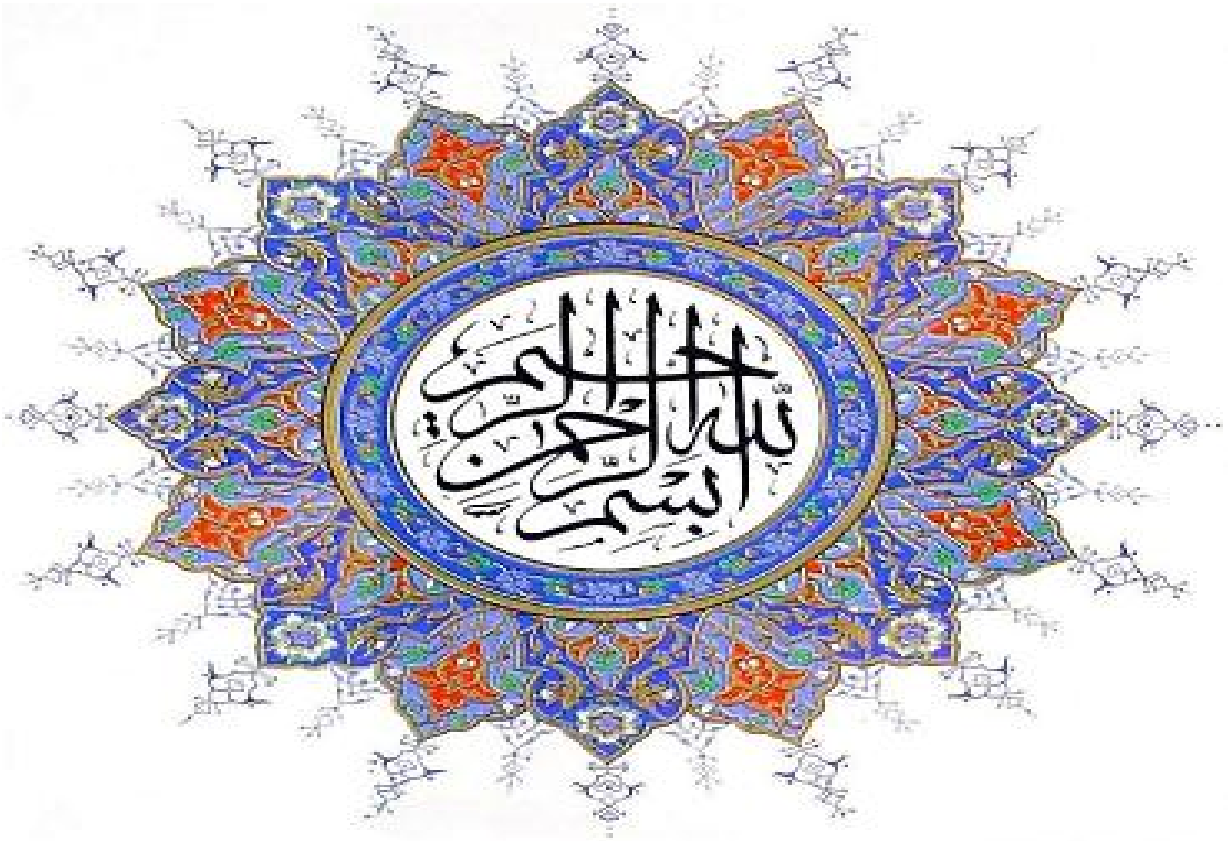
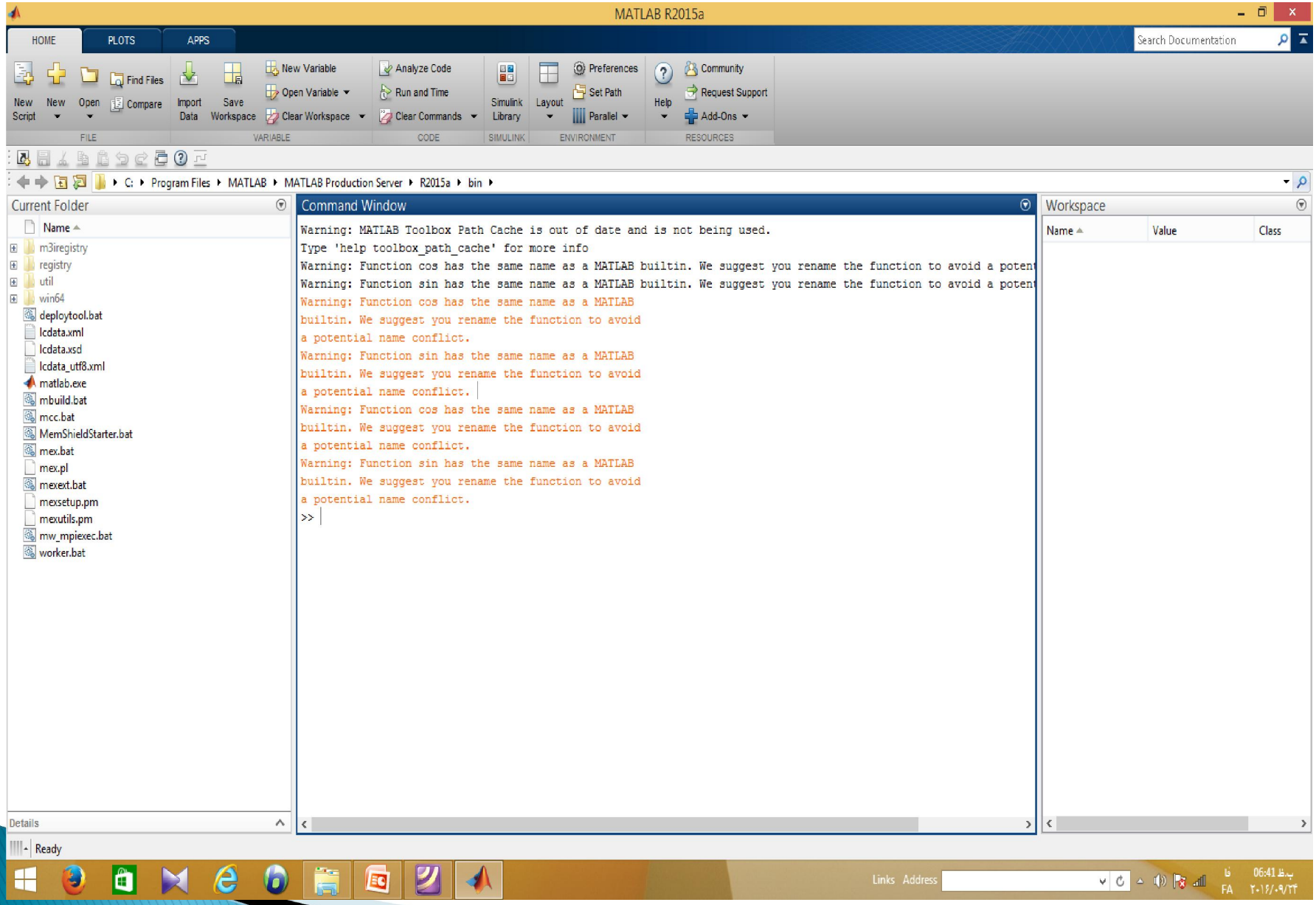

