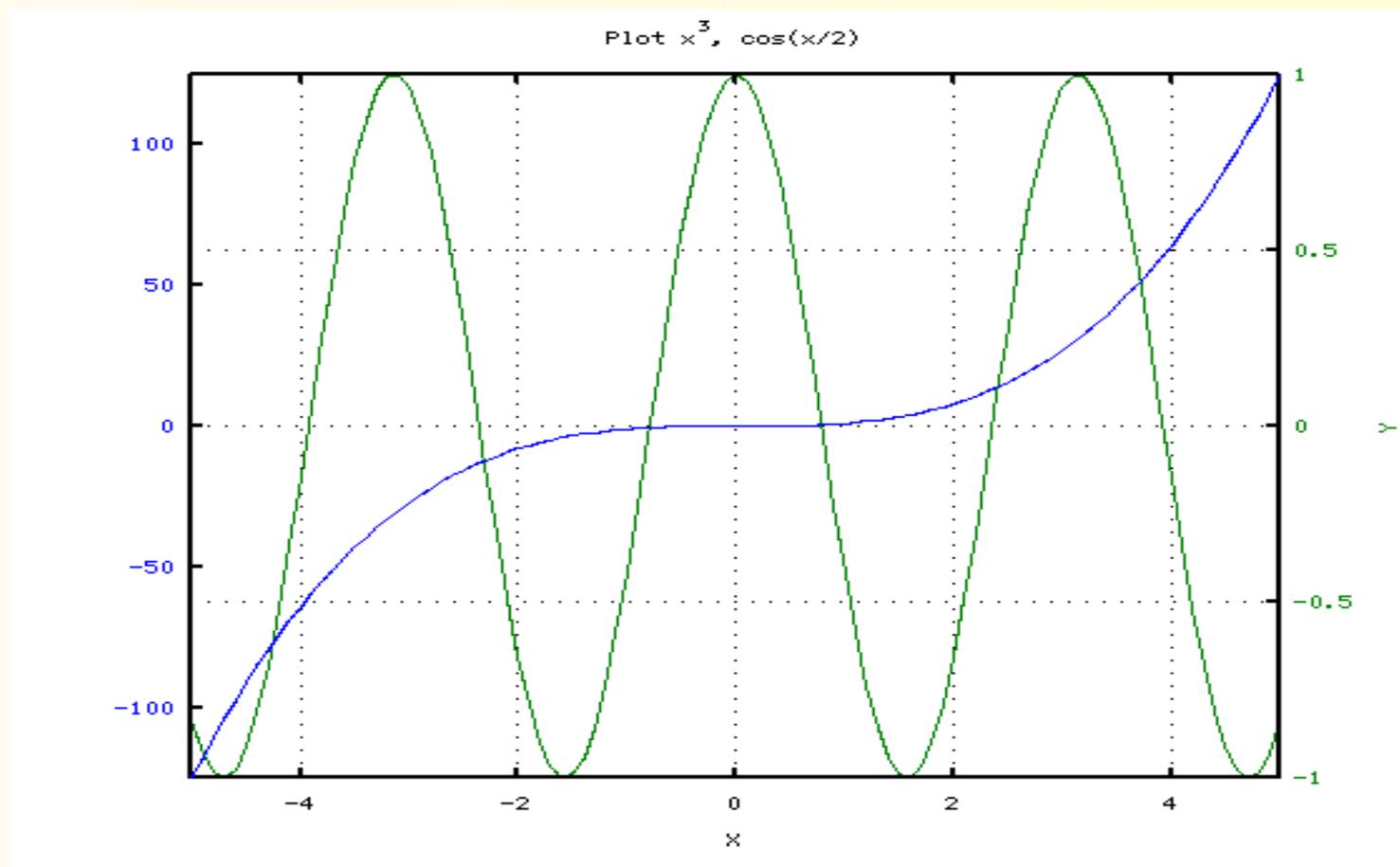
**grid on; % Линии сетки.**

# График функции

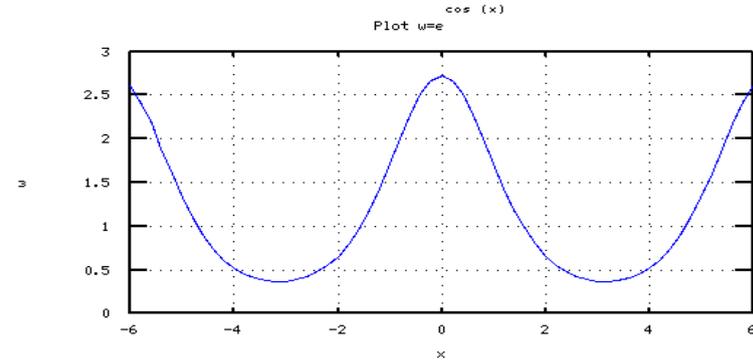
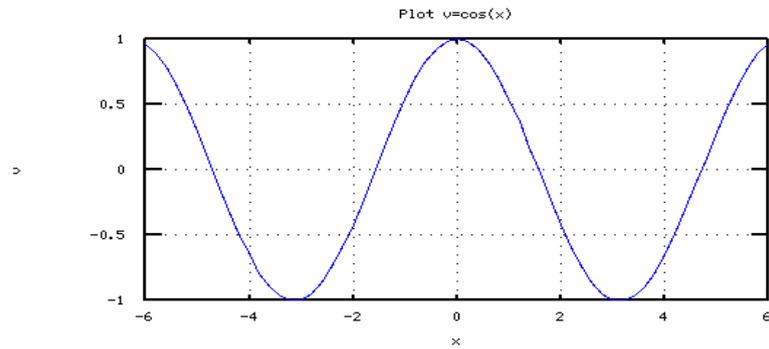
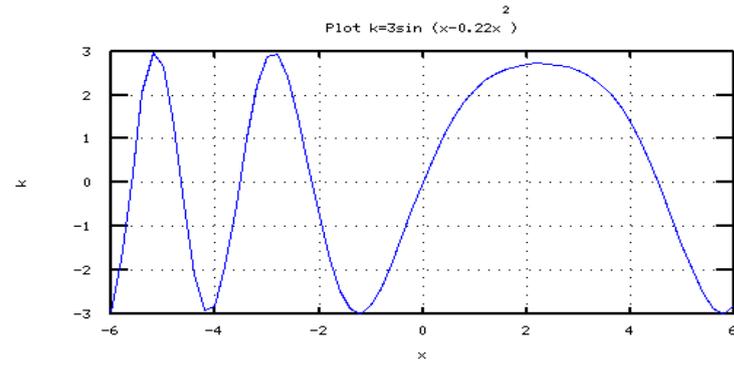
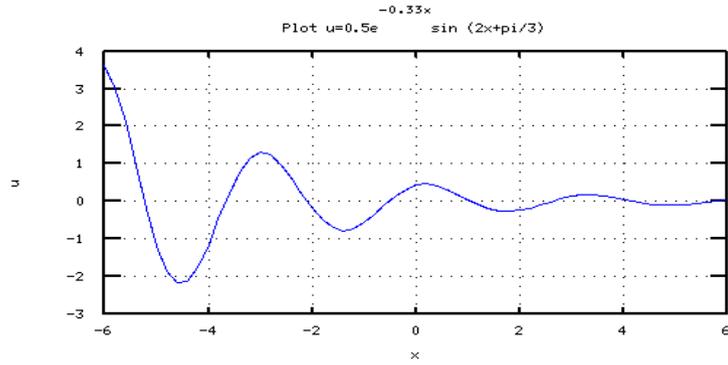
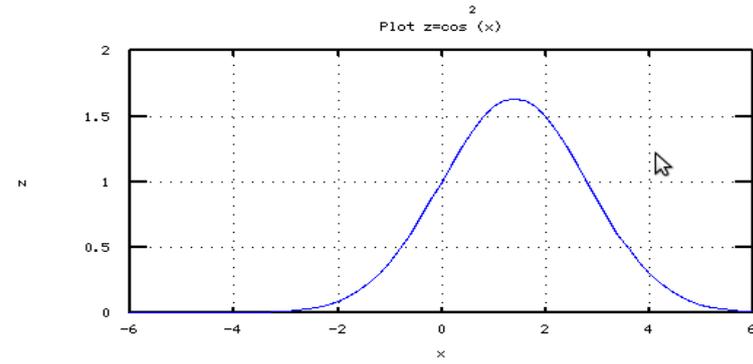
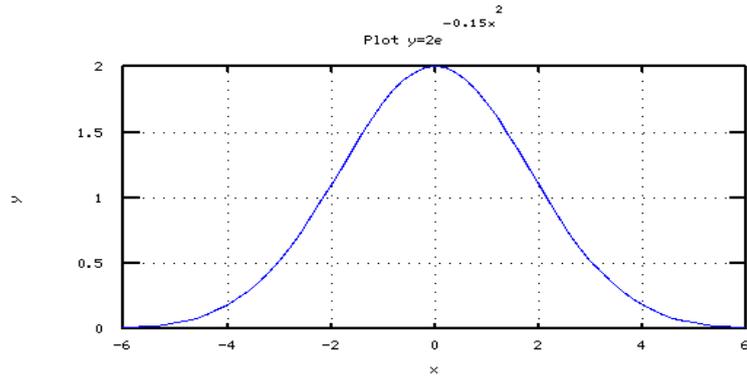


