



گرافیک کامپیوتری

پردازش تصویر در متلب

دانشکده فنی و حرفه ای
ملاصدرا - رامسر

مدرس: آمنه قنبری تلوکی

هیأت علمی موسسه آموزش عالی آیندگان تنکابن



جلسه پنجم

۹۸ / ۱۲ / ۲۶

آشنایی با ترسیم نمودار در متلب

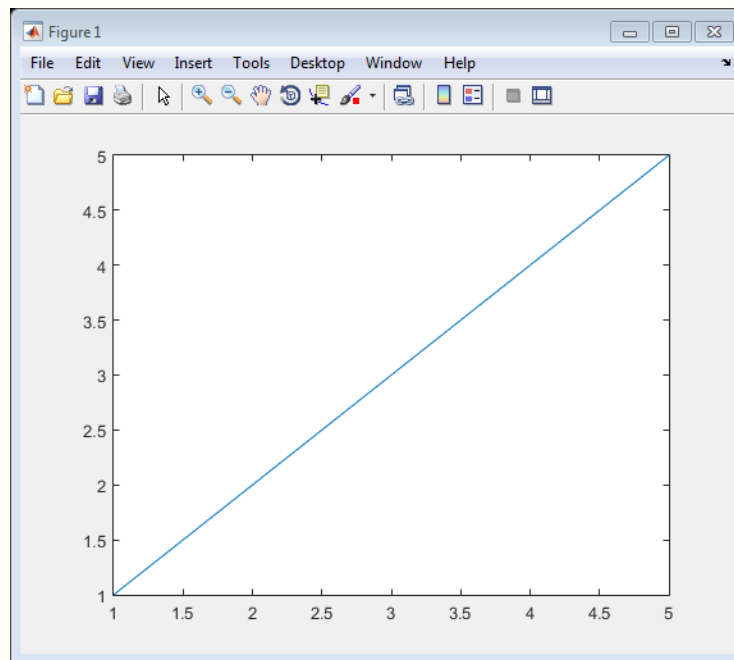
آشنایی با ترسیم نمودار در متلب

رسم نمودار :

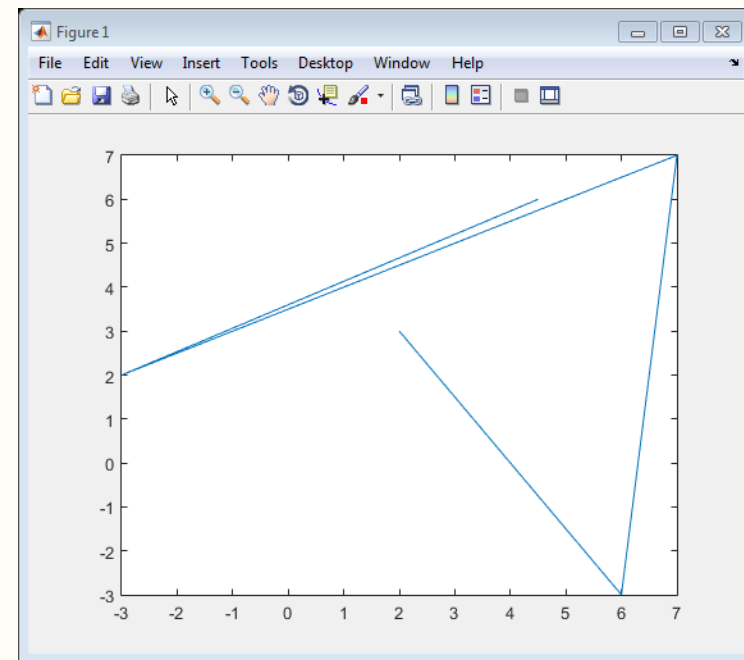
- رسم نمودار با تابع `plot` انجام می شود.
- پسوند نمودار رسم شده برای ذخیره کردن، `.fig` میباشد.
- برای ذخیره ی `plot`، در هنگام `save` کردن می توان فرمت آن را به `pdf` تغییر داد؛ از این روش برای ارائه یک `plot form` استفاده می شود.

آشنایی با ترسیم نمودار در متلب

```
Editor - F:\mat\t1.m
t1.m x +
1 - y=[1 2 3 4 5];
2 - plot(y)
```



```
Editor - F:\mat\t1.m
t1.m x +
1 - y=[2 6 7 -3 4.5];
2 - z=[3 -3 7 2 6];
3 - plot(y,z)
```



آشنایی با ترسیم نمودار در متلب

تابع `gtext`:

برای نوشتن بر روی نمودار `plot` مورد استفاده قرار می گیرد و به صورت های زیر به کار می رود:

`gtext({'text1','text2','text3',...})` □

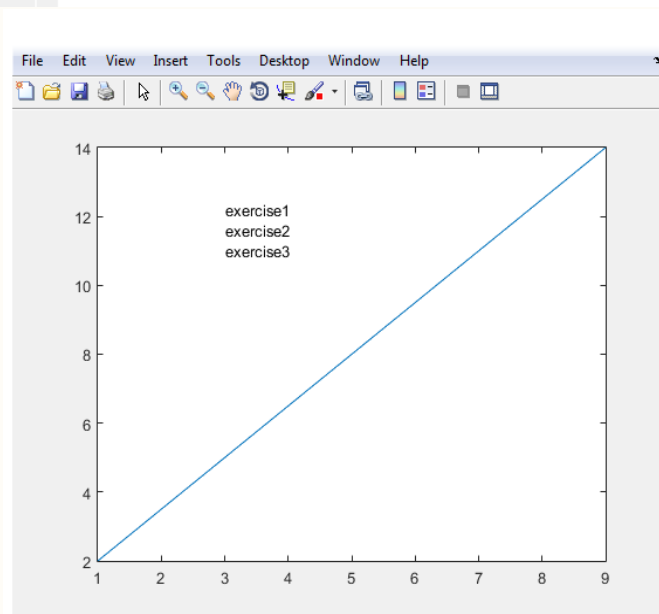
تمام نوشته ها را با هم در صفحه ی `plot` نشان می دهد

`gtext({'text1';'text2';'text3';...})` □

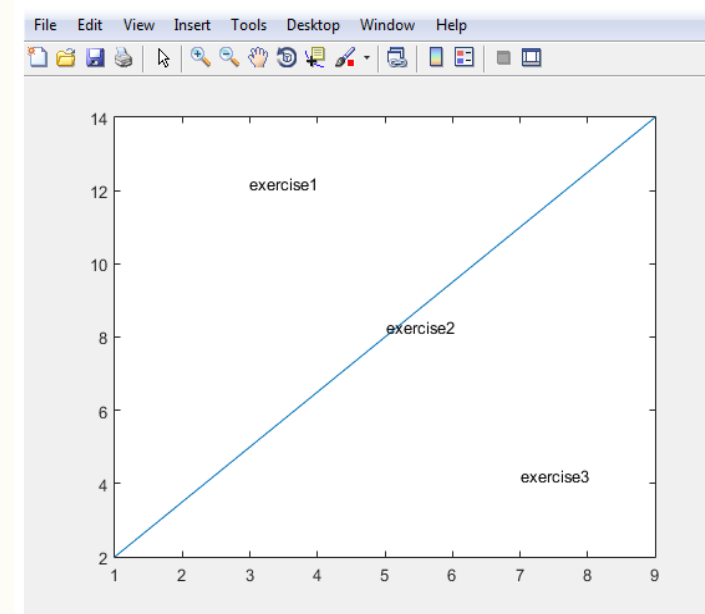
این امکان را فراهم می کند که نوشته ها را در هر جای صفحه که خواستیم، حک کند.

آشنایی با ترسیم نمودار در متلب

```
Editor - F:\mat\t1.m
t1.m x +
1 - clc
2 - clear
3 - x=1:2:10;
4 - y=2:3:15;
5 - plot(x,y)
6 - gtext({'exercise1','exercise2','exercise3'})
```



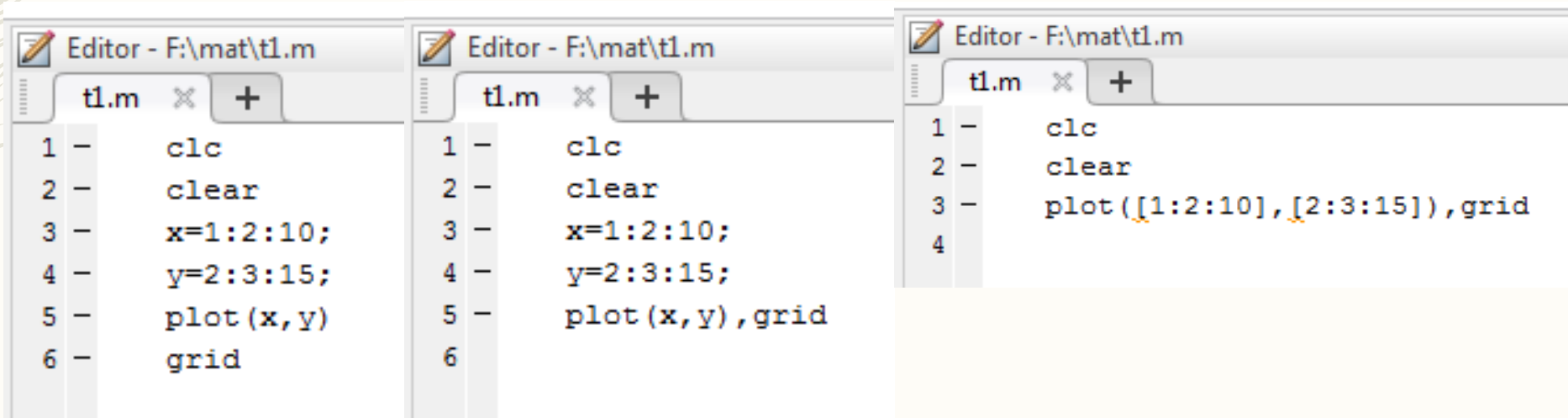
```
Editor - F:\mat\t1.m
t1.m x +
1 - clc
2 - clear
3 - x=1:2:10;
4 - y=2:3:15;
5 - plot(x,y)
6 - gtext({'exercise1','exercise2','exercise3'})
```



آشنایی با ترسیم نمودار در متلب

تابع `gread`:

