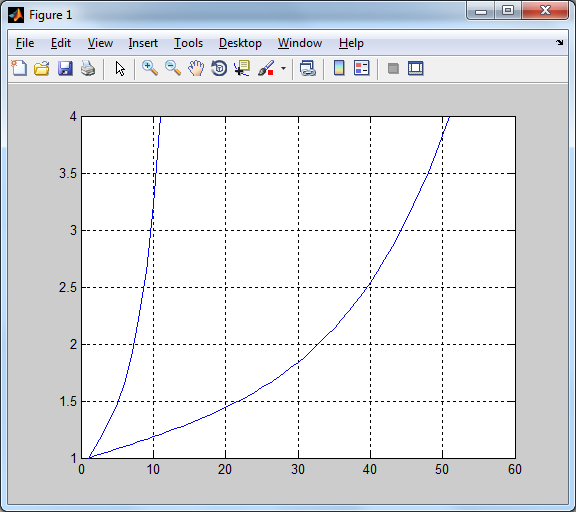
**Цель работы: получение базовых знаний для работы c двумерной и трёхмерной графикой в среде MATLAB, создание графиков функций одной переменной, построение поверхностей, использование М-файлов, создание пользовательских функций.**

**Задание №1**

**1.1:** в интервале [0.1, 1] с шагом 0.1 рассчитать таблицу значений для выражения из пункта №1 первой лабораторной работы, которое вы выбирали согласно вашему варианту



N=[0.1:0.1:1]

N =

0.1000 0.2000 0.3000 0.4000 0.5000 0.6000 0.7000 0.8000 0.9000 1.0000

>> A = sin (N\*100)

A =

-0.5440 0.9129 -0.9880 0.7451 -0.2624 -0.3048 0.7739 -0.9939 0.8940 -0.5064

>> B =tan(N)

B =

0.1003 0.2027 0.3093 0.4228 0.5463 0.6841 0.8423 1.0296 1.2602 1.5574

>>C= N.\*sqrt(N.^2+log(N))

C =

Columns 1 through 7

0 + 0.1514i 0 + 0.2506i 0 + 0.3166i 0 + 0.3479i 0 + 0.3328i 0 + 0.2330i 0.2556

Columns 8 through 10

0.5165 0.7555 1.0000

>> M = sqrt(A.^2+B.^2 ).\*(sin(A.^(B+C))).^2

M =

Columns 1 through 7

0.1573 + 0.0657i 0.6465 - 0.0194i -0.0454 + 0.1355i 0.5089 - 0.0764i -0.0144 + 0.0099i -0.0283 - 0.0198i 0.5368

Columns 8 through 10

-1.7702 - 0.7032i 0.7917 -0.0476 - 0.0181i

**1.2.** Выполнить задание 1.1, используя шаг -1.

>> N = [0.1:1:1]

N = 0.1000

A = sin (N\*100)

A = -0.5440

>> B =tan(N)

B = 0.1003

>>C= N.\*sqrt(N.^2+log(N))

C = 0 + 0.1514i

>> M = sqrt(A.^2+B.^2 ).\*(sin(A.^(B+C))).^2

M = 0.1573 + 0.0657i

**1.3.** Отобразить в линейном масштабе на одних осях два графика функций:

>> N = [0:0.1:1]

N =

Columns 1 through 8

0 0.1000 0.2000 0.3000 0.4000 0.5000 0.6000 0.7000

Columns 9 through 11

0.8000 0.9000 1.0000

>> M = [0:0.02:1]

M =

Columns 1 through 8

0 0.0200 0.0400 0.0600 0.0800 0.1000 0.1200 0.1400

Columns 9 through 16

0.1600 0.1800 0.2000 0.2200 0.2400 0.2600 0.2800 0.3000

Columns 17 through 24

0.3200 0.3400 0.3600 0.3800 0.4000 0.4200 0.4400 0.4600

Columns 25 through 32

0.4800 0.5000 0.5200 0.5400 0.5600 0.5800 0.6000 0.6200

Columns 33 through 40

0.6400 0.6600 0.6800 0.7000 0.7200 0.7400 0.7600 0.7800

Columns 41 through 48

0.8000 0.8200 0.8400 0.8600 0.8800 0.9000 0.9200 0.9400

Columns 49 through 51

0.9600 0.9800 1.0000

>> f1 = N.^5 +N.^3 +N +1

f1 =

Columns 1 through 8

1.0000 1.1010 1.2083 1.3294 1.4742 1.6563 1.8938 2.2111

Columns 9 through 11

2.6397 3.2195 4.0000

>> f1 = N.^5 +N.^3 +N +1 ;

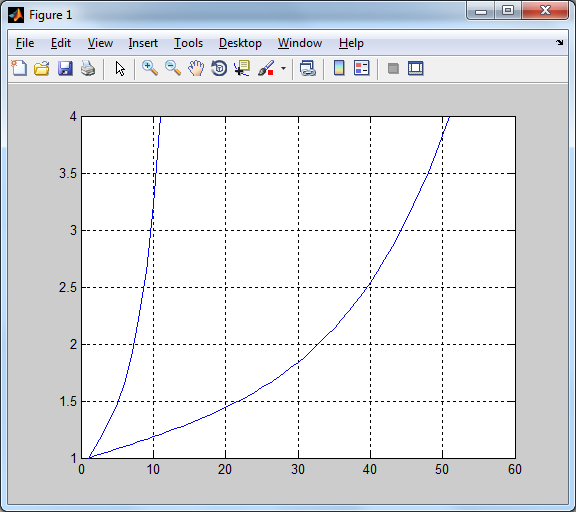
>> f2 = M.^5 +M.^3 + M + 1;

>> axes(handles.axes1);

>> plot(f1);

>> hold on;

>> grid on;

>> plot(f2);  


>>f3 = 10.^(-2) .\*f1 .\*sin(N);

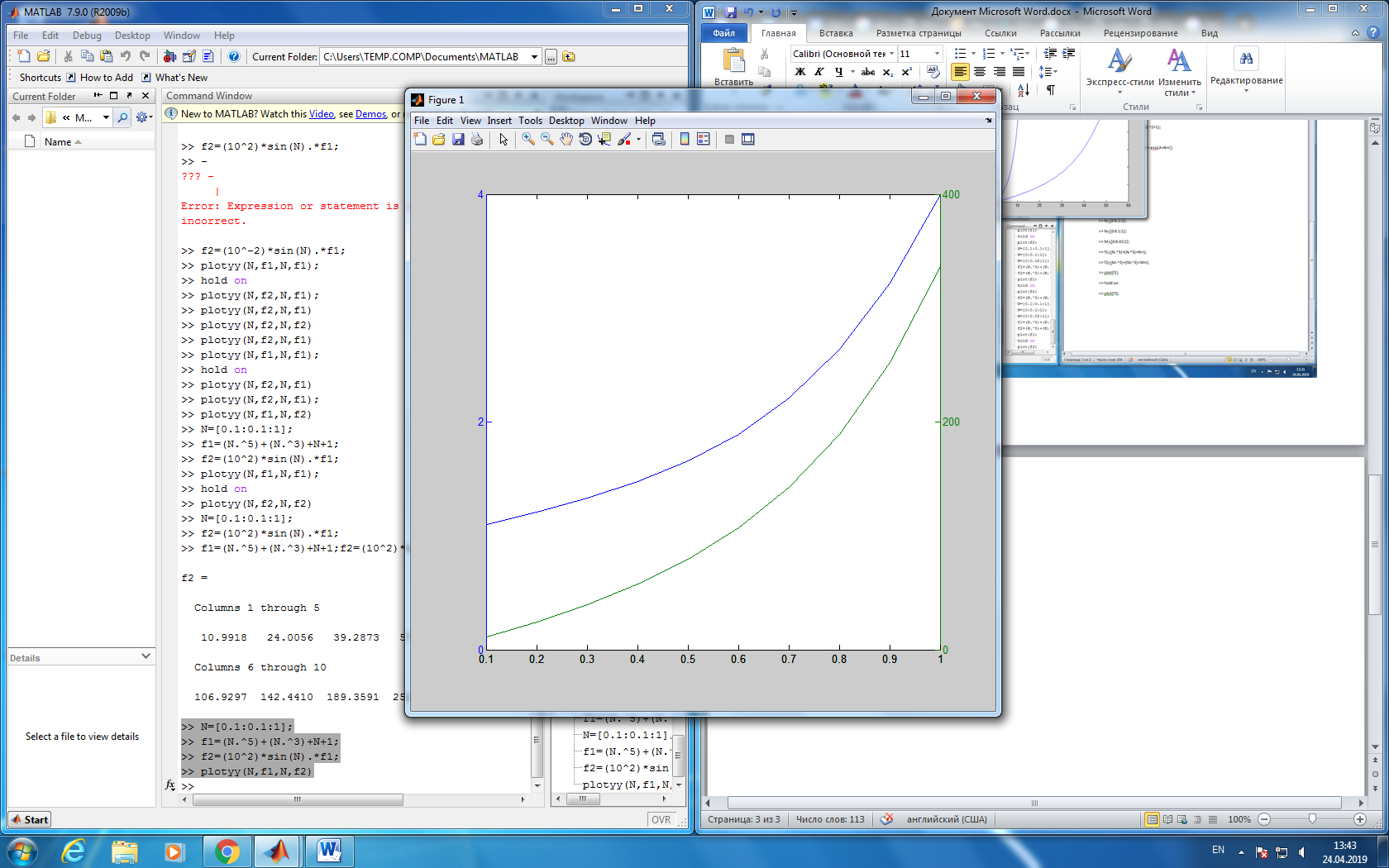
**1.4**. С помощью функции plotyy постройте графики функций 1 f N( ) и 2 1 10 ( ) sin( ) f N N − ⋅ . Что делает функция plotyy?

