ГРАФИЧНИ ФУНКЦИИ НА MATLAB

dimitrova@tu-sofia.bg
pct.tu-sofia.bg/dd/pik3

Графика в MatLab

- Предназначение – визуализация на данни
- Избор на графика
  - в зависимост от природата на данните
  - целта на изобразяването
- Типове
  - двумерна
    - линейна
    - бар
    - хистограма
    - кръгова
  - тримерна
    - повърхнини
    - мрежи
    - сечения

Програмни средства за графика

- Средства за изчертаване
  - интерктивни графични средства *Figure, plottools*
  - команди
  - комбинация от двата метода
- Средства за анализиране
  - анотиране
    - локализиране на точки
    - апроксимиране
    - маркиране на подмножества от данните
    - коментиране
- Средства за форматиране
  - обекти
    - компоненти

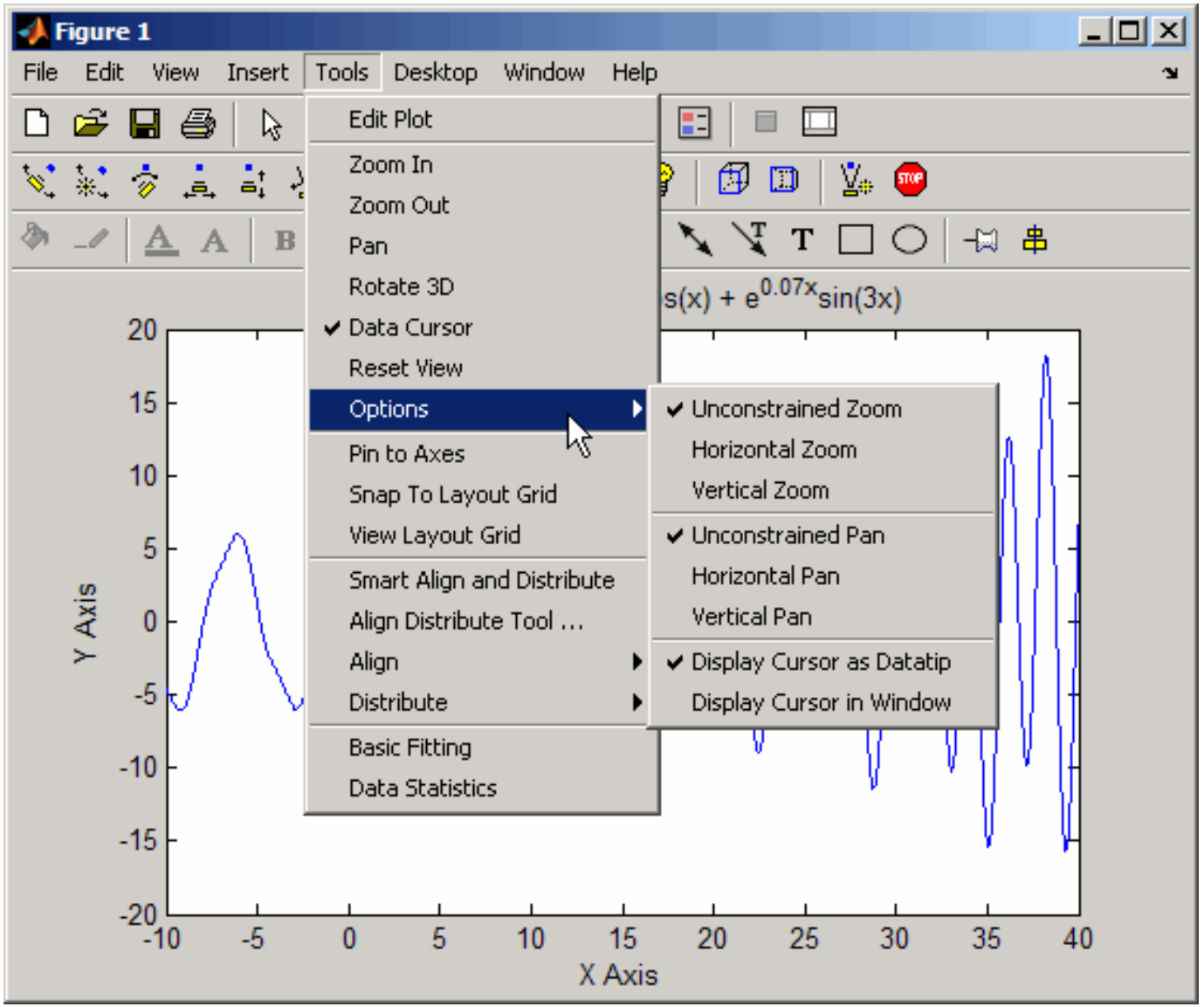
      свойства

      изглед

      поведение

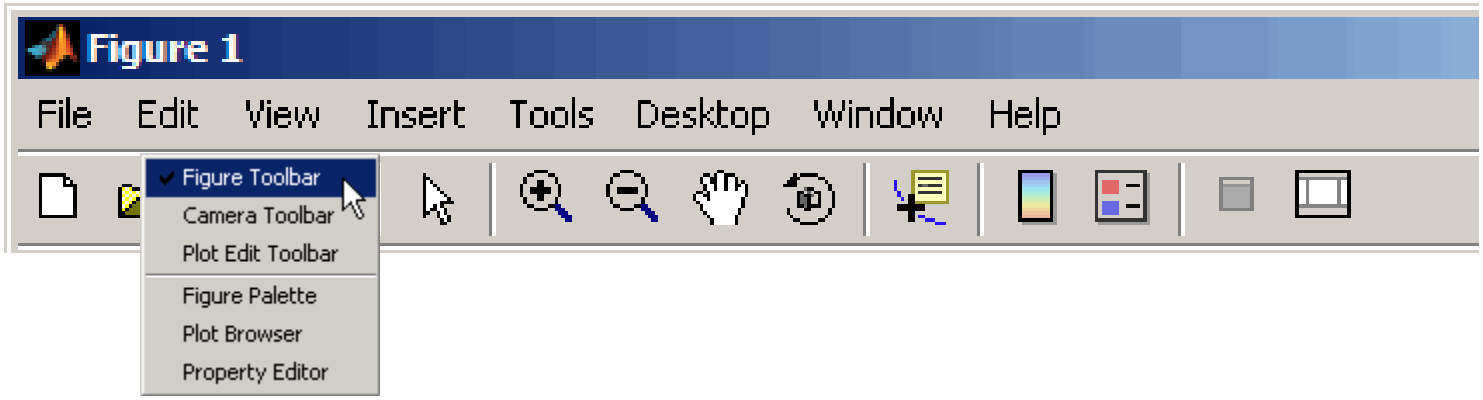
ИНТЕРАКТИВНА ГРАФИЧНА СРЕДА

Компоненти на графичната среда

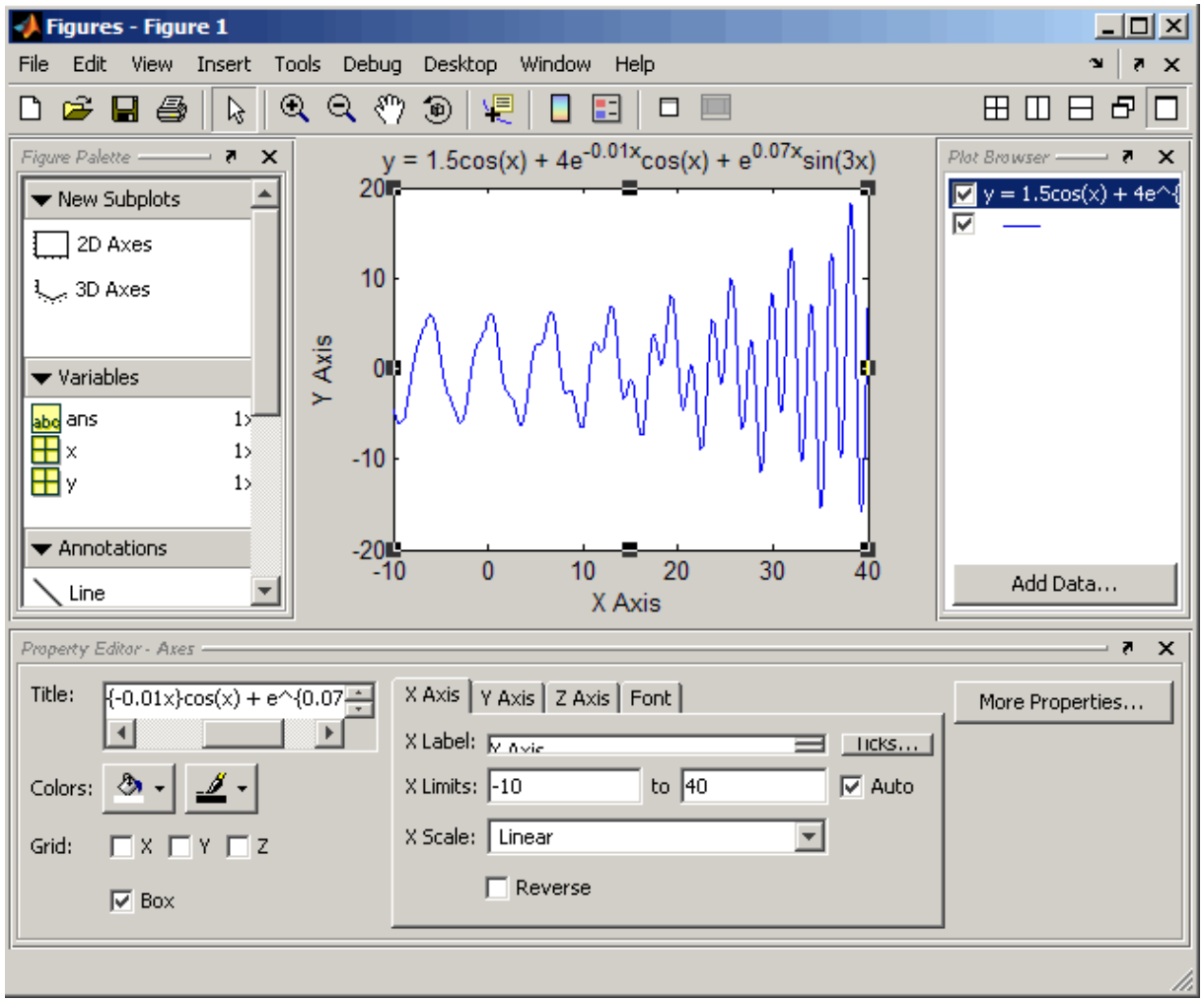


Чертожни средства

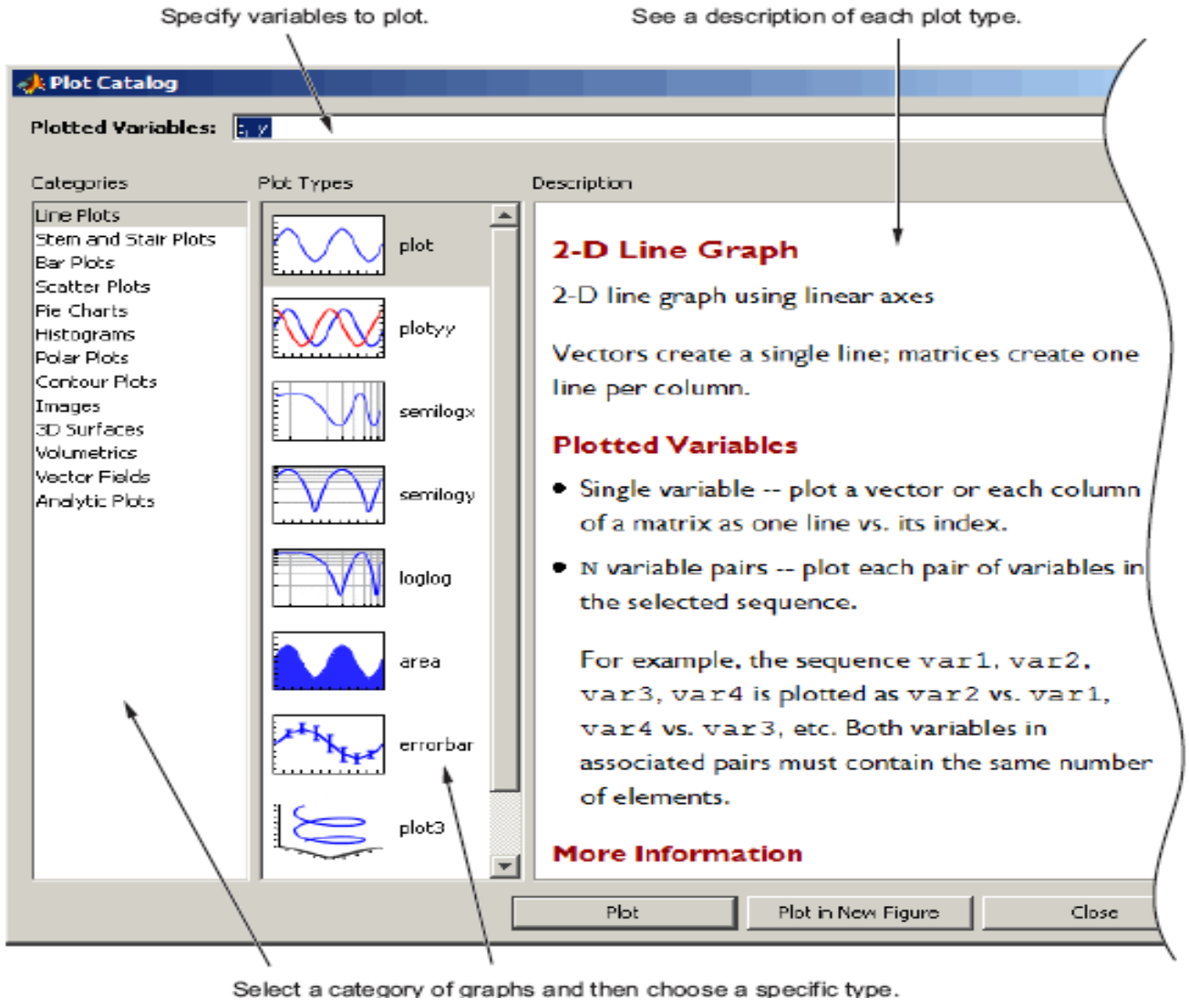
- **Предназначение**
  - **избор на тип графика**
  - **настройка на свойствата на графични обекти**
  - **анотиране на графика с текст, стрелки и др. символо**
  - **наслагване на графики**
  - **добавяне и премахване на данни в графика**
- **Достъп до чертожните средства**
  - plottools
  - figurepalette
  - plotbrowser
  - propertyeditor