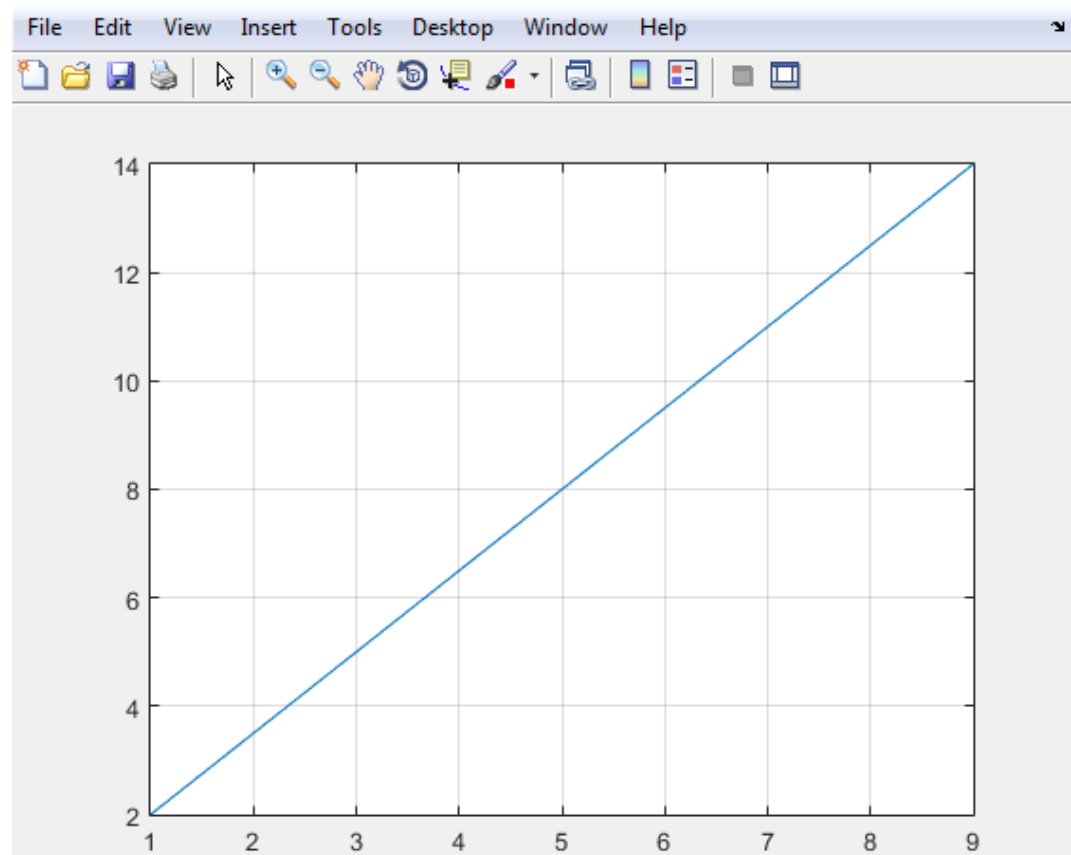
صفحه ی `plot` را به صورت شطرنجی نمایش می دهد.



The image displays three sequential screenshots of the MATLAB Editor window, showing the development of a script to create a grid plot. Each window has a title bar 'Editor - F:\mat\t1.m' and a tab 't1.m'.

- First Screenshot:** The script contains the following lines:

```
1 - clc
2 - clear
3 - x=1:2:10;
4 - y=2:3:15;
5 - plot(x,y)
6 - grid
```
- Second Screenshot:** The script is identical to the first, but the line numbers are visible on the left margin.
- Third Screenshot:** The script is identical to the second, but the line numbers are visible on the left margin.



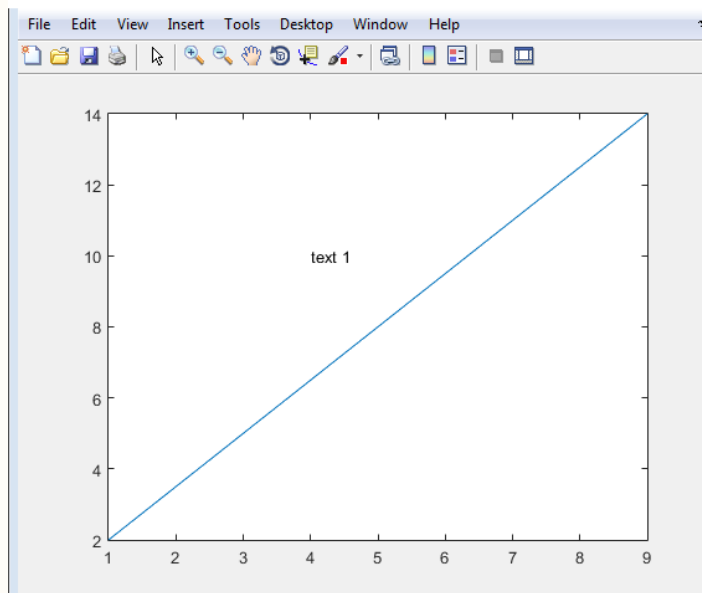
تابع text :

در این تابع مختصاتی که می خواهیم
مطلب مورد نظر را در آن جا نمایش داده
شود، می دهیم و سپس مطلب را در ادامه آن
می نویسیم.

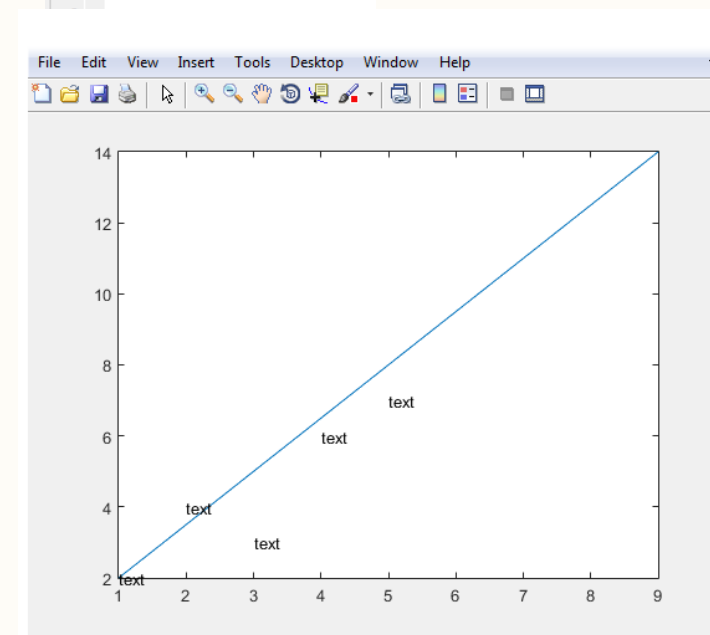
10

آشنایی با ترسیم نمودار در متلب

```
Editor - F:\mat\t1.m  
1 - clc  
2 - clear  
3 - x=1:2:10;  
4 - y=2:3:15;  
5 - plot(x,y)  
6 - text(4,10,'text 1')
```



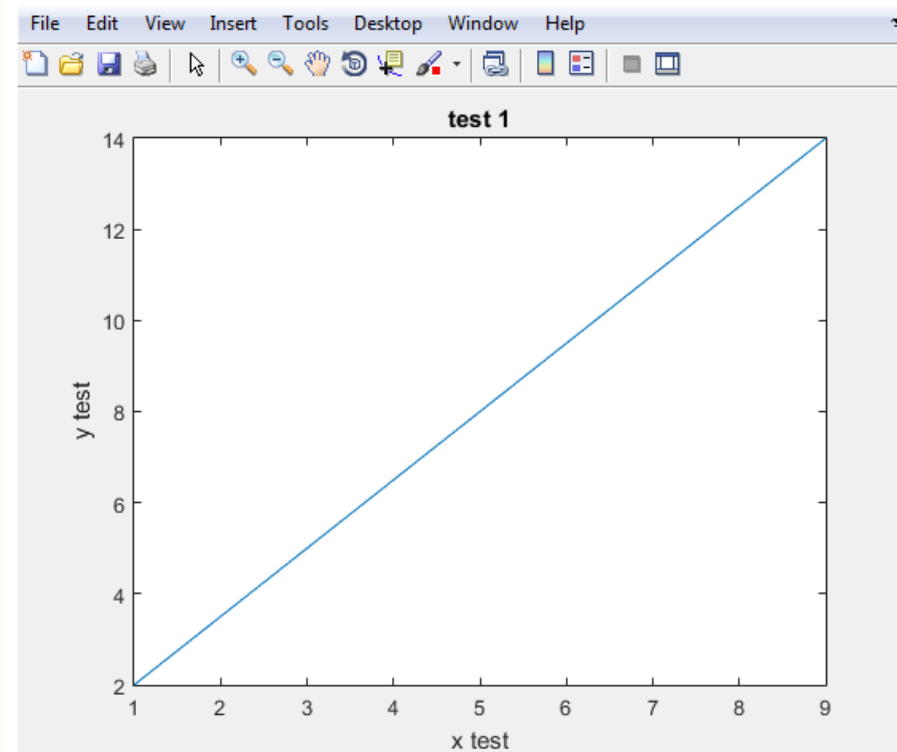
```
Editor - F:\mat\t1.m  
1 - clc  
2 - clear  
3 - x=1:2:10;  
4 - y=2:3:15;  
5 - plot(x,y)  
6 - xx=[1 2 3 4 5];  
7 - yy=[2 4 3 6 7];  
8 - text(xx,yy,'text')
```



نکته :

توابع `title` ، `xlabel` ، `ylabel` به ترتیب خود `plot`، محور افقی و محور عمودی را نام گذاری می کنند.

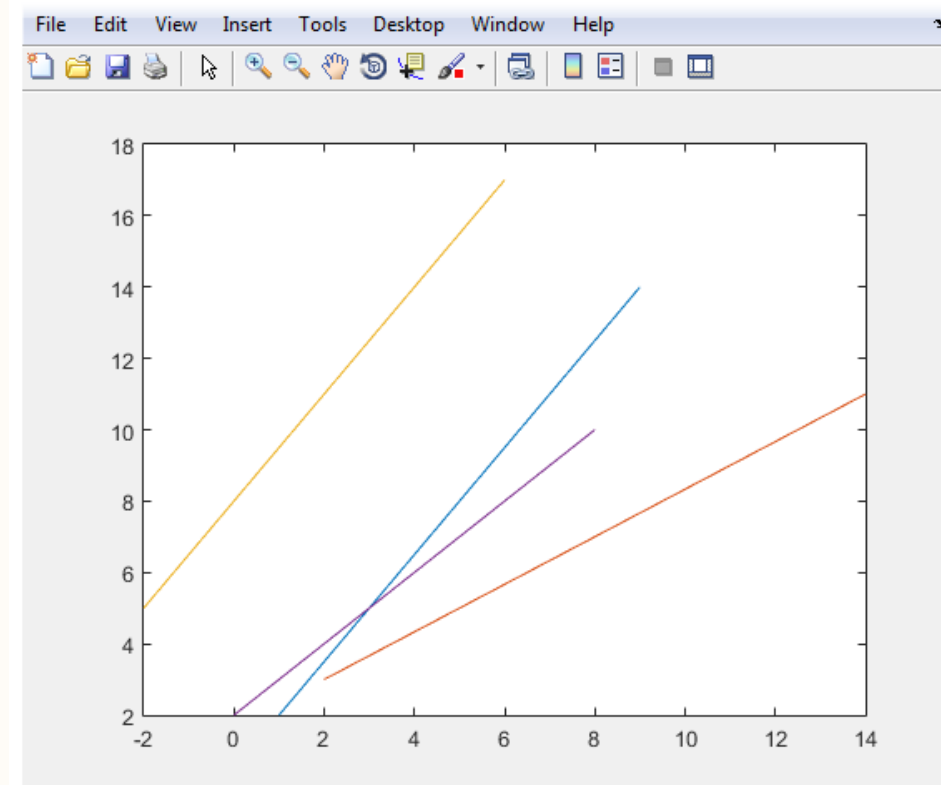
```
Editor - F:\mat\t1.m
t1.m x +
1 -   clc
2 -   clear
3 -   x=1:2:10;
4 -   y=2:3:15;
5 -   plot(x,y)
6 -   title('test 1');
7 -   xlabel('x test');
8 -   ylabel('y test');
9
```