>> N=[0.1:0.1:1];

>> f1=(N.^5)+(N.^3)+N+1;

>> f2=(10^2)\*sin(N).\*f1;

>> plotyy(N,f1,N,f2)



**1.5**. Выполните задание 1.4, используя:

• логарифмический масштаб по обеим осям;

• логарифмический масштаб по оси абсцисс;

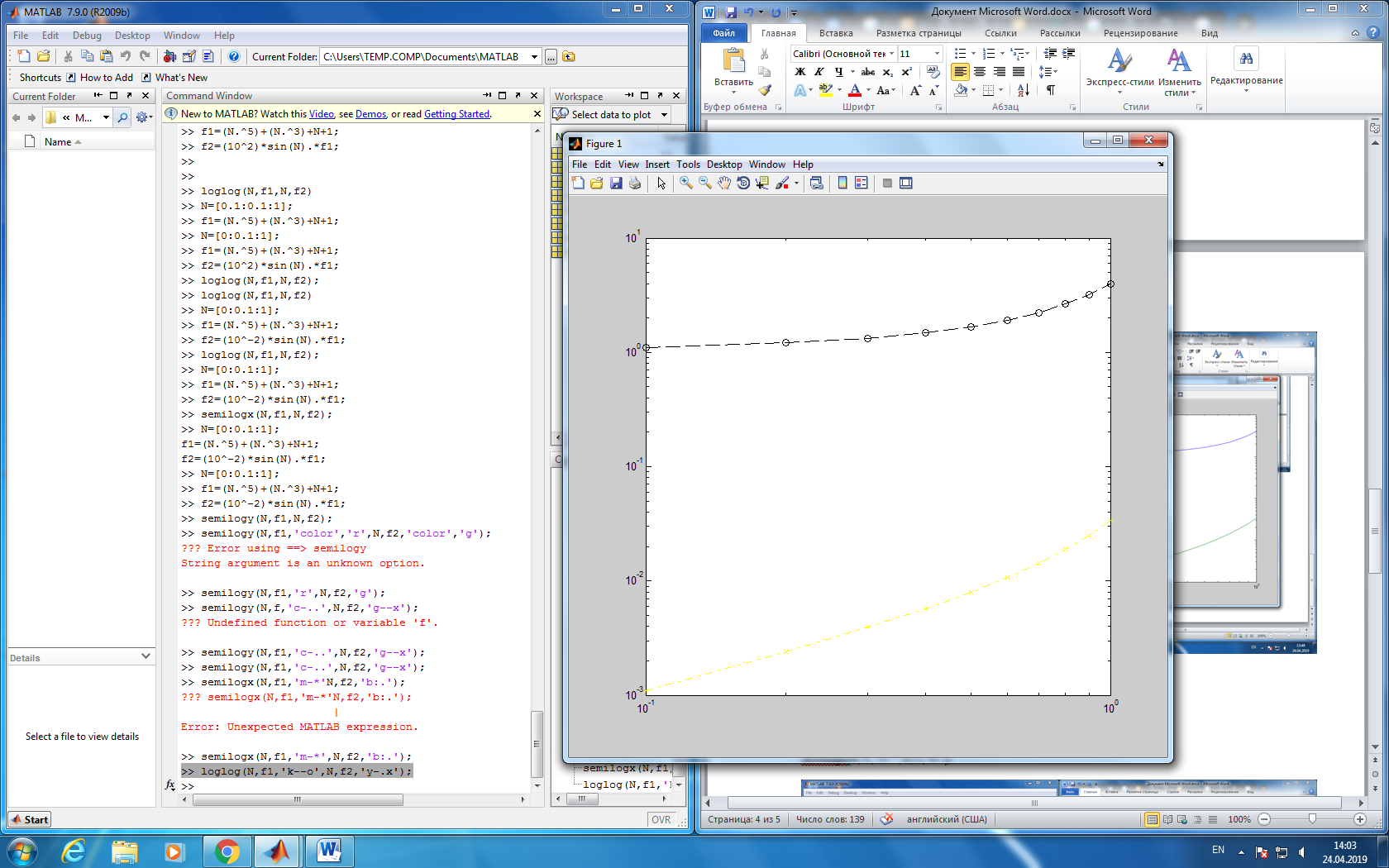
• логарифмический масштаб по оси ординат.

>> N=[0:0.1:1];

>> f1=(N.^5)+(N.^3)+N+1;

>> f2=(10^-2)\*sin(N).\*f1;

>> loglog(N,f1,'k--o',N,f2,'y-.x');

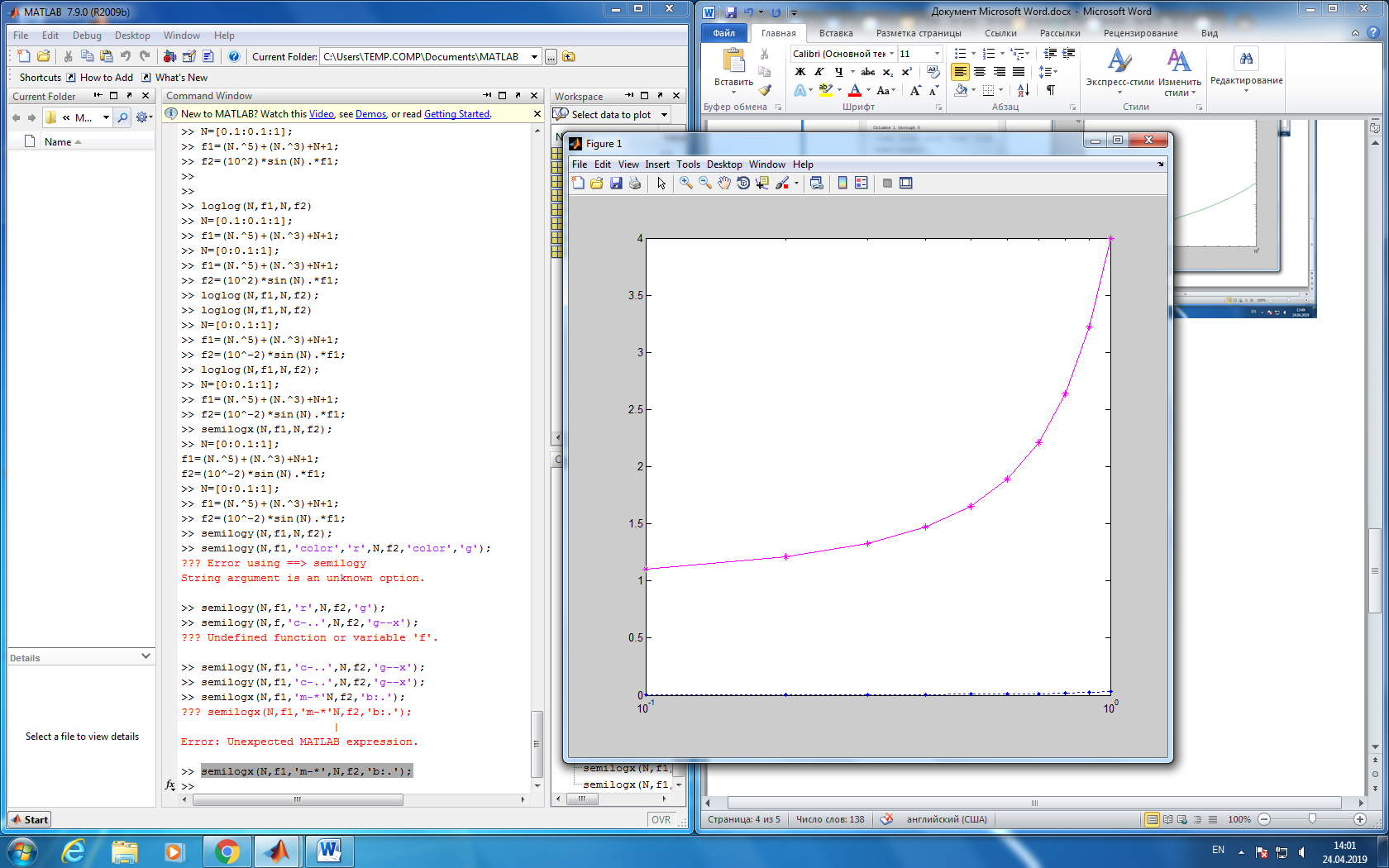


>> N=[0:0.1:1];

