



**Министерство науки и высшего образования  
Российской Федерации  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технологический университет «СТАНКИН»  
(ФГБОУ ВО «МГТУ «СТАНКИН»)**

---

Институт автоматизации и робототехники

Кафедра “Робототехника и мехатроника”

Учебный курс “Пакет прикладных программ MATLAB для  
исследований и разработок”

**Лабораторная работа №2  
«Построение графиков функций в MATLAB. М-файлы»**

Выполнил:  
студент группы АДБ-17-11

\_\_\_\_\_  
(подпись)

Абдулзагиров М.М.  
(ФИО)

Принял  
преподаватель:

\_\_\_\_\_  
(подпись)

Ким В.Л.  
(ФИО)

Оценка: \_\_\_\_\_

Дата: \_\_\_\_\_

Москва 2019

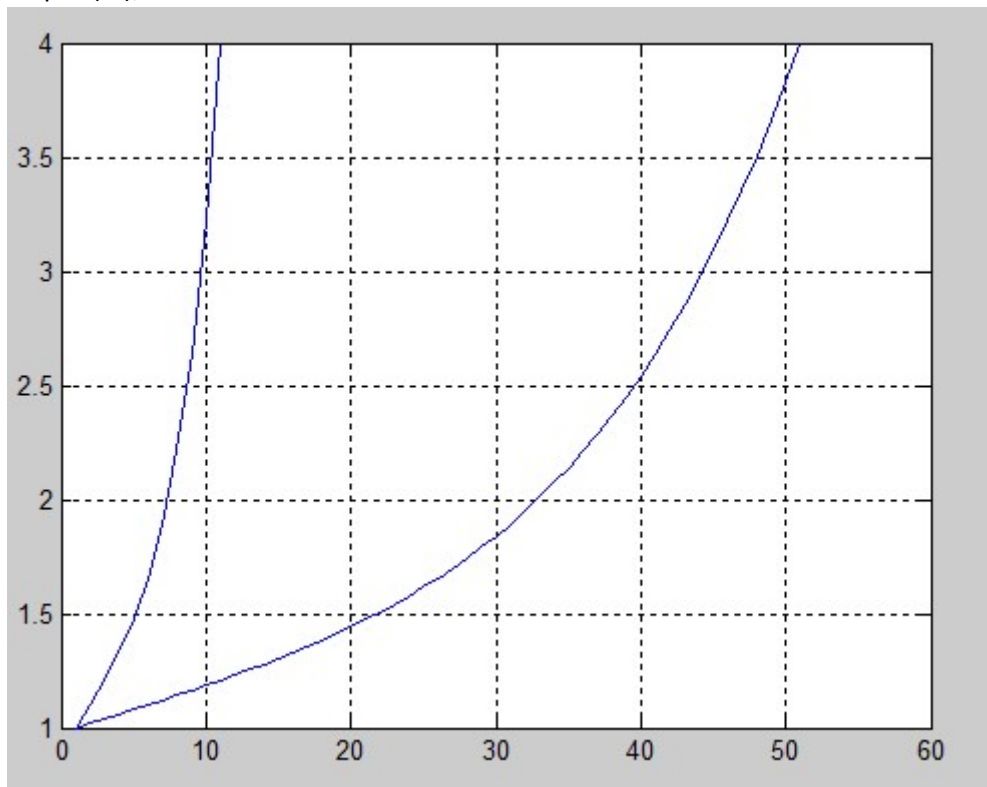
**Цель работы:** получение базовых знаний для работы с двумерной и трёхмерной графикой в среде MATLAB, создание графиков функций одной переменной, построение поверхностей, использование М-файлов, создание пользовательских функций.

## Задание №1

```
>> N = [0.1:0.1:1];
>> A = cos (N + 100) ;
>> C= log(5.*N +2) ;
>> B = sqrt(N)./(N.^3 +N.^2 + 1);
>> M = A.^2 .*tan(B+C.^3);
>> A = cos (N + 100);
>> B = sqrt(N)/(N.^3 +N.^2 + 1)
B =
    0.3128
>> C= log(5.*N +2)
C =
    0.9163
>> D = A.^2 .*tan(B+C.^3)
D =
    1.5525
>> N = [0:0.1:1];
>> M = [0:0.02:1];

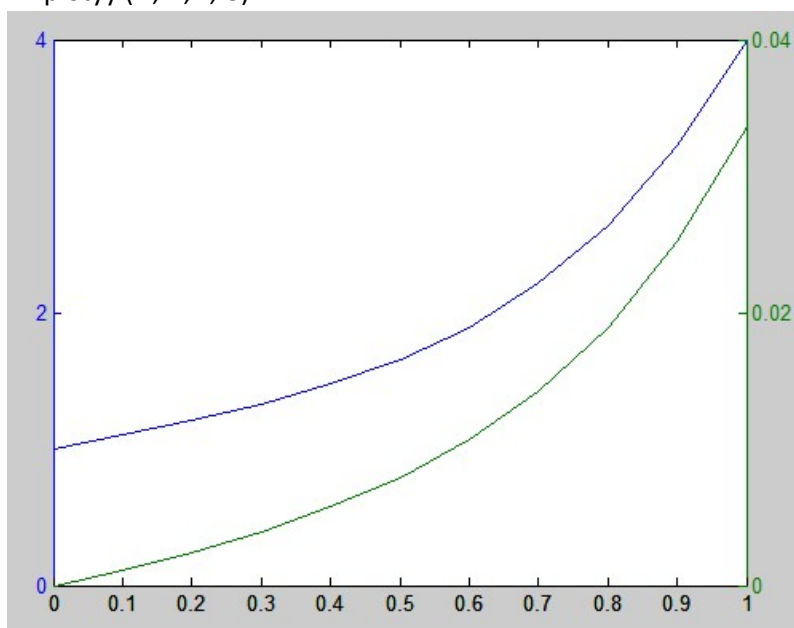
>> f1 = N.^5 +N.^3 +N +1 ;
>> f2 = M.^5 +M.^3 + M + 1;
>> plot(f1);
>> hold on;
>> grid on;
```

```
>> plot(f2);
```

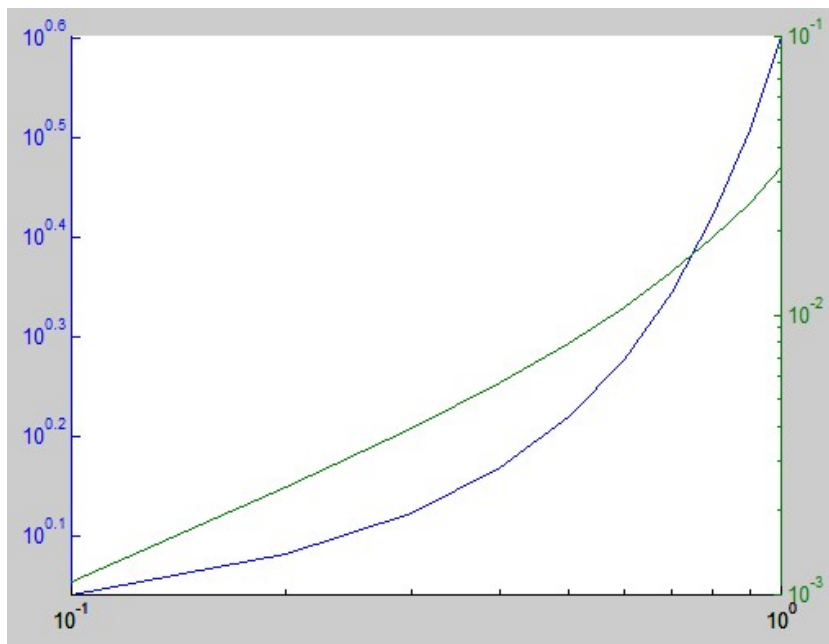


```
>> f3 = 10.^(-2) .* f1 .* sin(N);
```

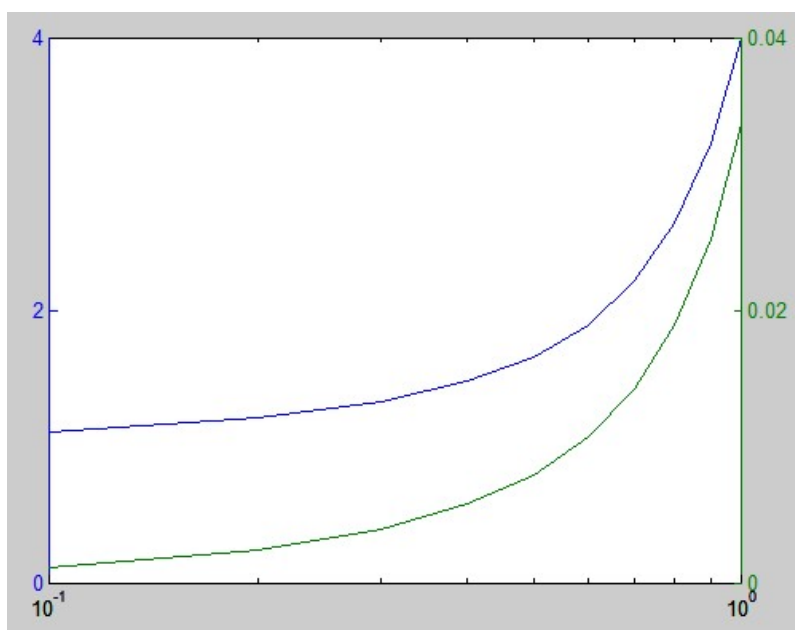
```
>> plotyy(N,f1,N,f3)
```



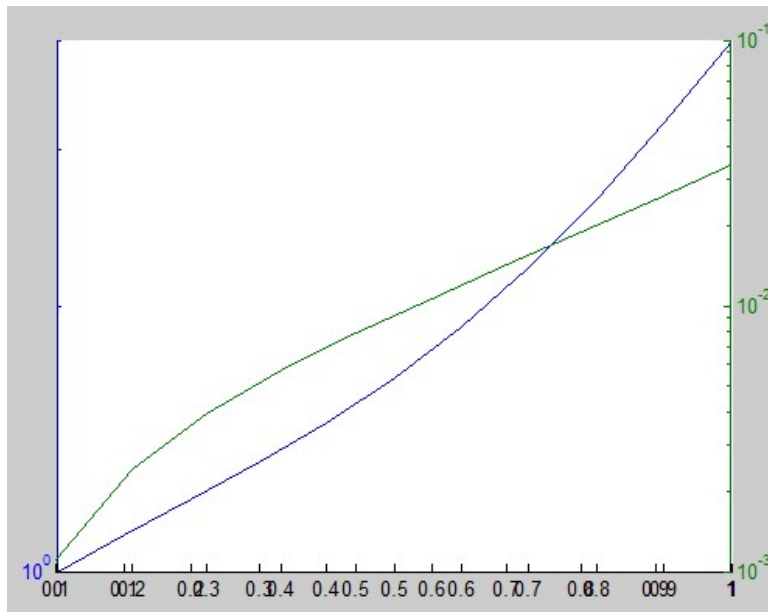
```
plotyy(N,f1,N,f3,'loglog')
```



```
>> plotyy (N,f1,N,f3,@semilogx)
```



```
>> plotyy (N,f1,N,f3,@semilogy)
```



```
>> J = [2 4 5 6 7 1; 2 6 7 5 3 2; 9 8 7 4 1 3; 7 6 5 8 9 8; 3 4 3 2 1 3]
```

```
J =
```

```

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

```
>> V = [2;4;6;5;3;7;6];
```

```
>> V= [ 2:8]
```

```
V =
```

```

2  3  4  5  6  7  8
```

```
>> imagesc(A)
```

```
>> plot(J);
```

```
>> hold on;
```

```
>> grid on;
```

```
>> imagesc(J);
```

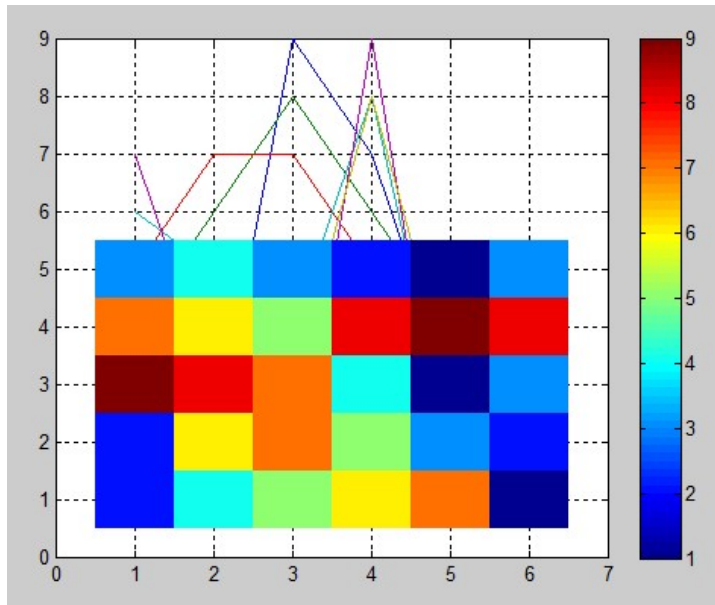
```
>> colorbar
```