بسم الله الرحمن الرحيم

آموزش نرم افزار MATLAB

معرفی

- MATLAB نرم افزاری برای انجام کارهای ریاضی ، آماری ، مهندسی و... می باشد.
- MATLAB=MATrix Laboratory
- اولین نگارشهای آن در دانشگاههای نیومکزیکو و استانفورد در سال 1970 در جهت حل مسائل تئوری ماتریسها، جبرخطی و آنالیزعددی بوجود آمد.
- مانند جاوا بصورت interpreted است. برای اجرا خط به خط اجرا می شود.
- این زبان autocomplete است. یعنی با تایپ قسمتی از دستور و استفاده از کلید tab می توان دستور مورد نیاز را انتخاب و کامل نمود.
- این نرم افزار case-sensitive است یعنی متغیر با حروف کوچک و متغیر با حروف بزرگ باهم فرق دارند.


MATLAB R2015a
HOME
PLOTS
APPS
Search Documentation
New Script
New
Open
Find Files
Compare
Import Data
Save Workspace
New Variable
Open Variable
Clear Workspace
Analyze Code
Run and Time
Clear Commands
Simulink Library
Layout
Preferences
Set Path
Parallel
Help
Community
Request Support
Add-Ons
FILE
VARIABLE
CODE
SIMULINK
ENVIRONMENT
RESOURCES
C: ▸ Program Files ▸ MATLAB ▸ MATLAB Production Server ▸ R2015a ▸ bin ▸
Current Folder
Name
m3iregistry
registry
util
win64
deploytool.bat
lcdata.xml
lcdata.xsd
lcdata_utf8.xml
matlab.exe
mbuild.bat
mcc.bat
MemShieldStarter.bat
mex.bat
mex.pl
mexext.bat
mexsetup.pm
mexutils.pm
mw_mpiexec.bat
worker.bat
Details
Command Window
Warning: MATLAB Toolbox Path Cache is out of date and is not being used.
Type 'help toolbox_path_cache' for more info
Warning: Function cos has the same name as a MATLAB builtin. We suggest you rename the function to avoid a potent
Warning: Function sin has the same name as a MATLAB builtin. We suggest you rename the function to avoid a potent
Warning: Function cos has the same name as a MATLAB
builtin. We suggest you rename the function to avoid
a potential name conflict.
Warning: Function sin has the same name as a MATLAB
builtin. We suggest you rename the function to avoid
a potential name conflict.
Warning: Function cos has the same name as a MATLAB
builtin. We suggest you rename the function to avoid
a potential name conflict.
Warning: Function sin has the same name as a MATLAB
builtin. We suggest you rename the function to avoid
a potential name conflict.
>>
Workspace
Name
Value
Class
Ready
Links
Address
FA
06:41 ب.ظ
۲۰۱۶/۰۹/۲۴

پنجره های متلب

command window – 1

پنجره دستور ...که می توانیم همه دستورات متلب را ، البته به صورت سطری(فقط یک دستور) در آن اجرا کنیم و همینطور پاسخ اجرای دستورات در اینجا نمایش داده می شود .

command history –2

پنجره ای است که همه دستورات اجرا شده در command window را بایگانی می کند .

work space – 3

مکانی است که همه پا رامترها وماتریسهای تعریف شده در آن نگهداری می شود .

current folder- 4

نمایش فایلهایی که در مسیر جاری قرار دارند.

کار در command window

- Command window یا پنجره فرمان توانایی انجام همه دستورات متلب را دارد و همچنین پاسخ همه دستورات و برنامه های اجرا شده در این پنجره نمایش داده می شود.
- وقتی متلب را باز کردید تا وقتی در Command window علامت >> ظاهر نشده است، کامپیوتر آماده نیست و باید منتظر شد.
- وارد کردن دستوری در این پنجره بدین صورت است که دستور را مقابل >> می نویسیم و Enter می کنیم

مقدمات کار در متلب

- ساده ترین کارهایی که با متلب میتوان انجام داد همان اعمالی است که یک ماشین حساب معمولی انجام میدهد. جمع، تفریق، ضرب و تقسیم:

```
>>2+4-1
ans=
      5
```

- راه دیگر انجام محاسبات این است که مقادیر را در چند متغیر ذخیره کرده و روی متغیرها عملیات محاسباتی را انجام دهیم:

```
>>a=2
a=
    2
>>b=3
b=
   3
>>a+b
ans=
      5
```

دقت کنید هرگاه نتیجه محاسبات در متغیری قرار داده نشود، نتیجه در متغیر ans قرار می گیرد.

قوانین نامگذاری متغیرها

- متلب نسبت به حروف کوچک و بزرگ حساس است.
- اسامی متغیرها حداکثر می تواند 31 کاراکتر باشد.
- اسامی متغیرها باید با حرف شروع شود.