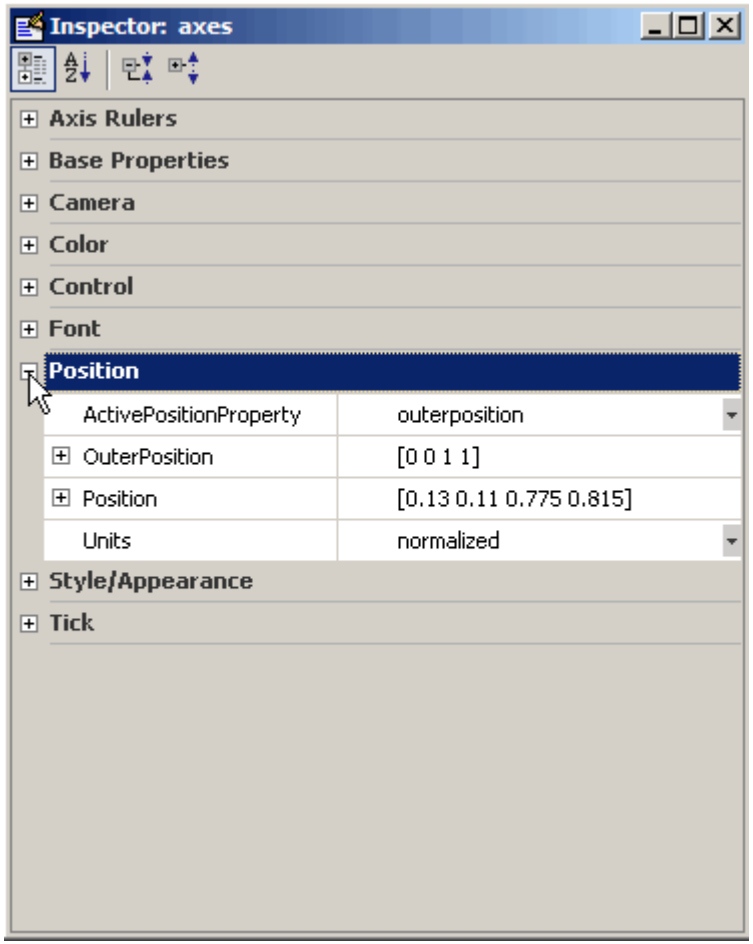
Чертожни средства

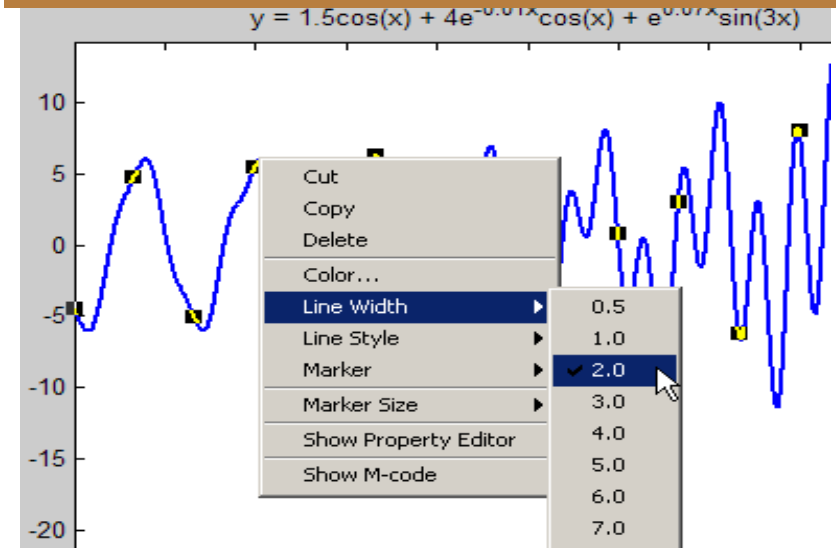


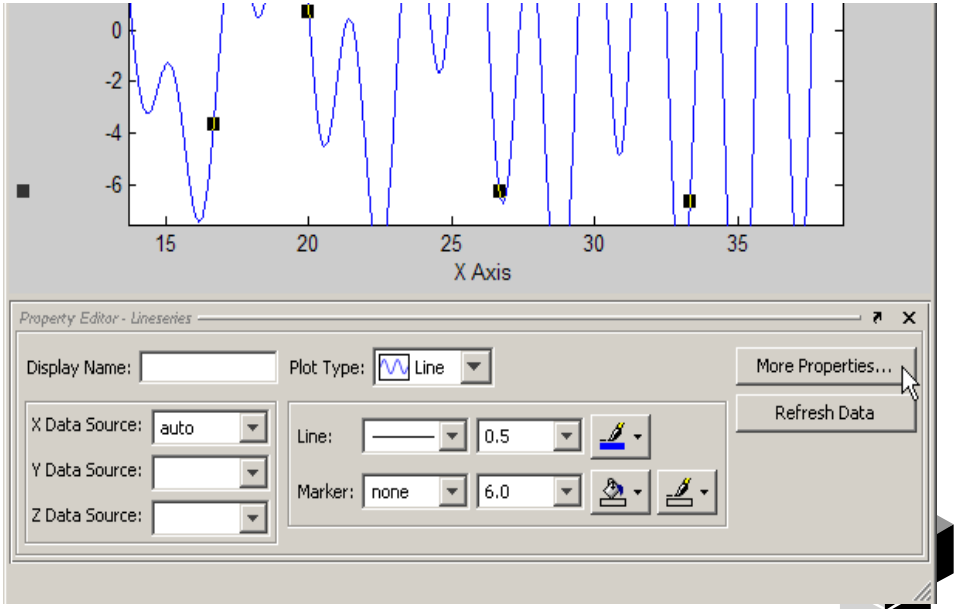
Избор на тип графика



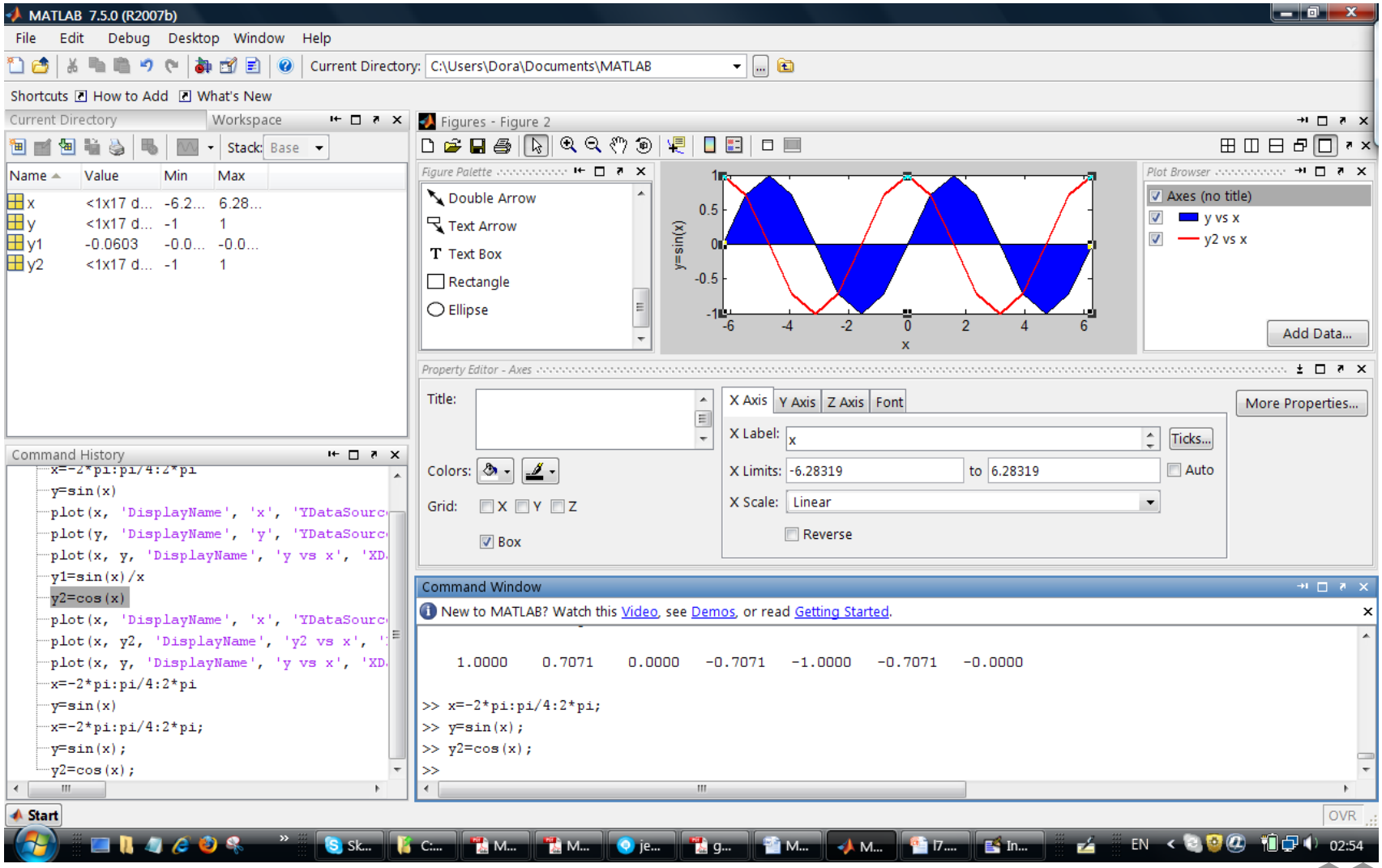
Атрибути на графиката







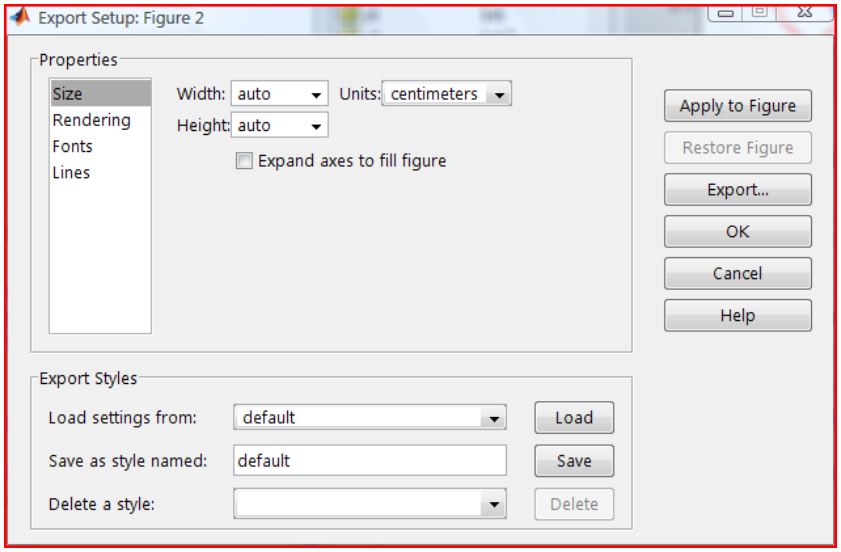
Пример

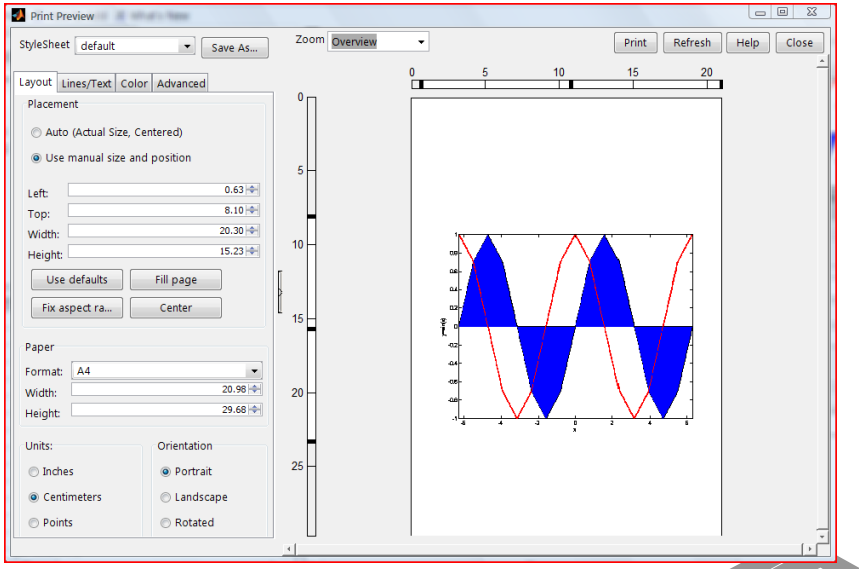

MATLAB 7.5.0 (R2007b)
File Edit Debug Desktop Window Help
Current Directory: C:\Users\Dora\Documents\MATLAB
Shortcuts How to Add What's New
Current Directory
Workspace
Stack: Base
Name Value Min Max
x <1x17 d... -6.2... 6.28...
y <1x17 d... -1 1
y1 -0.0603 -0.0... -0.0...
y2 <1x17 d... -1 1
Figures - Figure 2
Figure Palette
Double Arrow
Text Arrow
Text Box
Rectangle
Ellipse
y=sin(x)
x
Plot Browser
Axes (no title)
y vs x
y2 vs x
Add Data...
Property Editor - Axes
Title:
Colors:
Grid: X Y Z
Box
X Axis Y Axis Z Axis Font
X Label: x
Ticks...
X Limits: -6.28319 to 6.28319
Auto
X Scale: Linear
Reverse
More Properties...
Command History
x=-2*pi:pi/4:2*pi
y=sin(x)
plot(x, 'DisplayName', 'x', 'YDataSourc
plot(y, 'DisplayName', 'y', 'YDataSourc
plot(x, y, 'DisplayName', 'y vs x', 'XD
y1=sin(x)/x
y2=cos(x)
plot(x, 'DisplayName', 'x', 'YDataSourc
plot(x, y2, 'DisplayName', 'y2 vs x',
plot(x, y, 'DisplayName', 'y vs x', 'XD
x=-2*pi:pi/4:2*pi
y=sin(x)
x=-2*pi:pi/4:2*pi;
y=sin(x);
y2=cos(x);
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
1.0000 0.7071 0.0000 -0.7071 -1.0000 -0.7071 -0.0000
>> x=-2*pi:pi/4:2*pi;
>> y=sin(x);
>> y2=cos(x);
>>
Start
OVR
EN
02:54

Извеждане на графика

Видове изход

- на принтер
- във файл
  - стандартен графичен формат - TIFF, JPEG, EPS
  - FIG файл
  - команден файл





ГЕНЕРИРАРНЕ НА ГРАФИКА ЧРЕЗ КОМАНДИ И ФУНКЦИИ

Категории команди

- Команди и функции за изчертаване
- Команди за управление на координатната система
- Команди за анотиране на графиката

Команди за изчертаване

- **plot (y)**
  - изчертава графика на функция, чиито стойности са записани във вектора y. Като абсциси се използват индексите на отделните елементи на вектора;
- **plot(Y)**
  - изчертава едновременно графики на няколко функции, представени чрез отделните колонки на матрицата Y. За абсциси се използват първите индекси на елементите;
- **plot (x, y)**
  - изчертава графика на функцията y = f(x). Тук x е вектор с абсцисите, а y - вектор с ординатите;
- **plot(x, Y)**
  - изчертава едновременно графики на няколко функции на един и същ аргумент: y1 = f1(x), y2 = f2(x). Стойностите на функциите са записани в отделните колонки на матрицата Y;
- **plot(x, y1, x, y2, x, y3, ...)**
  - изчертава едновременно графики на няколко функции на един и същ аргумент, функциите са представени с двойки вектори:
    - x - на абсцисите, y1, y2, y3, - на ординатите;
- **plot(x, f1(x), x, f2(x), ...)**