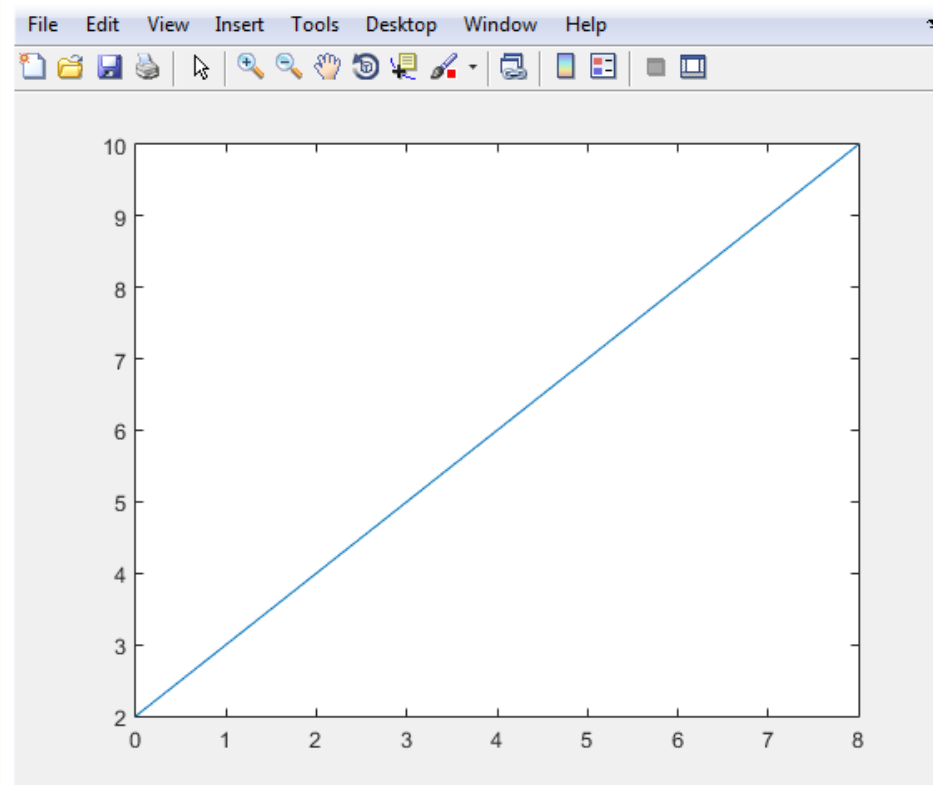
تابع hold :

این تابع امکان نمایش چند نمودار را در یک plot فراهم می کند و به صورت hold on و hold off به کار می رود.

```
Editor - F:\mat\t1.m
t1.m x +
1 -   clc
2 -   clear
3 -   x=1:2:10;
4 -   y=2:3:15;
5 -   plot(x,y)
6 -   hold on
7 -   plot(y,x+2)
8 -   plot(x-3,y+3)
9 -   plot(x-1,x+1)
```




```
Editor - F:\mat\tl.m
tl.m x +
1 -   clc
2 -   clear
3 -   x=1:2:10;
4 -   y=2:3:15;
5 -   plot(x,y)
6 -   hold on
7 -   plot(y,x+2)
8 -   plot(x-3,y+3)
9 -   hold off
10 -  plot(x-1,x+1)
```




در `help` ، `plot` را جست و جو کرده
و در آن قسمت، از سرتیتز `Description`
گزینه `LineSpec` را بزنید، صفحه ای باز می شود
که در آن، ویژگی هایی از خط نمودار را نشان
می دهد که امکان تغییر آن ها برای ما وجود دارد.

آشنایی با ترسیم نمودار در متلب

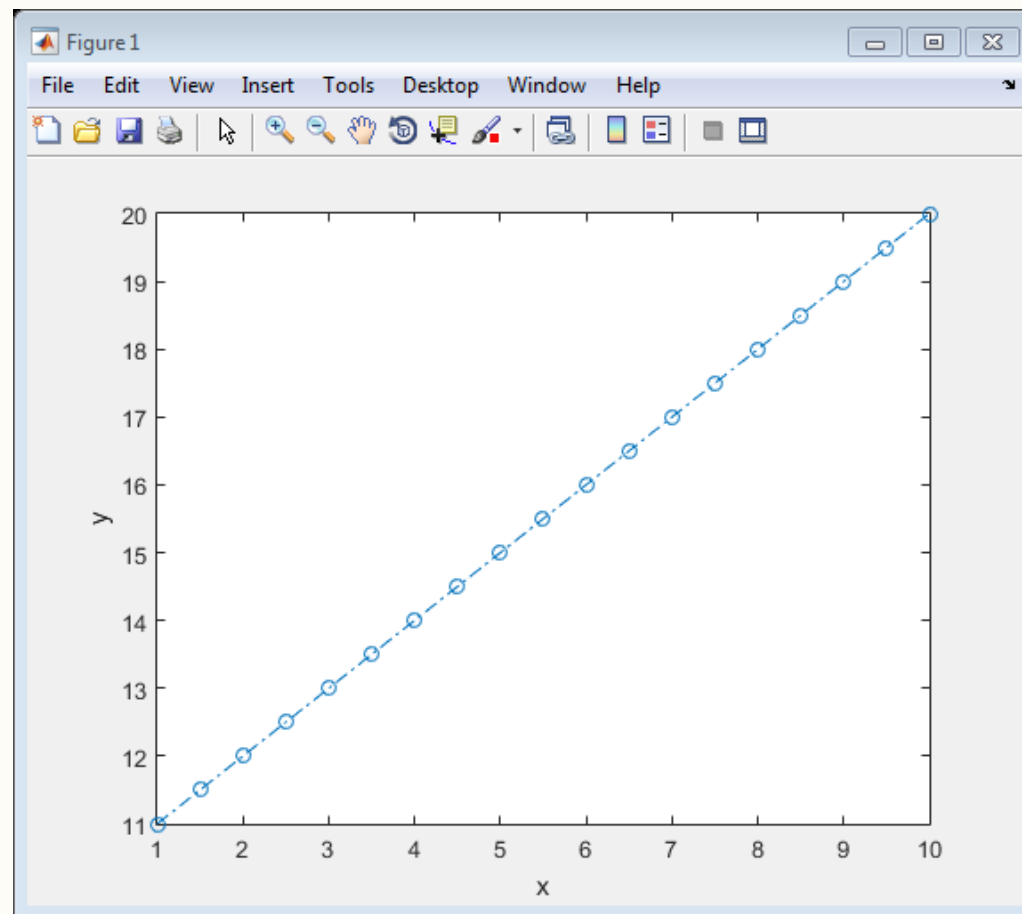
Specifier	Line Style
-	Solid line (default)
--	Dashed line
:	Dotted line
-.	Dash-dot line


Specifier	Marker
o	Circle
+	Plus sign
*	Asterisk
.	Point
x	Cross
s	Square
d	Diamond
^	Upward-pointing triangle
v	Downward-pointing triangle
>	Right-pointing triangle
<	Left-pointing triangle
p	Pentagram
h	Hexagram



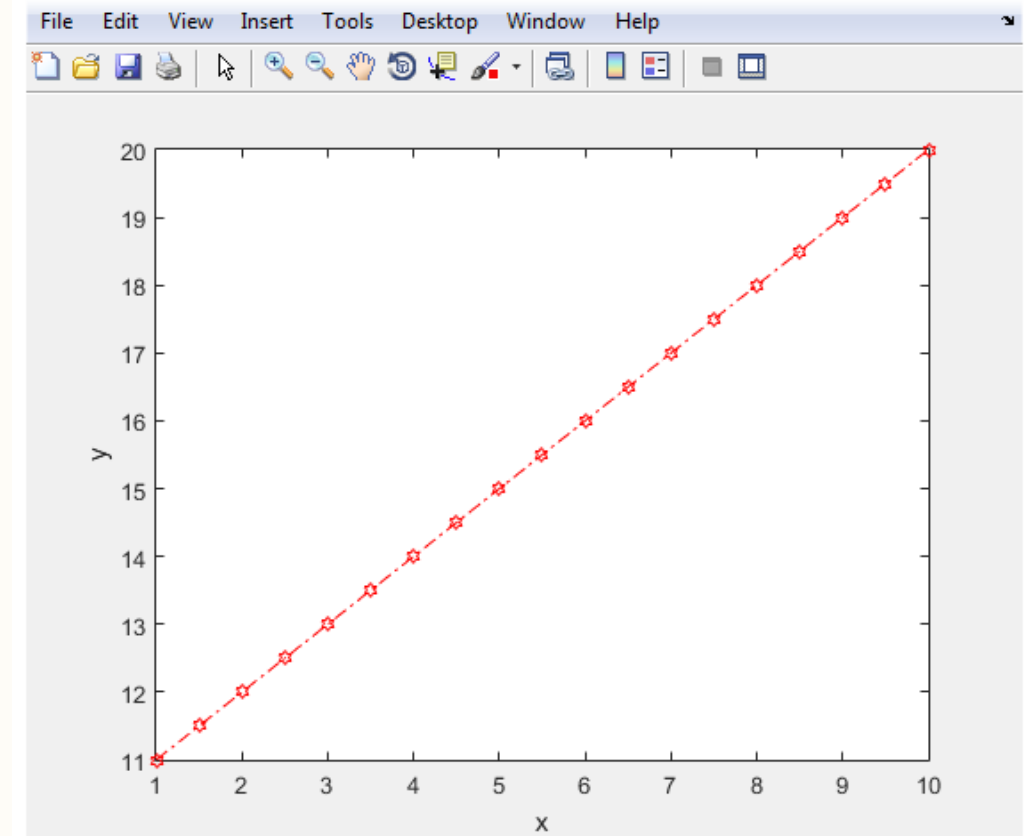
Specifier	Color
y	yellow
m	magenta
c	cyan
r	red
g	green
b	blue
w	white
k	black


```
Editor - F:\galase9\t1.m
t1.m x +
1 - clear
2 - clc
3 - x=1:0.5:10;
4 - y=11:0.5:20;
5 - plot(x,y,'-o')
6 - xlabel('x')
7 - ylabel('y')
```