- جز کلمات کلیدی متلب نباشد. برای شناسایی کلمات کلیدی متلب می توان از دستور iskeyword استفاده کرد. این تابع در صورتیکه عبارت یک کلمه کلیدی باشد مقدار یک ودر غیر این صورت مقدار صفر را برمی گرداند.

```
>>iskeyword('for')
ans=
    1
>>iskeyword('keyword')
ans=
      0
```

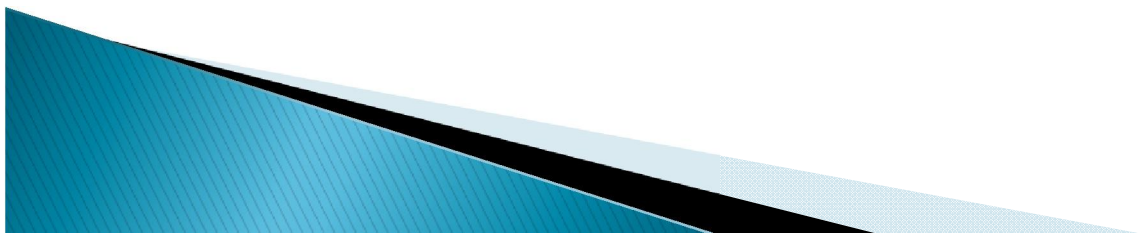
- تمام دستورات و عبارات کلیدی متلب با حروف کوچک نوشته می شود. بنابراین می توان اسامی آن ها را با حروف بزرگ برای نامگذاری متغیرها بکار برد.

در نامگذاری فایل ها در متلب ، بهتراست که از _ (under line) و حروف کوچک استفاده شود. مثال :

a_new

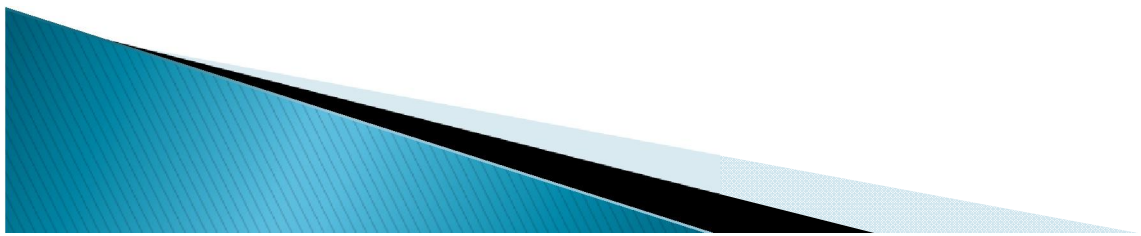
برخی از قابلیت های متلب

- نیازی به تعریف متغیرها نیست.
- همه متغیرها از نوع double(8 بایت) فرض میشوند.
- با دستور clear می توان متغیرها را از حافظه پاک کرد.
- دستور clc برای پاک کردن پنجره فرمان بکار میرود.
- دستور whos همه متغیرهای موجود در work space را با اطلاعاتی از قبیل اندازه، تعداد بایتها، کلاس و ... نشان میدهد.
- دستور who همه متغیرهای موجود را ... نشان میدهد.
- با دستور doc صفحه help متلب باز میشود.
- با دستور demo تعدادی از مثالهای متلب نمایش داده میشود.



روشهای گرد کردن اعداد

- fix گرد کردن به طرف صفر
- floor گرد کردن به طرف منهای بینهایت
- ceil گرد کردن به طرف مثبت بینهایت
- round گرد کردن به طرف نزدیکترین عدد صحیح



برخی از علائم و نشانه های پر کاربرد در متلب

NaN مبهم (not a number)

Pi $\pi = 3.14$

i و j $i = \sqrt{-1}$

eps (epsilon)

inf بینهایت

realmin ← کوچکترین مقدار

realmax ← بزرگترین مقدار

- در این نرم افزار متغیرها به صورت بردار میباشند.
- برای تعریف بردارهای سطری متغیر با علامت کروشه باز و بسته و نوشتن اعداد داخل آن تعریف میشود.

```
a=[1 2 3 4 5 6]
```

- برای جدا کردن عضوهای یک بردار سطری میتوان از کاما نیز استفاده کرد.

```
a=[1, 2, 3, 4, 5, 6]
```

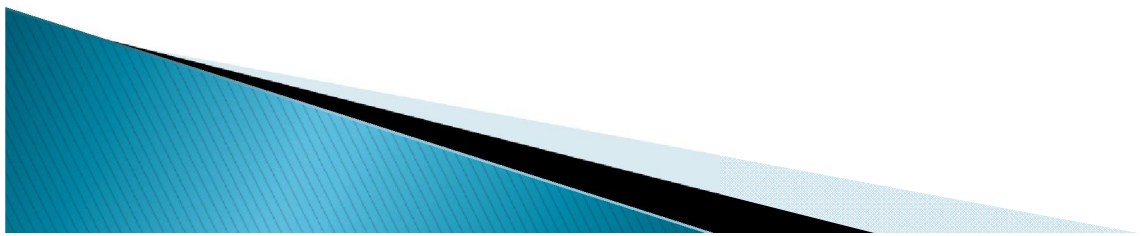
```
a=[1 2, 3, 4 5, 6]
```

- برای تعریف بردار ستونی بین عضوها از علامت سمی کالن استفاده میشود.

```
a=[1;2;3;4;5;6]
```

- برای تعریف ماتریس عضوهای یک سطر با کاما یا فاصله جدا میشوند و سطر های ماتریس با سمی کالن از هم جدا میشوند.

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



- اگر در انتهای یک دستور سمی کالن نگذاریم نتیجه محاسبات دیده میشود ولی اگر سمی کالن بگذاریم عملیات انجام میشود ولی نتیجه محاسبات دیده نمیشود.

```
>>a=2+3-5
a=
     0
>>a=2+3-5;
>>
```

- اگر یک سطرطولانی باشد می توان با گذاشتن سه نقطه ادامه دستور را در سطر بعد نوشت.

```
A=[1 2 3 ...
 4 5;6 7 8 9  2; 1...
2 3 4 5]
```

مثال

- نحوه تعریف ماتریس $\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$ به صورت زیر است:

[1;2;3]

- نحوه تعریف ماتریس [1 2 3] به صورت زیر است:

[1 2 3] یا [1,2,3]

- نحوه تعریف ماتریس $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$

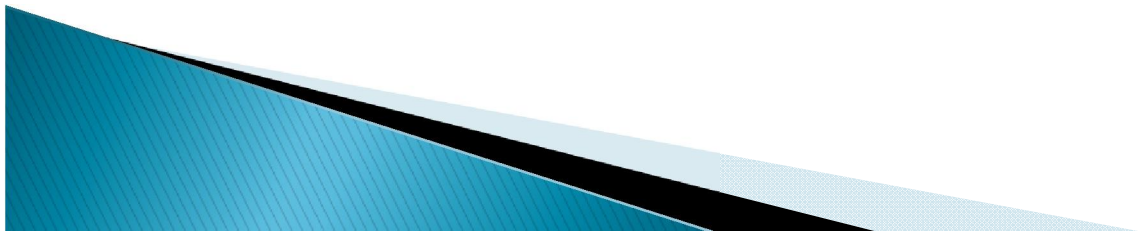
برای تعریف ماتریس فوق در متلب به صورت زیر عمل میکنیم: [1 2;3 4]