>> f1=(N.^5)+(N.^3)+N+1;

>> f2=(10^-2)\*sin(N).\*f1;

>>semilogx(N,f1,'m-\*',N,f2,'b:.');

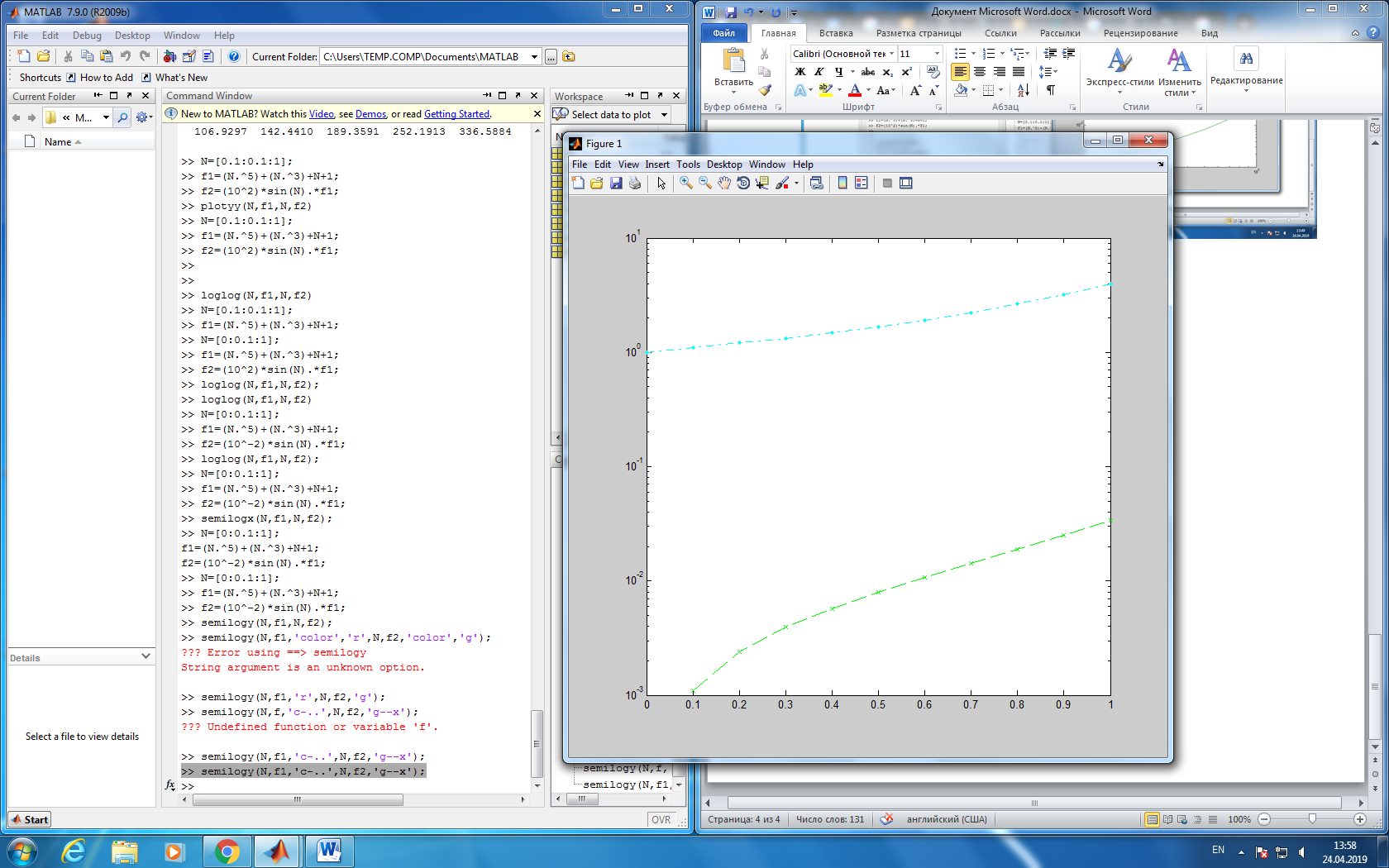


>> N=[0:0.1:1];

>> f1=(N.^5)+(N.^3)+N+1;

>> f2=(10^-2)\*sin(N).\*f1;

>> semilogy(N,f1,'c-..',N,f2,'g--x');



**1.6.** Сформируйте матрицу и вектор размерами соответственно не менее 5x6 и 1x7, первыми элементами которых является Ваш номер по списку в журнале группы. Постройте графики вектора и матрицы  
>>J = [3 4 8 6 7 1; 2 6 8 5 3 2; 2 8 7 4 1 3; 7 6 5 8 9 8; 3 4 3 2 1 3]

J =

3 4 8 6 7 1

2 6 8 5 3 2

2 8 7 4 1 3

7 6 5 8 9 8

3 4 3 2 1 3

>>V = [1;4;6;5;3;7;6]

V =

1

4

6

5

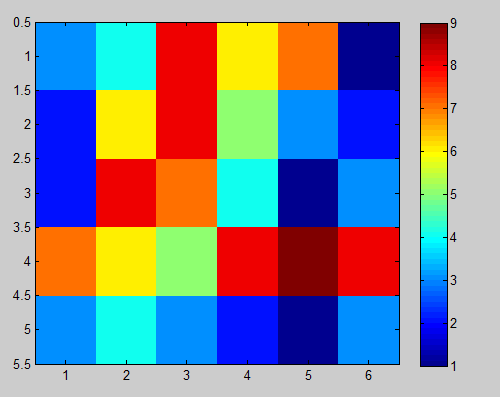
3

7

6

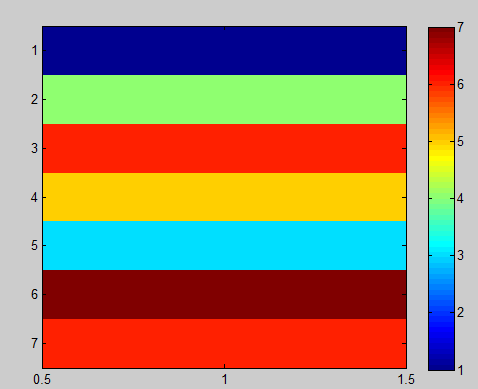
>> imagesc(A)

>> plot(J);



>> imagesc(V);

>> colorbar



**Задание №2 – двумерная графика**

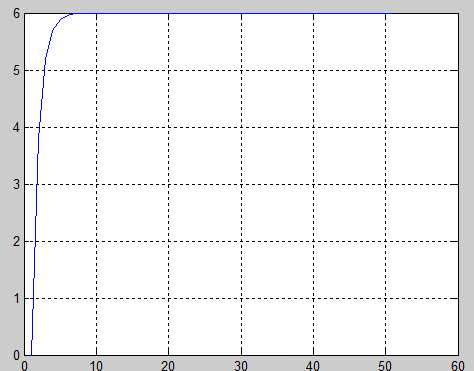
**2.1**. При подаче ступенчатого напряжения на двигатель постоянного тока скорость вращения его вала устанавливается не мгновенно, а с некоторым запозданием, вызванным инерционностью механической и электрической части двигателя. Типичный график изменения скорости двигателя при подаче ступенчатого напряжения представлен на рисунке 1.

>> x=0:0.1:5;

>> y1=6\*(1-exp(-x/0.1));

>> plot(y);

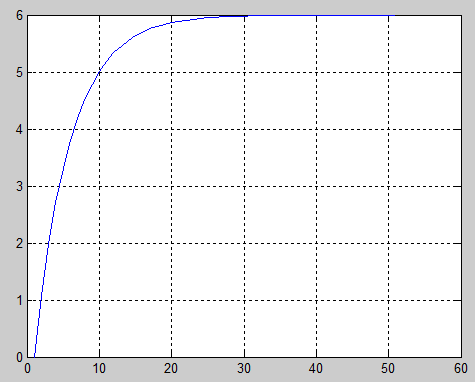
>> grid on



>> y2=6\*(1-exp(-x/0.5));

>> plot(y);

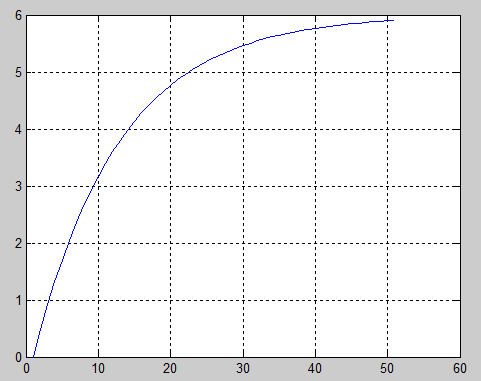
>> grid on



>> y3=6\*(1-exp(-x/1.2));

>> plot(y);

>> grid on



**2.2.** С помощью функции hold on отобразите все три графика в одном графическом окне. Используйте три различных типа линий и маркеров.