```
>> imagesc(A)
```

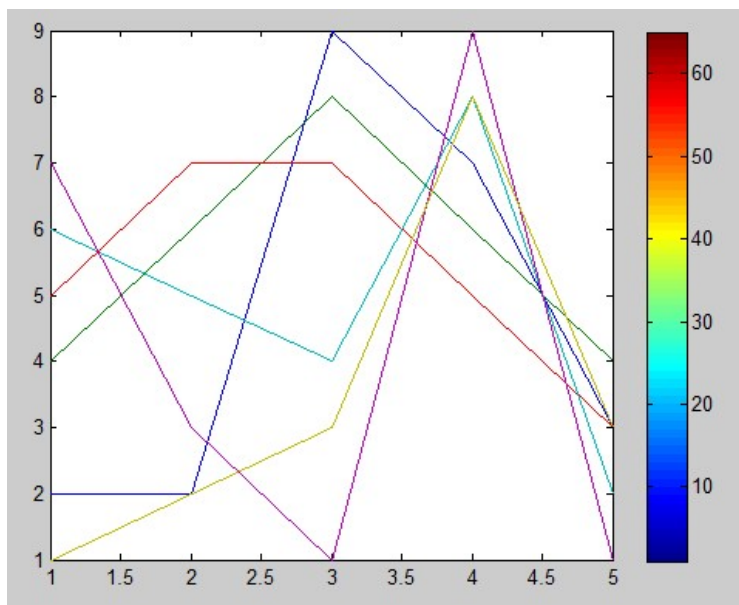
```
>> plot(J);
```

```
>> hold on;
```

```
>> grid on;
>> imagesc(J);
>> colorbar
```



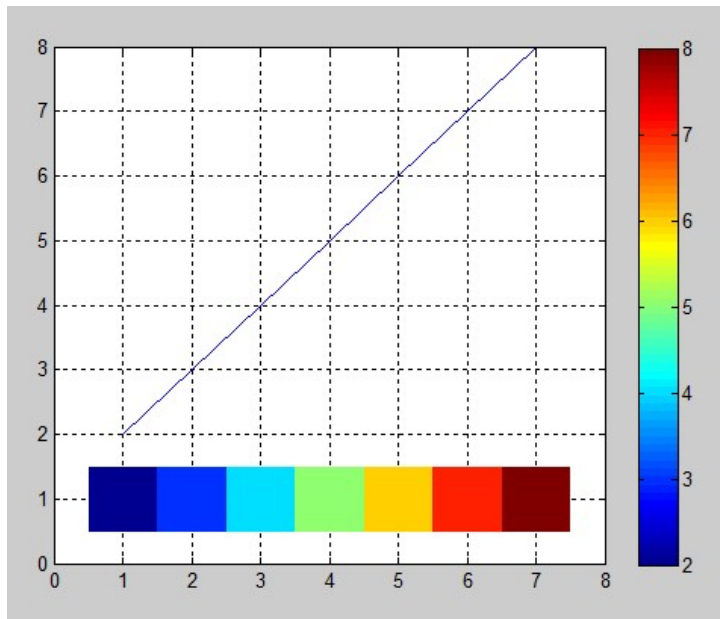
```
>> plot(J)
>> colorbar
```



```
>> plot(V)
>> hold on;
>> grid on;
```

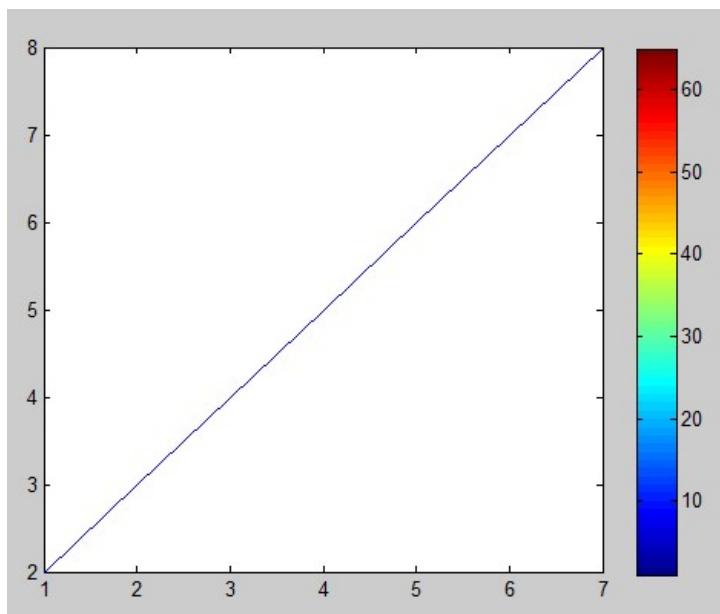
```
>> imagesc(V);
```

```
>> colorbar
```



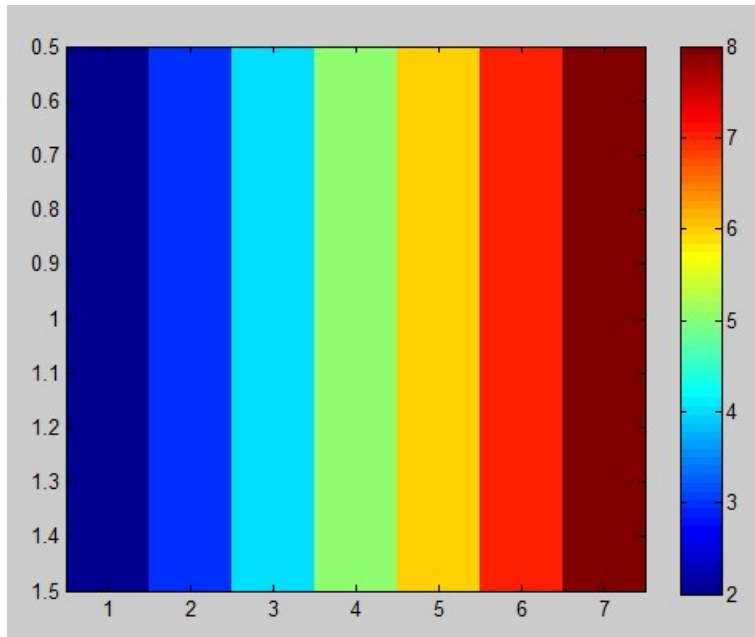
```
>> plot(V)
```

```
>> colorbar
```



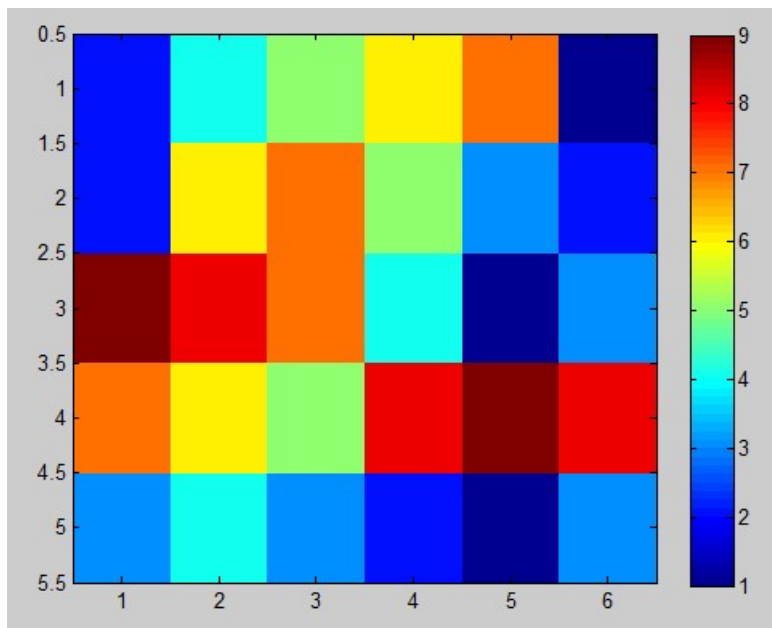
```
>> imagesc(V)
```

```
>> colorbar
```



```
>> imagesc(J)
```

```
>> colorbar
```





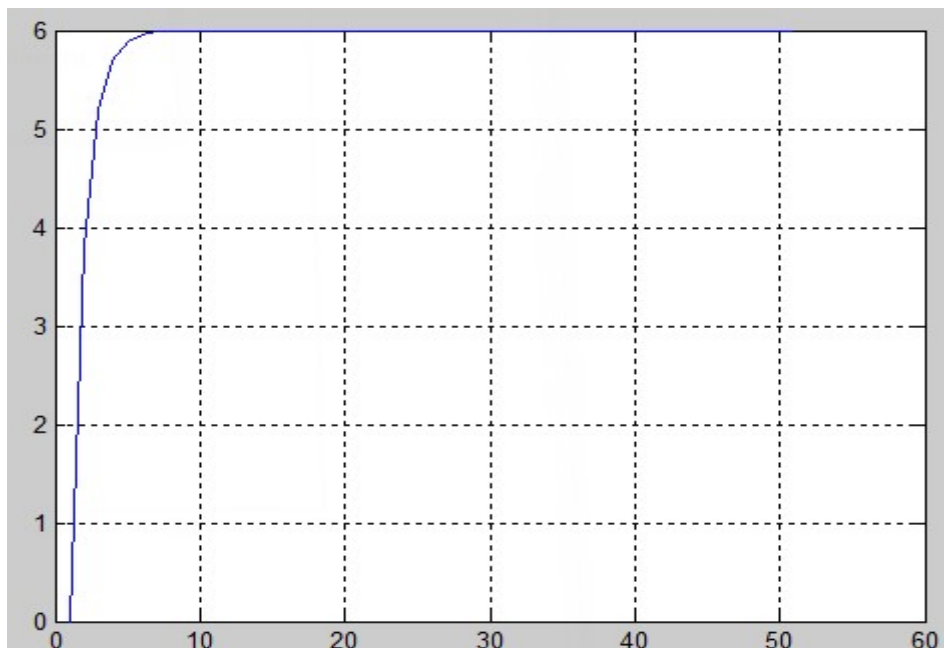
## Задание №2– двумерная графика

```
1)x=0:0.1:5;
```

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

```
>> plot(y);
```

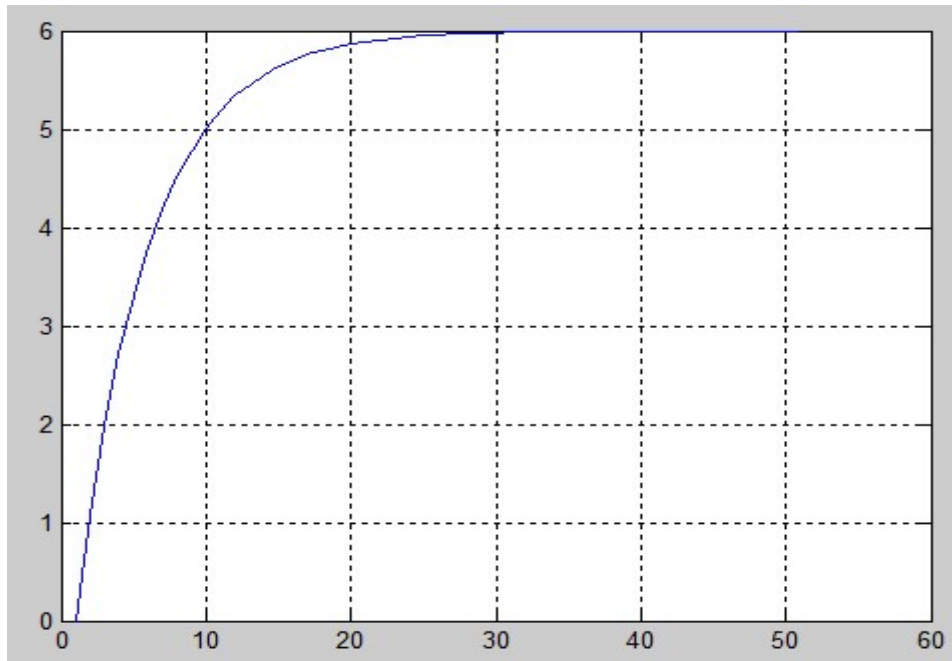
```
>> grid on
```



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