  - изчертава едновременно графики на няколко функции на един и същ аргумент , стойностите на отделните функции, за всички абсциси от вектора x, се пресмятат директно в командата plot;

Команди за изчертаване

- **plot(x, y1, 'str1', x, y2, 'str2', ...)**
  - аргументите са по "тройки ". Третият аргумент от всяка тройка е стринг, състоящ се от един до три *управляващи* символа. С помощта на тези символи може да се променят подразбиращите се стойности за *цвят и стил на линията и вида на маркера*
- **plot(x, y1, 'str1', 'param', value_of_par, ...)**
  - двойките или тройките аргументи могат да бъдат следвани от ***двойки*** *parameter/values* , задаващи ***допълнителни свойства*** на компонентите:
    - **'LineWidth', n**
      дебелина на линията в **pt**; **n** — цяло число;
    - **'MarkerEdgeColor', 'symbol'**
      цвят на маркера (ако е отворен) или цвят на рамката на маркера (ако е затворена фигура). Стрингът 'symbol' е един от горните символи за задаване на цвят;
    - **'MarkerFaceColor', 'symbol'**
      цвят за запълване на маркера (ако е затворена фигура);
    - **'MarkerSize', n**
      размер на маркера в pt; n - цяло число

  - **1 pt = (1 / 72)" ➔ (1 /72) * 25.4 мм ≈ 0.353 мм**

Атрибути на графичните компоненти

Type	Values	Meanings
Color	'c' 'm' 'y' 'r' 'g' 'b' 'w' 'k'	cyan magenta yellow red green blue white black
Line style	'-' '--' ':' '.-' no character	solid dashed dotted dash-dot no line
Marker type	'+' 'o' '*' 'x' 's' 'd' '^' 'v' '>' '<' 'p' 'h' no character or none	plus mark unfilled circle asterisk letter x filled square filled diamond filled upward triangle filled downward triangle filled right-pointing triangle filled left-pointing triangle filled pentagram filled hexagram no marker

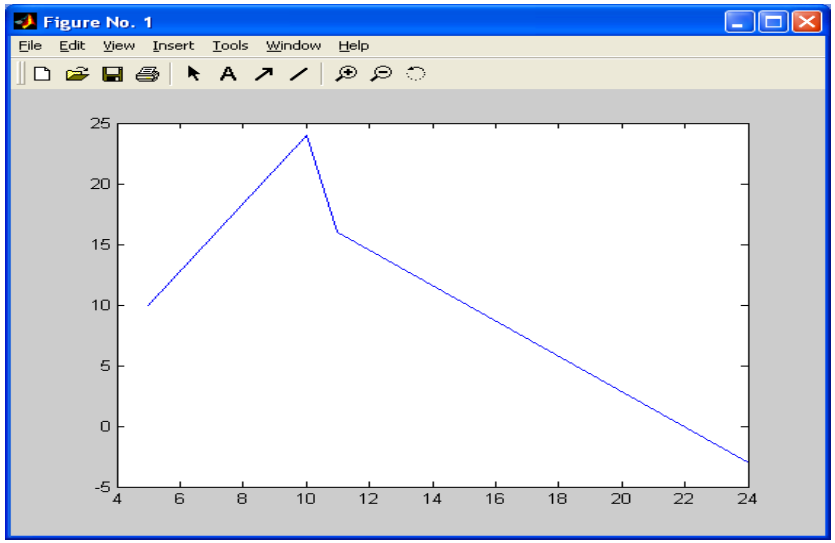
Други команди за изчертаване

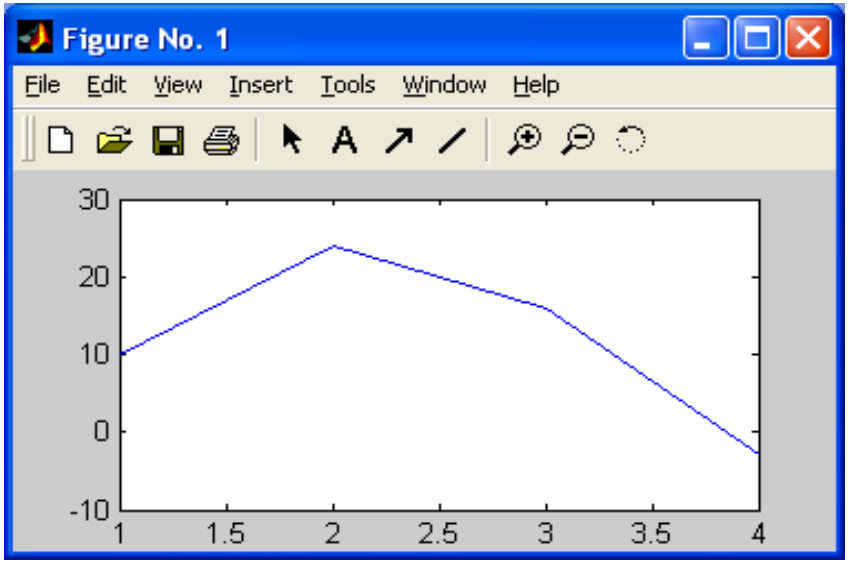
- **loglog(x,y)**
  - създава логаритмична графика $\log_{10}(y)$ спрямо $\log_{10}(x)$, използва логаритмичен мащаб по двете оси x и y
- **semilogx(x,y)**
  - създава полулогаритмична графика на вектора y спрямо $\log_{10}(x)$, използва логаритмичен мащаб по оста x
- **semilogy(x,y)**
  - създава полулогаритмична графика на $\log_{10}(y)$ спрямо вектора x, използва логаритмичен мащаб по оста y
- polar(theta,rho)
  - създава графика в полярни координати на радиуса rho спрямо ъгъла theta
- **polar (phi , r, 'str')**
  - графика на функцията r = r(ϕ) в полярни координати
    - **phi** - вектор със стойностите на полярния ъгъл в радиани
    - **r** - вектор със стойностите на радиуса
    - **'str'** - стринг, който задава стила на линията

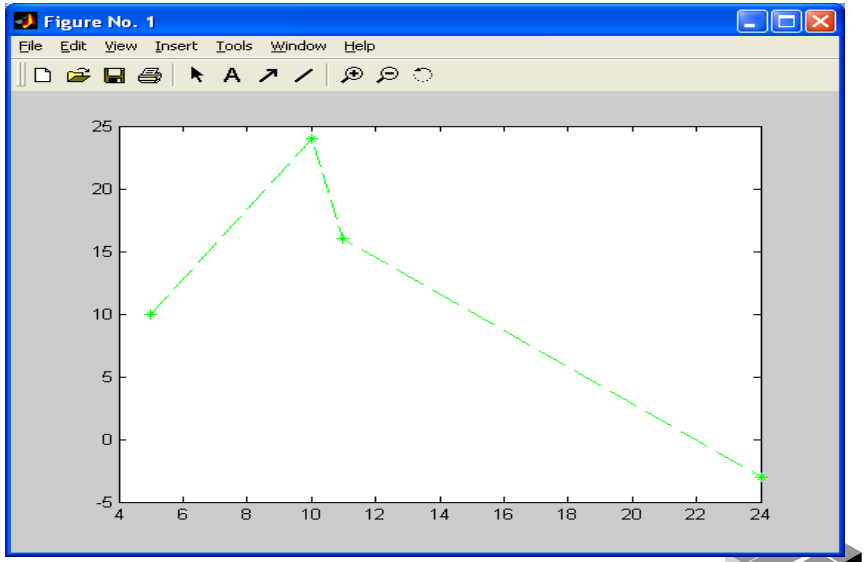
Примери

```
>> y=[10  24  16  -3];
>> plot(y)

>> x=[5  10  11  24];
>> plot(x,y)
>> plot(x,y,'g--*')
```