>> plot(x,y2,'m:+');

>> hold on

>> plot(x,y1,'r-.');

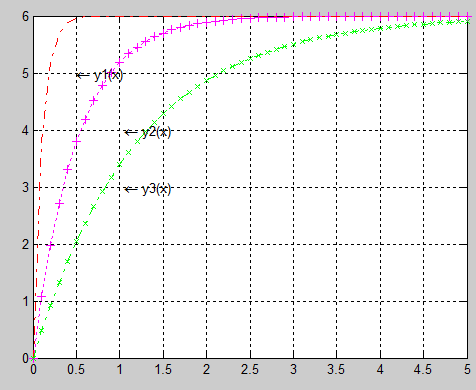
>> plot(x,y3,'g-.X');

>> grid on

>> text(0.5,5,'\leftarrow y1(x)');

>> text(1.05,4,'\leftarrow y2(x)');

>> text(1.05,3,'\leftarrow y3(x)');

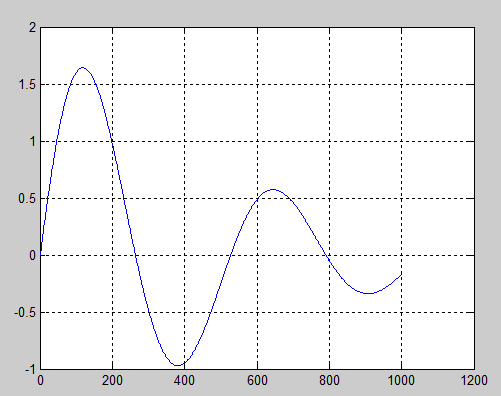


**2.3.** Математический маятник является классическим примером колебательной системы: при выводе его из равновесия, например, за счёт внешнего воздействия, угол отклонения ϕ нити маятника от вертикальной оси подчиняется колебательному закону движения.

>>f1=exp((-0.1/0.5).\*x).\*sin((sqrt(1-0.1.\*0.1)/0.5.\*x).\*6.\*0.1)/(0.5.\*(sqrt(1-0.1)));

>> plot(f1);

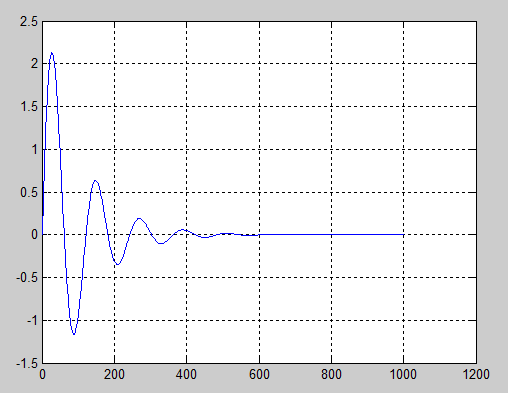
>> grid on



f2=exp((-0.5/0.5).\*x).\*sin((sqrt(1-0.5.\*0.5)/0.5.\*x).\*6.\*0.5)/(0.5.\*(sqrt(1-0.5)));

>> plot(f2);

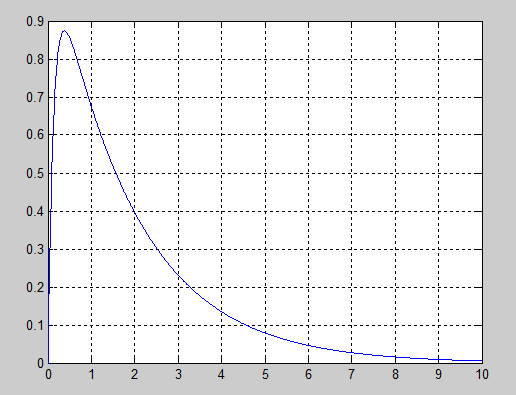
>> grid on



>> f3=exp((-2/0.5).\*x).\*sin(x.\*((1-2.\*2)^(0.5)/0.5)).\*1.\*2./(0.5.\*((1-2.\*2)^(0.5)));

>> plot(x,real(f3));

>> grid on



**2.4**. С помощью функции hold on отобразите все четыре графика в одном графическом окне. Используйте четыре различных типа линий и маркеров.

>> plot(x,f2,'g--X');

>> hold on

>> plot(x,f1,'m-.');

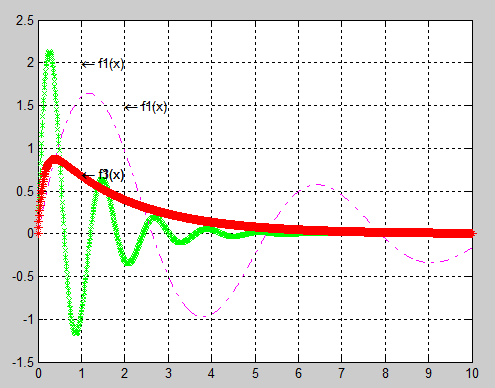
>> plot(x,f3,'r-\*');

Warning: Imaginary parts of complex X and/or Y arguments ignored

>> grid on

>> text(2,1.5,'\leftarrow f1(x)');

>> text(1,2,'\leftarrow f1(x)');

>> text(1,0.7,'\leftarrow f3(x)');

**Задание №3 – трёхмерная графика**

**3.1.** Постройте сетчатый график функции 2 2 z x y x y ( , ) = + на области определения в виде квадрата x∈[0,1] и y ∈[0,1] с шагом 0.2 и шагом 0.02.

>> [x,y]=meshgrid(0:0.2:1,0:0.02:1);

>> z=x.^2+y.^2

z =

0 0.0400 0.1600 0.3600 0.6400 1.0000

0.0004 0.0404 0.1604 0.3604 0.6404 1.0004

0.0016 0.0416 0.1616 0.3616 0.6416 1.0016

0.0036 0.0436 0.1636 0.3636 0.6436 1.0036

0.0064 0.0464 0.1664 0.3664 0.6464 1.0064

0.0100 0.0500 0.1700 0.3700 0.6500 1.0100

0.0144 0.0544 0.1744 0.3744 0.6544 1.0144

0.0196 0.0596 0.1796 0.3796 0.6596 1.0196

0.0256 0.0656 0.1856 0.3856 0.6656 1.0256

0.0324 0.0724 0.1924 0.3924 0.6724 1.0324

0.0400 0.0800 0.2000 0.4000 0.6800 1.0400

0.0484 0.0884 0.2084 0.4084 0.6884 1.0484

0.0576 0.0976 0.2176 0.4176 0.6976 1.0576

0.0676 0.1076 0.2276 0.4276 0.7076 1.0676

0.0784 0.1184 0.2384 0.4384 0.7184 1.0784

0.0900 0.1300 0.2500 0.4500 0.7300 1.0900

0.1024 0.1424 0.2624 0.4624 0.7424 1.1024

0.1156 0.1556 0.2756 0.4756 0.7556 1.1156

0.1936 0.2336 0.3536 0.5536 0.8336 1.1936

0.2116 0.2516 0.3716 0.5716 0.8516 1.2116

0.2304 0.2704 0.3904 0.5904 0.8704 1.2304

0.2500 0.2900 0.4100 0.6100 0.8900 1.2500

0.2704 0.3104 0.4304 0.6304 0.9104 1.2704

0.2916 0.3316 0.4516 0.6516 0.9316 1.2916

0.3136 0.3536 0.4736 0.6736 0.9536 1.3136

0.3364 0.3764 0.4964 0.6964 0.9764 1.3364

0.3600 0.4000 0.5200 0.7200 1.0000 1.3600

0.3844 0.4244 0.5444 0.7444 1.0244 1.3844

0.4096 0.4496 0.5696 0.7696 1.0496 1.4096

0.4356 0.4756 0.5956 0.7956 1.0756 1.4356

0.4624 0.5024 0.6224 0.8224 1.1024 1.4624

0.4900 0.5300 0.6500 0.8500 1.1300 1.4900

0.5184 0.5584 0.6784 0.8784 1.1584 1.5184

0.6400 0.6800 0.8000 1.0000 1.2800 1.6400

0.6724 0.7124 0.8324 1.0324 1.3124 1.6724

0.7056 0.7456 0.8656 1.0656 1.3456 1.7056

0.7396 0.7796 0.8996 1.0996 1.3796 1.7396

0.7744 0.8144 0.9344 1.1344 1.4144 1.7744

0.8100 0.8500 0.9700 1.1700 1.4500 1.8100

0.8464 0.8864 1.0064 1.2064 1.4864 1.8464

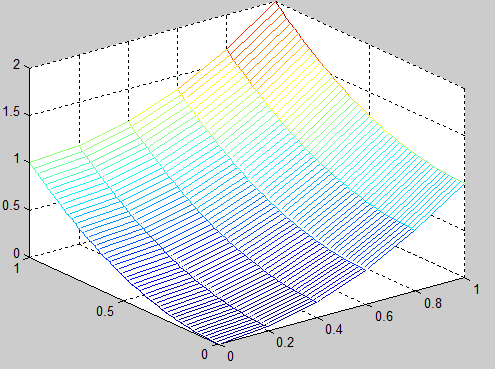
0.8836 0.9236 1.0436 1.2436 1.5236 1.8836