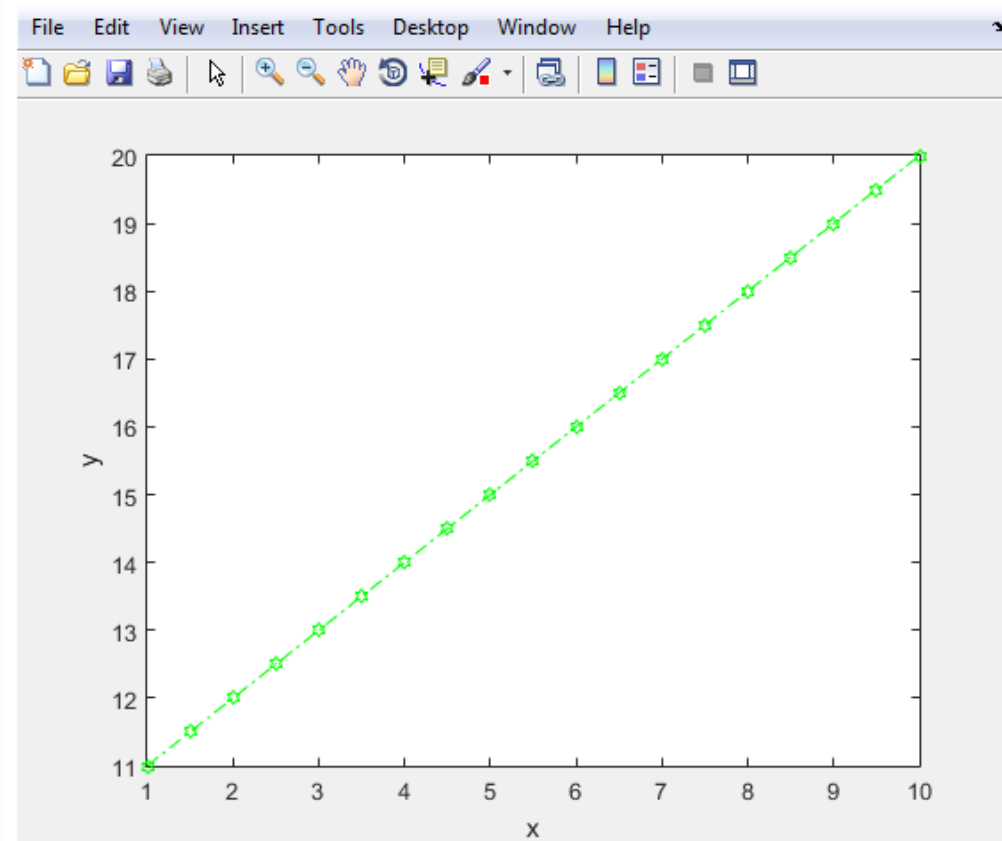


```
Editor - F:\galase9\t1.m
t1.m x +
1 - clear
2 - clc
3 - x=1:0.5:10;
4 - y=11:0.5:20;
5 - plot(x,y,'-.hr')
6 - xlabel('x')
7 - ylabel('y')
```

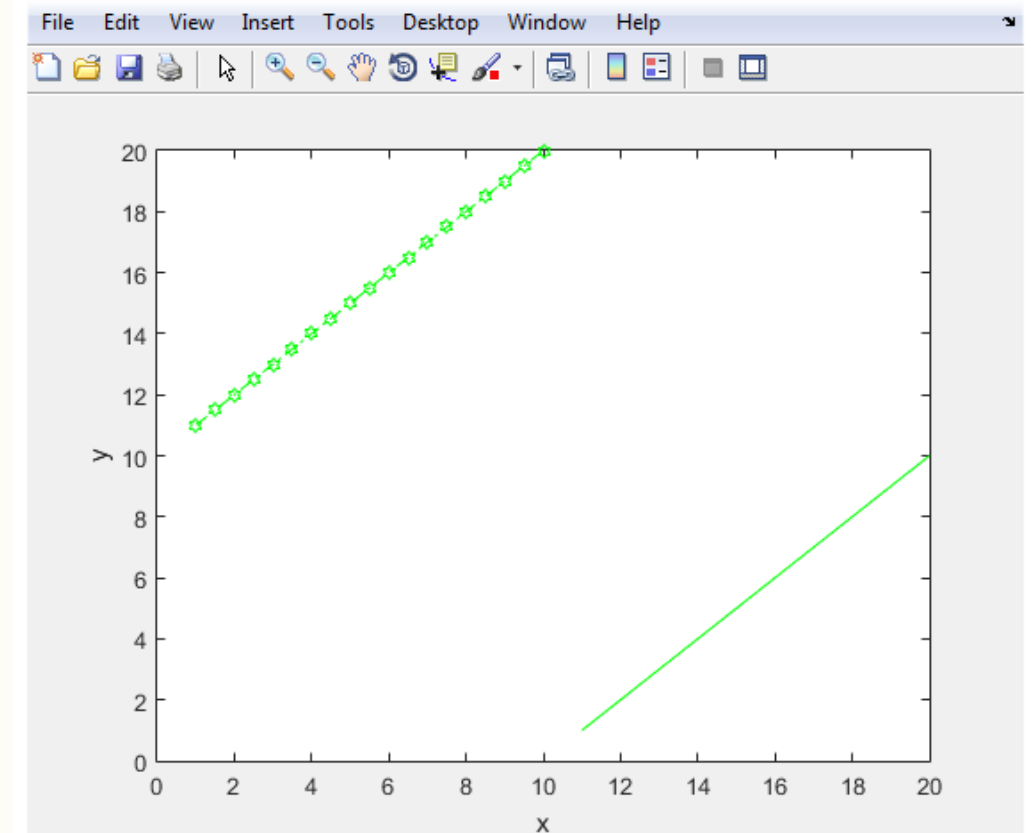




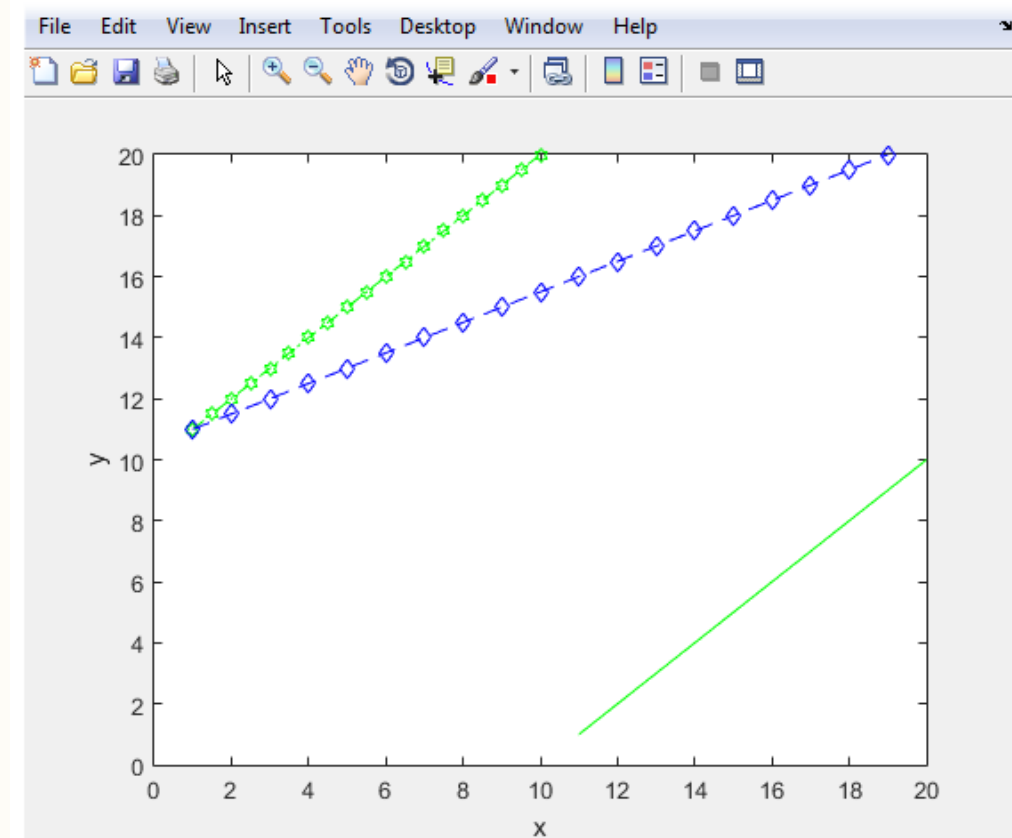
```
Editor - F:\galase9\t1.m
t1.m x +
1 - clear
2 - clc
3 - x=1:0.5:10;
4 - y=11:0.5:20;
5 - plot(x,y,'-.hg')
6 - xlabel('x')
7 - ylabel('y')
```



```
Editor - F:\galase9\t1.m
t1.m x +
1 - clear
2 - clc
3 - x=1:0.5:10;
4 - y=11:0.5:20;
5 - plot(x,y,'-.hg',y,x,'g')
6 - xlabel('x')
7 - ylabel('y')
```



```
Editor - F:\galase9\t1.m
t1.m x +
1 - clear
2 - clc
3 - x=1:0.5:10;
4 - y=11:0.5:20;
5 - plot(x,y,'-.hg',y,x,'g')
6 - hold on
7 - plot(y,'d--b')
8 - xlabel('x')
9 - ylabel('y')
```