Изображение 2-х осей на графике

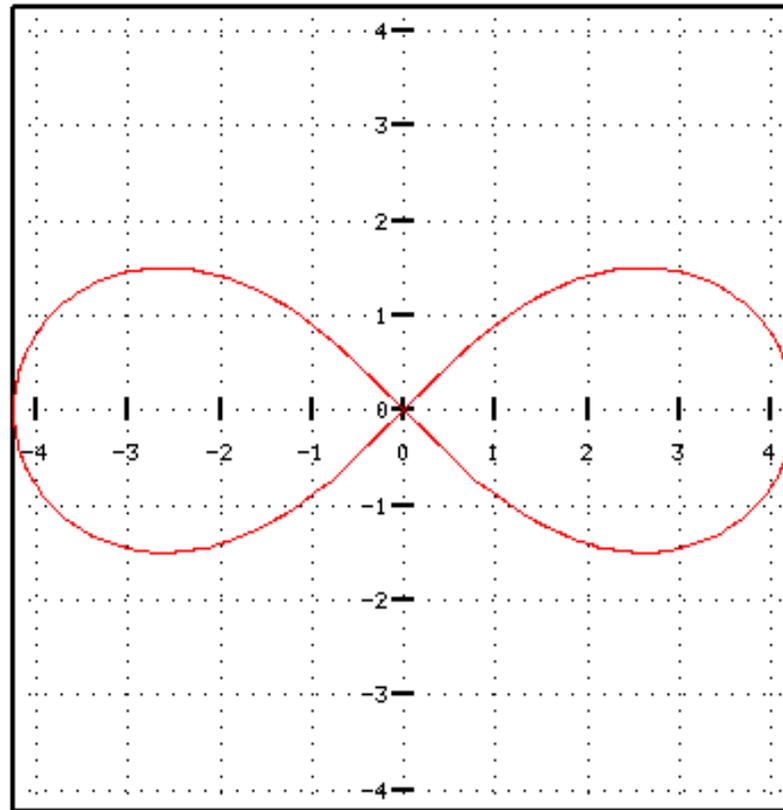
```
x=-5:0.1:5; y=x.^3; z=cos(2*x);  
plotyy(x,y,x,z); grid on;  
title('Plot x^3, cos(x/2)');  
xlabel('X'); ylabel('Y');
```



# Построение графиков с помощью функции `subplot(row, col, cur);`

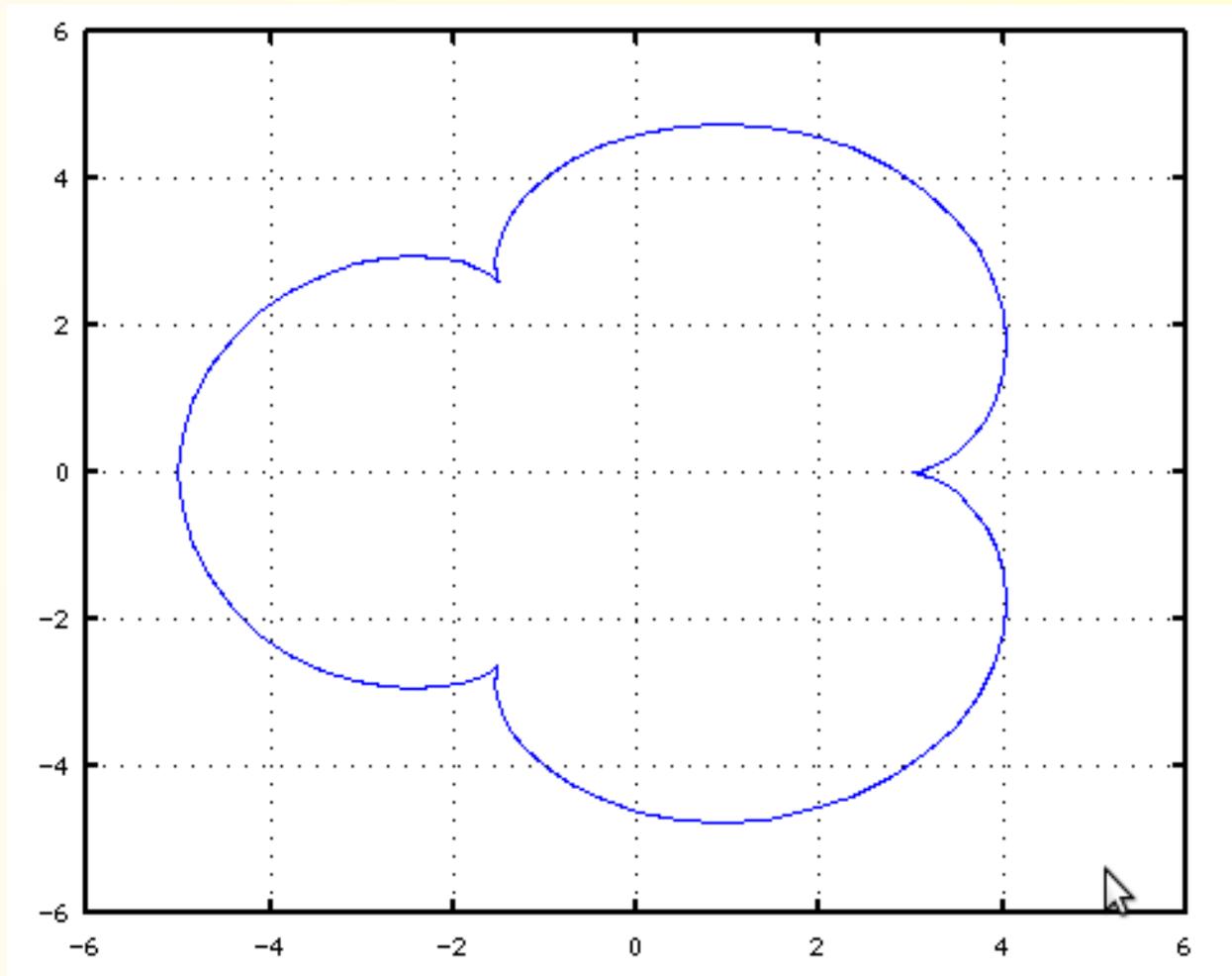


```
% График в полярной системе координат.  
fi=-pi/4:pi/200:pi/4;ro=3*sqrt(2*cos(2*fi));  
polar(fi,ro,'r');hold on;  
polar(fi,-ro,'r');grid on;
```