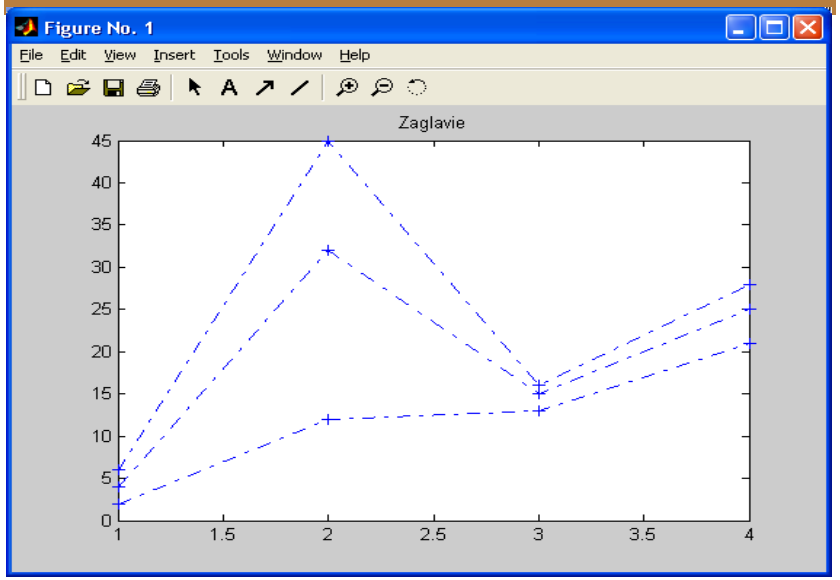






Примери

```
>> B= [2 4 6;12 32 45;13 15 16;21 25 28];
>> plot(B,'b-.+')
>> title('Zaglavie')
```

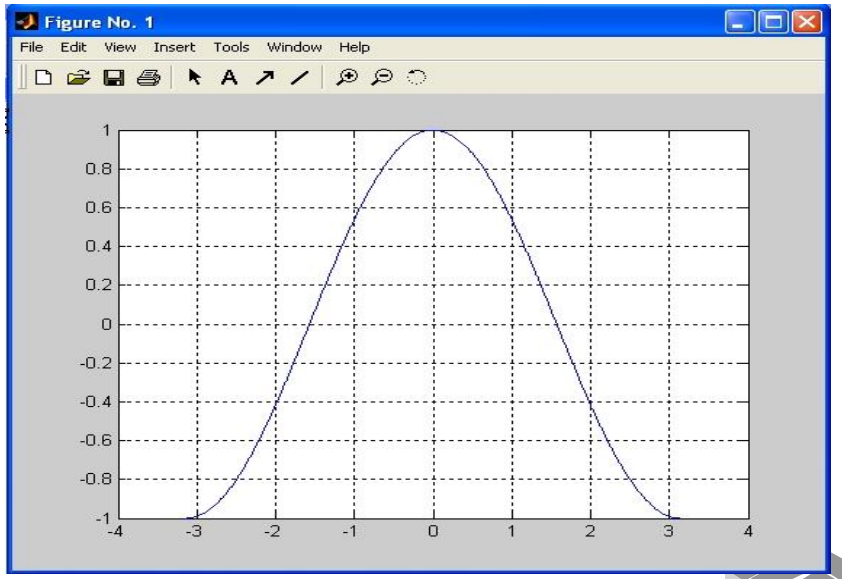



```
>> x=-pi:0.01:pi;
```

% задаване на интервал (начална точка, стъпка, крайна точка)

```
>> plot (x,cos(x)), grid on
```

% изчертаване на функцията с включена мрежа



Команди за управление на координатната система

- Управление на мащабирането и изгледа на координатните оси на текущата графика:
  - **axis ([xmin xmax ymin ymax])**
    - задава интервалите на изменение на променливите по координатните оси
  - **axis auto**
    - връщане към подразбиращи се стойности
  - **axis equal**
    - установява еднакъв мащаб по двете оси, така че графиката да не е деформирана
  - **axis square**
    - установява еднакъв диапазон на изменение на променливите по двете оси
  - **axis normal**
    - премахва всякакви ограничения по мащабирането
- Управление на мрежата на чертожното поле
  - **grid on**
    - добавя мрежа към текущата графика
  - **grid off**
    - премахва мрежата
- Установяване на свойства на чертожното поле за последователно изчертаване на графики
  - **hold on**
    - задържа зададените стойности
  - **hold off**
    - премахва задържането

Команди за управление на координатната система

- Едновременно изобразяване на няколко графики в различни части на един прозорец
  - **subplot** - подготвя прозорчето, в което ще попадне графиката от непосредствено следващата я функция за изчертаване
  - **subplot (m, n, k)** - разделя основния прозорец на прозорчета, които може да се представят като матрица с размери m x n
    - аргумент k указва номера на прозорчето, в което ще се изчертае графиката от следващата я команда за изчертаване.
    - Номерацията на прозорчетата е по редове
- Обединяване на разделени части
  - за трети аргумент на функцията subplot() се подава вектор с номерата на обединяваните прозорчета, например subplot (3,3,[3 6 9])

Надписване на координатните оси

- **xlabel ('string')**
  - нанасяне надпис по оста x;
- **ylabel ('string')**
  - нанасяне надпис по оста y;
- **title ('string')**
  - нанасяне заглавие в горната част на графиката;
- **legend ('str1', 'str2', ...)**
  - изобразява легенда в горната дясна част на графичен прозорец
  - използва се, когато прозорецът съдържа няколко графики
  - за всяка графика в легендата е начертана права линия с цвета и стила на линията на графиката, с пояснителен текст, взет от поредния аргумент на функцията **legend()**
  - последователността на аргументите-стрингове трябва да отговаря на последователността на построяване на съответните графики
  - броят на стринговете трябва да е равен на броя на графиките
  - допуска се и допълнителен аргумент, позволяващ да се задават различни места на легендата

Анотиране на графиката

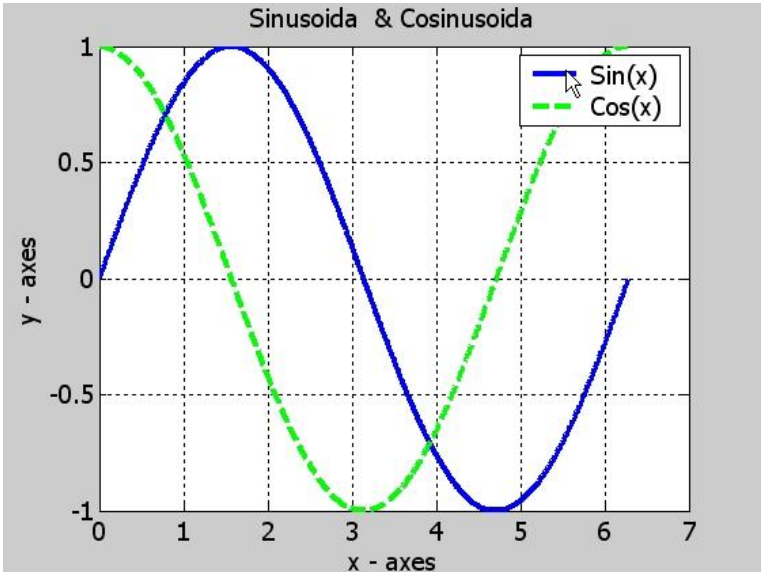
- Програмно нанасяне на текст на произволно зададено място в текущия графичен (или текстов) прозорец
  - text(x, y, 'string', 'FontName', '<Име на шр.>', 'FontSize', <Размер на шр. в pt>)
    - x, y - декартови координати на началната позиция, от която ще се нанесе текста *string* с указания стил и размер на шрифта
- Два типа координати:
  - *абсолютни* - единици за измерване: **inches, centimeters, points, pixels,** отчитат се от долния ляв ъгъл на координатните оси
  - *нормализирани* – **normalized - о**тчитат се от горния ляв ъгъл на графичното поле и приемат стойности в диапазона от 0 до 1.

Анотиране на графиката

- gtext ('string')
  - позволява чрез позициониране курсора да се посочи мястото на текста от аргумента 'string';
- gtext ({'This is the first line', 'This is the second line'})
- gtext ({'First line','Second line'}, 'FontName', 'Times', 'Fontsize', 12)

Пример

```
x=0:0.01:2*pi;                          % интервал и стъпка на аргумента
plot(x,sin(x),'-b',x,cos(x),'--g','LineWidth',3)% стил, цвят на линиите
                                                % и дебелина на линиите
set(gca,'FontName','Tahoma','FontSize',14) % шрифт
grid on, xlabel('x - axes'), ylabel('y - axes')
title('Sinusoida  & Cosinusoida')
legend('Sin(x)','Cos(x)')
```

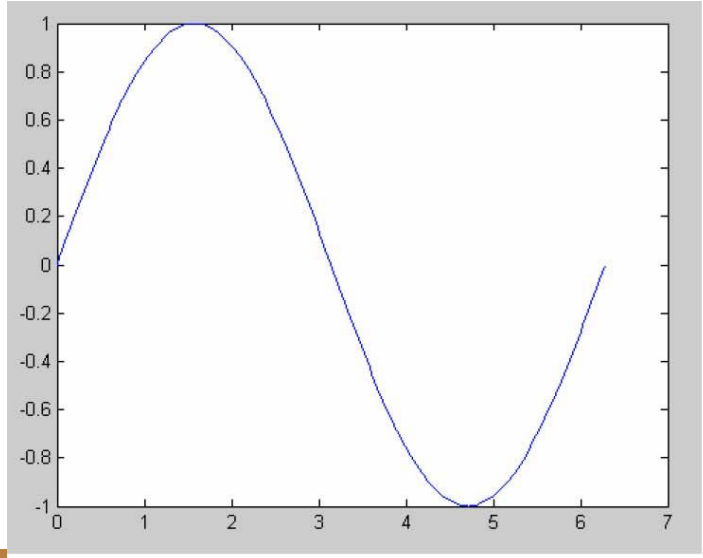


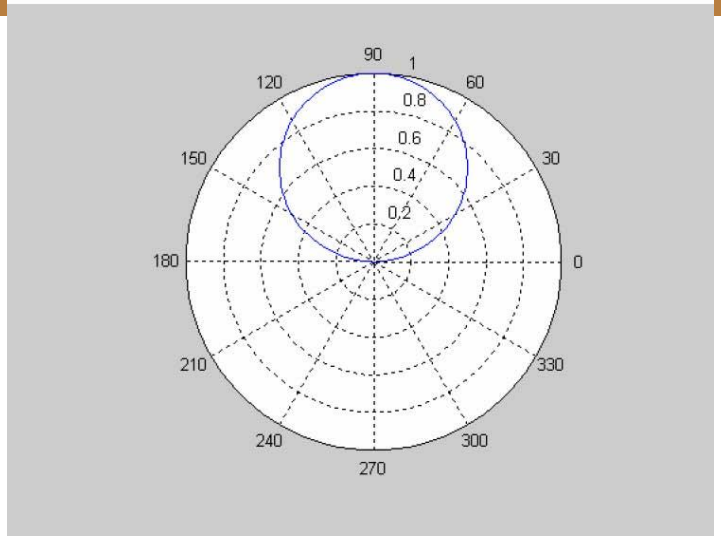
Пример

Изчертаване на синусоида в интервала

$t \in [0, 2\pi]$

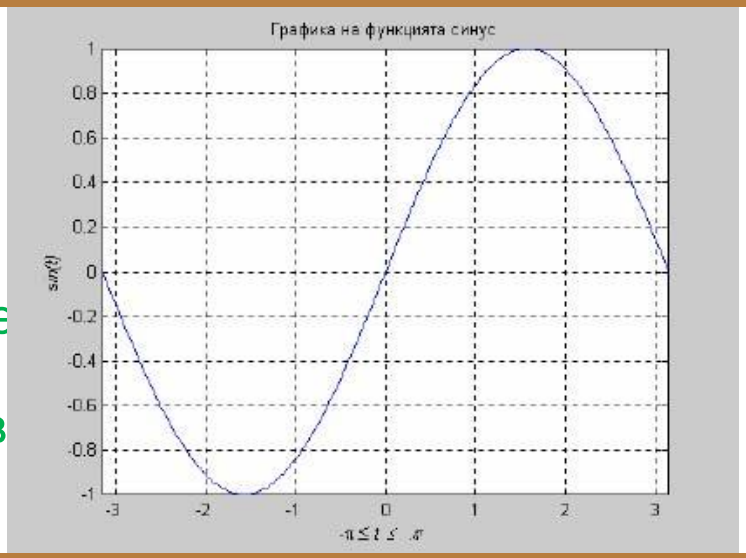
```
» t=0:pi/100:2*pi;
» y=sin(t);
» plot(t,y)
» polar(t,y)
```