```
>> plot(y);
```

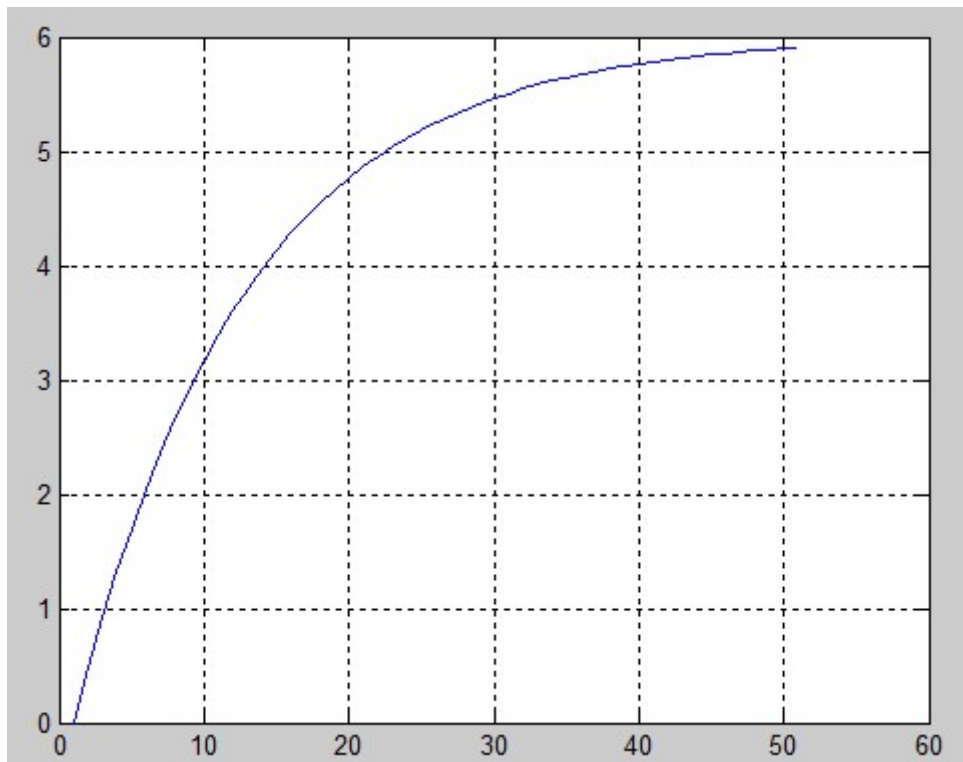
```
>> grid on
```



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

```
>> plot(y);
```

```
>> grid on
```



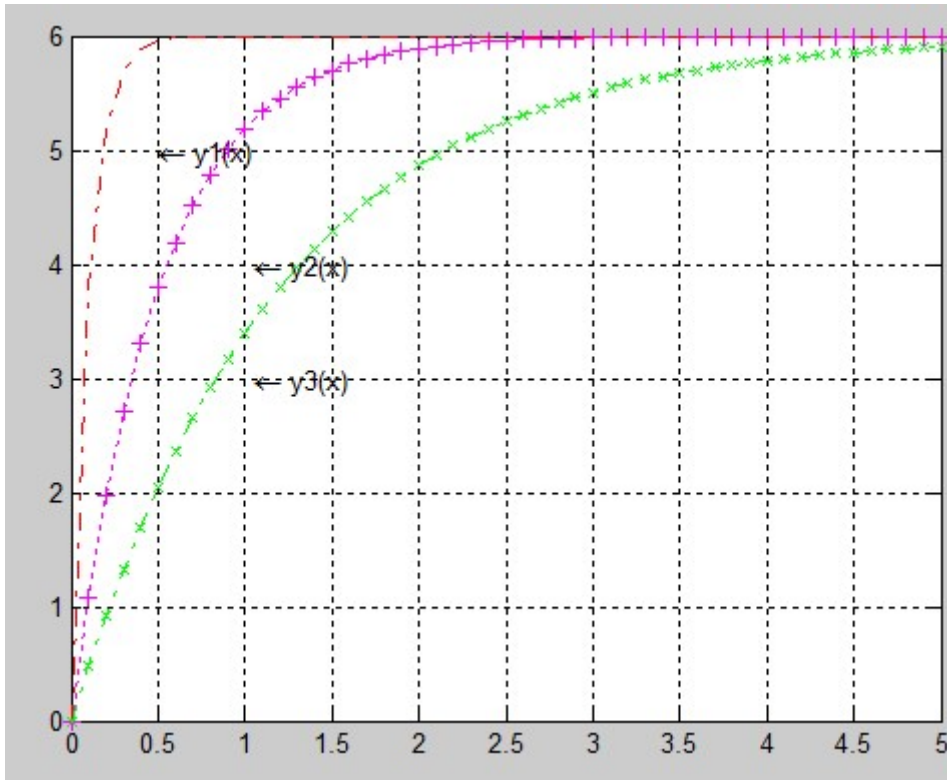
```
2) >> plot(x,y2,'m:+' );
```

```
>> 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)');

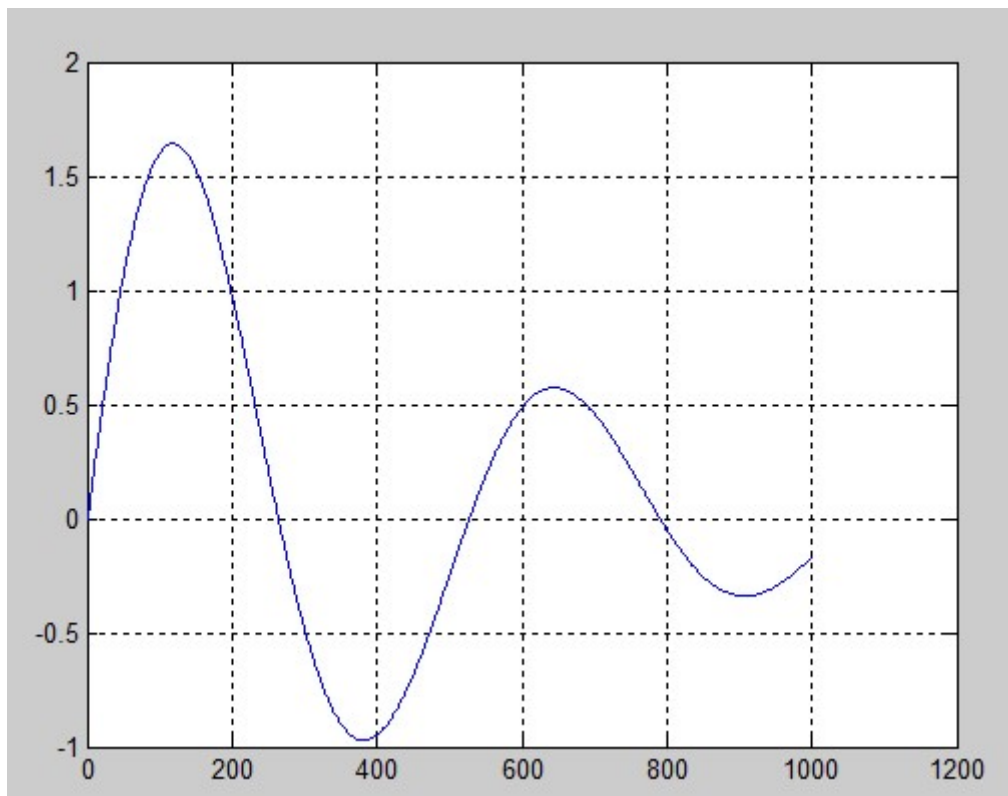
```



```

3) 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

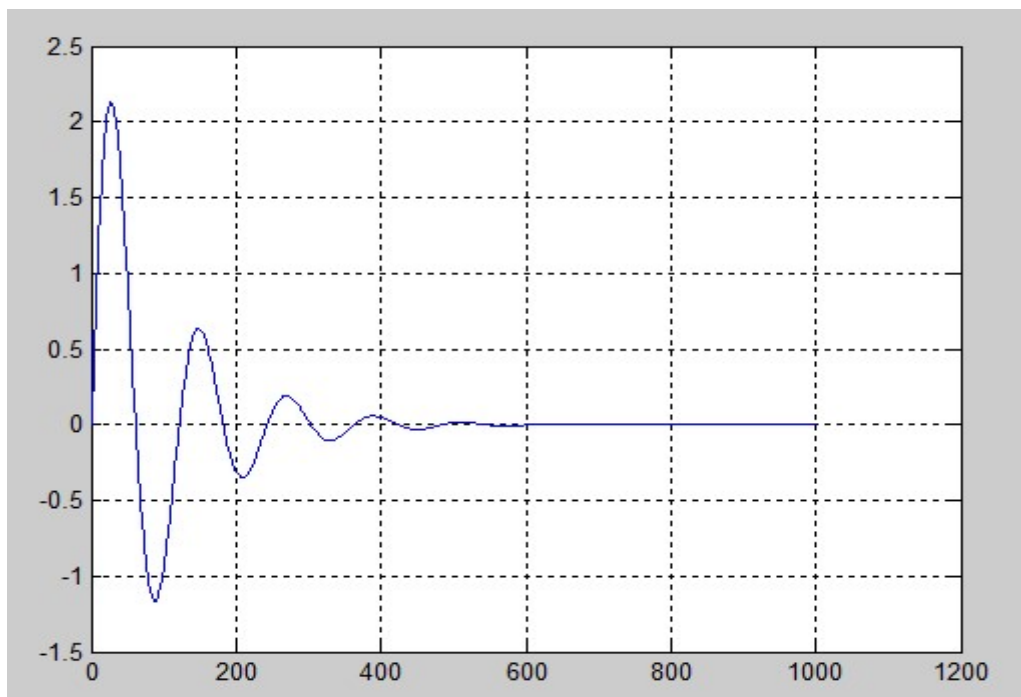
```



```
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);
```

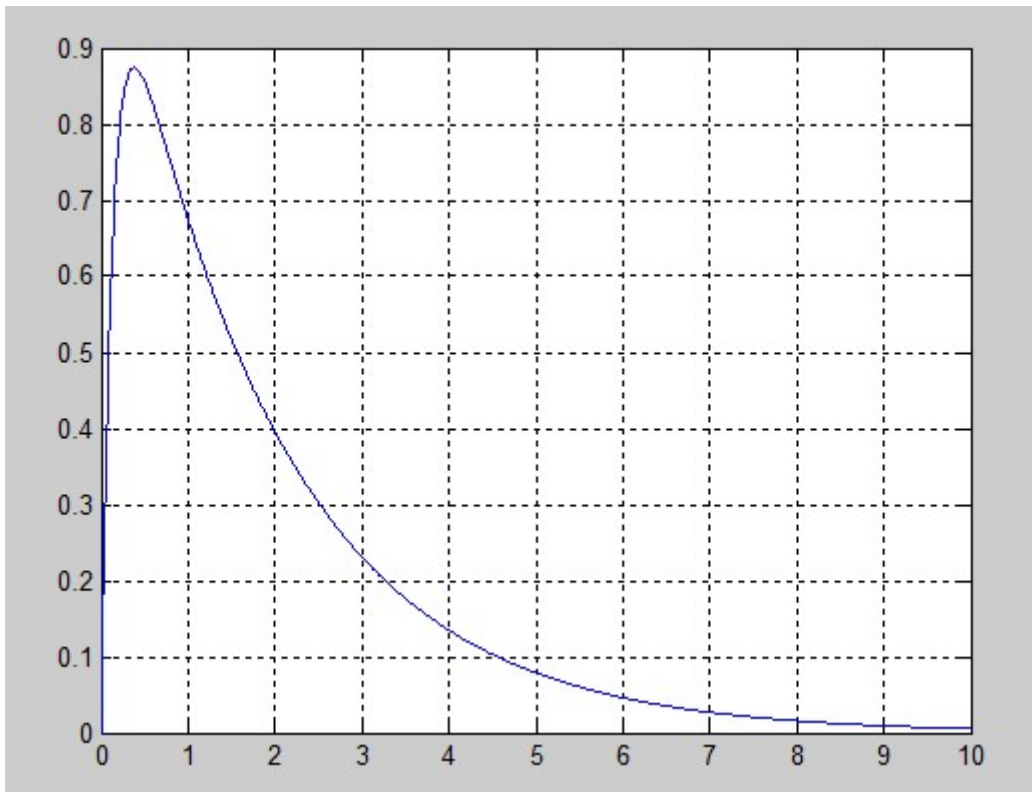
```
>> grid on
```



```
>> 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
```



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

```
>> hold on
```

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

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

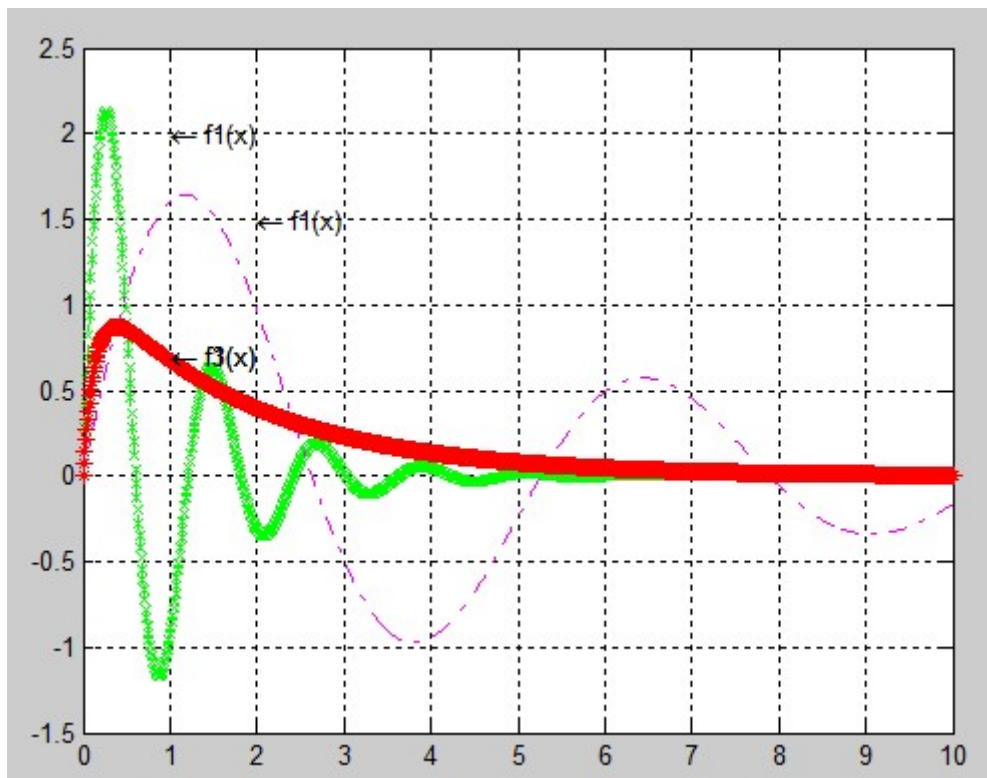
Warning: Imaginary parts of complex X and/or Y arguments ignored