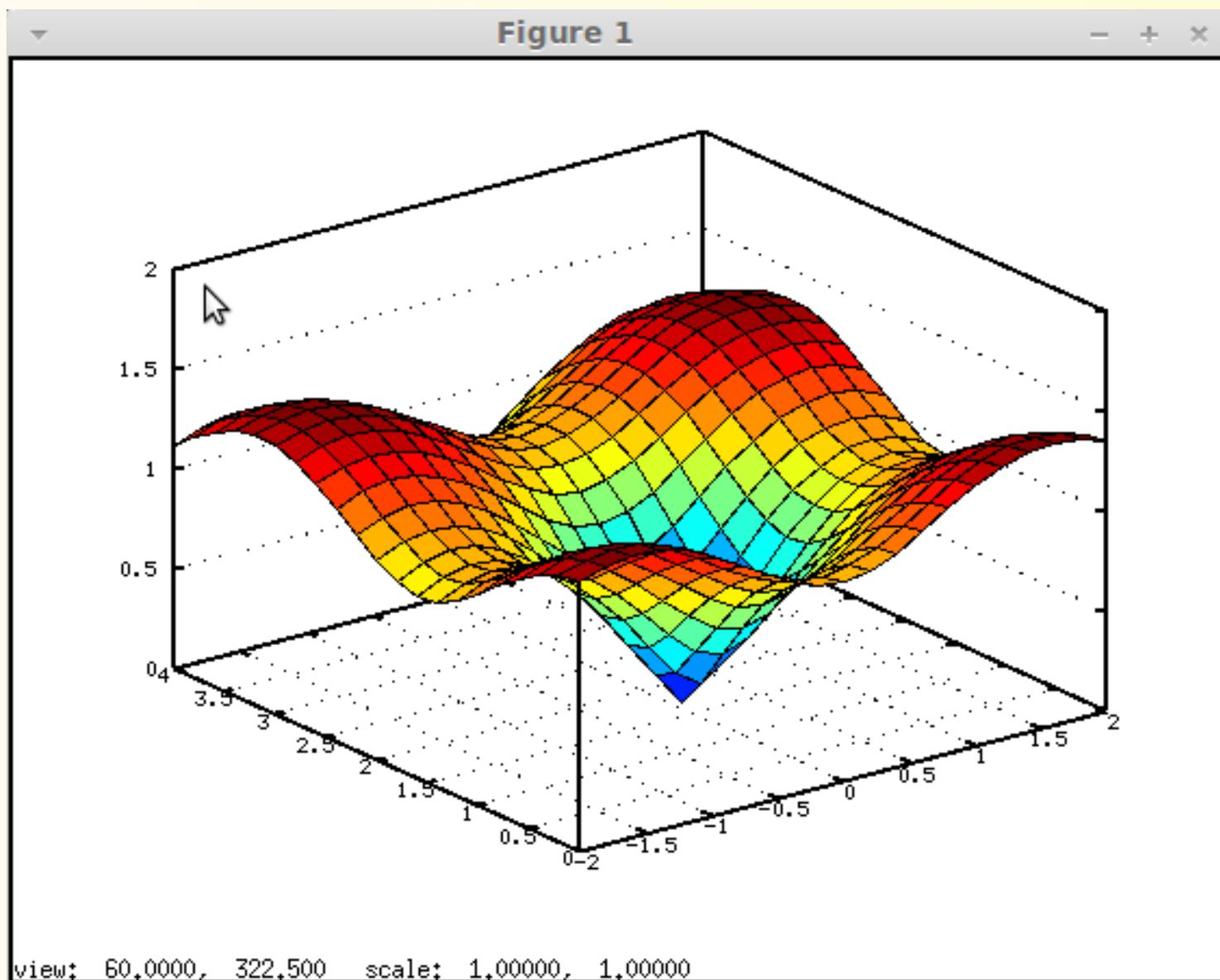
4.82912, 5.01311

```
%График функции, заданной параметрически  
t=0:pi/50:2*pi;  
x=4*cos(t)-cos(4*t);y=4*sin(t)-sin(4*t);  
plot(x,y);grid on;
```