تذکر:برای تعریف یک ماتریس با زدن Enter میتوان سطرها را از هم جدا کرد.مثل

```
[1 2
 3 4]
```

محاسبات ماتریسی

- جمع و تفریق با + و –
- ضرب دو ماتریس (با ابعاد m*n , n*p) با *
- تقسیم دو ماتریس با /
- ^ برای به توان رساندن ماتریس بکار میرود. A^2
- ترانهاده ماتریس با ‘ A’
- اگر بخواهیم عناصر دو ماتریس را نظیر به نظیر در هم ضرب کنیم به جای * از .* استفاده می کنیم.
- /. برای تقسیم نظیر به نظیر استفاده می شود.
- ^. به توان رسانی نظیر به نظیر A.^B



مثال

```
A=[1 2 3;4 5 6;7 8 9];
B=[2 3;1 1;4 5];
```

A یک ماتریس 3*3 و B یک ماتریس 3*2 است.

دستورزیر ضرب ماتریسی A در B را انجام میدهد و حاصل که ماتریسی 3*2 است در C قرار میگیرد.

```
C=A*B
```

این دستور اشتباه است چون نمیتوان یک ماتریس 3*2 را در ماتریس 3*3 ضرب کرد. `D=B*A`

این دستور ماتریس A 3*3 را در خودش ضرب میکند حاصل ماتریس E 3*3 است. `E=A^2`

این دستور اشتباه است چون نمیتوان یک ماتریس 3*2 را در 3*2 ضرب کرد. `F=B^2`

این دستور درست است و معادل B.*B یعنی ضرب نظیر به نظیر عناصر **B است.** `G=B.^2`

این دستور از تمام درایه های A دو واحد کم میکند. `H=A-2`

برخی از انواع فرمت ها در متلب

```
>>format short
>>pi
ans=3.1416
```

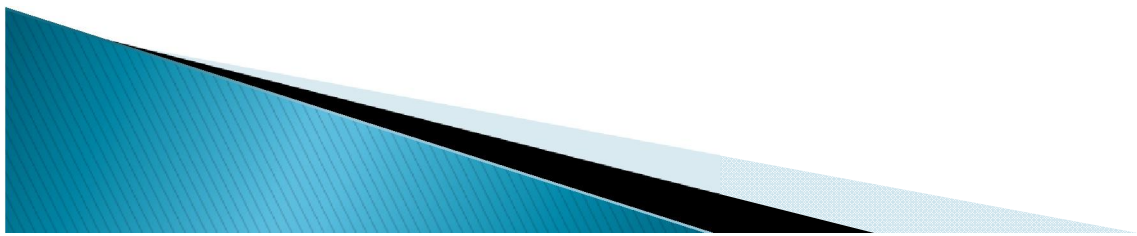
عدد را تا 4 رقم اعشار نشان میدهد.

```
>>format long
>>pi
ans=3.141592653589793
```

عدد را تا 15 رقم اعشار محاسبه میکند.

```
>>format rat
>>pi
ans=355/113
```

نشاندهنده اعداد اعشاری به صورت یک تقسیم گویا



در این فرمت عدد و حروف باهم هستند.

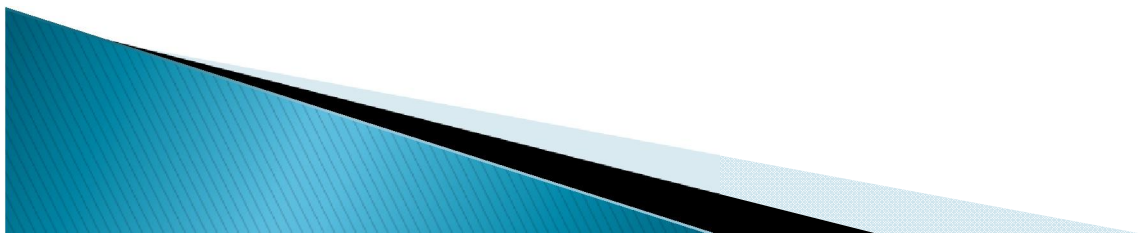
```
>>format hex
>>pi
ans= 400921fb54442d18
```

اعداد را گرد می کند.

```
>>format bank
>>pi
ans=        3.14
```

دستورvpa : با این دستور فرمان می دهیم که تا چند رقم اعشار نشان داده شود.

```
>>Vpa(pi,11)
ans=3.1415926536
```



نمایش عدد موهومی :

```
>> a=complex(1,3)
a=1.0000+3.0000i
```

قسمت حقیقی:

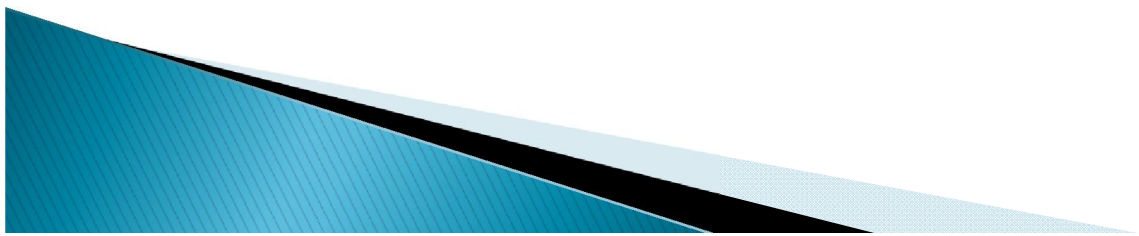
```
>>real(complex(1,3))
ans=1
```

قسمت موهومی:

```
>>imag(complex(1,3))
ans=3
```

قدر مطلق:

```
>>abs(-1)
ans=1
```



مزدوج :

```
>>conj(a)
ans=1.0000-3.0000i
```

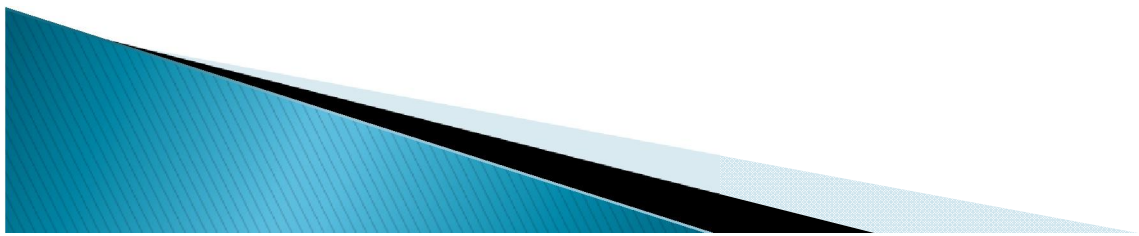
تابع نمایی:

>>exp(x) $\longrightarrow e^{x}$

>>atan(x) $\longrightarrow \tan^{-1}(x)$

ones(n):ماتریسی که همه درایه های آن برابر یک است.

```
>>ones(3)
ans=
     1   1   1
     1   1   1
     1   1   1
```

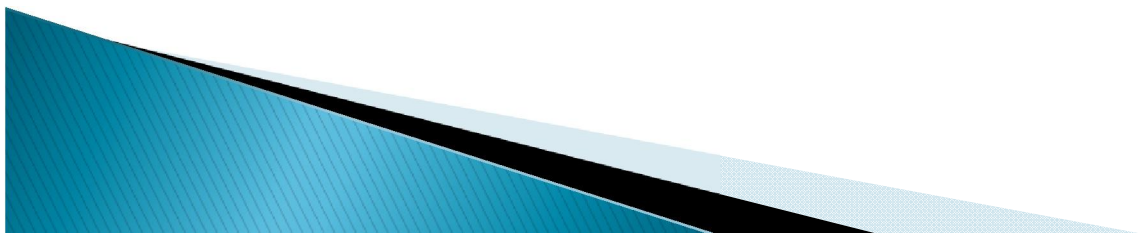


- ones(n,m):ماتریس n*m که همه درایه های آن برابر یک است.

```
>>ones(2,3)
ans=
     1   1   1
     1   1   1
```

- eye(n):ماتریس واحد:

```
>>eye(3)
ans=
     1   0   0
     0   1   0
     0   0   1
```

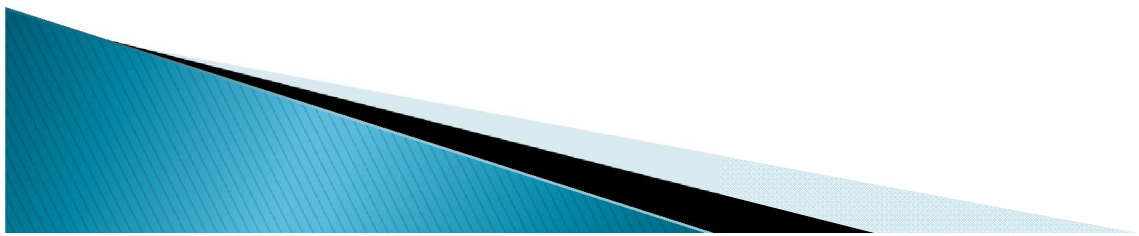


eye(n,m):ماتریس n*m که درایه های روی قطر اصلی آن یک است:

```
>>eye(3,2)
ans=
      1   0
      0   1
      0   0
```

zeros(n): ماتریس صفر

```
>>zeros(3)

ans=
      0   0   0
      0   0   0
      0   0   0
```

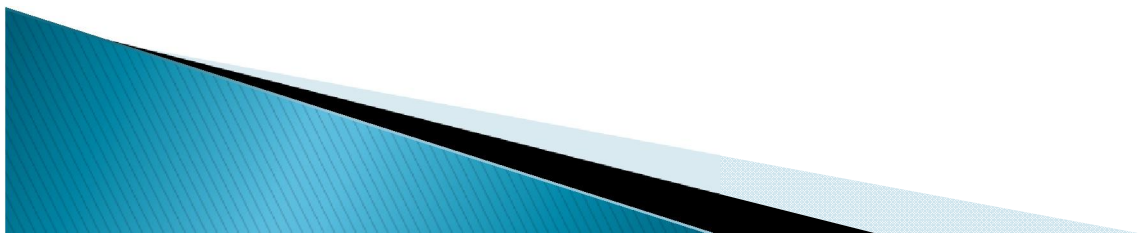


- یک ماتریس با درایه های تصادفی بین صفر و یک ایجاد میکند:

```
>>rand(2,3)
```

- اعداد 1 تا n را به صورت تصادفی در یک بردار سطری قرار میدهد

```
>>randperm(8)
ans=
      2  4  1  5  8  6  3  7
```

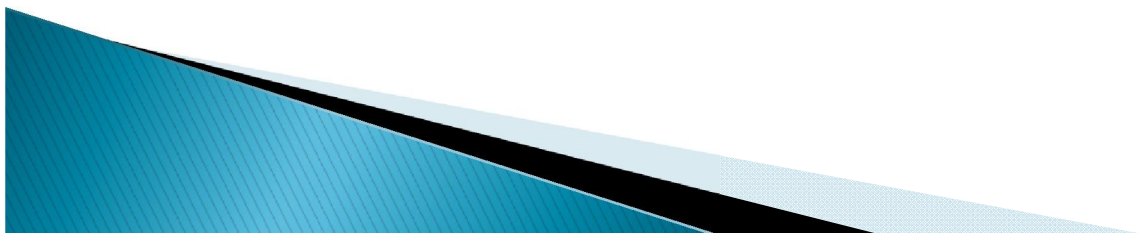


length: نشاندهنده بیشترین طول یک ماتریس یا بردار

```
>>x=[1 4 3 6 7 2 8 9]
>>length(x)
ans=  8
```