```
>> 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

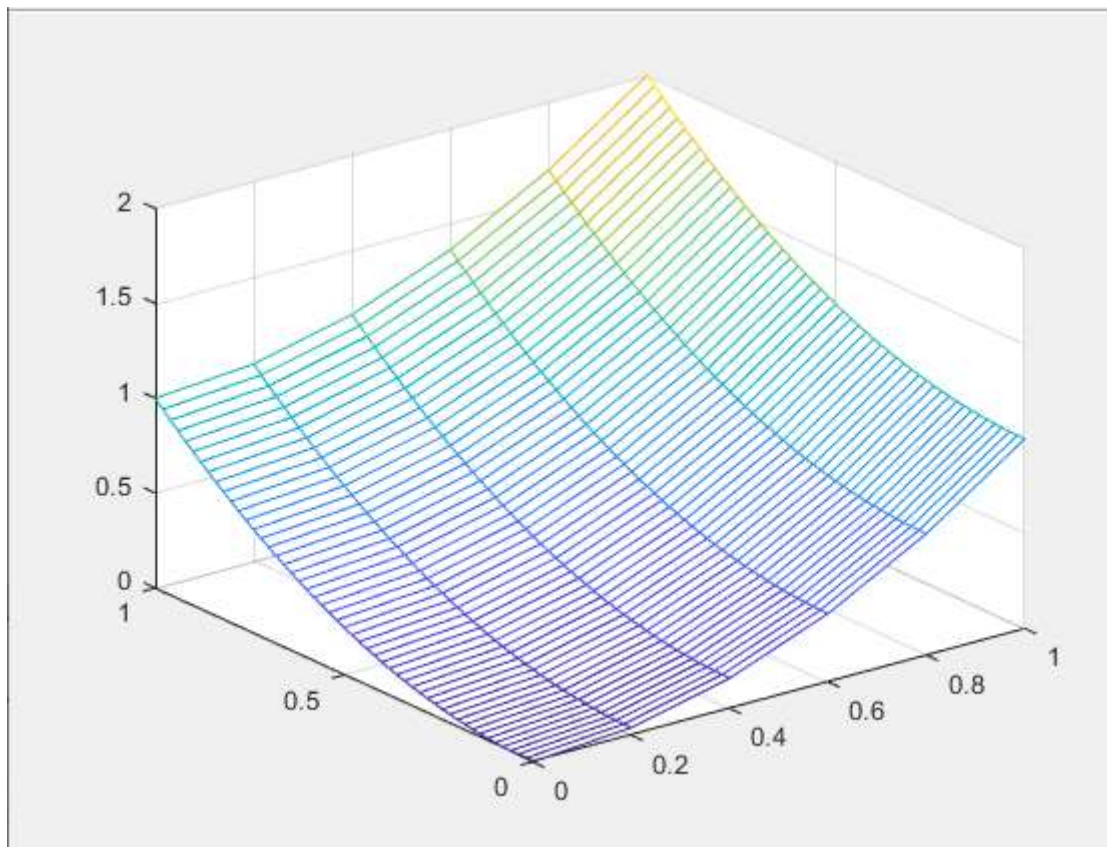
3,1

Постройте сетчатый график функции  $z = x^2 + y^2$  на области определения в виде квадрата  $x \in [0,1]$  и  $y \in [0,1]$  с шагом 0.2 и шагом 0.02.

```
>> [x,y]=meshgrid(-1:0.1:1,-1:0.1:1);
```

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

```
>> mesh(x,y,z)
```

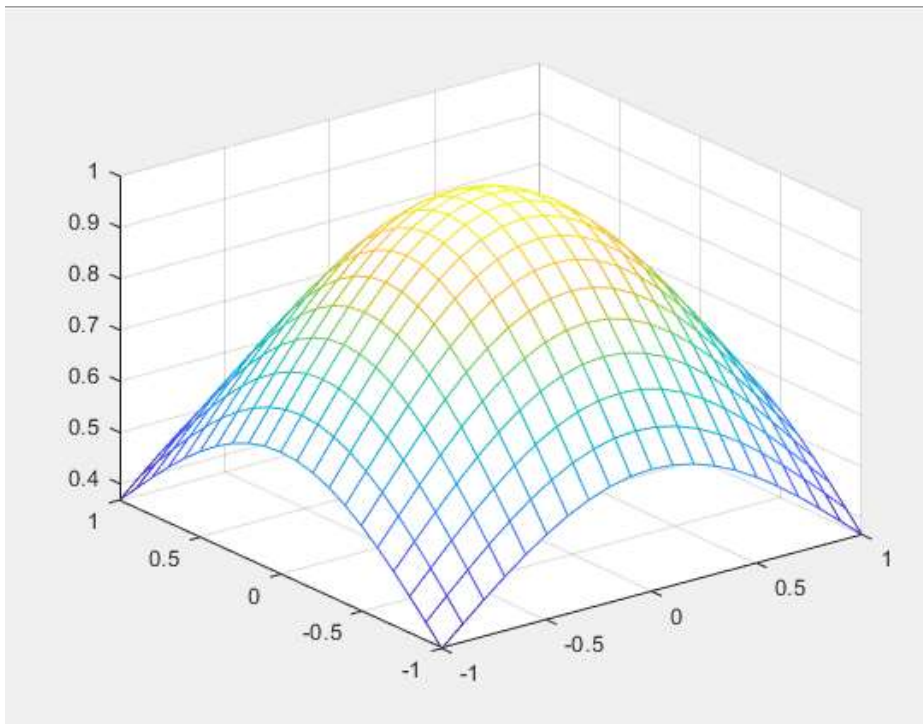


3,2

Постройте сетчатый график функции.

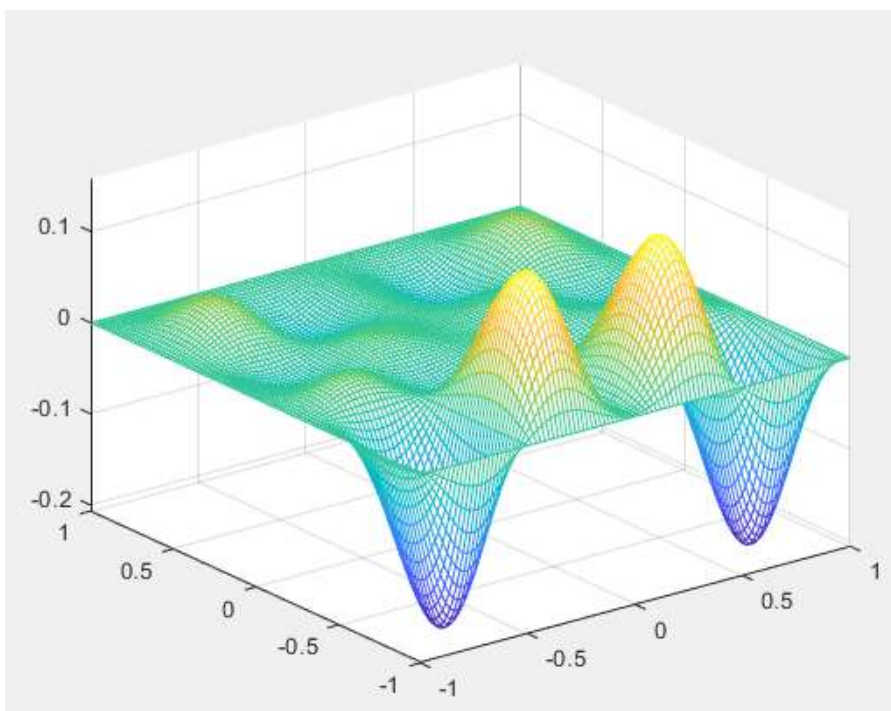
```
>> [x,y]=meshgrid(-1:0.1:1,-1:0.1:1);  
>> z=exp(-(((x-0).^2)/2)+(((y-0).^2)/2));  
>> mesh(x,y,z)
```





3,3

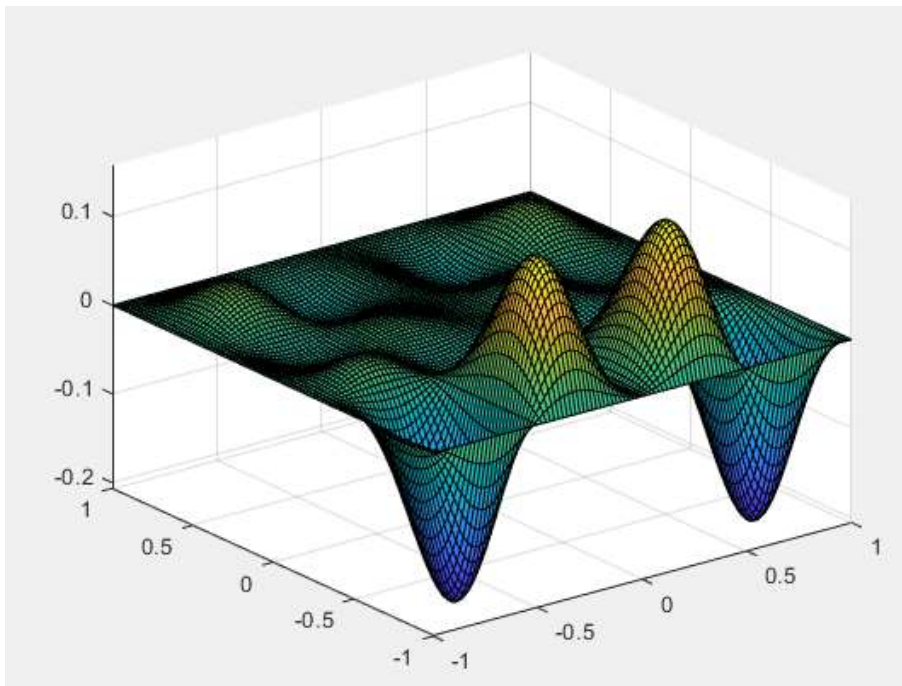
```
>> N=1;
>> [x,y]=meshgrid(-1:0.02:1, -1:0.02:1);
>> z=(N/2)*sin(2*pi*x).*cos(1.5*pi*y).*(1-x.^2).*y.*x.*(1-y);
>> mesh(x,y,z)
```