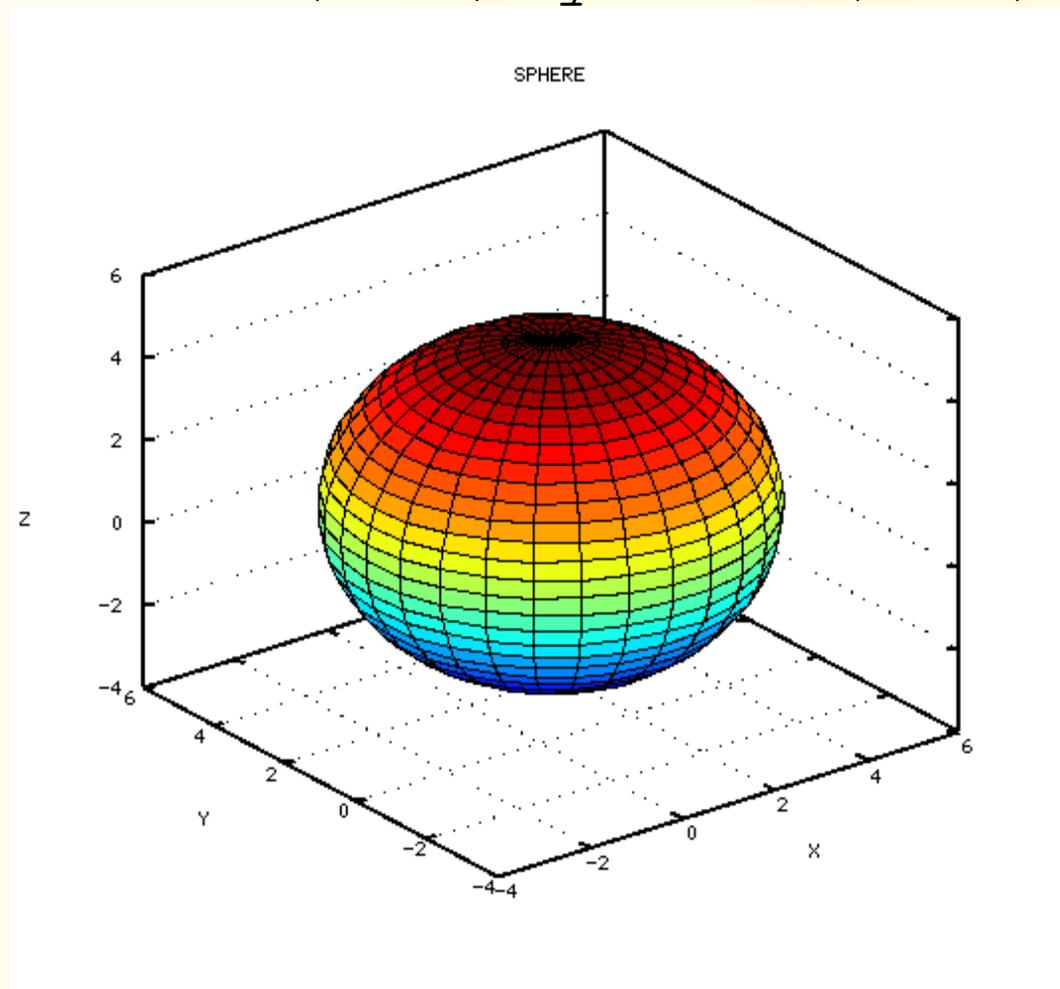
## % Изображение поверхности

```
[x y]=meshgrid(-2:0.2:2,0:0.2:4);  
z=sqrt(sin(x).^2+cos(y).^2);  
surf(x,y,z);
```



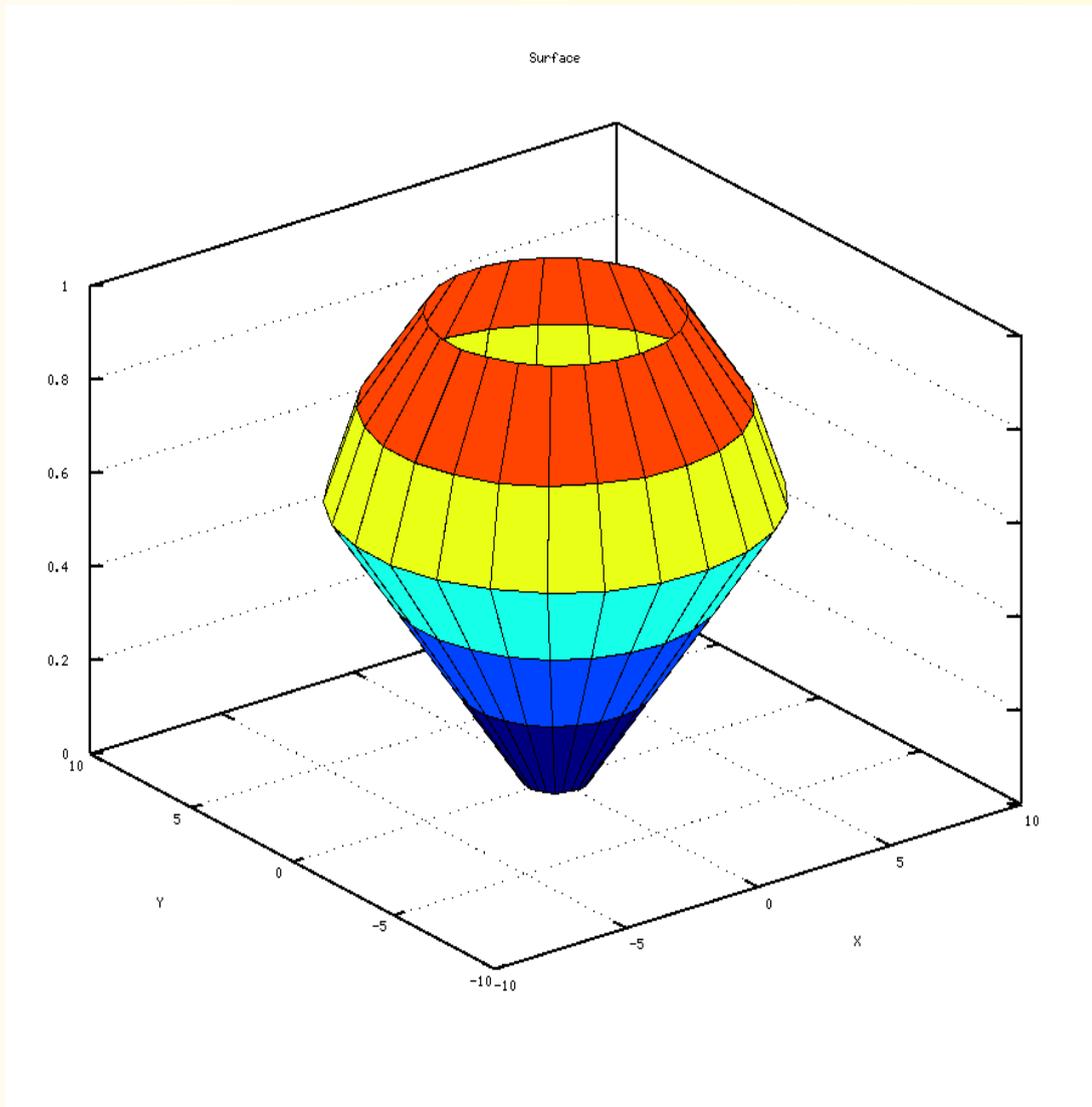
%Изображение поверхности сферы

```
h=pi/30; u=[-0:h:pi]'; v=[0:2*h:2*pi];  
x=1+4*sin(u)*cos(v); y=1+4*sin(u)*sin(v);  
z=1+4*cos(u)*ones(size(v));  
surf(x,y,z); grid on; title('SPHERE');  
xlabel('X'); ylabel('Y'); zlabel('Z');
```



%Поверхность цилиндра

```
[x, y, z] = cylinder([1,3,5,7,6,4],25);  
surf(x, y, z);  
title ("Surface"); xlabel('X');  
ylabel('Y'); zlabel('Z');
```