Пример

```
xlabel('текст')
        % Добавя текст до x–оста
ylabel('текст')
        %  Добавя текст до y-оста
title('текст')
        % Добавя текст като заглавие
text(x,y,'текст')
        % Добавя текст, който започв
от т. (x,y)
```



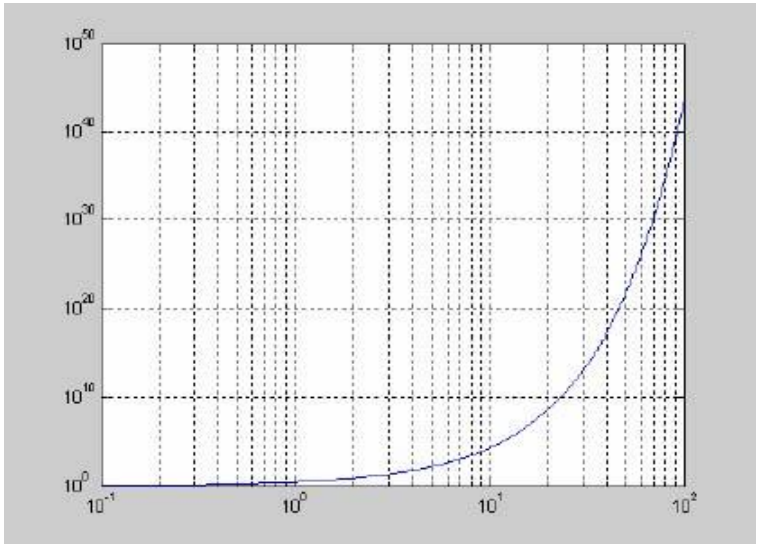

```
» t=-pi:pi/100:pi;
» y=sin(t);
» plot(t,y);
» axis([-pi pi -1 1]);
» xlabel('-\pi \leq \itt \leq \pi');
» ylabel('\itsin(t)');
» title('Графика на функцията синус');
» grid on
```

\pi	стойност на π
\it	курсив
\leq	≤

Пример

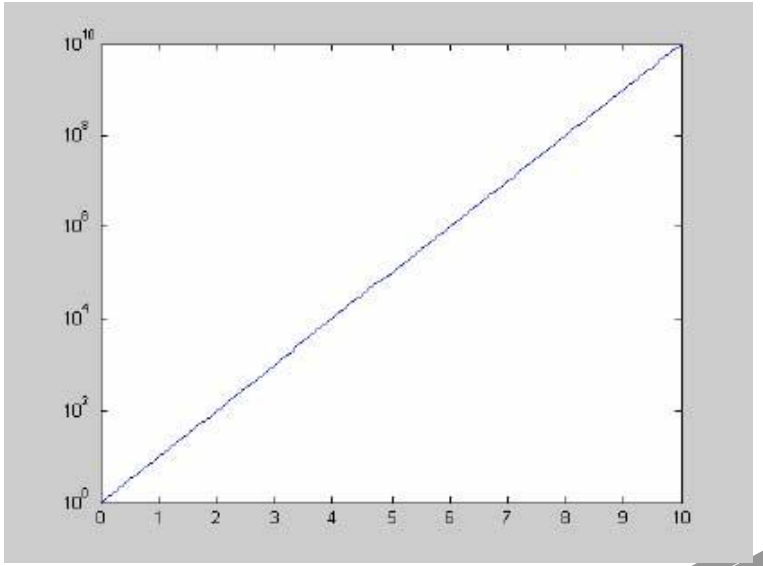
%Изчертава $\log_{10}(e^x)$ спрямо $^{\log}10^{(x)}$в интервала $x \in [0.1,100]$

```
» x=0.1:0.1:100;
» loglog(x,exp(x)),grid on
```



%Изчертава $^{\log}{}_{10}\,10^x$ спрямо x в интервала $x \in [10,0\ ]$

```
» x = 0:.1:10;
» semilogy(x,10.^x)
```



Пример

subplot(m,n,p)

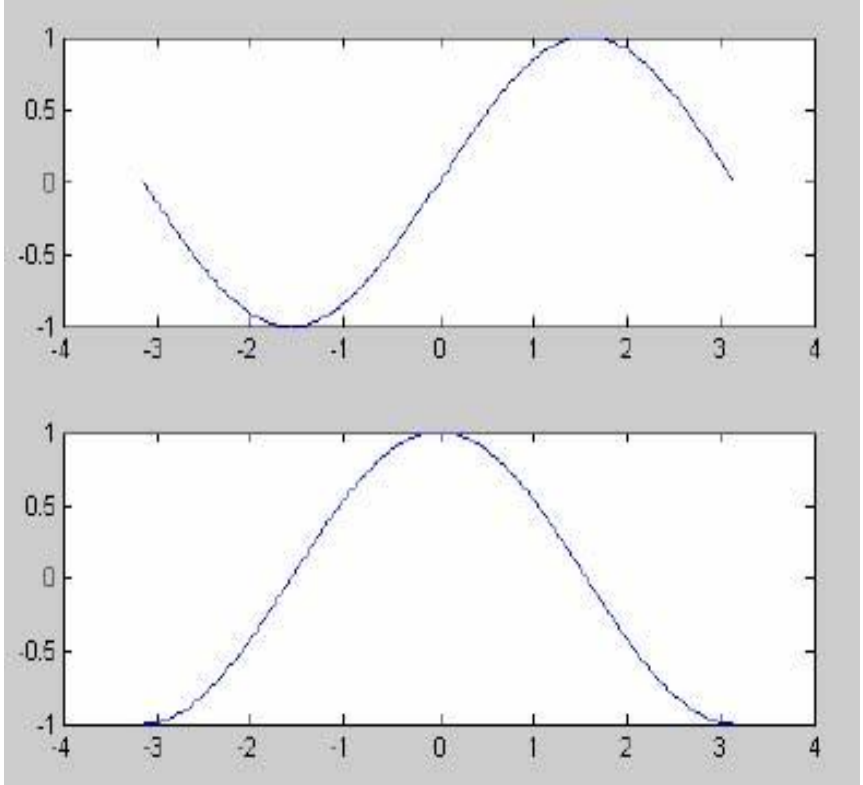
subplot(mnp)

Изчертава няколко графики в един прозорец, като разделя прозореца на m x n матрица от координатни системи и избира p-тата система

subplot(111)

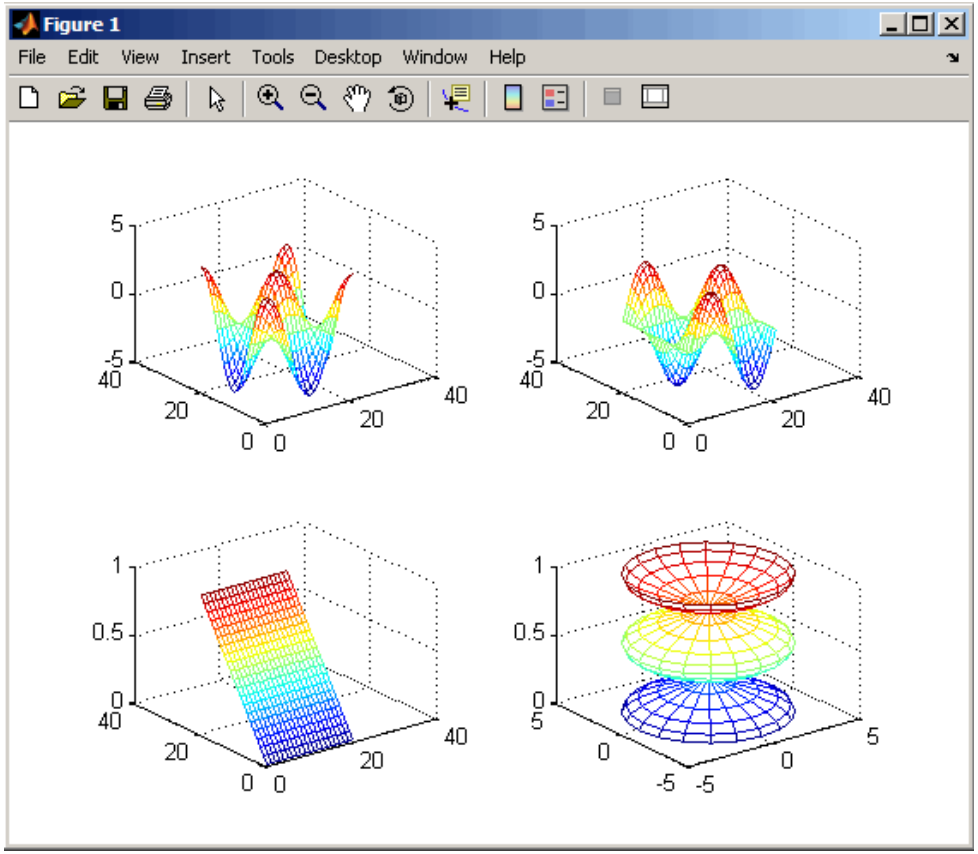
Връща към пълен екран за изчертаване

```
» y1=sin(t);
» y2=cos(t);
» subplot(211),plot(t,y1)
» subplot(212),plot(t,y2)
» subplot(111)
```



Пример

```
t = 0:pi/10:2*pi;
[X,Y,Z] = cylinder(4*cos(t));
subplot(2,2,1); mesh(X)
subplot(2,2,2); mesh(Y)
subplot(2,2,3); mesh(Z)
subplot(2,2,4); mesh(X,Y,Z)
```

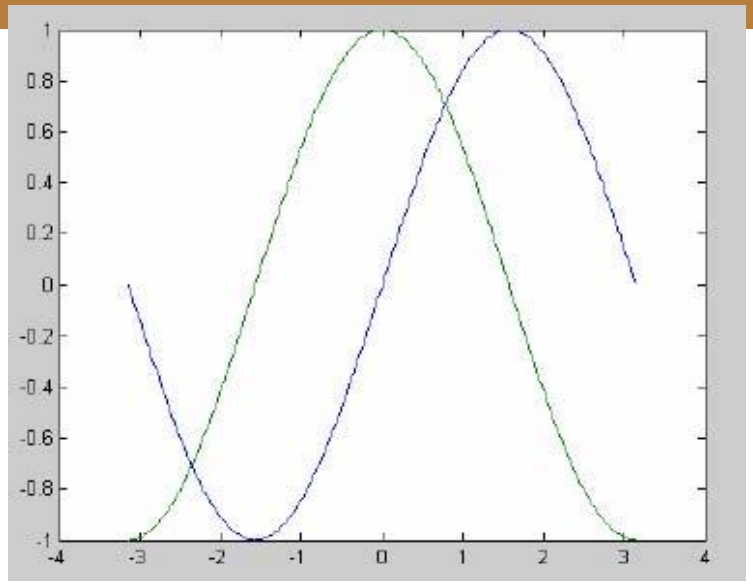


Пример

plot(x1,y1,x2,y2,…)

%Изчертава няколко графики в един прозорец върху една координатна система

» plot(t,y1,t,y2)



plot3(x,y,z)

- 3-D аналог на plot()
- генерира 3-D линия с координати елементите на трите вектора x, y и z (векторите трябва да бъдат с еднаква дължина)
- създава 2-D проекция на линията върху екрана