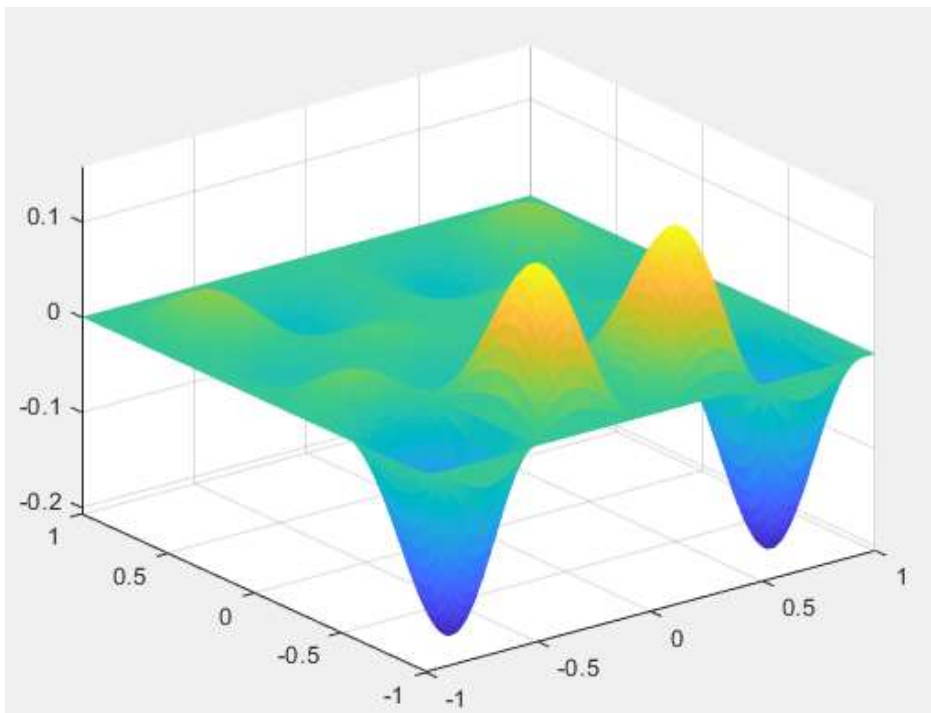
3.4

```
>> surf(x,y,z)
```

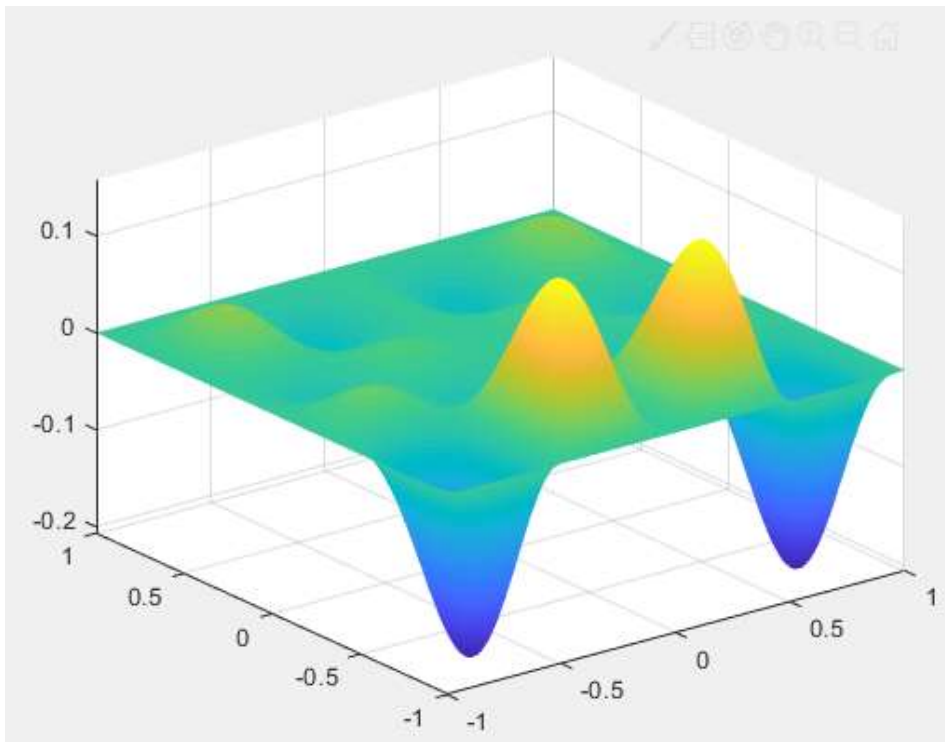




>> shading flat



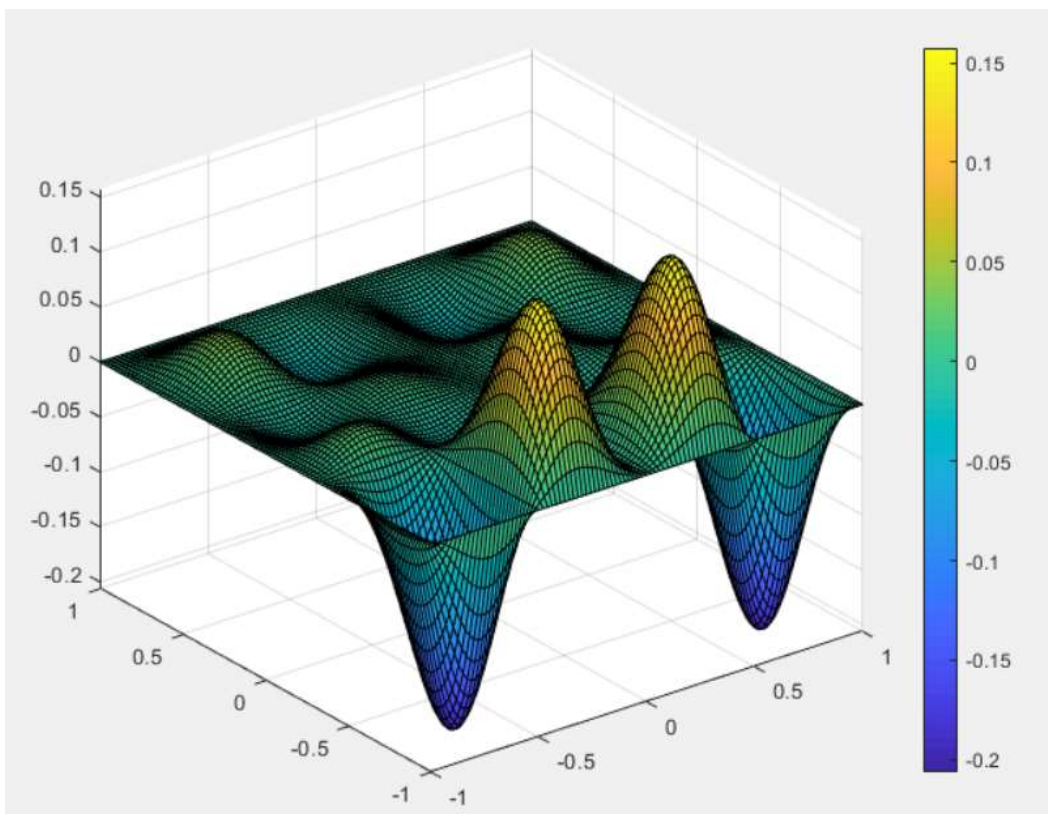
>> shading interp



3.5

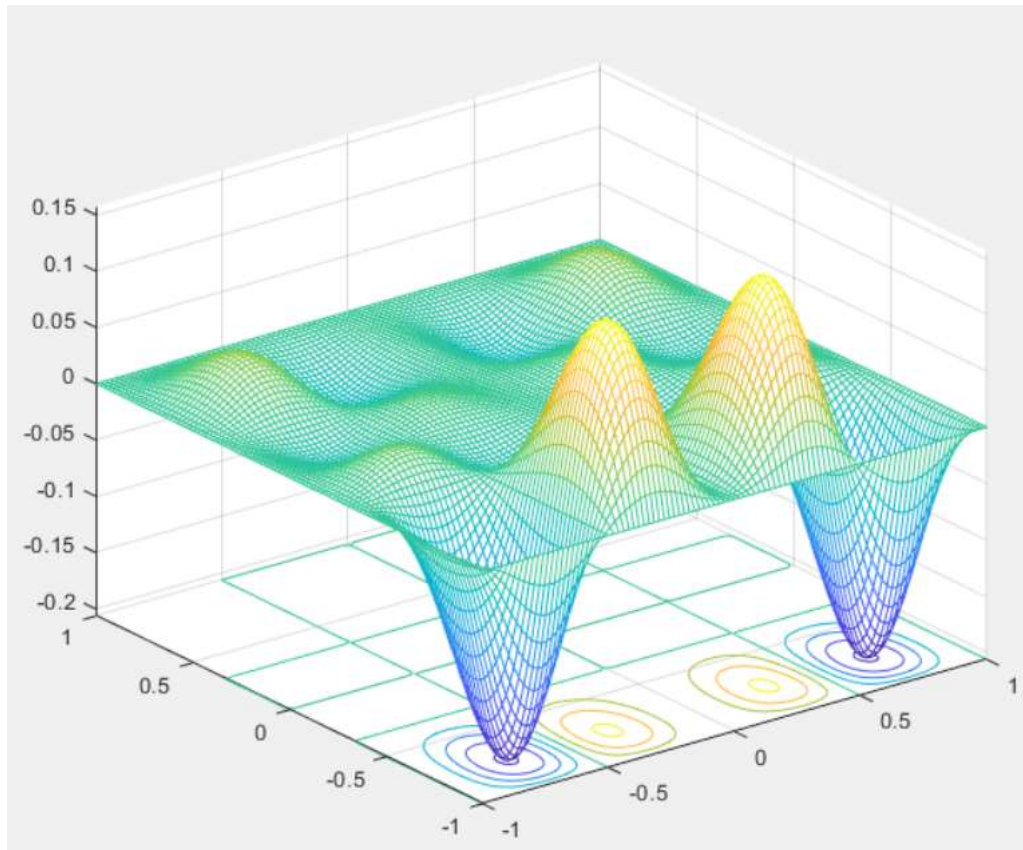
```
>> surf(x,y,z)
```

```
>> colorbar
```