# Решение задач линейной алгебры.

```
>>> %Решение матричного уравнения  $A \cdot X = B$ 
```

```
>>> A=[ 2 3;-2 6];B=[2 5;2/3 5/3]
```

```
>>> %Первый способ
```

```
>>> X=A\B
```

```
X =
```

```
    0.55556    1.38889
```

```
    0.29630    0.74074
```

```
>>> %Второй способ
```

```
>>> X=inv(A)*B
```

```
X =
```

```
    0.55556    1.38889
```

```
    0.29630    0.74074
```

```
>>> %Проверка  $A \cdot X - B = 0$ 
```

```
>>> A*X-B
```

```
ans =
```

```
    0.0000e+00    0.0000e+00
```

```
   -3.3307e-16   -6.6613e-16
```

```
%Решение СЛАУ методом Гаусса
disp('Решение СЛАУ методом Гаусса');
disp('Матрица системы:');
A=[2 1 -5 1;1 -3 0 -6;0 2 -1 2;1 4 -7 6]
disp('Вектор свободных коэффициентов:');
b=[8;9;-5;0]
disp('Расширенная матрица системы:');
C=rref([A b])
disp('Размерность матрицы C:');
n=size(C)
disp('Вектор решений СЛАУ Ax=b');
x=C(:,n(2))
disp('Проверка Ax-b');
A*x-b
```

%Решение СЛАУ методом LU-разложения

```
disp('Матрица системы:');  
A=[2 1 -5 1;1 -3 0 -6;0 2 -1 2;1 4 -7 6]  
disp('Вектор свободных коэффициентов:');  
b=[8;9;-5;0]  
disp('LU-разложение:');  
[L,U,P]=lu(A)  
Y=rref([L P*b])  
n=size(Y)  
y=Y(:,n(2))  
X=rref([U y])  
n=size(X)  
x=X(:,n(2))  
disp('Проверка Ax-b');  
A*x-b
```

```
% Собственные значения
% и собственные векторы матрицы
disp('Введите матрицу:');
A=input('A=');
[n,m]=size(A);
disp('Вектор собственных значений:');
d=eig(A)
[L, D]=eig(A);
disp('L- Матрица собственных векторов:');
L
disp('D - Матрица собственных значений:');
D
disp('Проверка:');
for i=1:n
    (A-D(i,i)*eye(n))*L(:,i)
end;
```

```
%Норма и число обусловленности
disp('Введите матрицу:'); A=input('A=');
[n,m]=size(A);
disp('Первая норма:');
n_1=norm(A,1)
N_1=max(sum(abs(A)))
disp('Число обусловленности в 1-й норме:');
c_1= cond(A,1)
C_1= norm(A,1)*norm(inv(A),1)
disp('Евклидова норма:');
n_e= norm(A,'fro')
N_e= sqrt(sum(diag(A*A')))
disp
('Число обусловленности в евклидовой норме:');
c_e= cond(A,'fro')
C_e= norm(A,'fro')*norm(inv(A),'fro')
```

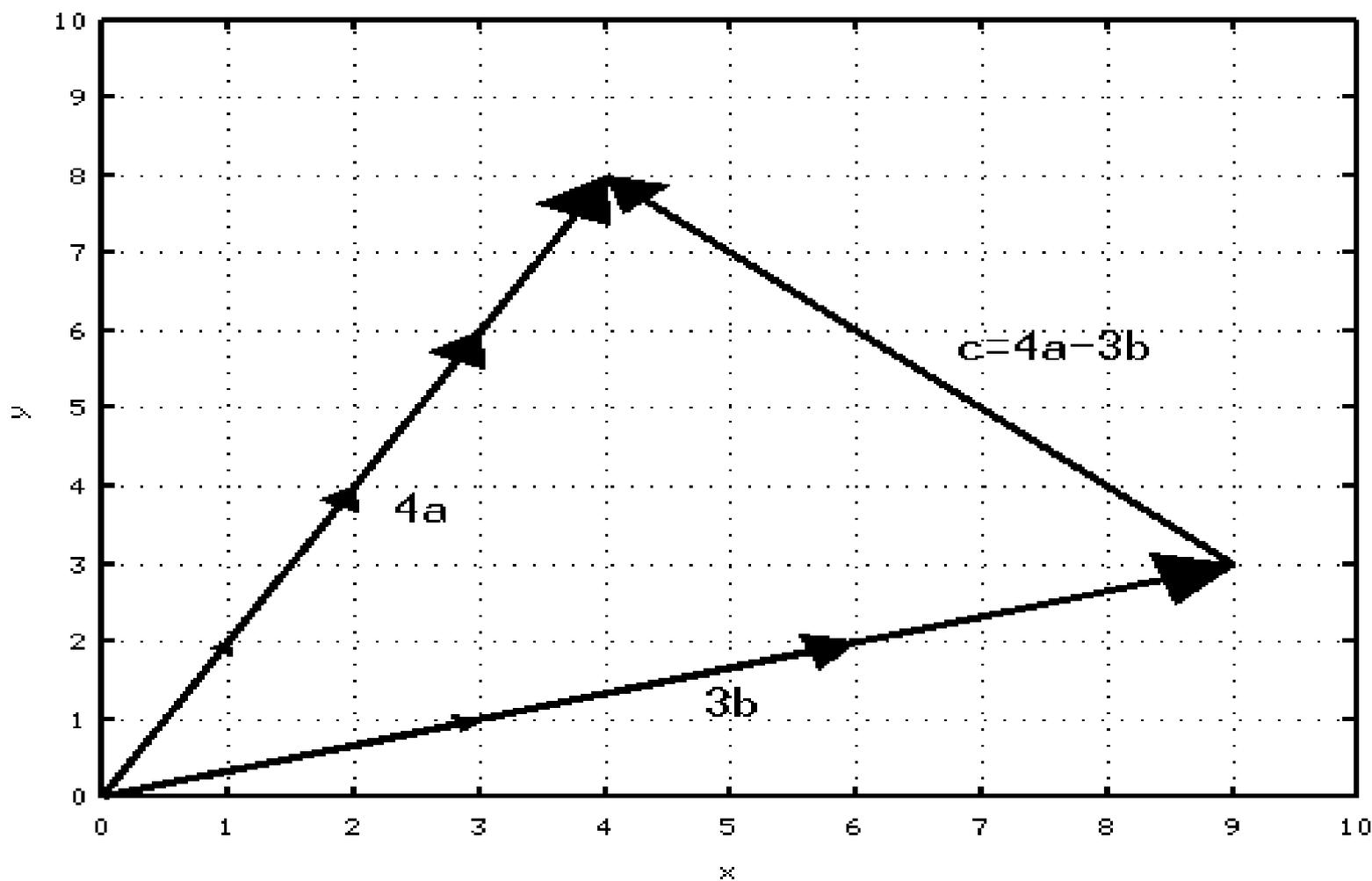
# Линейная алгебра в СИМВОЛЬНЫХ ВЫЧИСЛЕНИЯХ