0.9216 0.9616 1.0816 1.2816 1.5616 1.9216

0.9604 1.0004 1.1204 1.3204 1.6004 1.9604

1.0000 1.0400 1.1600 1.3600 1.6400 2.0000

>> mesh(x,y,z)

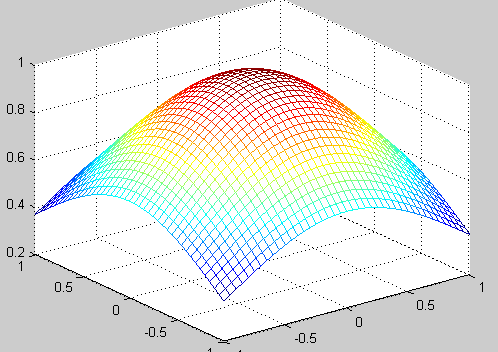


**3.2**. Постройте сетчатый график функции.

>> [x,y]=meshgrid(-1:0.05:1,-1:0.05:1);

>> z=exp(-((((x-0).^2)/2)+(((y-0).^2)/2)));

>> mesh(x,y,z)



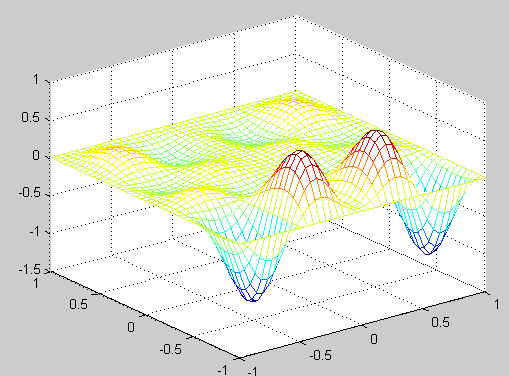
**3.3.** Постройте прозрачную и непрозрачную сетчатую поверхность для функции

>> [x,y]=meshgrid(-1:0.05:1, -1:0.05:1);

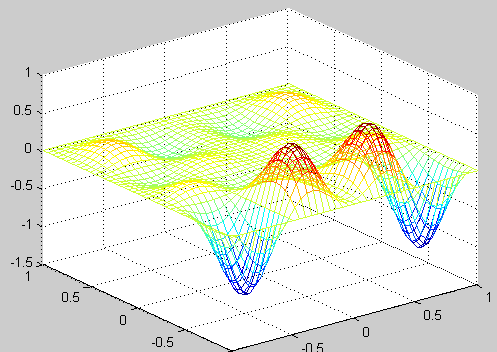
>> N=5;

>> z=(N/2)\*sin(2\*pi\*x).\*cos(1.5\*pi\*y).\*(1-x.^2).\*y.\*x.\*(1-y);

>> mesh(x,y,z)



>>hidden off



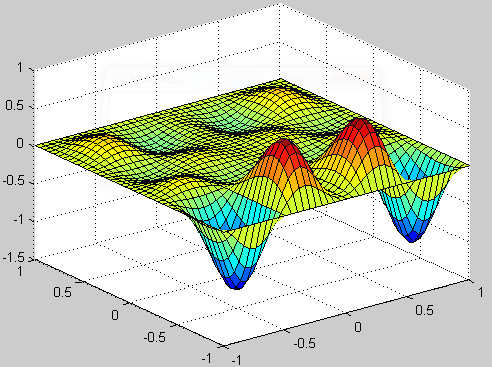
**3.4.** Постройте каркасную поверхность функции z x y ( , )

>> [x,y]=meshgrid(-1:0.05:1, -1:0.05:1);

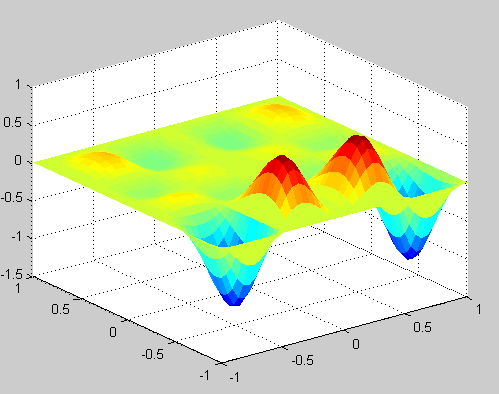
>> N=5;

>> z=(N/2)\*sin(2\*pi\*x).\*cos(1.5\*pi\*y).\*(1-x.^2).\*y.\*x.\*(1-y);

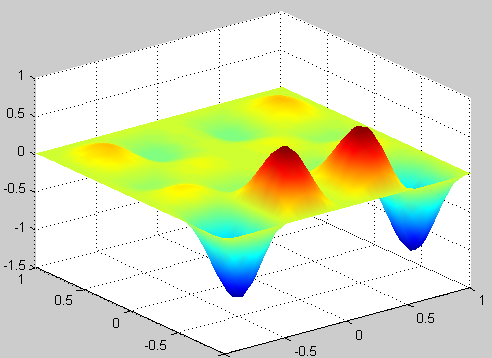
>> surf(x,y,z)



>> shading flat



>> shading interp



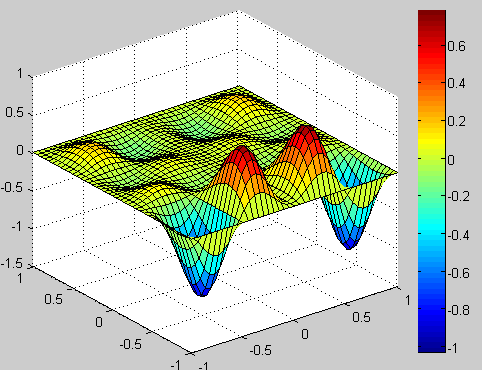
**3.5.** Постройте каркасную поверхность функции z x y ( , ) с помощью команд surf(x, y, z)и colorbar.

>> [x,y]=meshgrid(-1:0.05:1,-1:0.05:1);

>> z=(N/2)\*sin(2\*pi\*x).\*cos(1.5\*pi\*y).\*(1-x.^2).\*y.\*x.\*(1-y);

>> surf(x,y,z)

>> colorbar

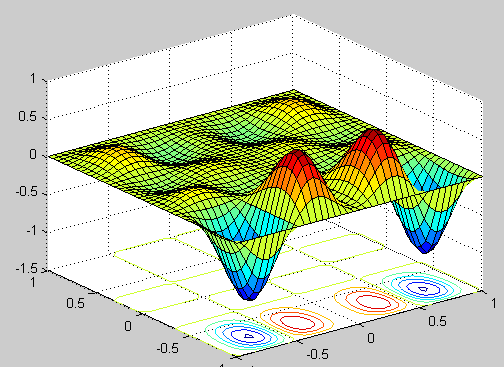


**3.6.** Постройте каркасную поверхность функции z x y ( , ) с помощью команд surfc, meshc.

>> [x,y]=meshgrid(-1:0.05:1,-1:0.05:1);

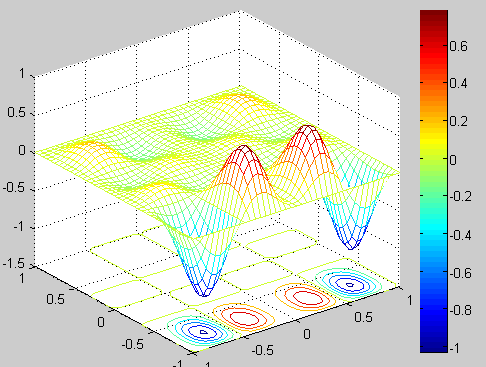
>> z=(N/2)\*sin(2\*pi\*x).\*cos(1.5\*pi\*y).\*(1-x.^2).\*y.\*x.\*(1-y);

>> surfc(x,y,z)



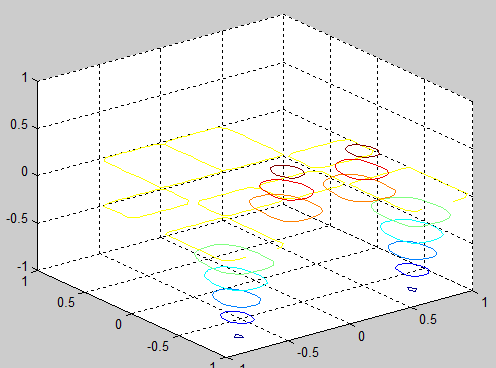
>> meshc(x,y,z)