آشنایی با ترسیم نمودار در متلب

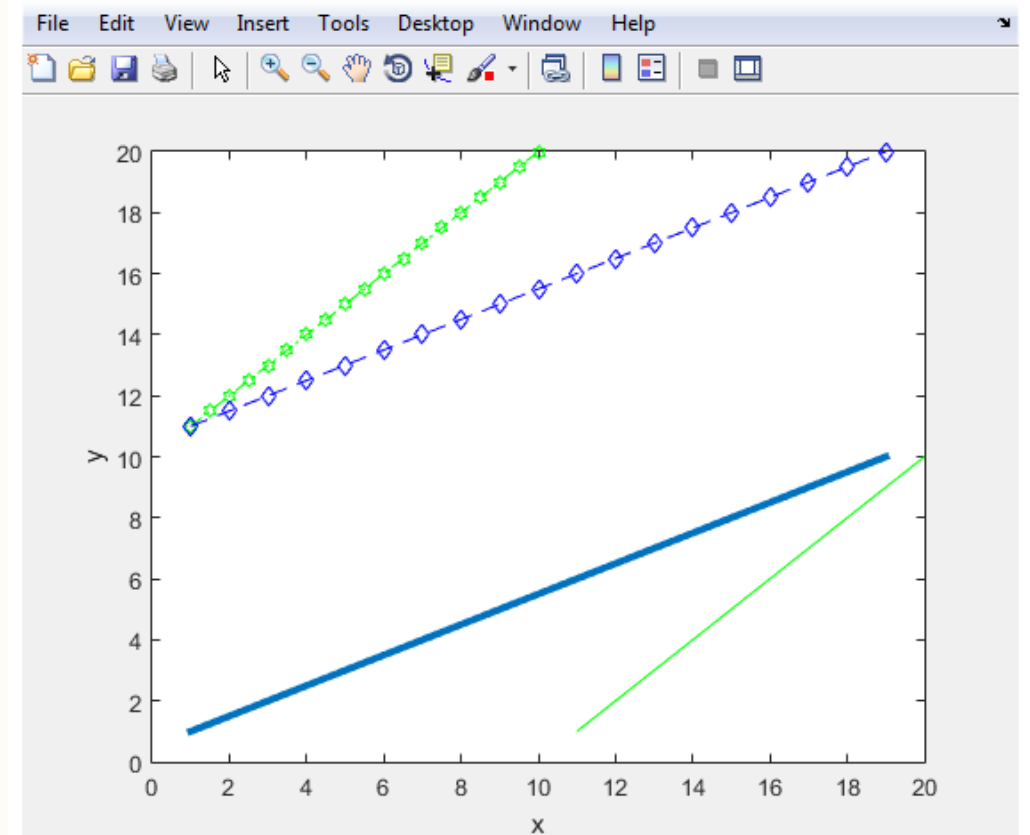
– linewidth برای نشان دادن ضخامت خط

– markeregecolor برای نشان دادن رنگ ستاره

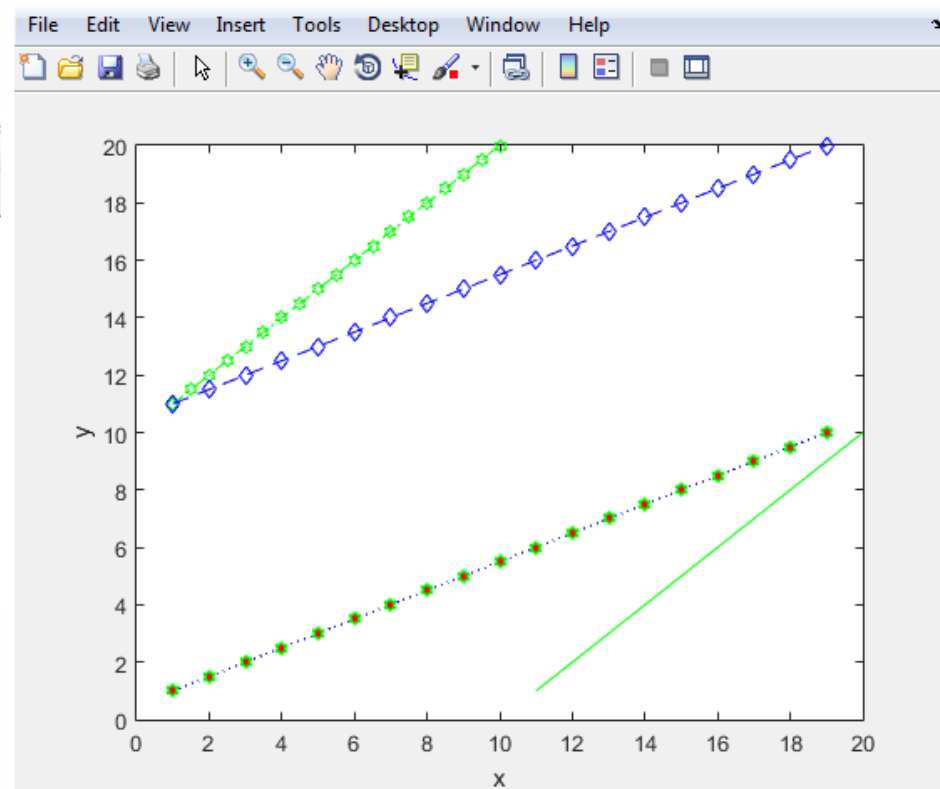
– markerfacecolor برای نشان دادن رنگ داخل ستاره

به کار می رود.....

```
Editor - F:\galase9\t1.m
t1.m x +
1 - clear
2 - clc
3 - x=1:0.5:10;
4 - y=11:0.5:20;
5 - plot(x,y,'-.hg',y,x,'g')
6 - hold on
7 - plot(y,'d--b')
8 - plot(x,'linewidth',3)
9 - xlabel('x')
10 - ylabel('y')
```



```
Editor - F:\galase9\t1.m
t1.m x +
1 - clear
2 - clc
3 - x=1:0.5:10;
4 - y=11:0.5:20;
5 - plot(x,y,'-.hg',y,x,'g')
6 - hold on
7 - plot(y,'d--b')
8 - plot(x,':hb','linewidth',1,'MarkerEdgeColor','g','MarkerFaceColor','r')
9 - xlabel('x')
10 - ylabel('y')
```

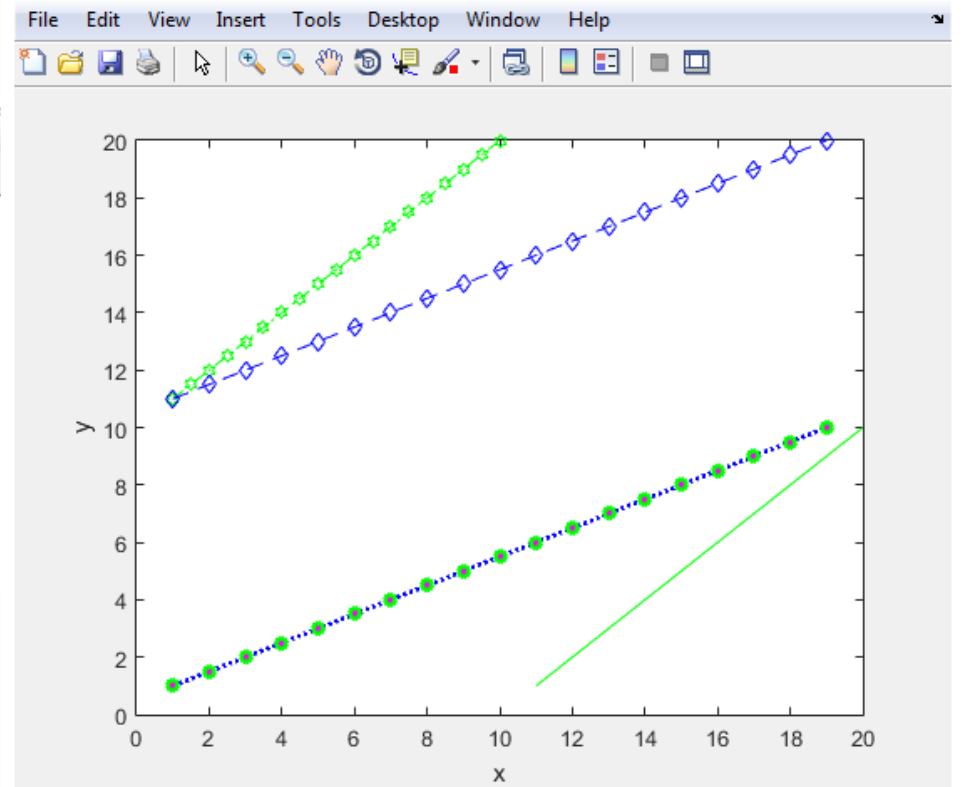


–[1 0 1] نشان دهنده رنگ قرمز هست

–در کل هشت رنگ داریم پس می توان به صورت باینری رنگ ها را به کار برد



```
Editor - F:\galase9\t1.m
t1.m x +
1 - clear
2 - clc
3 - x=1:0.5:10;
4 - y=11:0.5:20;
5 - plot(x,y,'-.hg',y,x,'g')
6 - hold on
7 - plot(y,'d--b')
8 - plot(x,':hb','linewidth',2,'MarkerEdgeColor','g','MarkerFaceColor',[1 0 1])
9 - xlabel('x')
10 - ylabel('y')
11
```



: Subplot

امکان نمایش چند نمودار به صورت همزمان را عملی می کند.

`subplot(2,2,1)`=(اولین جدول , تعداد ستون , تعداد سطر) subplot

یعنی چهار جدول داریم که در دو سطر و دو ستون قرار دارد و از این چهار نمودار اولین نمودار را انتخاب کردیم.

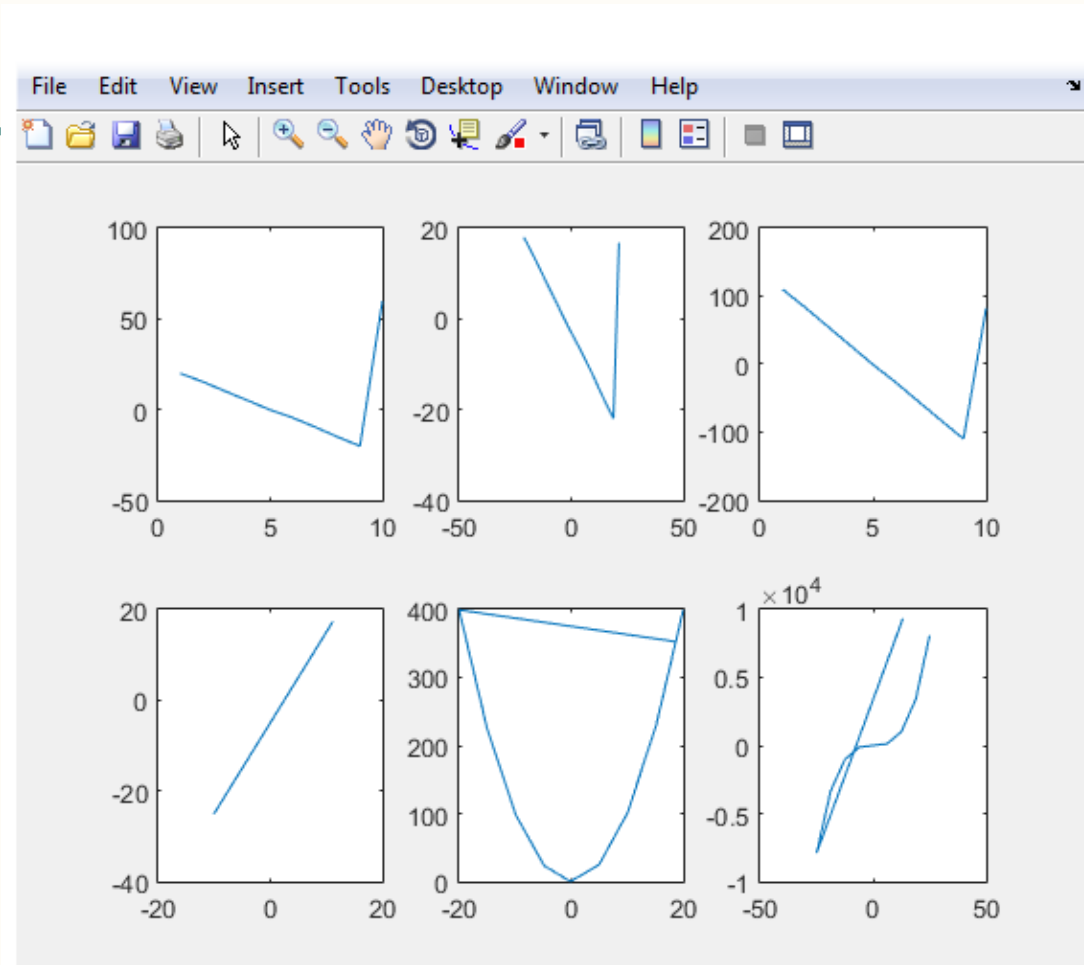
: Ginput


یک ورودی گرافیکی است که با آن مقادیر متغیرهای نمودار را وارد می کنیم.