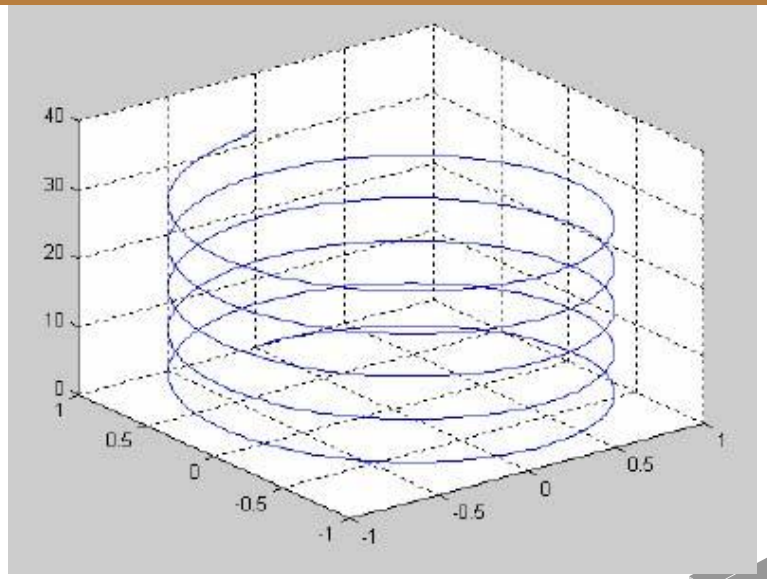
» t = 0:pi/100:10*pi;

» y1 = sin(t);

» y2 = cos(t);

» plot3(y1, y2, t);

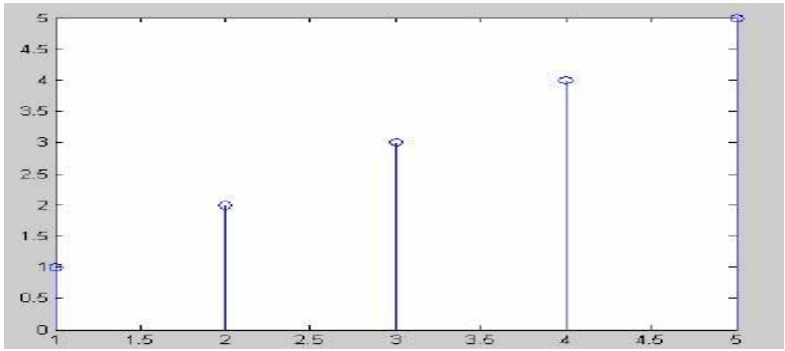
» grid on; stem(y)



Примери

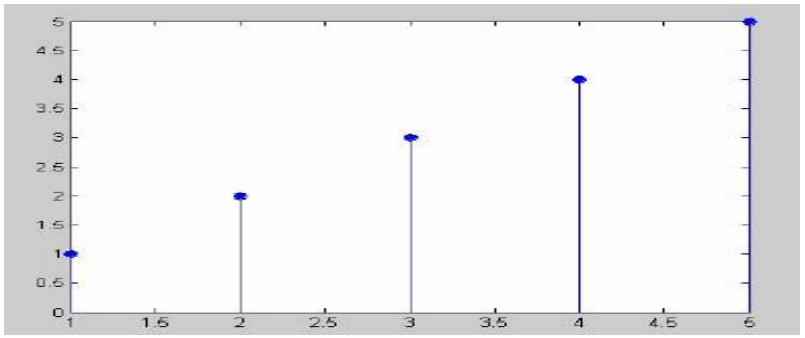
stem(x,y)

•За дискретни сигнали изчертава данните във вектора y спрямо остта x , като всяка точка е малка окръжност и права линия



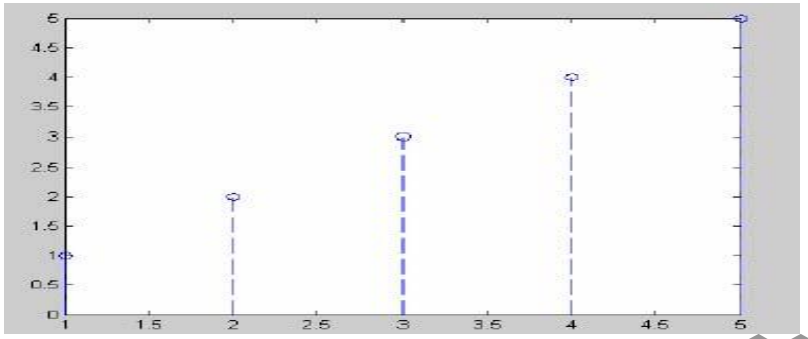
stem(x,y,'filled')

•Изчертава данните във вектора y спрямо вектора x
Използва запълнени маркери



stem(x,y,'тип_линия')

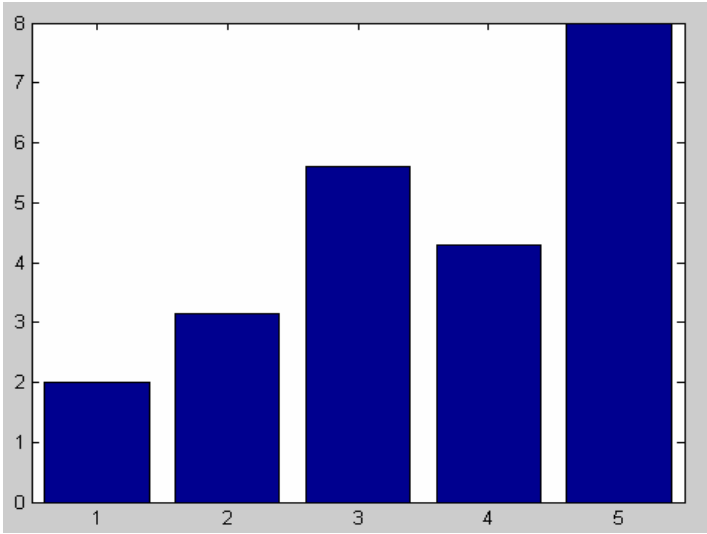
•Използва тип_линия за изчертаването



Специални графики

bar - изчертава лентова графика;
Синтаксис:
bar(y), bar(x,y)

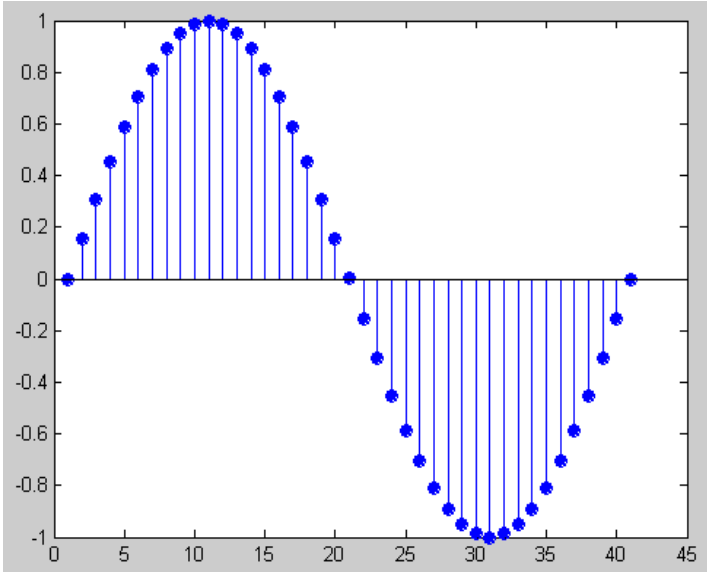
```
>> y=[2, pi, 5.6, 4.3, 8]
y =
2.0000    3.1416    5.6000    4.3000
8.0000
>> bar(y)
```



stem - графика само с кръгчета, свързани с прави линии с абсцисната ос.

stem(y); stem(x, y); stem(x, y, 'filled');
Кл. дума 'filled' - запълване на кръгчетата

```
>>x=0:pi/20:2*pi;
>>stem(sin(x),'filled')
```



Специални графики

pie - за построяване на кръгови диаграми

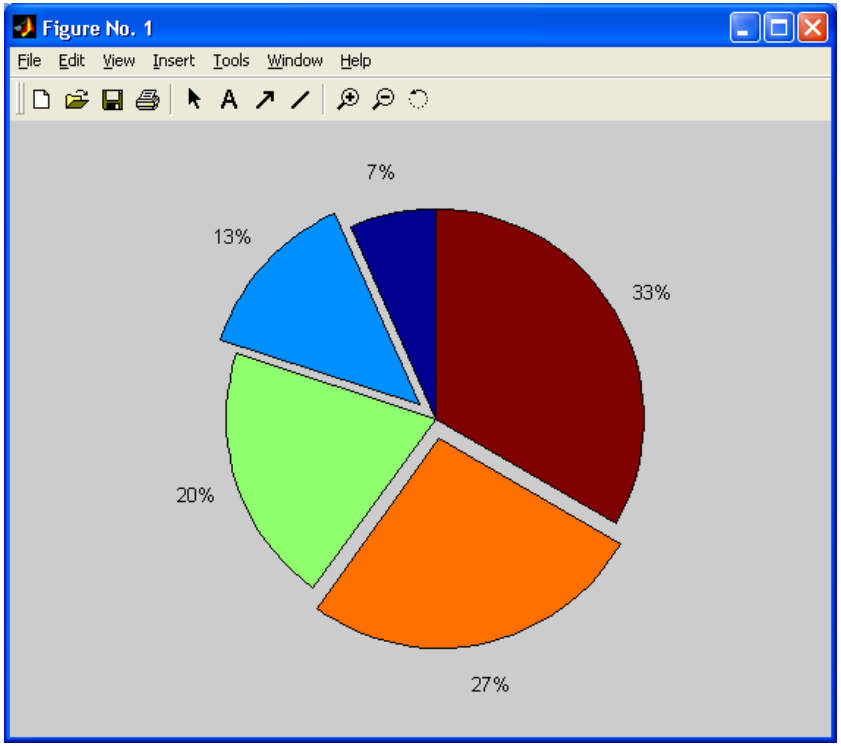
pie (y);
pie (y, explode);

Вторият аргумент explode е вектор с дължината на y, състоящ се от единици и нули.

Секторите, съответстващи на елементите на explode със стойности, равни на 1, се изобразяват изместени от центъра навън.

Номерацията на секторите започва от най-горния вляво, в посока обратна на часовата стрелка.

```
>> y=[1, 2, 3, 4 ,5 ];
>> pie(y, [0 1 0 1 0])
```

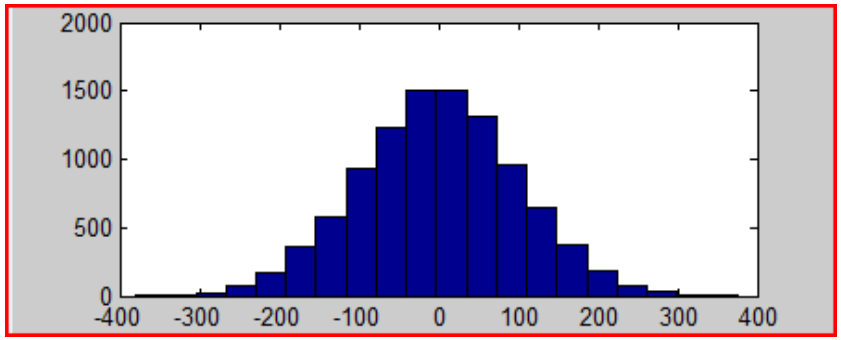


Специални графики

hist() - построява хистограми, даващи нагледна представа за разпределението на случайна величина в отделните интервали от стойности

hist (y); **%n=10**

hist (y, n); %n — брой интервали




```
>> y=round(100*randn(1,10000));
>> hist(y,20)
```

Специални графики

начертаване графиките на функции, зададени с аналитични си изрази

fplot

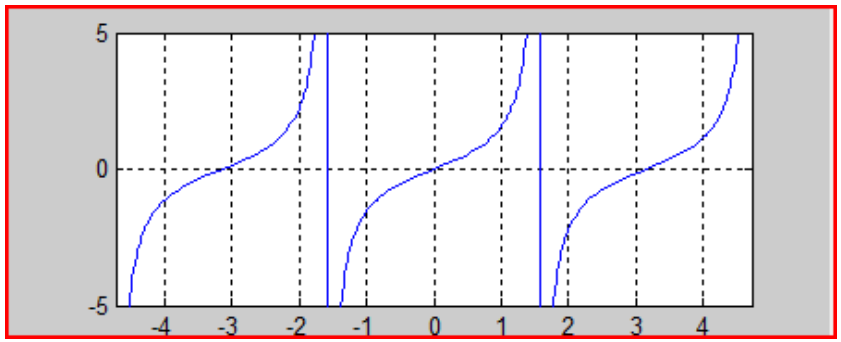
- изчертава графиката на функцията y = f(x);

fplot ('f(x)',[xmin xmax])

- задава се интервала на аргумента x;

fplot ('f(x)',[xmin xmax ymin ymax])