%Решение СЛАУ

```
>>> x = sym ("x"); y = sym ("y");  
>>> a = sym ("a"); b = sym ("b");  
>>> c = sym ("c"); d = sym ("d");  
>>> sols =  
symIsolve ({a*x+b*y==c, x+y==d}, {x, y})  
sols =  
( [1] = -(a-b)^(-1)*(d*b-c)  
  [2] = -(a-b)^(-1)*(c-d*a)  
)
```

# Векторная алгебра.

Геометрическое представление действий с векторами

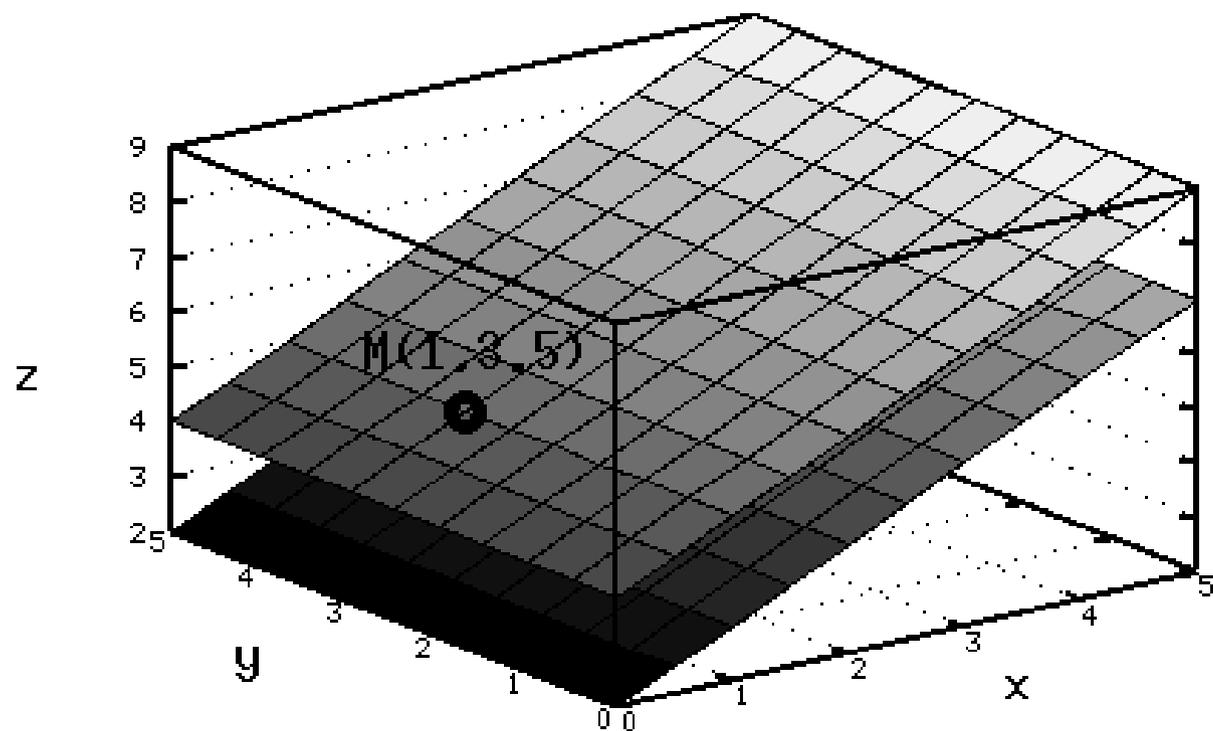


4.25084, -1.68550

# Аналитическая геометрия.

Плоскость, проходящая через точку параллельно заданной плоскости.

$$3x - 3z + 6 = 0, \quad 3x - 3z + 12 = 0$$



# Действия над многочленами.

%Умножение

```
>>> p1=[3 0 -7 0 5]; p2=[1 0 2 -1];
```

```
p=conv(p2,p1)
```

```
>>>p =      3      0     -1     -3     -9      7     10     -5
```

%Деление (частное и остаток)

```
>>>p1=[1 -1 3 0 -8 1 -10];p2=[1 0 1 -1];
```

```
[q,r]=deconv(p1,p2)
```

```
>>>q =      1     -1      2      2
```

```
r =      0      0      0      0     -11      1     -8
```

%Значение многочлена в точке

```
>>> p=[1 -1 3 0 -8 1 -10]; x=[-1,1];
```

```
polyval(p,x)
```

```
>>>ans =     -14     -14
```

%Определение многочлена по вектору корней

```
>>> x=[-2 3]; poly(x)
```

```
>>>ans =      1     -1     -6
```

# Решение алгебраических уравнений

```
>>> p=[2 -3 -12 -5];
```

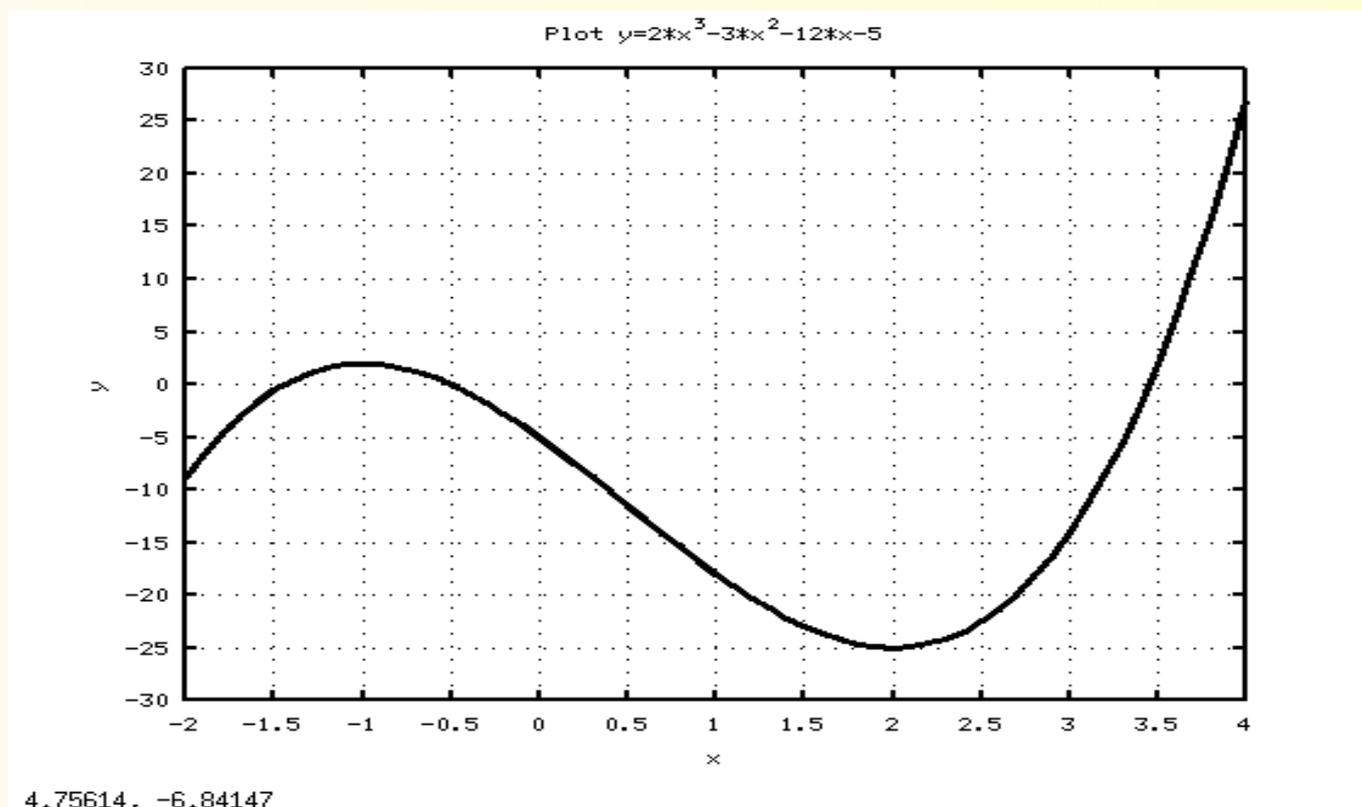
```
x=roots(p)
```

```
>>>x =
```

```
3.44949
```

```
-1.44949
```

```
-0.50000
```