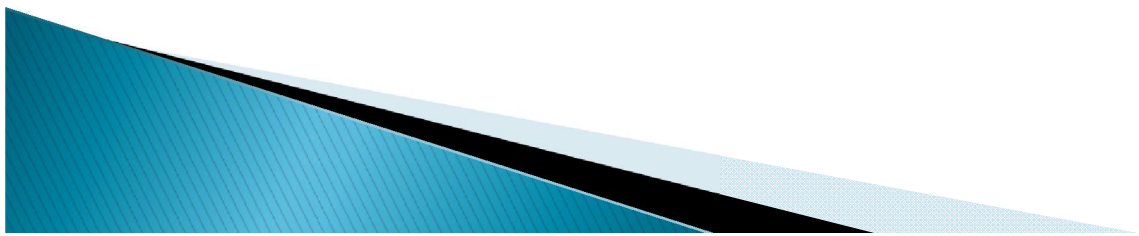
diag:این دستور روی قطر اصلی ماتریس کار میکند.

```
>>a=[1 2 3];
>>diag(a)
ans=
      1 0  0
      0  2  0
      0  0  3
```



دسترسی به عناصر یک آرایه دو بعدی

- دسترسی به عنصر i و j ماتریس A : >>A(i,j)
- دسترسی به سطر i تا k و ستون j تا h ماتریس A : >>A(i:k,j:h)
- دسترسی به ستون j ام: >>A(:,j)
- دسترسی به ستونهای j تا h: >>A(:,j:h)
- دسترسی به سطر iام: >>A(i,:)
- دسترسی به سطرهای i تا k: >>A(i:k,:)
- برای دسترسی به یک عنصر ماتریس از اندیس گذاری یک بعدی نیز میتوان استفاده کرد. ایندکس به صورت ستونی شمرده میشود.

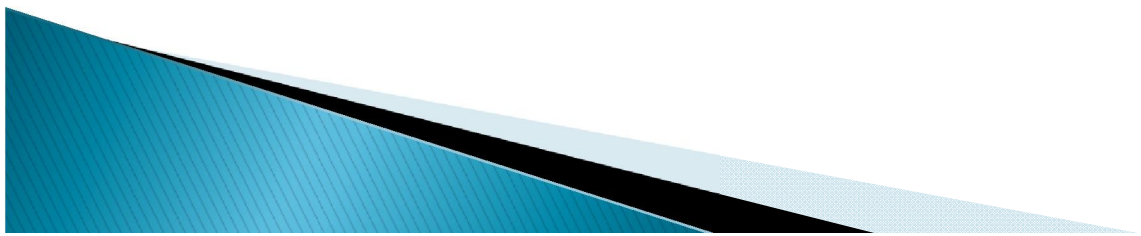


ایجاد تغییر در ماتریسها

- برای مقداردهی عدد n به یک عنصر i و j ماتریس A به صورت زیر عمل میشود.

```
>>A(i,j)=n
```

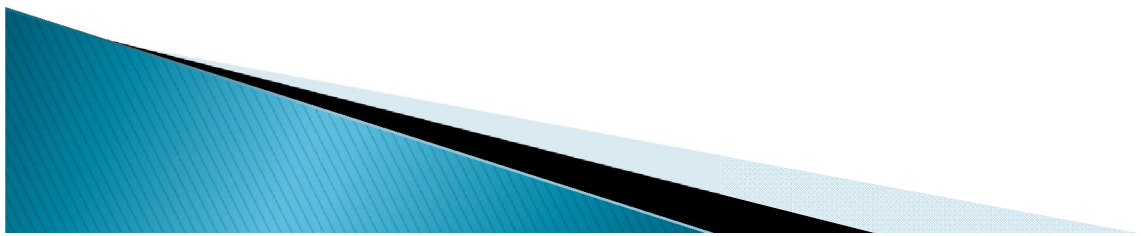
- حذف سطر i ماتریس A: `>>A(i,:) = []`
- حذف ستون j ماتریس A: `>>A(:,j)= []`



مثال:

```
A=
    2  3  4
    6  8  9
    7  1  5
>>A(2,3)
ans=
       9
>>A(1:3,2:3)
ans=
       3  4
       8  9
       1  5
```

```
>>A(3)
ans=
        7
```



مرتب کردن آرایه ها

- یک ماتریس A با درایه های صحیح بین 0 و 20 تولید میکنیم و سپس با استفاده از دستور sort(A,n) که n مشخص کننده سطر یا ستون میباشد آن را مرتب میکنیم.

```
>>A=fix(20*rand(2*3))
A=
    8     8    7
    18   7     3
>> sort(A,1)
ans=
        8     7    3
        18    8    7

>>sort(A,2)
ans=
        7    8    8
        3    7    18
```

الحاق عناصر به آرایه

- الحاق عناصر X=[1 2 3] به آرایه Y=[X -5] ، Y=[1 2 3 -5] میشود.
- اگر X=[1 2 3] و بنویسیم Y=[X;3 4 5] دراینصورت :

$$Y = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 4 & 5 \end{bmatrix}$$

- اگر X=[1 2 3] و بنویسیم Y=[X;5] خطا رخ میدهد.

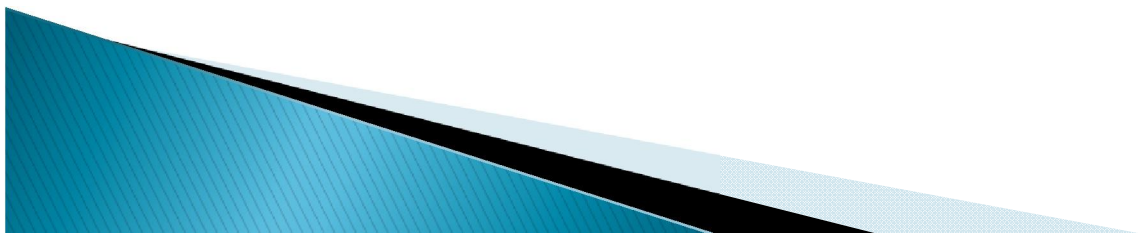
دو تابع برای الحاق عناصر

- horzcat :اتصال عناصر به صورت افقی

```
>>A=horzcat([1  2  3],[4  5  6])
A=[1  2  3  4  5  6]
```

- Vertcat :اتصال عناصر به صورت عمودی

```
>>A=vertcat([1  2  3],[4  5  6])
A =[1  2  3;4  5  6]
```



محاسبه مجموع و میانگین آرایه

- تابع sum برای محاسبه مجموع عناصر آرایه به کار میرود.

- اگر یک بردار داشته باشیم مثلا x=[1 2 3] , sum(x) مجموع عناصر آن را برمیگرداند.