- задава се и интервала на изменение на функцията y (за изобразяването ѝ на графиката)




```
>> fplot('tan(x)',[-3*pi/2 3*pi/2 -5 5])
>> grid on
```

Специални графики

ezplot - изчертава графиката на явно, неявно и параметрично зададена функция;

ezplot ('f(x)')

- по подразбиране -2π < x < 2 π;

ezplot ('f(x)', [xmin xmax]);

ezplot ('f(x, y)')

- по подразбиране -2 π < x < 2 π и -2 π < y < 2 π;

ezplot ('f(x, y)',[xmin xmax ymin ymax]);

ezplot ('f1(t)', 'f2(t)')

- по подразбиране 0 < t < 2 π;

ezplot ('f1(t)', 'f2(t)',[tmin tmax]);

```
ezplot('cos');
ezplot('cos(x)');
ezplot('x^3+y^3-5*x*y+1/5',[-3, 3]);
ezplot('t*cos(t)','t*sin(t)',[0 4*pi]);
grid on
```

ezpolar ()

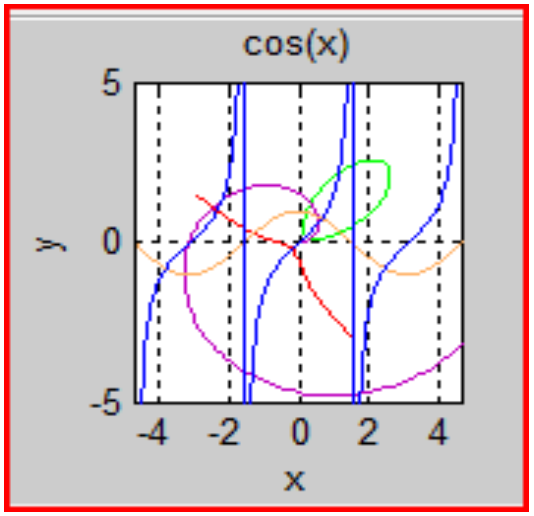
- чертае графиката на функция, зададена в полярни координати r = r(φ).

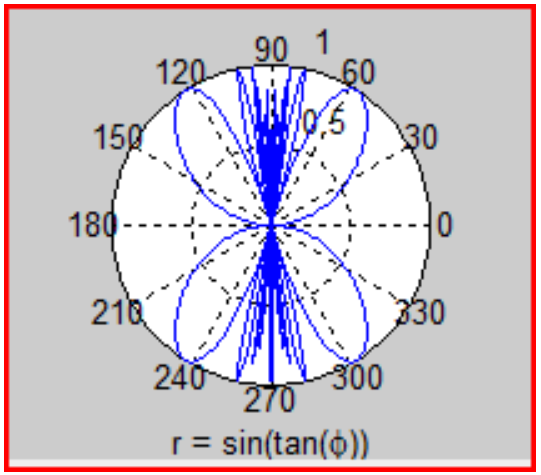
ezpolar ('f(φ)')

- по подразбиране 0 < φ < 2 π;

ezpolar ('f(φ)', [φmin φmax]);

```
ezpolar('sin(tan(phi))');
```





Тримерна графика

- Видове
  - **mesh** - изобразяване с помощта на мрежа от линии, съединяващи координати на точки;
  - **surf** - изобразяване на непрекъснати повърхнини;
  - **surfl** - използва се допълнително осветяване (l от light -светлина).
- Функции за изчертаване

  plot3(x,y,z) - 3D аналог на plot()

  x, y и z са 3 вектора с еднаква дължина – линия в 3D пространството

  plot3(X,Y,Z),

  X, Y и Z са 3 матрици – няколко линии - по колони,

  plot3(X,Y,Z,s)

  s е стринг, задаващ тип на линия, цвят и маркер
- Изобразяване на повърхнини, описвани от функцията **z = F(x, y)**, зададена в аналитичен или табличен вид
- Функции за управление на атрибути на чертожното поле
  - както при двумерната графика

Построяване на тримерна графика

1) Пресмятат се матриците X и Y на координатите x и y на възлите на мрежата с помощта на функцията meshgrid()

[X,Y] = meshgrid (x,y);

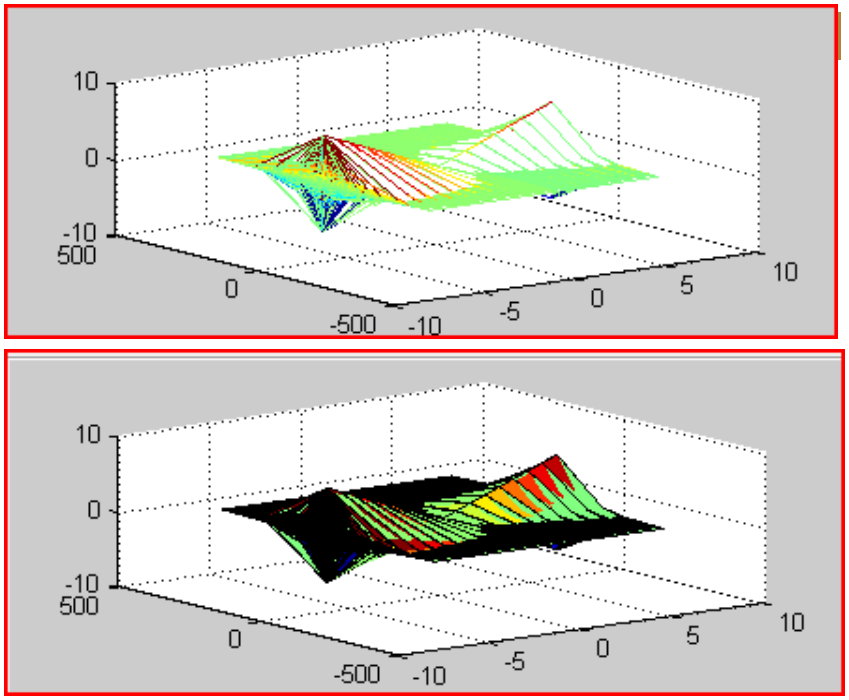
x и y са вектори с абсцисите и ординатите;

2) Пресмята се матрицата Z със стойностите на функцията във всички възли:

Z = F(X,Y);

F(X,Y) е функция с използване на поелементните оператори **.*, ./, .^** !

3) Извиква се една от трите функции **mesh(), surf()** или **surfl()** за построяване на графиката:



mesh(X,Y,Z);

surf(X,Y,Z);

surfl (X,Y,Z);

Функции за тримерна графика

- **colorbar**
  - извежда се цветова лента;
- **colormap('colmap')**
  - цветова карта, определяща оцветяването на изображението
  - 'colmap' може да има стойности **'grey', 'copper', 'pink', 'spring', 'summer', …**
- **shading interp()**
  - плавен преход на сенките (командата се въвежда без аргументи)

Функции за тримерна графика

- ezmesh()
- ezsurf()
- ezgraph3()
  - начертаване на повърхнина, зададена с функцията z = f(x,y)
  - използва една от трите функции - mesh, surf или surfl, указана от потребителя в първия аргумент 'plotfun'.

ezgraph3 ('plotfun', 'f(x.y)');

ezgraph3 ('plotfun', 'f(x,y)', [a b])

в областта a<x<b, a<y<b;

ezgraph3 ('plotfun', 'f(x.y)', [xmin xmax ymin ymax]);

ezgraph3 ('plotfun', 'f(x,y)', [a b], 'domstile')

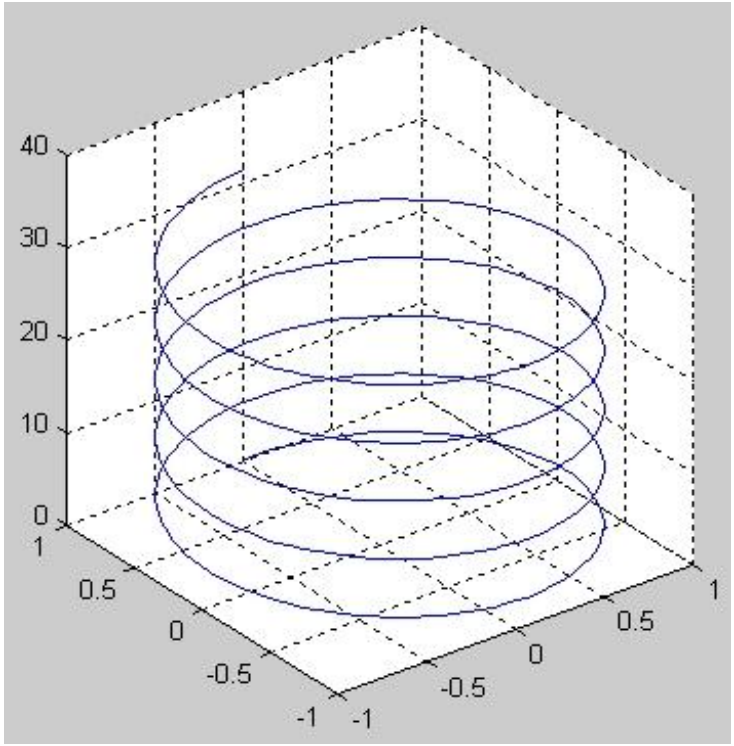
последният аргумент задава вида на областта (domain):

'rect' - правоъгълна

'circ' -кръгова

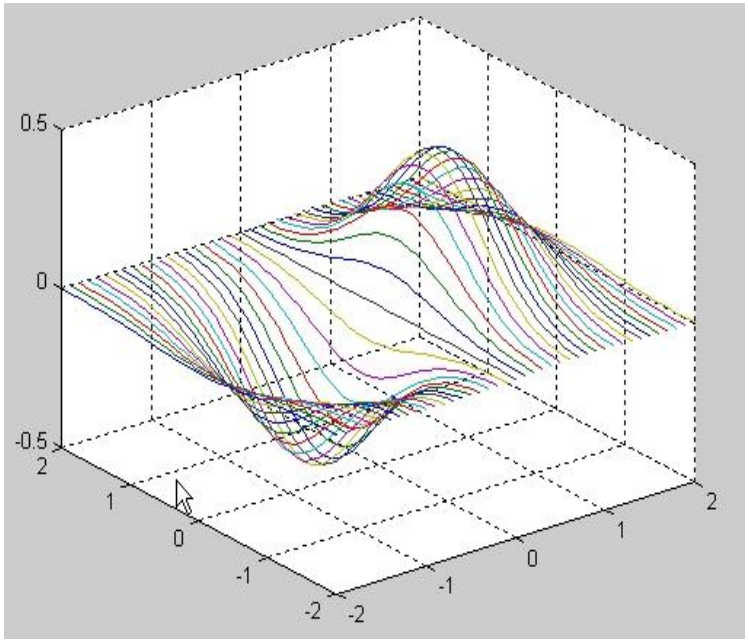
Пространствена крива

```
t = 0:pi/50:10*pi;
plot3(sin(t),cos(t),t)
axis square;
grid on
```

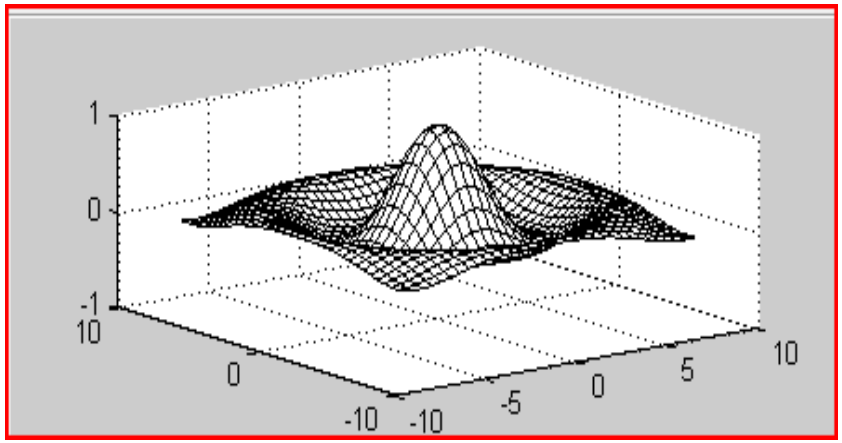


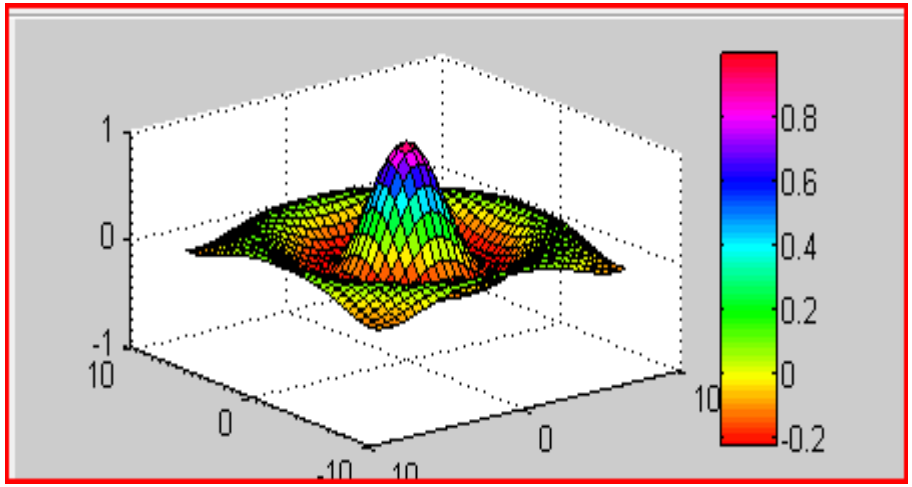
Изчертаване на функция

```
[X,Y] = meshgrid([-2:0.1:2]);
Z = X.*exp(-X.^2-Y.^2);
plot3(X,Y,Z)
grid on
```



```
>> [X,Y] = meshgrid(-8:.5:8);
R = sqrt(X.^2 + Y.^2) + eps;
Z = sin(R)./R;
mesh(X,Y,Z,'EdgeColor','black')
```