% Действительные и мнимые корни полинома

```
p=[1 4 4 0 -9];
```

```
x=roots(p)
```

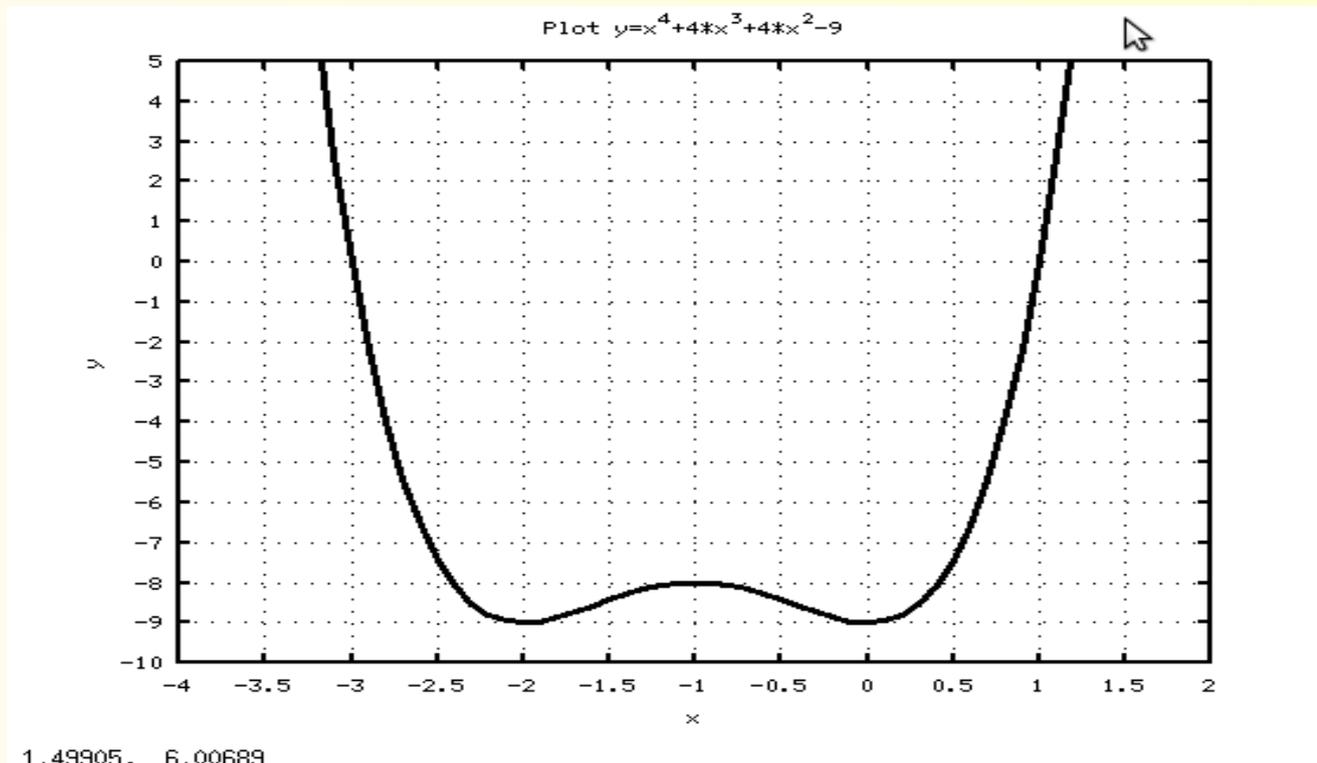
```
>>>x =
```

```
-3.000000 + 0.000000i
```

```
-1.000000 + 1.41421i
```

```
-1.000000 - 1.41421i
```

```
1.000000 + 0.000000i
```