`[x,y]=ginput()` با وارد کردن این دستور figure اجرا می شود و مقادیر مورد نظر را با کلیک بر روی figure وارد می کنیم و با زدن enter خواندن مقادیر تمام می شود.

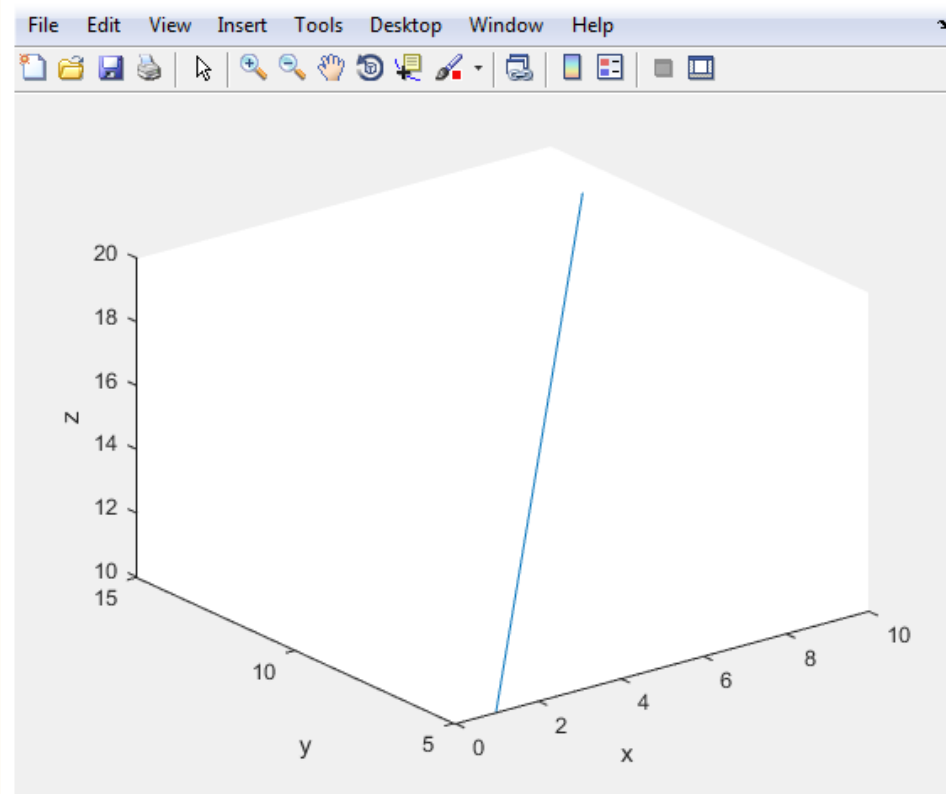
`[x,y]=ginput(3)` با وارد کردن این دستور، صفحه ی figure باز می شود و فقط سه مقدار ورودی را می توانیم با کلیک بر روی صفحه ی figure وارد کنیم.

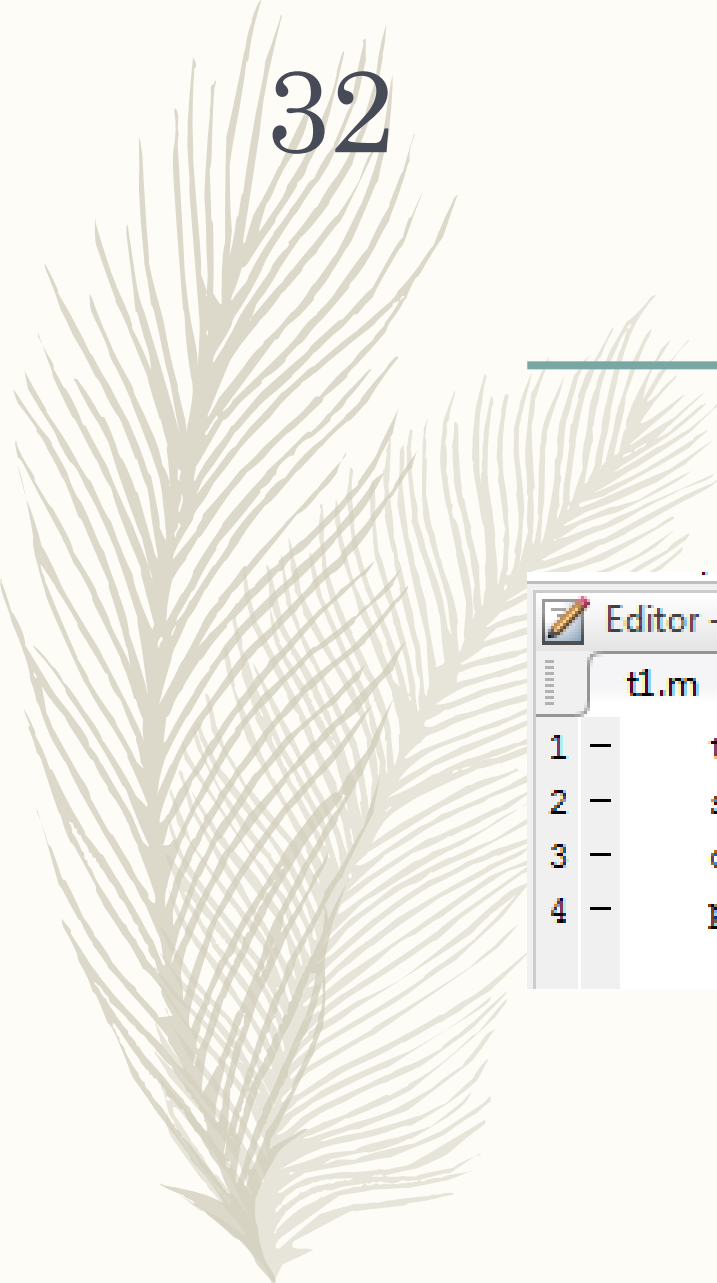
```
Editor - F:\galase9\t1.m
t1.m
1 - clear
2 - clc
3 - axis([-25,25,-23,23])
4 - [x,y]=ginput(10);
5 - subplot(2,3,1)
6 - plot(x+2.*y)
7 - subplot(2,3,2)
8 - plot(x-1,y-2)
9 - subplot(2,3,3)
10 - plot(y*5-x/2)
11 - subplot(2,3,4)
12 - plot(x/2,x-5)
13 - subplot(2,3,5)
14 - plot(y,y.^2)
15 - subplot(2,3,6)
16 - plot(y-x/4,y.*x.^2)
17
```



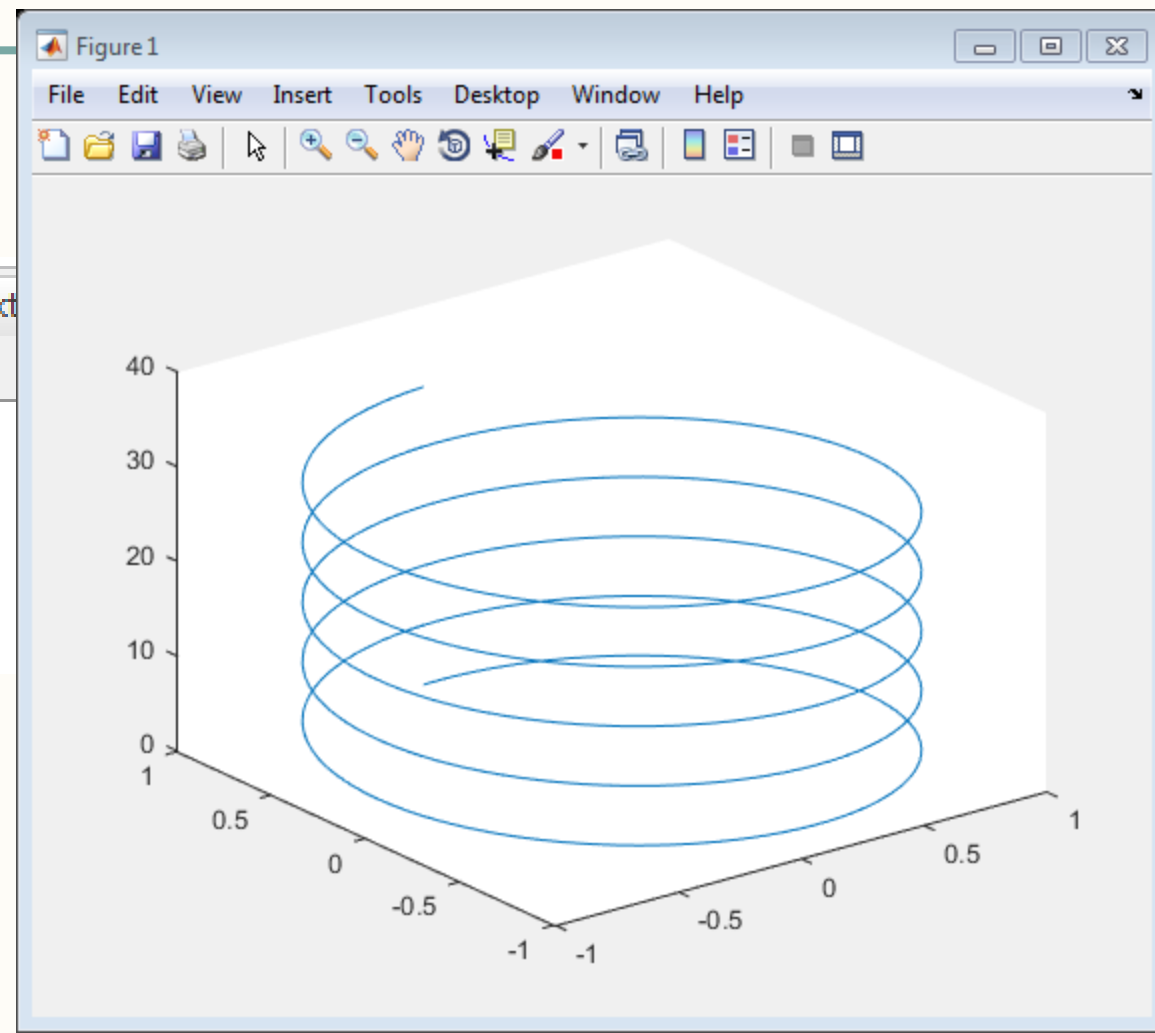


```
Editor - F:\galase9\t1.m
t1.m x +
1 - clear
2 - clc
3 - x=1:0.5:10;
4 - y=5:0.5:14;
5 - z=10:0.5:19;
6 - plot3(x,y,z)
7 - xlabel('x')
8 - ylabel('y')
9 - zlabel('z')
```





```
Editor - C:\Users\Hosein Abdollahpoor\Desktop
t1.m x +
1 - t = 0:pi/50:10*pi;
2 - st = sin(t);
3 - ct = cos(t);
4 - plot3(st,ct,t)
```



تابع `meshgrid` :

از اطلاعاتی که به آن می دهیم یک
ماتریس در ابعاد بزرگتر به ما می دهد.