>> colorbar

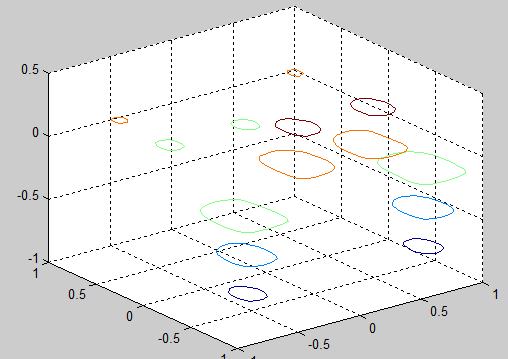


**3.7.** Постройте поверхности функции, которые состоят из линий уровня, с помощью функции contour3

>> contour3(x,y,z)



>>contour3(x,y,z,5)

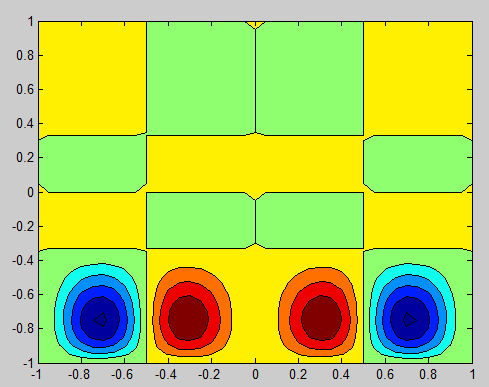


**3.8**. Постройте контурный график функции с помощью функций contour, contourf, clabel

>> contour(x,y,z)

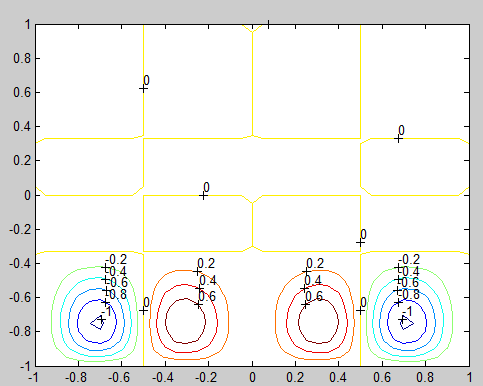
### 

>> contourf(x,y,z)



>> C=contour(x,y,z);

>> clabel(C)



**3.9**. Выполните три разных цветовых оформления графика функции.

>> [x,y]=meshgrid(-1:0.05:1, -1:0.05:1);

>> N=5;

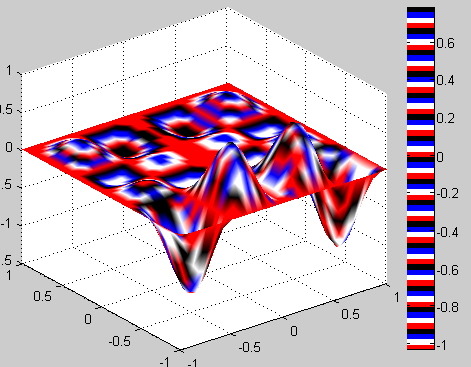
>> z=(N/2)\*sin(2\*pi\*x).\*cos(1.5\*pi\*y).\*(1-x.^2).\*y.\*x.\*(1-y);

>> surf(x,y,z)

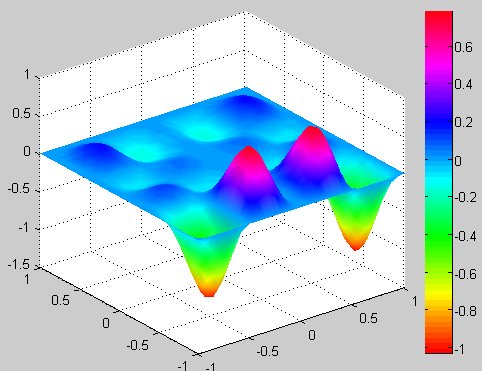
>> colorbar

>> shading interp

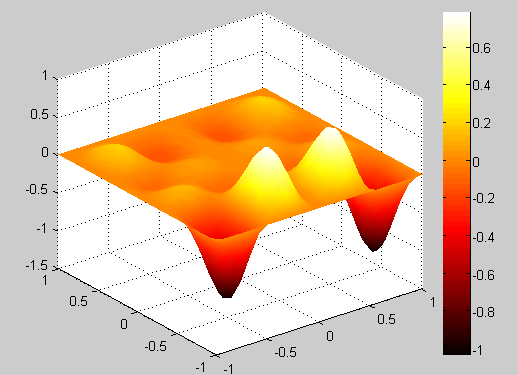
>> colormap('flag')



>> colormap('hsv')



>> colormap('hot')



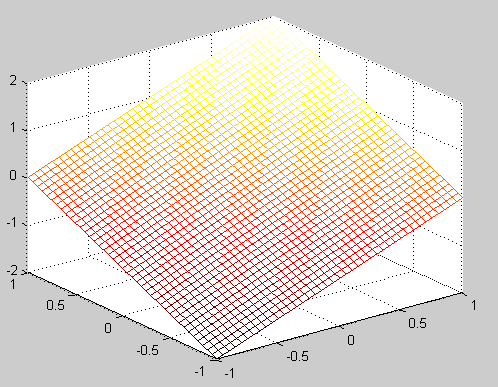
**3.10**. Выполните построение конуса и плоскости в соответствии с рекомендациями пунктов 3.3 и 3.4

>> [x,y]=meshgrid(-1:0.05:1, -1:0.05:1);

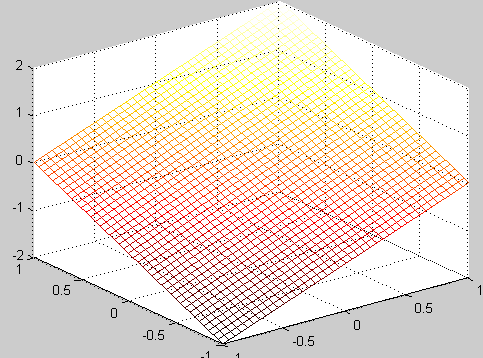
>> N=5;

>> z=x+y;

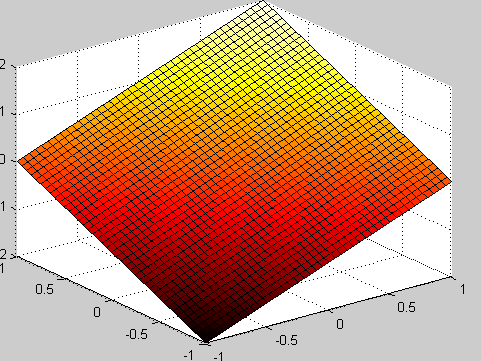
>> mesh(x,y,z)



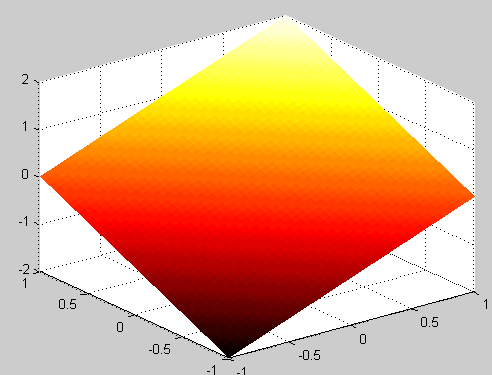
>> hidden off



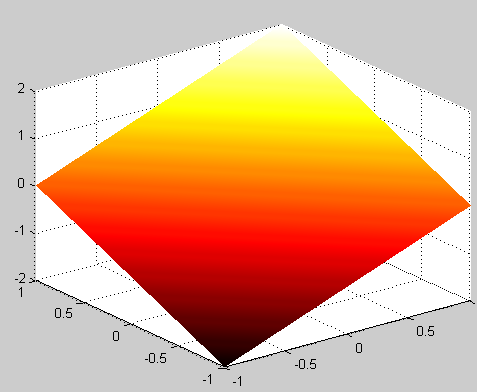
>> surf(x,y,z)



>> shading flat



>>shading interp



>> t=0:2\*pi/20:2\*pi;

>> z=0:0.2:10;

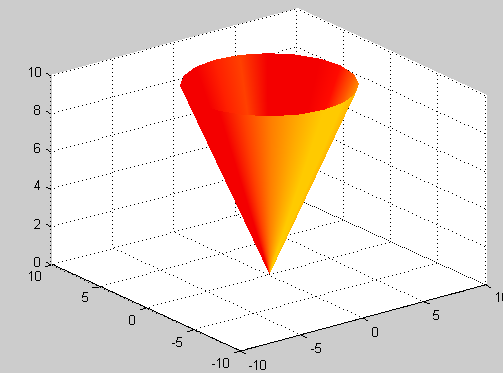
>> [t,z]=meshgrid(t,z);

>> x=z.\*cos(t)\*tan(pi/6);

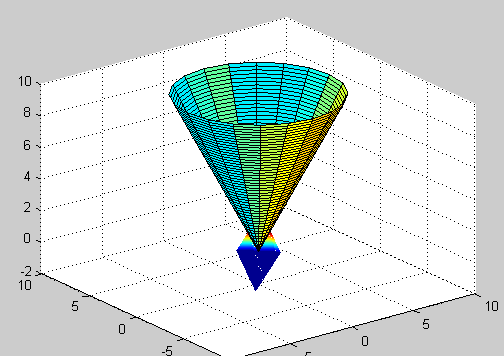
>> y=z.\*sin(t)\*tan(pi/6);

>> surfl(x,y,z)

>> shading interp



**3.11.** Пересеките конус плоскостью.



**Задание №4 – М-файлы и функции Работа с М-файлами и пользовательскими функциями.**

**4.1** Работа с М-файлами и пользовательскими функциями.

1. Создайте в корневом каталоге диска каталог со своей фамилией, например, Work\_ivanov.