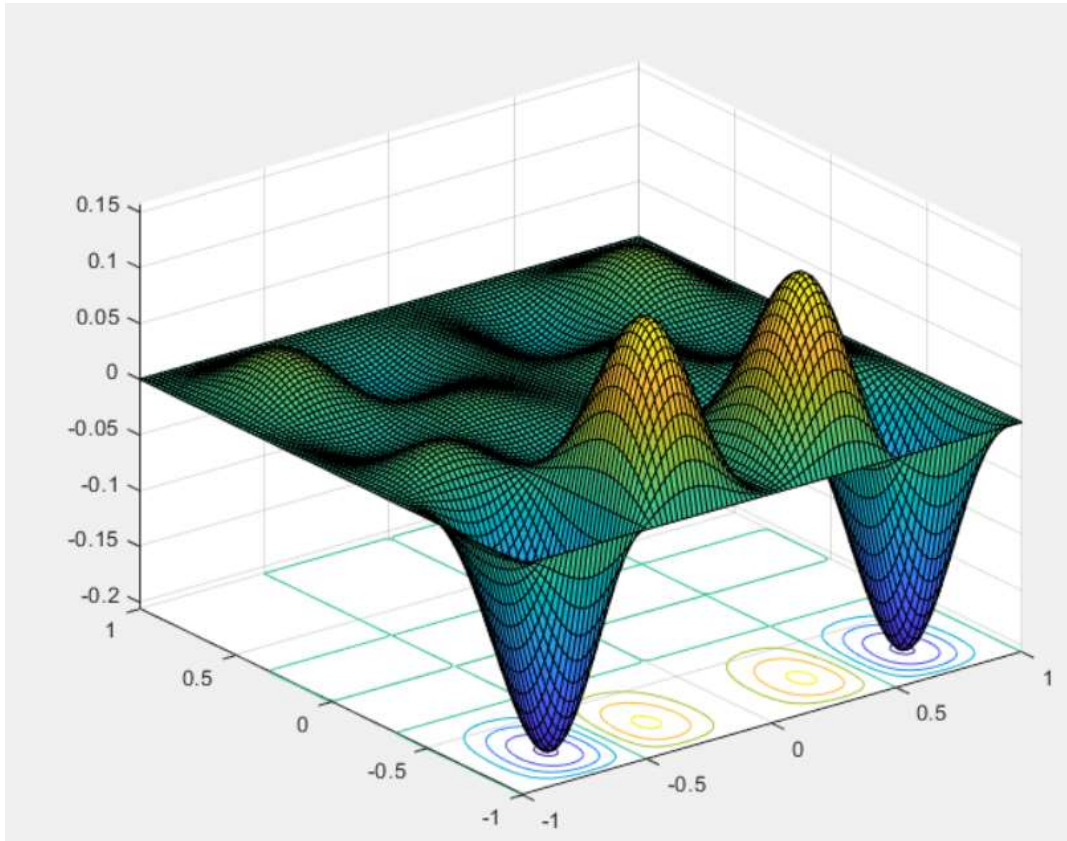


3.6

meshc

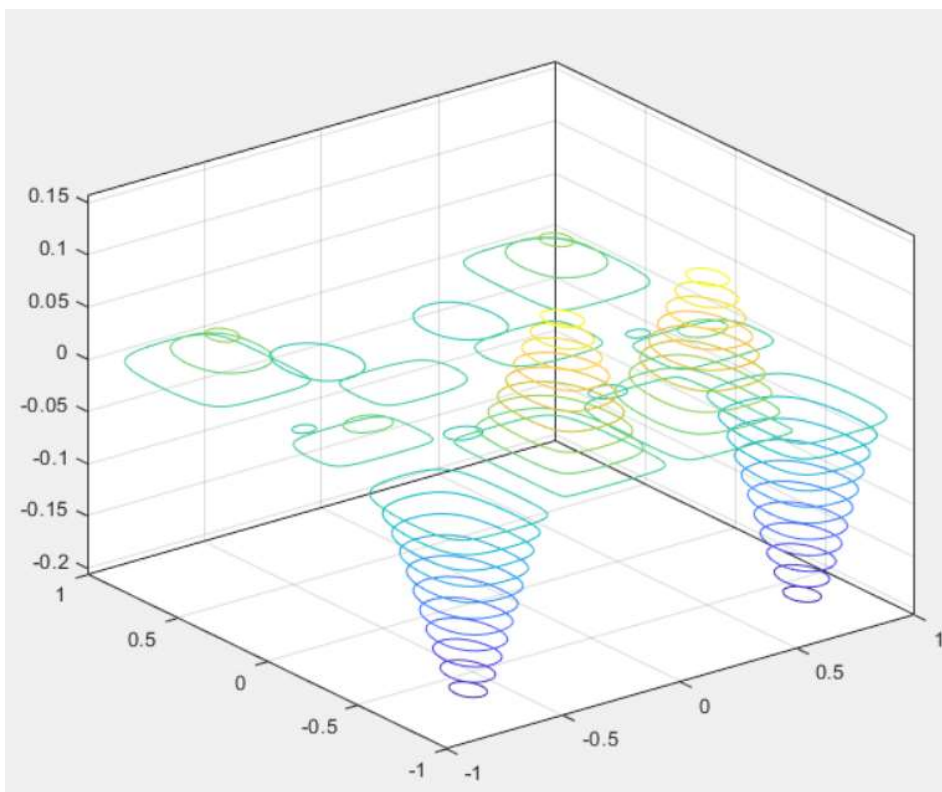


surf



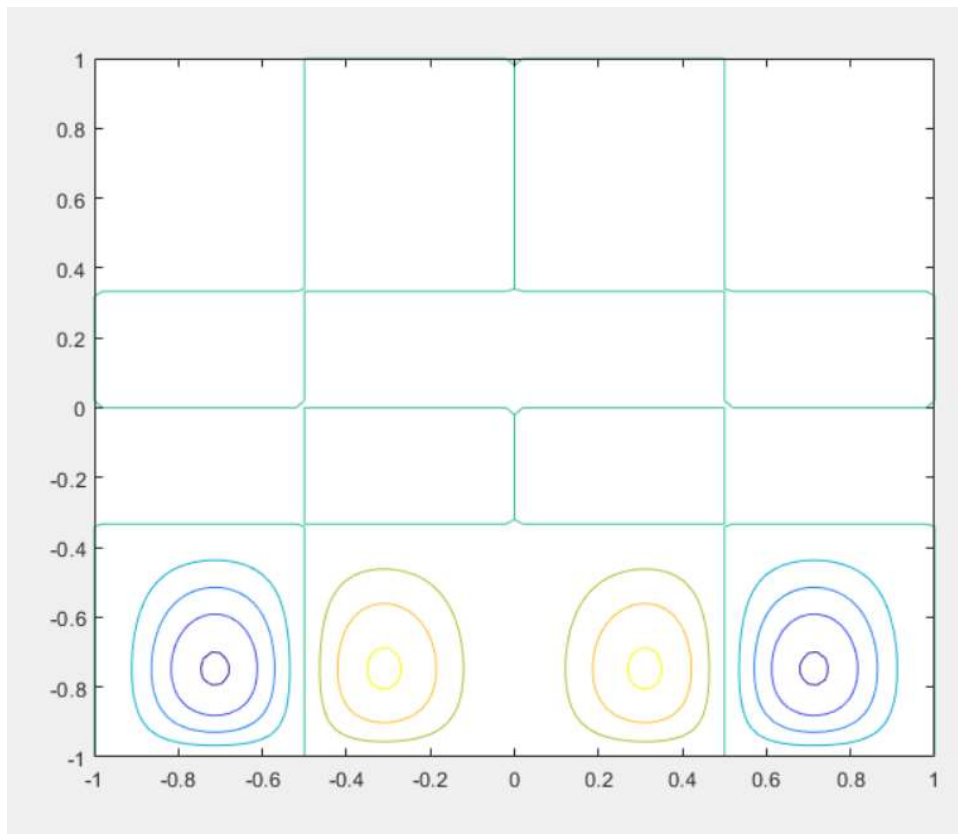
3.7

>> contour3(x,y,z,20)

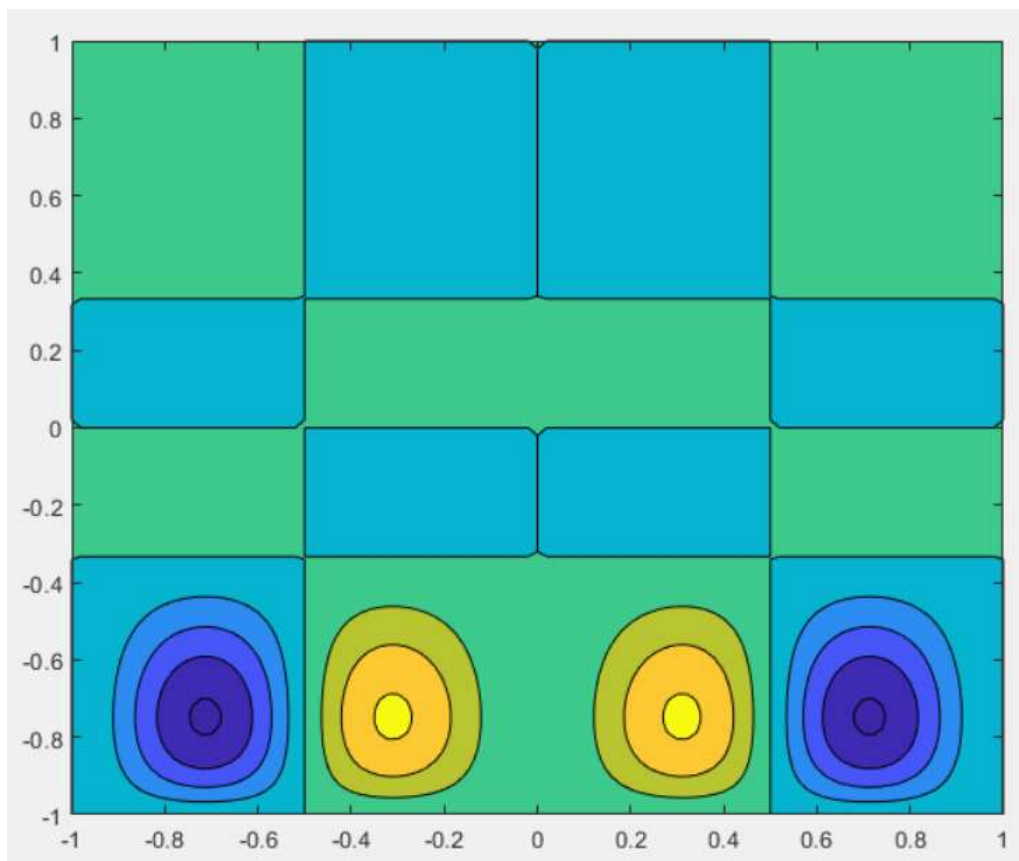


3.8

contour



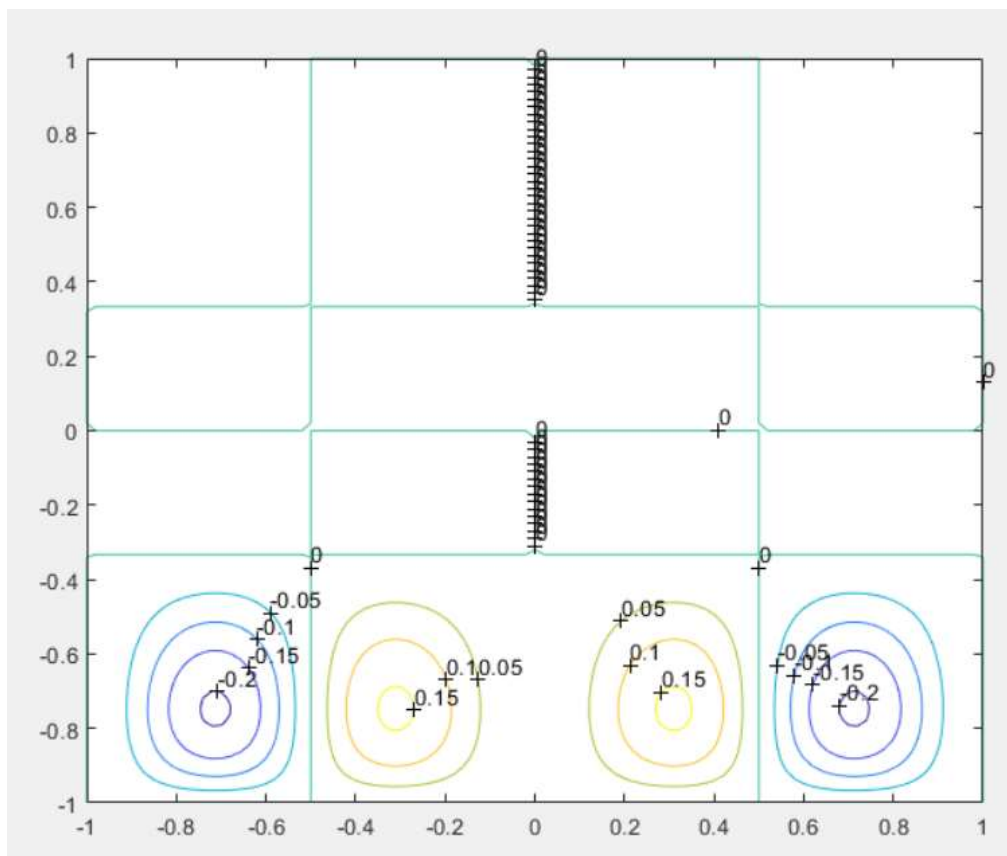
contourf





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

```
>> clabel(C)
```



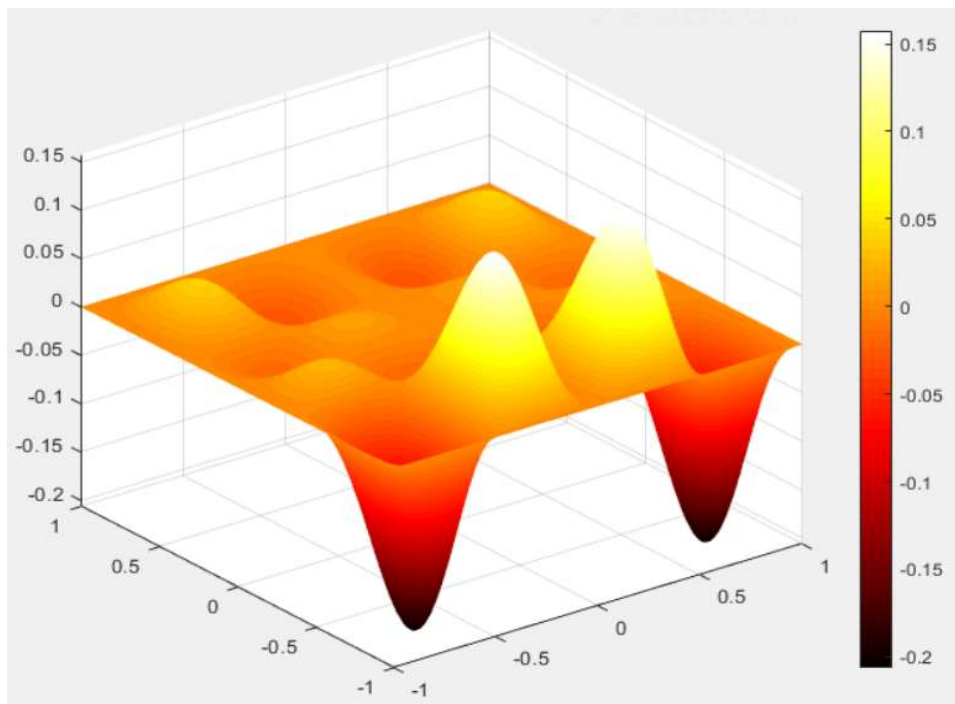
3.9. Выполните три разных цветовых оформления графика функции.

```
>> surf(x,y,z)
```

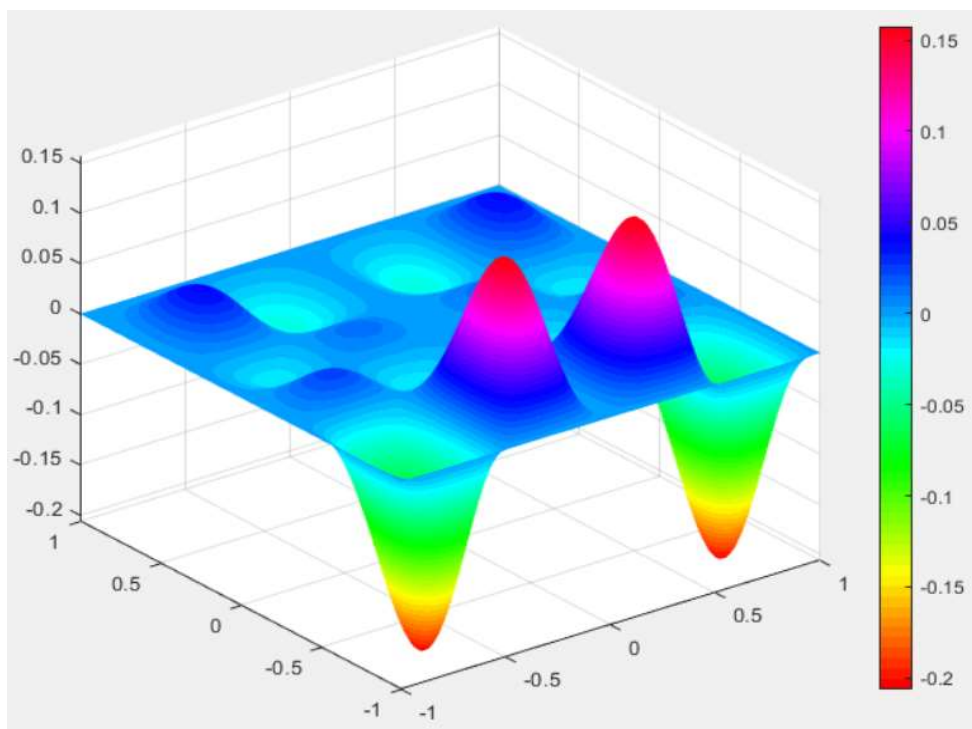
```
>> colorbar
```

```
>> shading interp
```

```
>> colormap('hot')
```