- اگر x=[1 2 3;4 5 6] , sum(x) مجموع عناصر هر ستون را به طور مجزا حساب میکند و [5 7 9] را برمیگرداند. با نوشتن به صورت sum(sum(x)) مجموع همه عناصر حساب میشود.

- sum(x,1) معادل sum(x) عمل میکند.

- sum(x,2) مجموع هر سطر را به طور مجزا حساب میکند. در مثال بالا [6;15]

- تابع mean برای محاسبه میانگین آرایه بکار میرود.

- طرز نوشتن و کار با ان مشابه sum است. در اینجا به جای مجموع میانگین محاسبه میشود.

جستجوی مقدار در آرایه

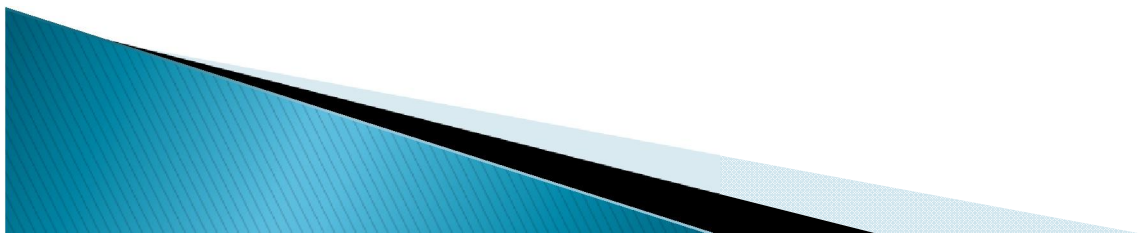
- تابع find برای جستجوی عناصر در آرایه به کار می رود.

```
>>X=[1 2 3;4 5 6] ;
>>[a b]=find(x==3)
```

در این صورت a=1 (شماره سطر عنصر 3) و b=3 (شماره ستون) میشود.

```
>>X=[1 2 3;4 3 6] ;
```

در این صورت a=1,2 و b=3,2 میشود.

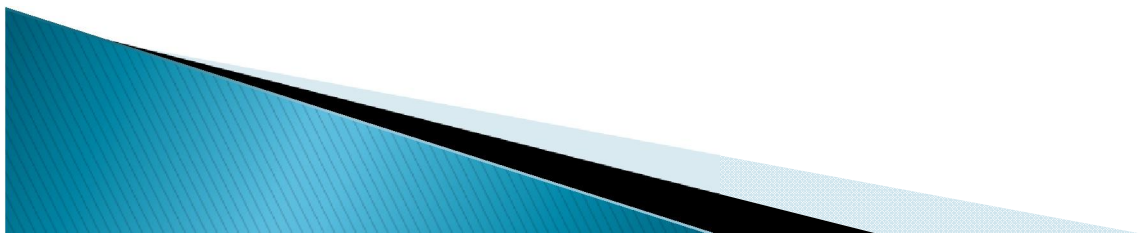


مقایسه دو ماتریس

فرض کنید x و y دو ماتریس باشند آنگاه دستور زیر برای مقایسه دو ماتریس بکار میرود:

```
>>isequal(x,y)
```

در صورتیکه دو ماتریس برابر باشند مقدار یک را برمی گرداند در غیر این صورت مقدار صفر را برمی گرداند.

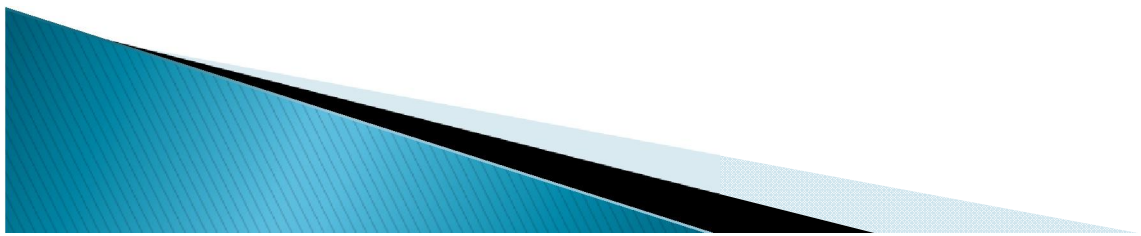


توابع ریاضی

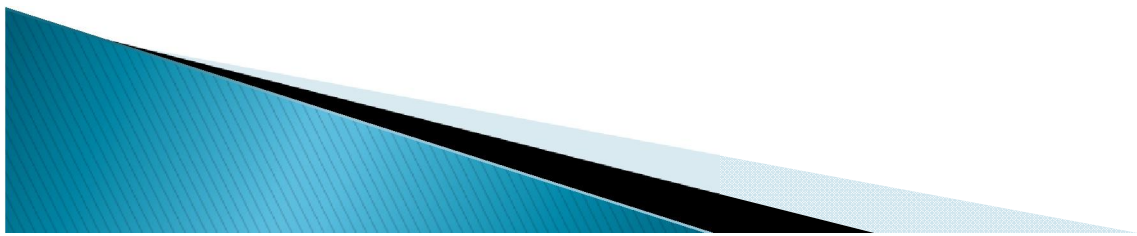
- نرم افزار matlab تعدادی تابع ریاضی آماده دارد که برای معرفی انها از دستور زیر استفاده کنید.

Help elfun

باتایپ دستور فوق کلیه تابعهای ریاضی اولیه با مختصر توضیحی ارائه می شود.



تابع	توضیح
sin	سینوس مثلثاتی
sind	سینوس (آرگومان بر حسب درجه)
sinh	سینوس هیپربولیک
asin	آرک سینوس
asind	آرک سینوسی (درجه)
asinh	آرک سینوس هیپربولیک



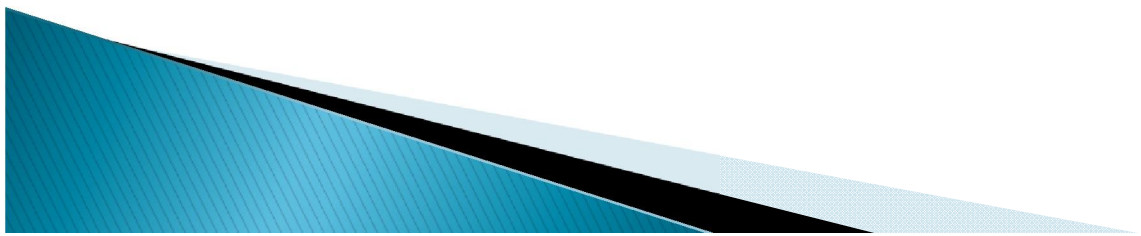
تابع	توضیح
hypot	جذر مجموع مربعات
exp	تابع نمایی
expm1	تابع نمایی منهای یک
log	لگاریتم نپرین
log1p	لگاریتم عدد به اضافه یک
log10	لگاریتم مبنای ده
log2	لگاریتم مبنای دو
pow2	دو به توان یک عدد