2. Создайте М-файл и сохраните его в только что созданный каталог под именем mydemo.

3. В этот файл вставьте следующий код:

>>x = [0:0.1:7];

>>f = exp(-x);

>>subplot(1, 2, 1)

>>plot(x, f)

>>g = sin(x);

>>subplot(1, 2, 2)

>>plot(x, g)

Объясните, что делает данная программа и добавьте комментарии к каждой строке.

4. Продемонстрируйте доступность созданного файла из командной строки:

>> mydemo

Объясните результат.

x = [0:0.1:7]; инициализация переменной x

f = exp(-x); задание первой функции

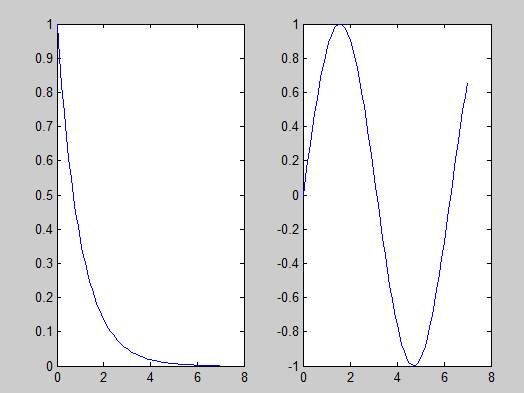
subplot(1, 2, 1)%функция, позволяющая построить несколько графиков функций в одном окне, первая цифра - количество частей на которое разбивается окно по горизонтали, второе - по вертикали, третье - номер текущего подокна

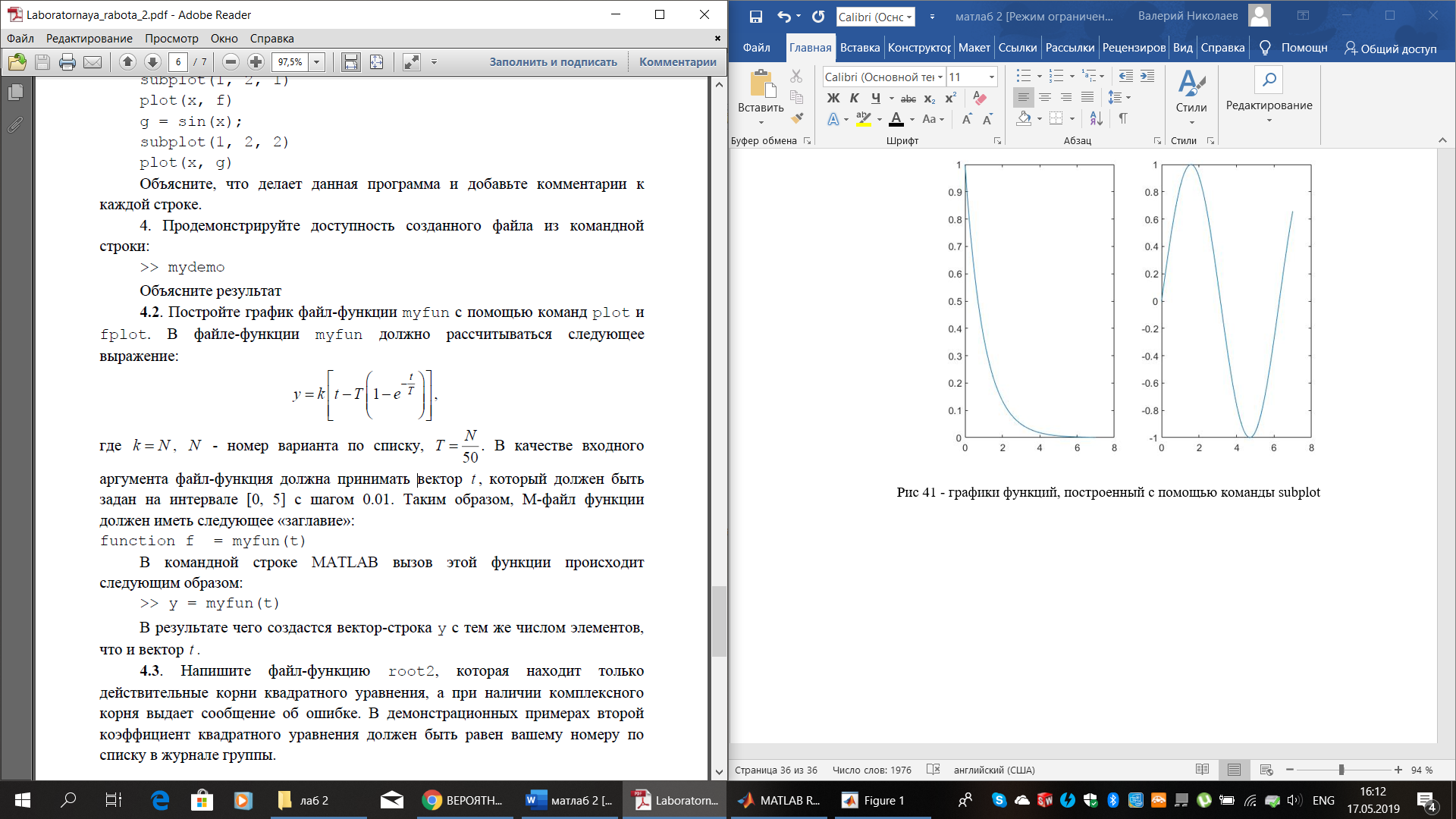
>>plot(x, f)

>>g = sin(x);

>>subplot(1, 2, 2)

>>plot(x, g)





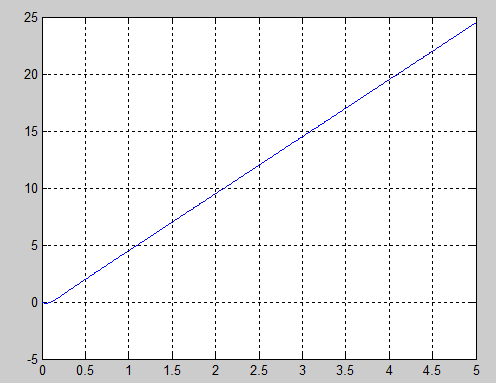
function f = myfun(t);

f=5\*(t-5/50)\*(1-exp(-(t./(5/50)))));

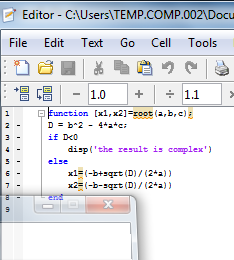
>> y=myfun(t);

>> plot(t,y)

>> grid on



**4.3**. Напишите файл-функцию root, которая находит только действительные корни квадратного уравнения, а при наличии комплексного корня выдает сообщение об ошибке. В демонстрационных примерах второй коэффициент квадратного уравнения должен быть равен вашему номеру по списку в журнале группы.