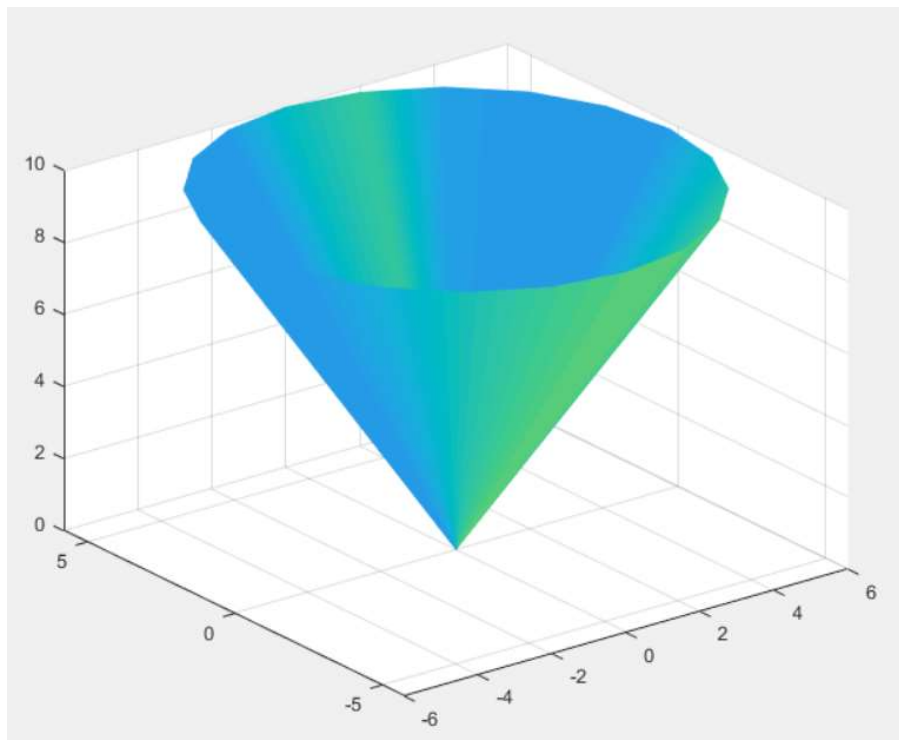


```
>> colormap('hsv')
```



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

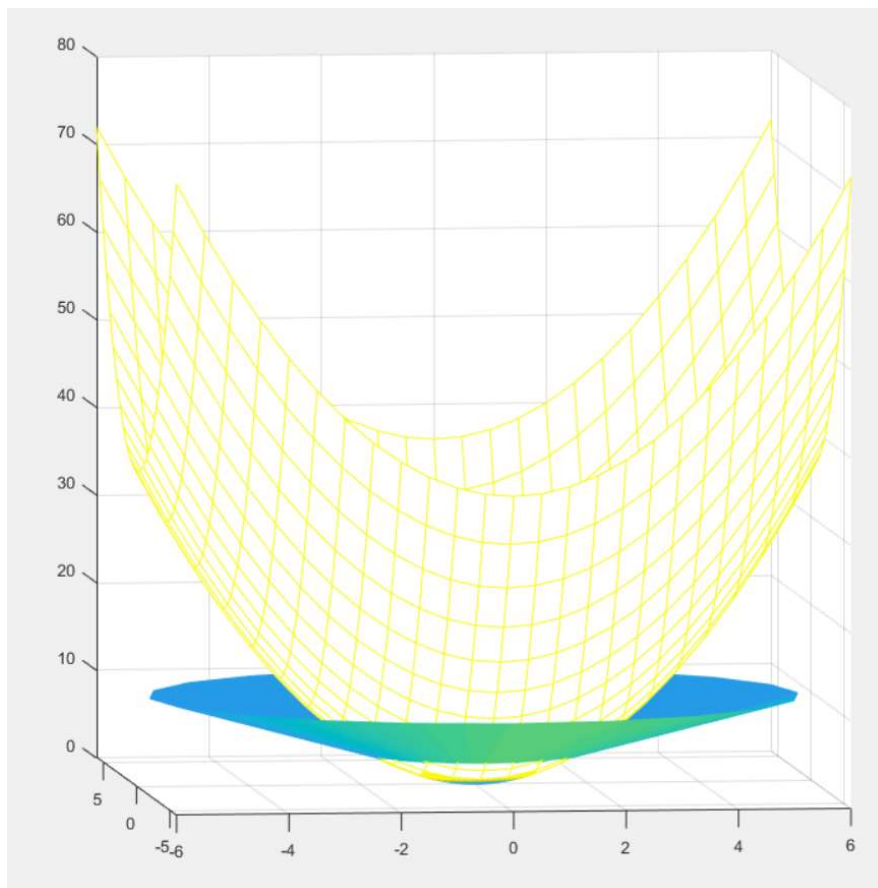
```
>> t=0:2*pi/20:2*pi;  
>> z=0:0.2:10;  
>> [t,z]=meshgrid(t,z);  
>> x=z.*cos(t)*tan(pi/6);  
>> y=z.*sin(t)*tan(pi/6);  
>> surf(x,y,z)  
>> shading interp  
>> hidden off  
>> shading interp
```



3.11. Пересеките конус плоскостью.

```
>> hold on  
>> [x,y]=meshgrid(-1:0.1:1,-1:0.1:1);  
z=x.^2+y.^2;  
mesh(x,y,z)
```





#### Задание №44. 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)
```

1)

В mu\_file:

`x = [0:0.1:7]; %задаем значения для построения графика`

`f = exp(-x); %задаем 1-ую функцию`

`subplot(1, 2, 1) %разбиваем окно графика на подокна`

`plot(x, f)%строим 1-ый график`

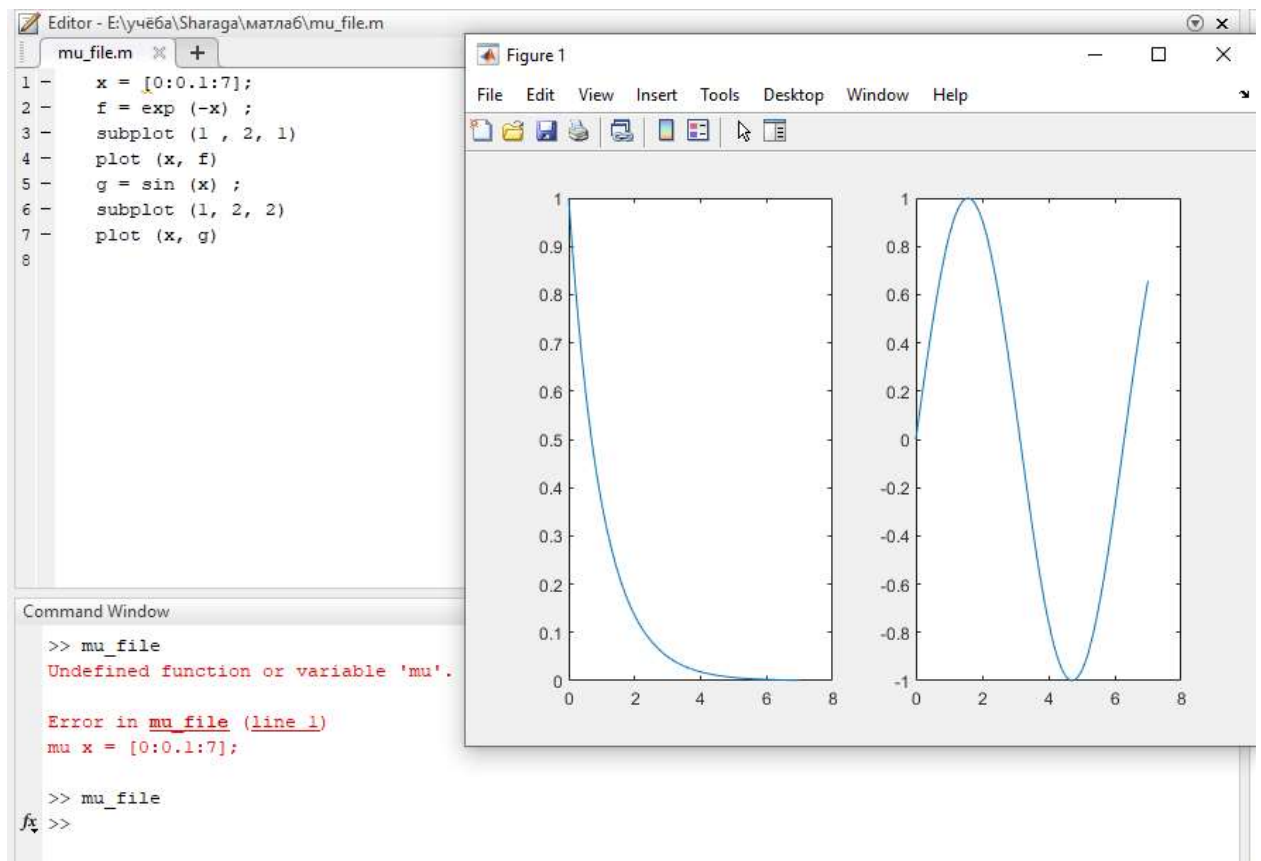
`g = sin(x); %задаем 2-ую функцию`

`subplot(1, 2, 2)%разбиваем окно графика на подокна`

`plot(x, g)%строим 2-ой график`

Пример использования:

`>>mu_file`



М-файлы позволяют сохранять множество команд программы MATLAB в одном файле, а затем запускать их одной командой.

4.2. Постройте график файл-функции `myfun` с помощью команд `plot` и `fplot`. В файле-функции `myfun` должно рассчитываться следующее выражение:

$$y = k \left[ t - T \left( 1 - e^{-\frac{t}{T}} \right) \right],$$

где  $k = N$ ,  $N$  - номер варианта по списку,  $T = \frac{N}{50}$ . В качестве входного аргумента файл-функция должна принимать вектор  $t$ , который должен быть задан на интервале  $[0, 5]$  с шагом 0.01. Таким образом, М-файл функции должен иметь следующее «заглавие»:

```
function f = myfun(t)
```

В командной строке MATLAB вызов этой функции происходит следующим образом:

```
>> y = myfun(t)
```

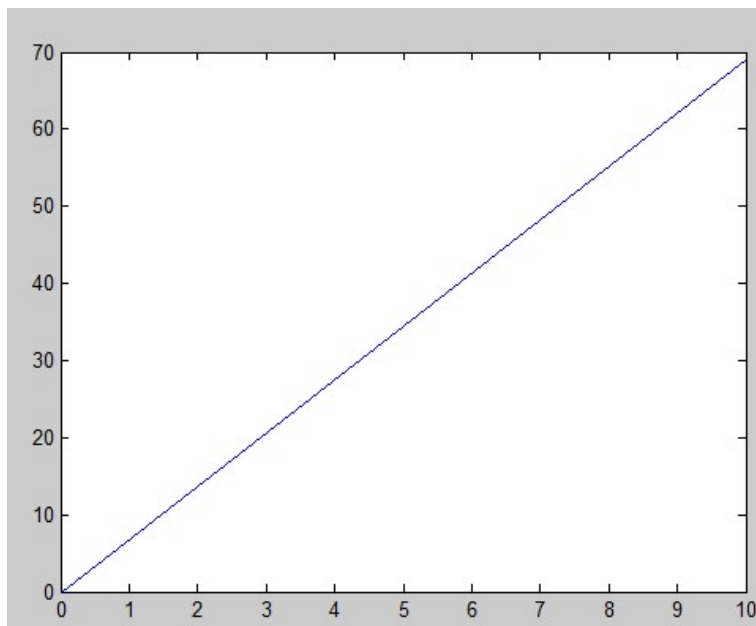
В результате чего создастся вектор-строка  $y$  с тем же числом элементов, что и вектор  $t$ .

2) В `myfun`:

```
Function f= myfun(t);
N=1;
k = N;
T=N/50;
y=k*(t-T*(1-exp(-t/T)));
plot(t,y)
%fplot(t,y)
```

Пример использования:

```
>> myfun([0:0.1;10])
```



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.