تابع `peaks` :

X و Y را به صورت پارامتری می گیرد و Z را به صورت
پارامتری به ما می دهد.

Command Window

New to MATLAB? See resources for [Getting Started](#).

>> [x y]=meshgrid(-2:1:2)

x =

-2	-1	0	1	2
-2	-1	0	1	2
-2	-1	0	1	2
-2	-1	0	1	2
-2	-1	0	1	2

y =

-2	-2	-2	-2	-2
-1	-1	-1	-1	-1
0	0	0	0	0
1	1	1	1	1
2	2	2	2	2

Command Window

New to MATLAB? See resources for [Getting Started](#).

>> [X,Y] = meshgrid(1:3,10:14)

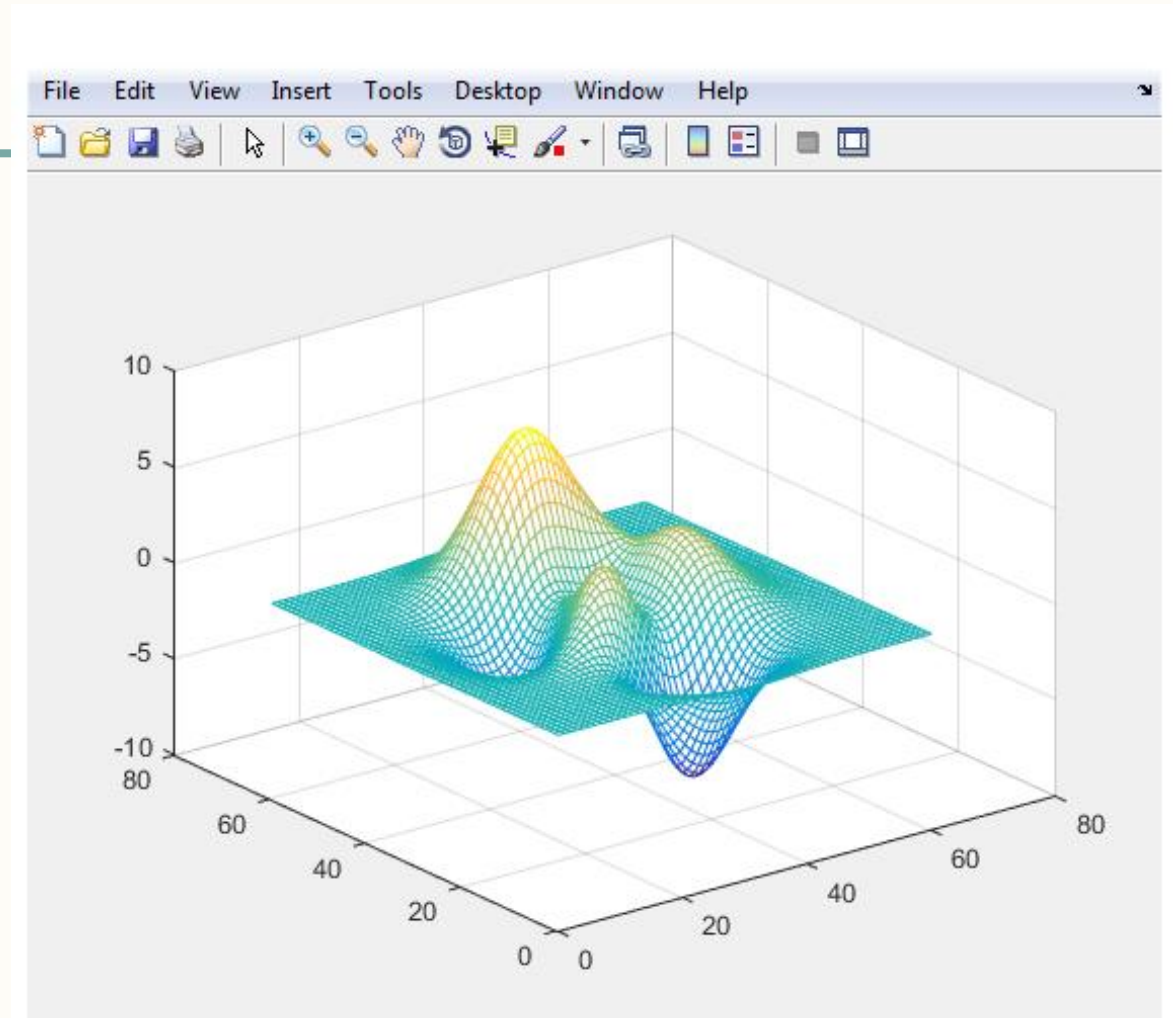
X =

1	2	3
1	2	3
1	2	3
1	2	3
1	2	3

Y =

10	10	10
11	11	11
12	12	12
13	13	13
14	14	14

```
Editor - F:\galase10\t1.m
t1.m x +
1 - clear
2 - clc
3 - [x y]=meshgrid(-3:0.1:3);
4 - z=peaks(x,y);
5 - mesh(z)
6 |
```



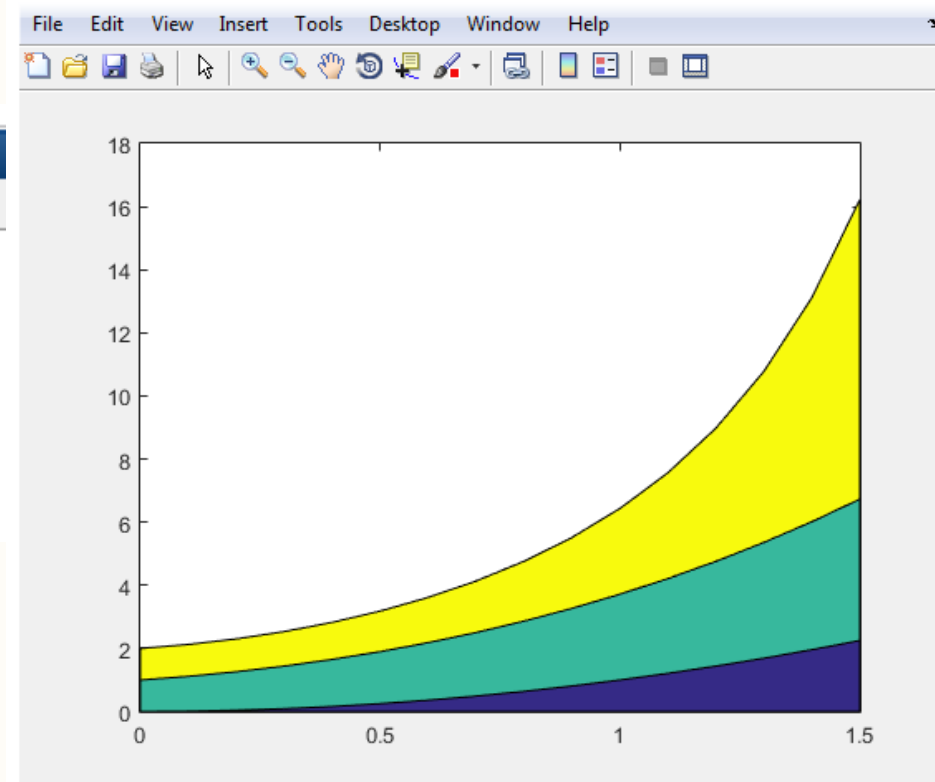
✓ Exp برای نمایش توابع نمایی مورد استفاده قرار می گیرد.


✓ area برای نمایش نمودار به صورت رنگی استفاده می شود.

✓ bar برای نمایش نمودار به صورت میله ای مورد استفاده قرار می گیرد.

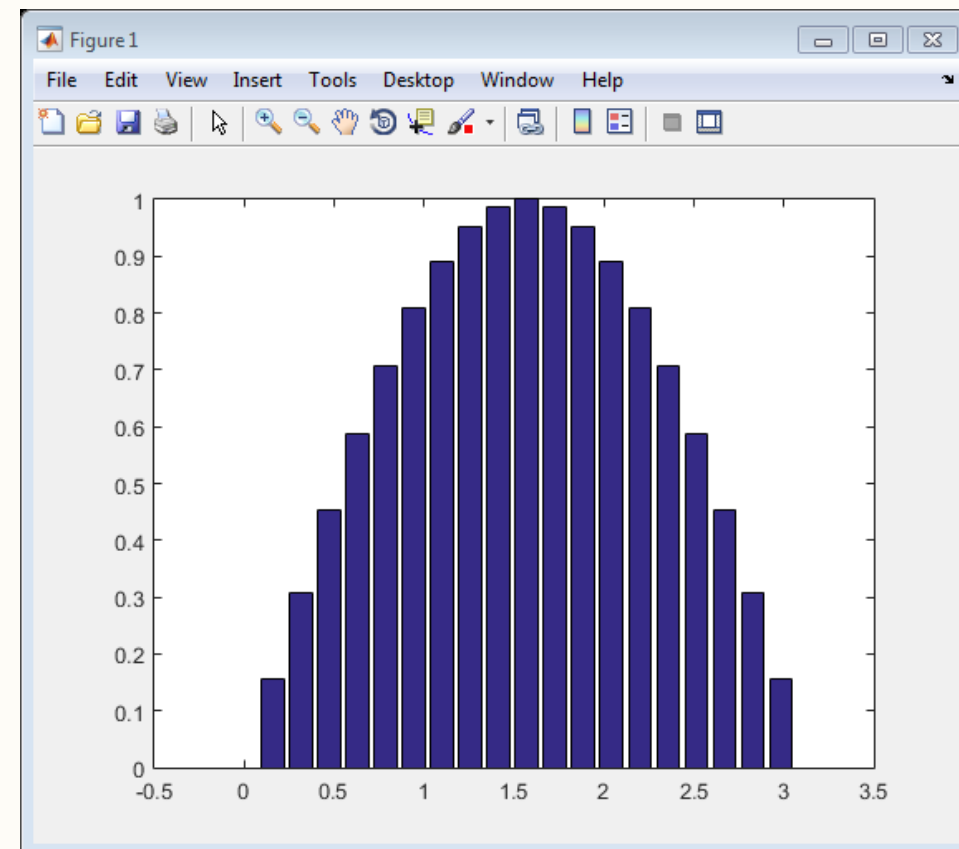


```
Editor - F:\galase10\t1.m
t1.m x +
1 - clear;
2 - clc
3 - x=0:.1:1.5;
4 - area(x, [(x.^2)' exp(x)' exp(x.^2)'])
5 -
```





```
Editor - F:\galase10\t1.m
t1.m x +
1 - clear;
2 - clc
3 - x=0:pi/20:pi;
4 - bar(x,sin(x))
5 |
```