1.49905, 6.00689

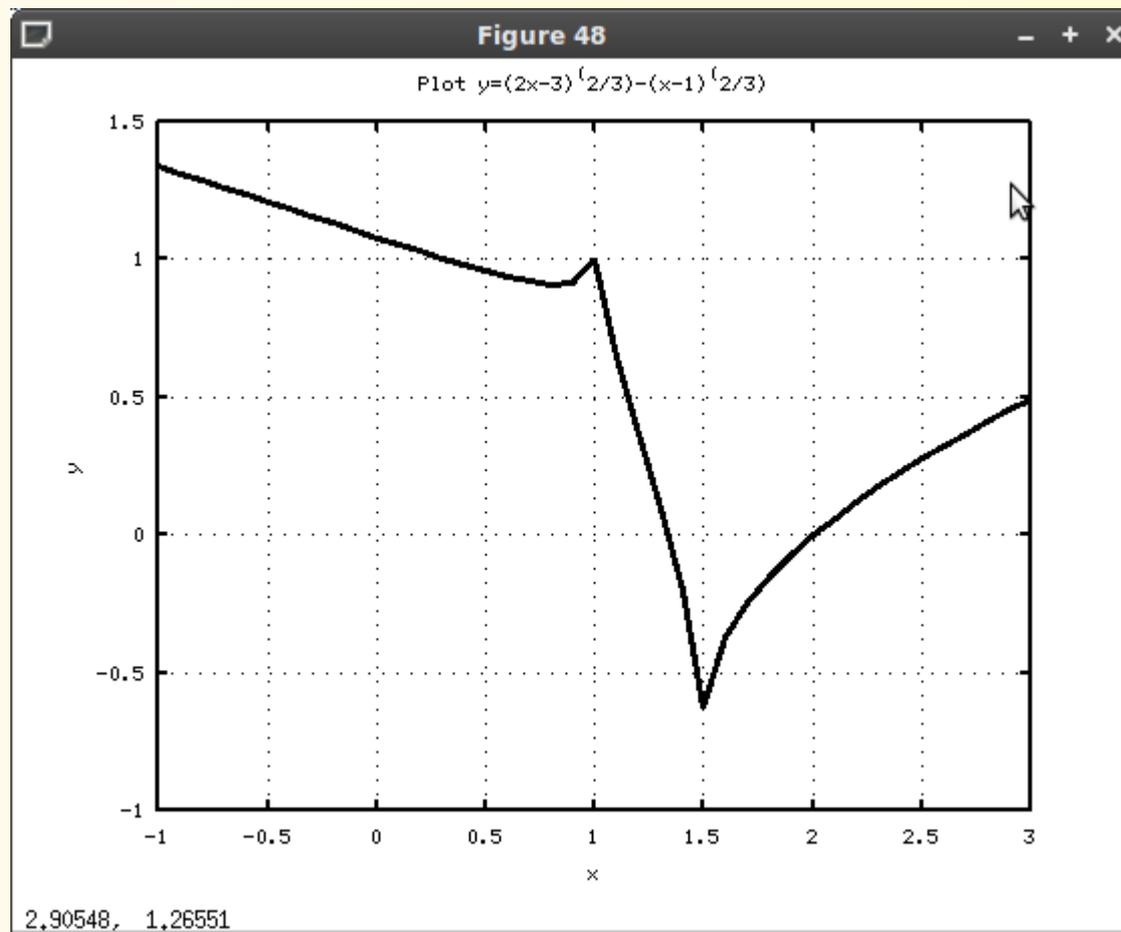
# Решение нелинейного уравнения

## График функции

```
function y=f1(x)
```

```
y=((2*x-3).^2).^(1/3)-(x-1).^2.^(1/3);
```

```
end;
```



# Решение нелинейного уравнения

```
[X(1), Y(1)] = fzero('f1', [1 1.5]);  
[X(2), Y(2)] = fzero('f1', [1.5 2.5]);
```

```
X %Решение уравнения
```

```
Y %Значения функции в точке X
```

```
>>>X = 1.333 2.000
```

```
>>>Y =
```

```
 -2.3870e-15    0.0000e+00
```

# Решение системы нелинейных уравнений

```
function f=Y(x)
f(1)=x(1)^2+x(2)^2+x(3)^2-1;
f(2)=2*x(1)^2+x(2)^2-4*x(3);
f(3)=3*x(1)^2-4*x(2)+x(3)^2;
end
[x,f,ex]=fsolve('Y',[0.5 0.5 0.5])
>>>x =
    0.78520    0.49661    0.36992
f =
    1.7571e-08    3.5199e-08    5.2791e-08
ex = 1
```

# Дифференцирование

```
%Производная многочлена
```

```
>>> p=[1 -1 3 0 -8 1 -10]; polyder(p)
```

```
>>>ans = 6 -5 12 0 -16 1
```

```
%Производная произведения многочленов
```

```
>>> p1=[3 0 -7 0 5]; p2=[1 0 2 -1];
```

```
polyder(p1,p2)
```

```
>>>ans = 21 0 -5 -12 -27 14 10
```

```
%Производная частного многочленов
```

```
>>> p1=[1 0 0 1]; p2=[1 -3 3 -1 0];
```

```
[q,r]=polyder(p1,p2)
```

```
>>> q = -1 -2 0 -4 1
```

```
      r = 1 -4 6 -4 1 0 0
```

# Дифференцирование

%Производная функции

```
symbols
```

```
x=sym("x");
```

```
f=(5*Sin(2*x))/Sqrt(Cos(2*x));
```

```
f1=differentiate(f,x)
```

```
>>>f1 =
```

```
(5.0)*sin((2.0)*x)^2*cos((2.0)*x)^(-/2)  
+ (10.0)*sqrt(cos((2.0)*x))
```

# Численное интегрирование

1. По методу трапеций

```
x=2:0.05:5;  
y=sqrt(2*x-1);  
I=trapz(x, y)  
>>>I 4 = 7.2679
```

2. По методу Симпсона

```
function y=G(x)  
y=(4-x^2).^ (1/2);  
end;  
[F1,K1]=quadv('G',0,1)  
  
>>>F1 = 1.91322288999134 K1 = 17
```

### 3. Интегрирование по квadrатурным формулам Гаусса

```
function y=f(x)  
y = tan(x).^4;  
end;  
format long  
[F]=quadl('f',-pi/3,pi/3,1.0e-05)  
>>>F = 2.09439512983937  
[F,err]=quadgk('f',-pi/3,pi/3,1.0e-05)  
>>>F = 2.09439510239319  
err = 1.02555919485880e-12
```

# Решение дифференциальных уравнений и систем.

Функции решения обыкновенных нежёстких дифференциальных уравнений (или систем) методом Рунге-Кутты 2-3-го и 4-5-го порядка точности соответственно:

```
ode23(@f, interval, X0, options)
ode45(@f, interval, X0, options)
```

Функции решения обыкновенных жёстких дифференциальных уравнений (или систем):

```
ode5r(f, interval, X0, options),
ode2r(f, interval, X0, options)
```

# Входные параметры функций решения дифференциальных уравнений (систем):

`f` – вектор-функция для вычисления правой части дифференциального уравнения или системы;

`interval` – массив из двух чисел, определяющий интервал интегрирования дифференциального уравнения или системы;

`X0` – вектор начальных условий системы дифференциальных систем;

`options` – параметры управления ходом решения дифференциального уравнения или системы.



# Метод наименьших квадратов.

`polyfit(x, y, k)` – подбор коэффициентов полинома  $k$ -й степени методом наименьших квадратов ( $x, y$  – экспериментальные точки,  $k$  – степень полинома) ;

`cor(x, y)` – вычисление коэффициента корреляции ( $x, y$  – экспериментальные точки) ;

`sqr(x0, phi, g, h, lb, ub, maxiter, tolerance)`  
– поиск минимума ;

# Интерполяция сплайнами

```
interp1(x, y, xi, '<метод>')
```

$x$  ,  $y$  – экспериментальные данные,  
 $x_i$  – точки, в которых необходимо  
вычислить значение с помощью сплайна,  
`method` – определяет метод построения  
сплайна.

Параметр `method` может принимать одно из  
следующих значений:

'linear' – линейная интерполяция,  
'spline' – кубический сплайн.

# Проблемы при использовании Octave

- Отсутствие документации на русском языке.
- Документация на английском языке недостаточно полная.
- Есть проблемы со стабильностью графических оболочек и пакетов расширений (<http://octave.sourceforge.net/packages.php>).

# Перспективы использования Octave

- Пакет стоит в первую очередь рекомендовать для использования при проведении научных исследований, со специальными функциями из пакетов расширений легче разбираться не IT-специалисту, а специалисту в предметной области.

# Выводы

- Авторами написана книжечка «GNU OCTAVE для студента и преподавателя»
- Черновик книжки можно увидеть на сайте <http://gnu-octave.narod2.ru>
- Дальнейшие планы – попробовать использовать Octave в серьёзных инженерных и математических расчётах
- Конкуренты Octave: Scilab, Maxima и компиляторы с Фортрана

СПАСИБО ЗА ВНИМАНИЕ!!!!