```
function [x1, x2] =root(a, b, c) ;  
    D = b^2 - 4*a*c;  
    if D<0  
        disp(' the result is complex')  
    else  
        x1=-b+sqrt(D) / (2*a)  
        x2=-b-sqrt(D) / (2*a)  
    end
```

Пример использования:

```
>> [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.

```
function N=Nod(a, b)  
    while (a~=0 && b~=0)  
        if a>b  
            a=mod(a, b) ;  
        else  
            b=mod(b, a) ;  
        end;  
    end;  
    N=a+b;
```

Пример использования:

```
>> N= Nod (45,3)
```

```
N =3
```

4.5. В мобильной робототехнике в задачах планирования движения робота в среде широкое применение находят кривые Безье. Благодаря своей непрерывности и гладкости кривые Безье используются как идеальные траектории, которые должен отслеживать мобильный робот – осуществлять перемещение по ним. Для построения кривых Безье используются опорные точки и параметр  $t$ , который изменяется в диапазоне  $[0, 1]$ . Рассмотрим **кубические** кривые Безье. Это параметрические кривые, которые задаются следующим выражением:

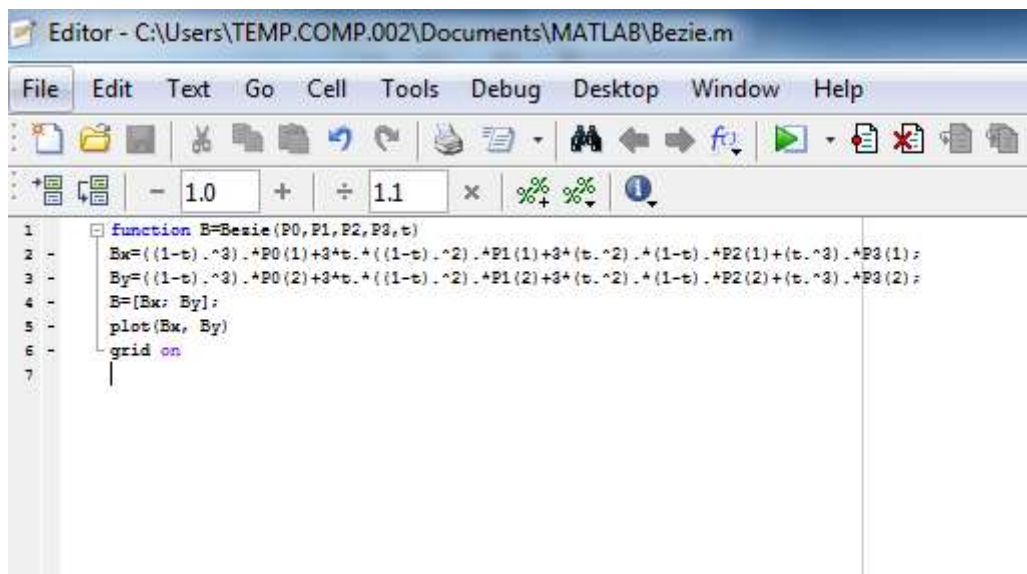
$$\mathbf{B}(t) = (1-t)^3 \mathbf{P}_0 + 3t(1-t)^2 \mathbf{P}_1 + 3t^2(1-t) \mathbf{P}_2 + t^3 \mathbf{P}_3, \quad t \in [0, 1],$$

где  $\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3$  - опорные точки, которые определяют форму кривой: кривая берёт начало из точки  $\mathbf{P}_0$  и заканчивается в точке  $\mathbf{P}_3$ .

Напишите файл-функцию, которая в качестве входных аргументов принимает четыре опорные точки и шаг изменения параметра  $t$ , а возвращает координаты точек кривой Безье ( $\mathbf{B}(t)$ ) и строит эту кривую:

```
>> B = Bezie(P0,P1,P2,P3,step)
```

В отчёте приведите не менее 5 графиков кривых Безье для различных комбинаций опорных точек  $\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3$ .

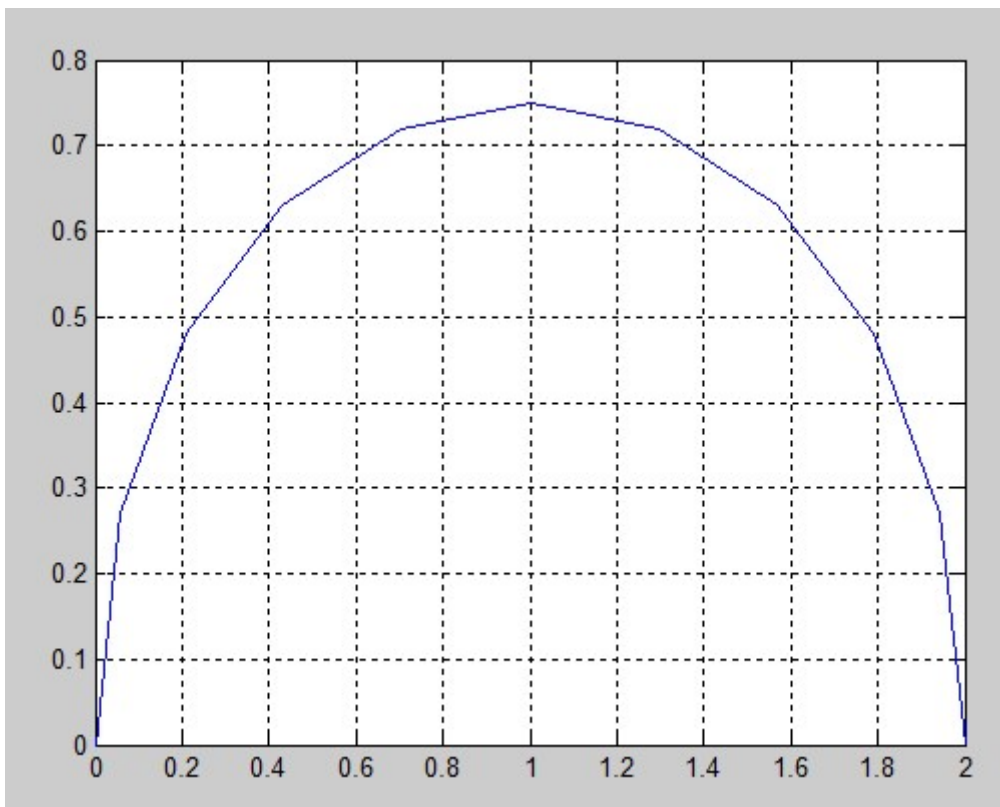


```
Editor - C:\Users\TEMP.COMP.002\Documents\MATLAB\Bezie.m
File Edit Text Go Cell Tools Debug Desktop Window Help
1 function B=Bezie(P0,P1,P2,P3,t)
2     Bx=((1-t).^3).*P0(1)+3*t.*((1-t).^2).*P1(1)+3*t.^2.*(1-t).*P2(1)+(t.^3).*P3(1);
3     By=((1-t).^3).*P0(2)+3*t.*((1-t).^2).*P1(2)+3*t.^2.*(1-t).*P2(2)+(t.^3).*P3(2);
4     B=[Bx; By];
5     plot(Bx, By)
6     grid on
7
```

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

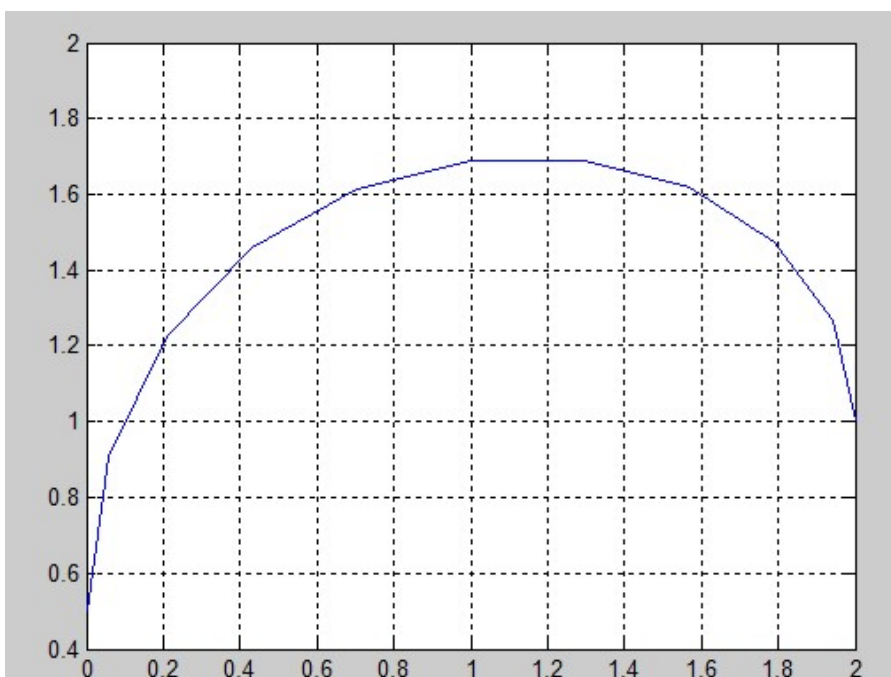
```
>> P0=[0.0, 0.0]; P1=[0.0, 1]; P2=[2, 1]; P3=[2, 0];
```

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



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

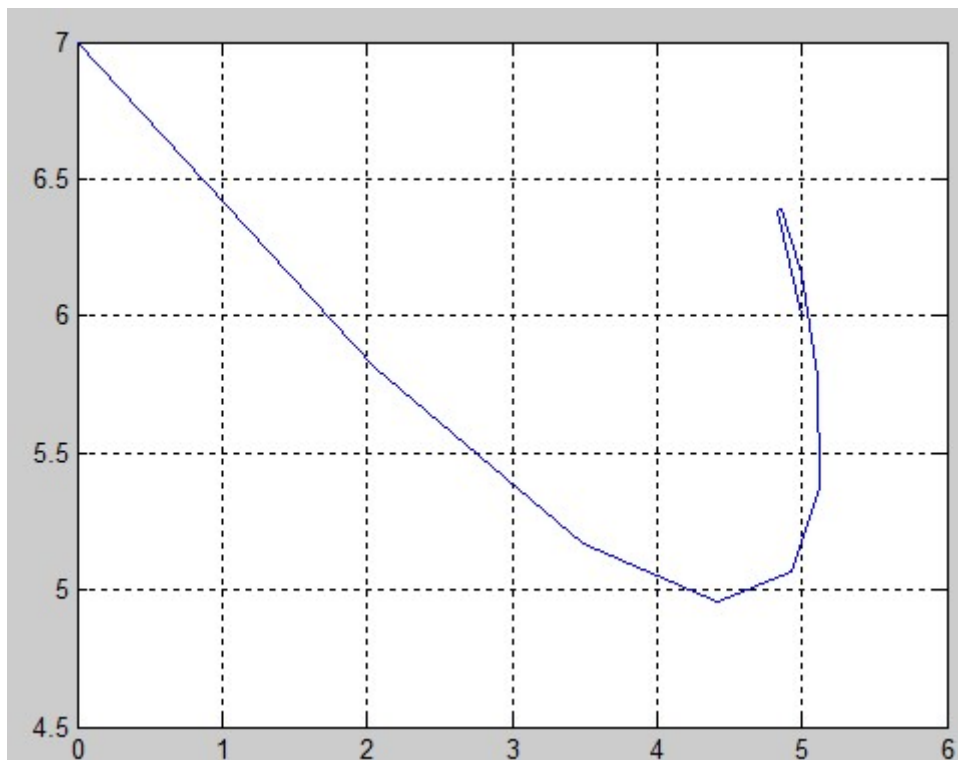
```
>> B=Bezier(P0,P1,P2,P3,t);
```



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

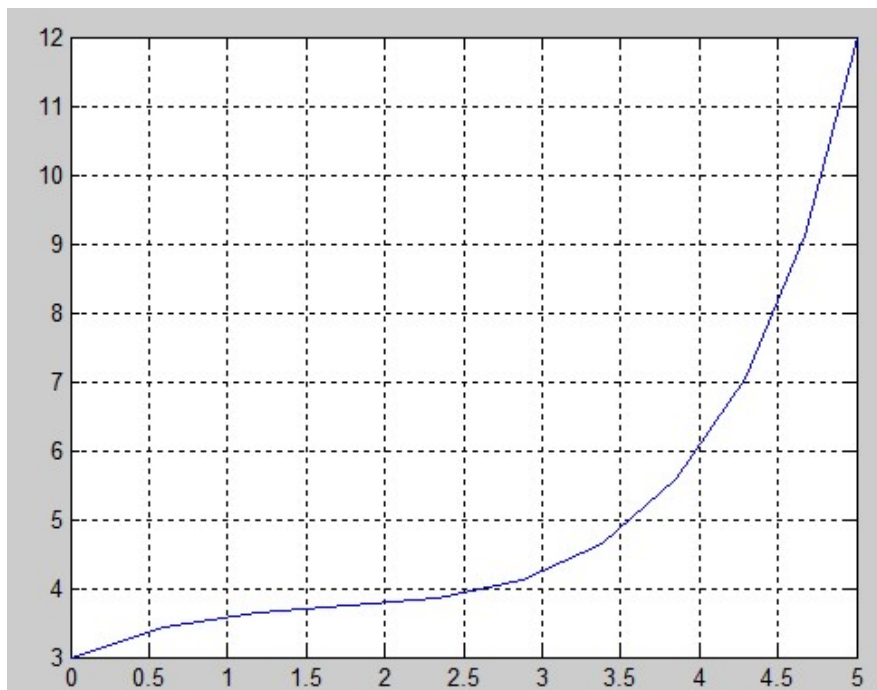
```
>> B=Bezier(P0,P1,P2,P3,t);
```





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

```
>> B=Bezier(P0,P1,P2,P3,t);
```



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

```
>> B=Bezier(P0,P1,P2,P3,t);
```