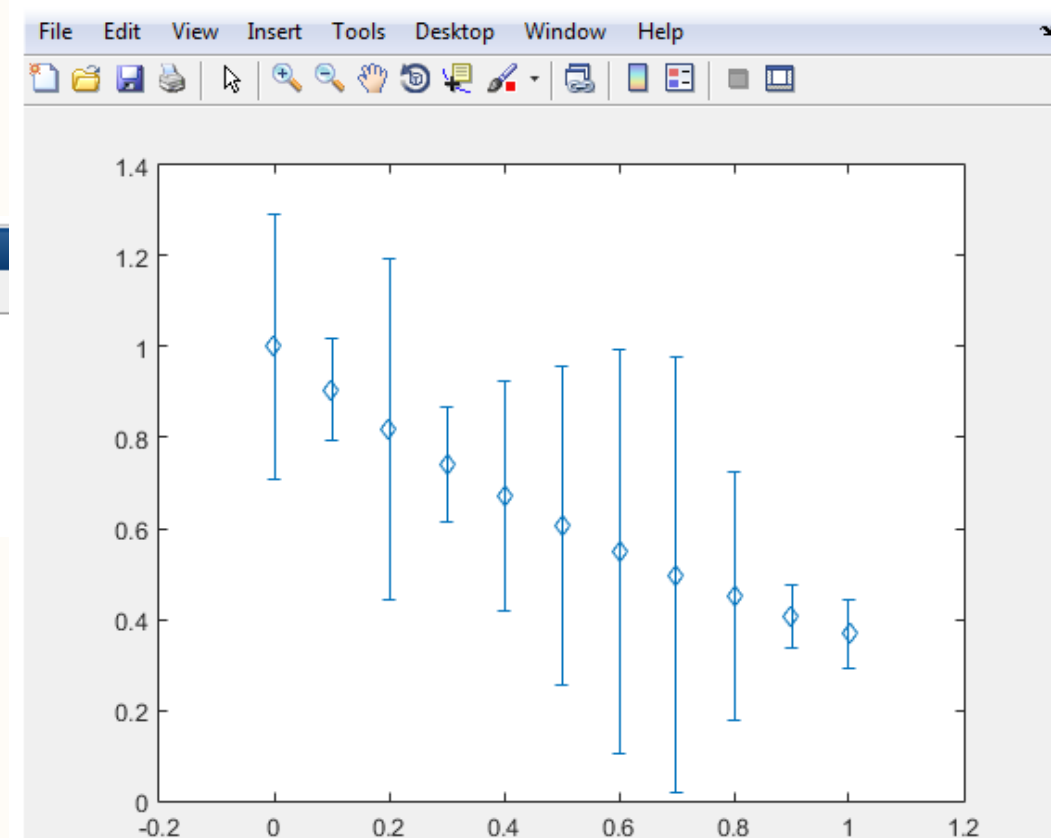


تابع errorbar :

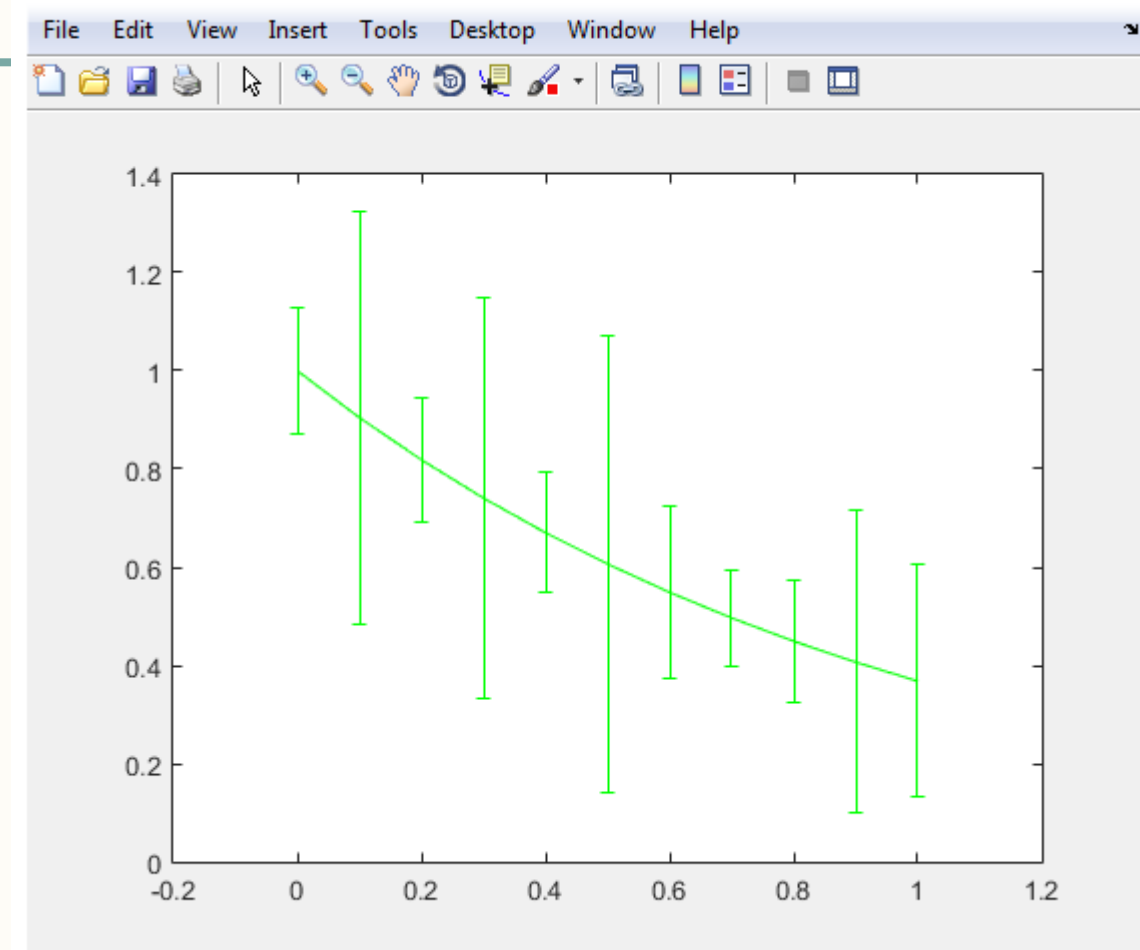
خطاها را به ما اعلام می کند و
نشان می دهد که کجا ها خطا
وجود دارد.




```
Editor - F:\galase10\t1.m
t1.m x +
1 - clear;
2 - clc
3 - x=0:0.1:1;
4 - errorbar(x,exp(-x),0.5*rand(1,length(x)),'d')
5 -
```



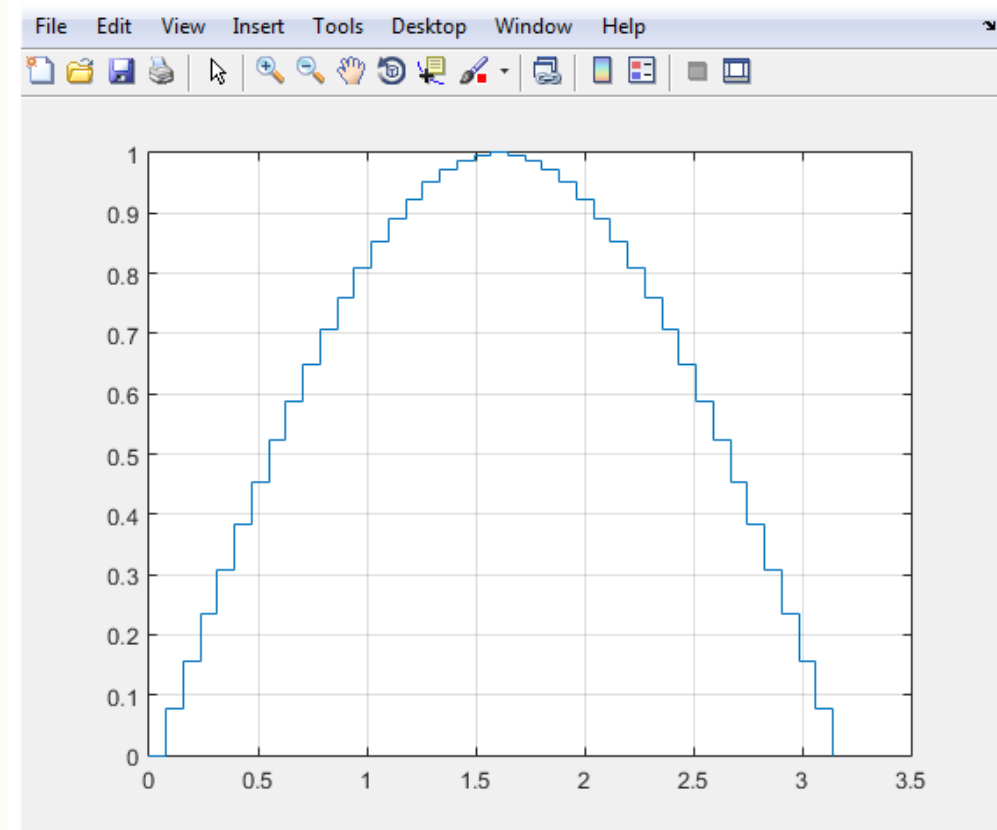
```
Editor - F:\galase10\t1.m  
t1.m x +  
1 - clear;  
2 - clc  
3 - x=0:0.1:1;  
4 - errorbar(x,exp(-x),0.5*rand(1,length(x)), 'g')  
5 |
```



تابع stairs :

- نمودار رسم شده را به صورت پلکانی نمایش می دهد.
- برای نمایش دادن توابع به صورت گسسته مورد استفاده قرار می گیرد.

```
Editor - F:\galase10\t1.m  
t1.m x +  
1 - clear;  
2 - clc  
3 - x=0:pi/40:pi;  
4 - stairs(x,sin(x)),grid  
5 |
```



```
Editor - F:\galase10\t1.m  
t1.m x +  
1 - clear;  
2 - clc  
3 - t=0:pi/50:2*pi;  
4 - r=exp(-.05*t);  
5 - stem(r.*sin(t),r.*cos(t)),grid  
6 |
```