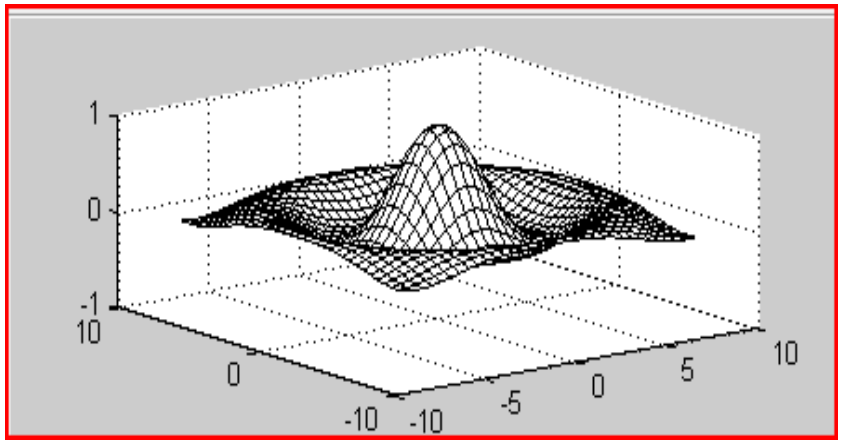


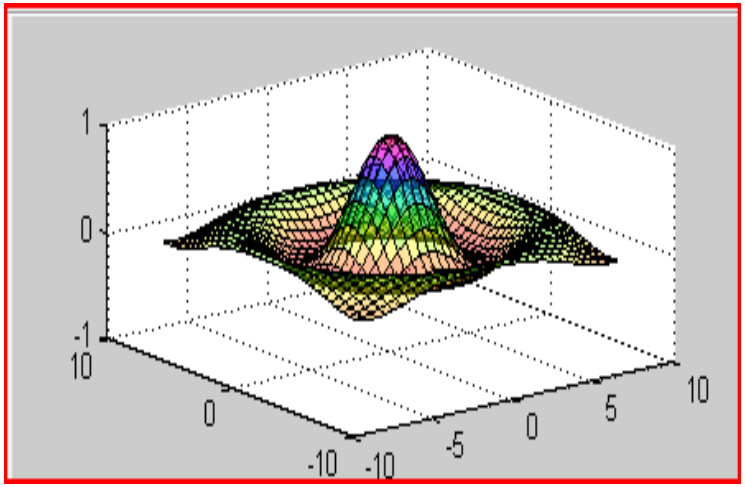



```
surf(X,Y,Z)
colormap hsv
colorbar
```

color map – списък от цветове, стойности на Z

```
>> [X,Y] = meshgrid(-8:.5:8);
R = sqrt(X.^2 + Y.^2) + eps;
Z = sin(R)./R;
mesh(X,Y,Z,'EdgeColor','black')
```

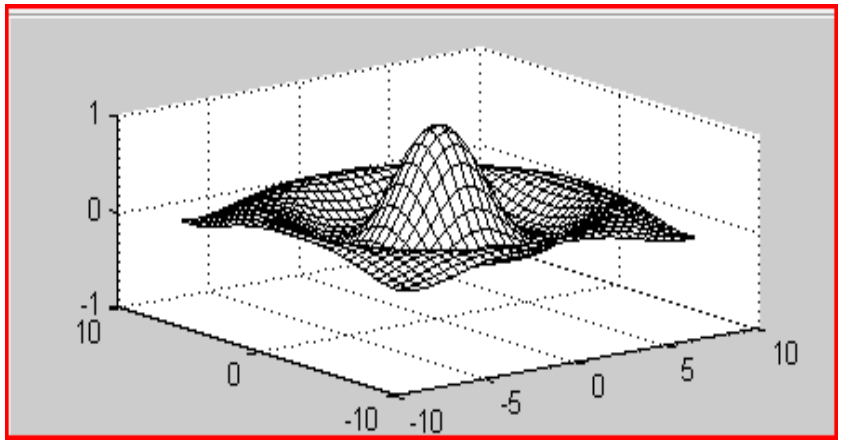




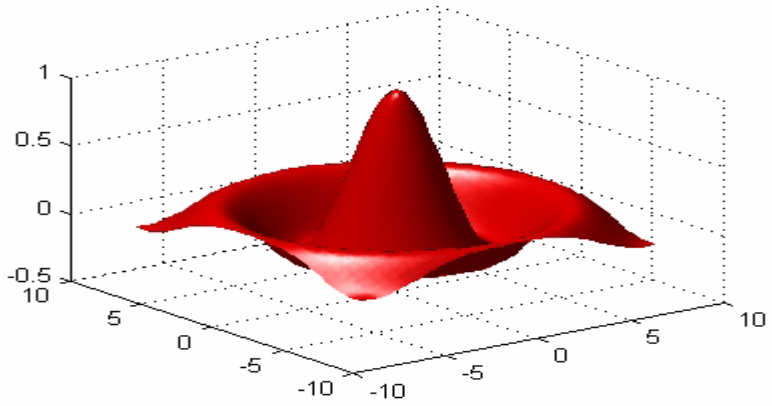

```
surf(X,Y,Z)
colormap hsv
colorbar
alpha(.4)
```

alpha(.4) – задава степен на прозрачност, [0,1]

```
>> [X,Y] = meshgrid(-8:.5:8);
R = sqrt(X.^2 + Y.^2) + eps;
Z = sin(R)./R;
mesh(X,Y,Z,'EdgeColor','black')
```

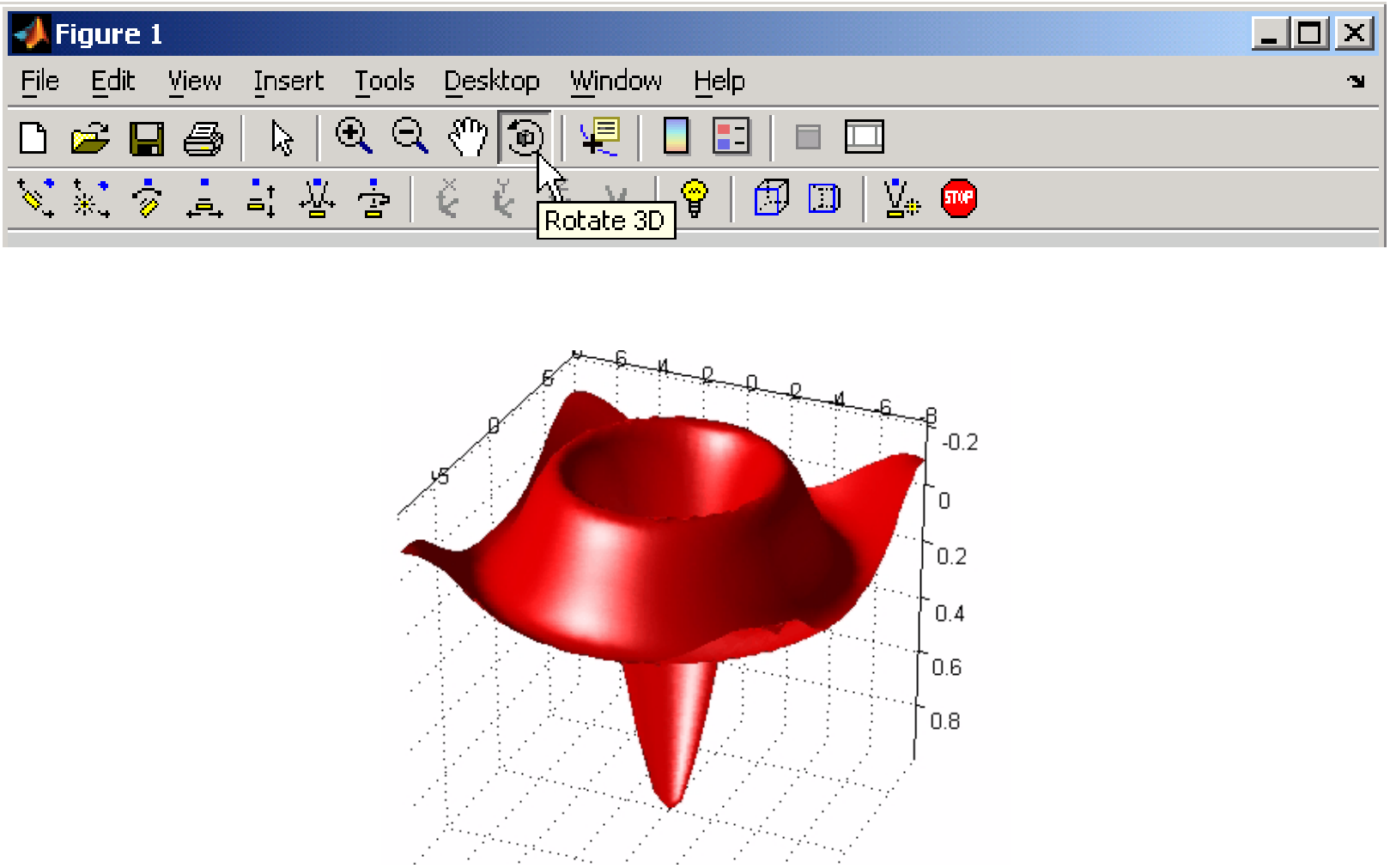



```
surf(X,Y,Z,'FaceColor','red','EdgeColor','none')
camlight left; lighting phong
```



осветяване

Операции с повърхнини



Представяне на образи

load durer

%зарежда бинарен файл x , съдържащ матрица с индекси към матрица map на цветовете на изображение

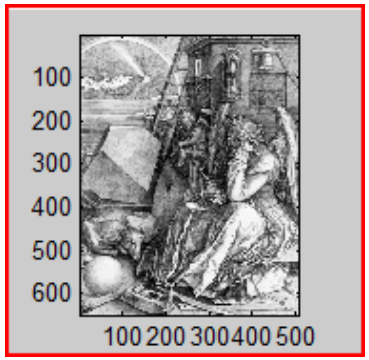
whos

Name	Size	Bytes	Class
X	648x509	2638656	double array
caption	2x28	112	char array
map	128x3	3072	double array

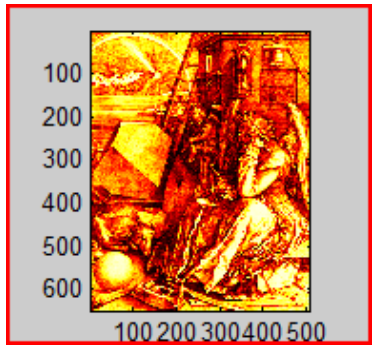
image(X)

colormap(map)

axis image



colormap(map)



colormap(hot)

Анимация

- Два метода
  1. Предварителна подготовка и запис на кадрите (movein, getframe, movie)
     - M=movein(nfr) дефинира фрейм-матрица M със стълбове за всеки кадър на движещи се обекти от графиката
     - отделните кадри се генерират с plot() и се записват като стълбове на фрейм-матрицата с M(:,i)=getframe,
     - проиграват се зададен брой пъти с movie(M,к)
  2. Всеки кадър се изчислява и изобразява в реално време със set и drawnow, докато старият се изтрива