**>> [a,b]=root2(1,5,3);**

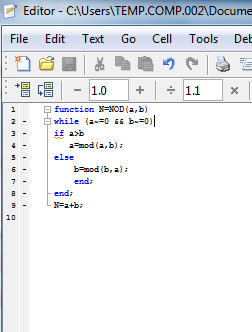
**x1 =**

**-3.1972**

**x2 =**

**-6.8028**

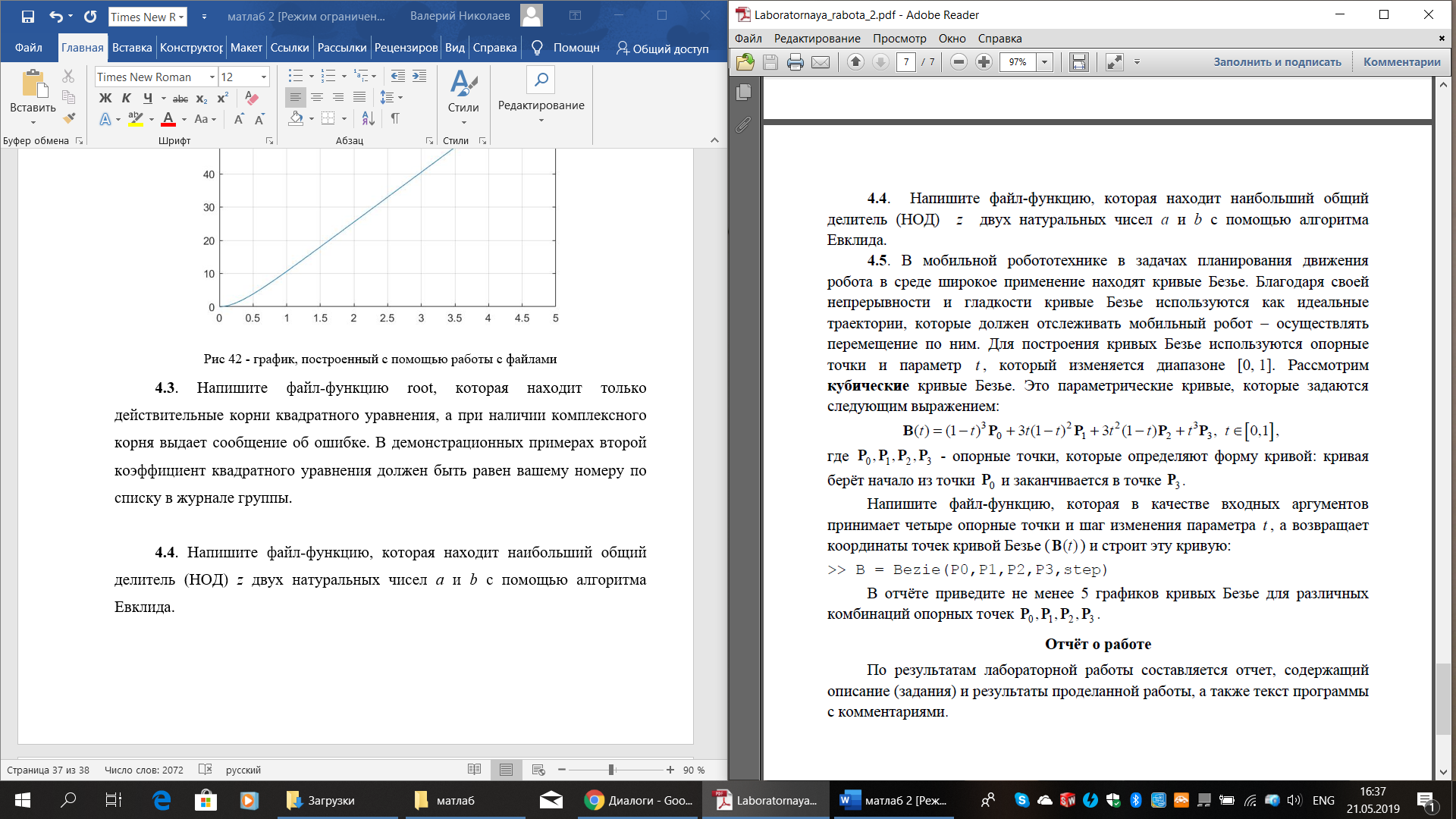
**4.4**. Напишите файл-функцию, которая находит наибольший общий делитель (НОД) *z* двух натуральных чисел *a* и *b* с помощью алгоритма Евклида.

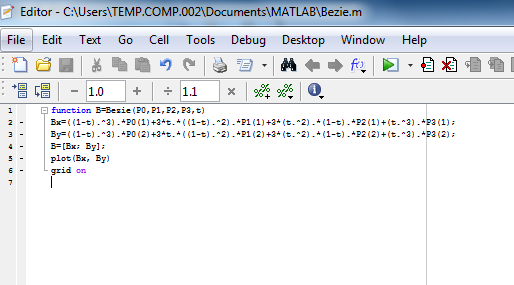


**>> N=NOD(45,3)**

**N =**

**3**





**>> t=[0:0.1:1];**

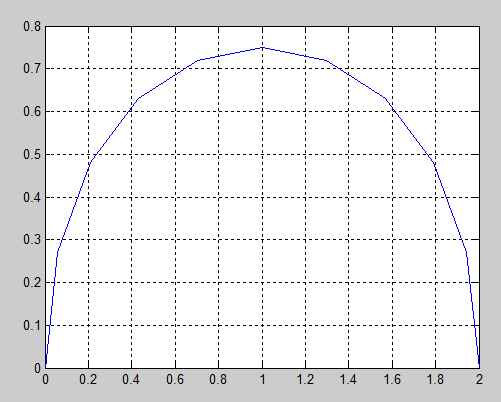
**>> P0=[0.0, 0.0]; P1=[0.0, 1]; P2=[2, 1]; P3=[2, 0];**

**>> B=Bezie(P0,P1,P2,P3,t)**

**B =**

**0 0.0560 0.2080 0.4320 0.7040 1.0000 1.2960 1.5680 1.7920 1.9440 2.0000**

**0 0.2700 0.4800 0.6300 0.7200 0.7500 0.7200 0.6300 0.4800 0.2700**



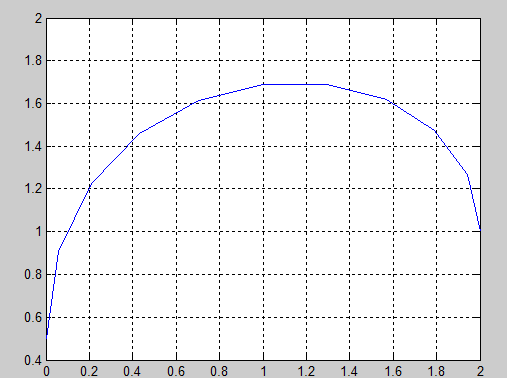
**>> P0=[0.0, 0.5]; P1=[0.0, 2]; P2=[2, 2]; P3=[2, 1];**

**>> B=Bezie(P0,P1,P2,P3,t)**

**B =**

**0 0.0560 0.2080 0.4320 0.7040 1.0000 1.2960 1.5680 1.7920 1.9440 2.0000**

**0.5000 0.9055 1.2240 1.4585 1.6120 1.6875 1.6880 1.6165 1.4760 1.2695 1.0000**



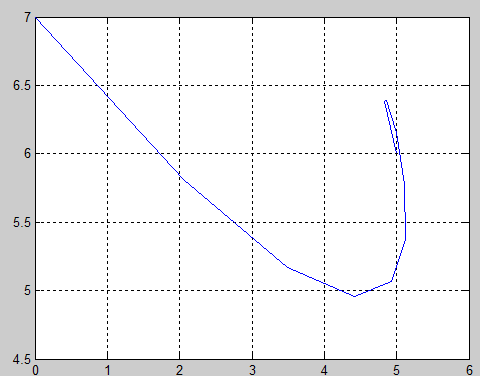
**>> P0=[0.0, 7]; P1=[8, 2]; P2=[4, 8]; P3=[5, 6];**

**>> B=Bezie(P0,P1,P2,P3,t)**

**B =**

**0 2.0570 3.4960 4.4190 4.9280 5.1250 5.1120 4.9910 4.8640 4.8330 5.0000**

**7.0000 5.8110 5.1680 4.9570 5.0640 5.3750 5.7760 6.1530 6.3920 6.3790 6.0000**



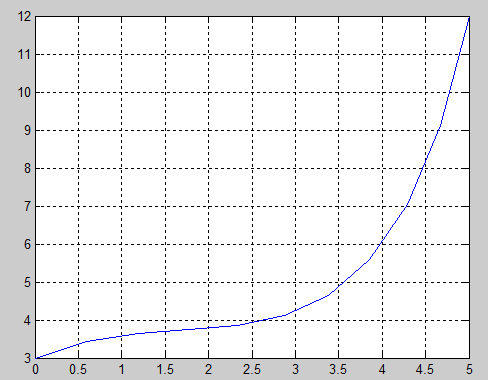
**>>P0=[0.0, 3]; P1=[2, 5]; P2=[4, 1]; P3=[5, 12];**

**>> B=Bezie(P0,P1,P2,P3,t)**

**B =**

**0 0.5990 1.1920 1.7730 2.3360 2.8750 3.3840 3.8570 4.2880 4.6710 5.0000**

**3.0000 3.4410 3.6480 3.7470 3.8640 4.1250 4.6560 5.5830 7.0320 9.1290 12.0000**



**P0=[0.0, 3]; P1=[0.2, 9]; P2=[2, 9]; P3=[1, 0.0];**

**>> B=Bezie(P0,P1,P2,P3,t)**

**B =**

**0 0.1036 0.2768 0.4932 0.7264 0.9500 1.1376 1.2628 1.2992 1.2204 1.0000**

**3.0000 4.6170 5.8560 6.6990 7.1280 7.1250 6.6720 5.7510 4.3440 2.4330 0**