تابع	توضیح
sqrt	جذر یک عدد
nthroot	ریشه یک عدد
nextpow2	نزدیک‌ترین توانی از دو
abs	قدرمطلق
angle	زاویه
complex	ساخت یک عدد مختلط
image	قسمت موهومی
real	قسمت حقیقی
isreal	آیا عدد حقیقی است؟

تابع	توضیحات
fix	عدد را به سمت صفر گرد می‌کند.
floor	عدد را به سمت منهای بینهایت گرد می‌کند.
ceil	عدد را به سمت مثبت بینهایت گرد می‌کند
round	عدد را به سمت نزدیک‌ترین عدد صحیح گرد می‌کند.
mod	مد عدد اول به عدد دوم را برمی‌گرداند.
rem	باقی مانده تقسیم دو عدد را برمی‌گرداند
sign	علامت عدد را برمی‌گرداند.

- نرم افزار matlab تعدادی تابع اولیه مربوط به ماتریسها دارد که برای معرفی انها از دستور زیر استفاده کنید.

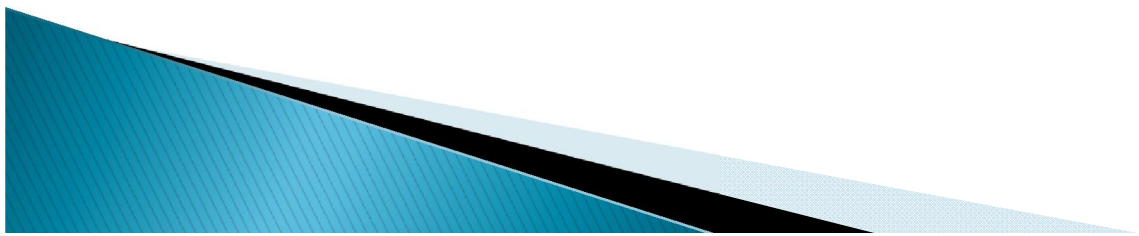
help elmat



تابع	توضیحات
zeros	یک ماتریس پر از صفر تولید می‌کند.
ones	یک ماتریس پر از یک تولید می‌کند.
eye	یک ماتریس واحد تولید می‌کند.
repmat	یک ماتریس را تکرار می‌کند
rand	ماتریس تصادفی یکنواخت استاندارد
randn	ماتریس تصادفی نرمال استاندارد

تابع	توضیح
linspace	تولید اعداد با فواصل یکسان
meshgrid	تولید یک شبکه
size	ابعاد یک متغیر
length	طول یک متغیر
ndims	تعداد بعد
numel	تعداد عناصر
display	برای نمایش
isempty	آیا متغیر خالی است؟
isequal	آیا متغیر مساوی است؟

تابع	توضیح
cat	چسباندن
reshape	تغییر شکل
diag	قطری
tril	پایین مثلثی
trilu	بالا مثلثی
fliplr	چرخش چپ و راست
flipud	چرخش بالا و پایین
flipdim	چرخش حول یک بعد
rot90	چرخش ۹۰ درجه

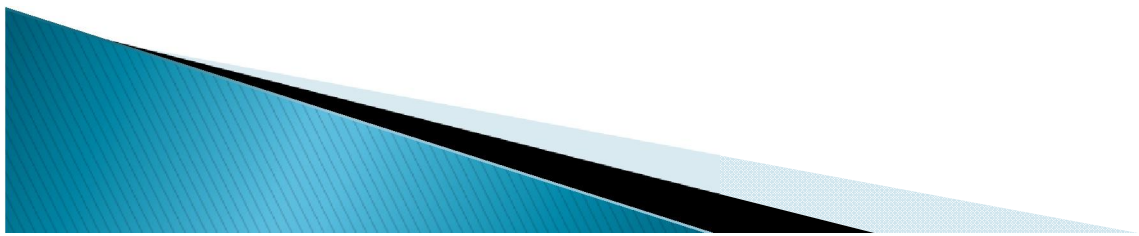


تابع	توضیح
find	برای یافتن عناصر غیر صفر
end	پایان
circshift	چرخش دایره‌ای
eps	اپسیلون
realmax	بزرگترین عدد حقیقی
realmin	بزرگترین عدد حقیقی
hadamard	تولید ماتریس هادامارد
magic	تولید ماتریس جادویی

مثال

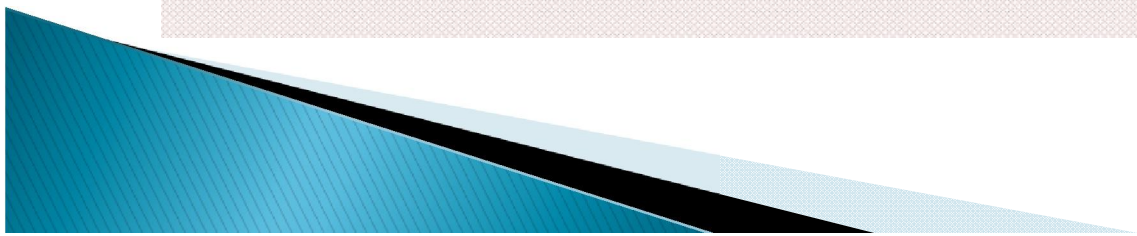
```
rho = (1+sqrt(5))/2
a = abs(3+4i)
a = 5*ones(3,3)
z = zeros(3,4)
i = ones(3)
n = round(10*rand(1,10))
```

```
r = randn(3)
x = -5:0.1:5
y = randn(1000,1)
```



```
b = [1 2; 3 4]
c = [b b, b+4 b-1]
c(:,2) = []
n = (0:10)'
pows = [n, n.^2, 2.^n]
x = (1:0.1:2)'
logs = [x, log10(x)]
```

```
c = [b, b; b+4 b-1]
c(1:3:4, :) = []
c(:, 1:3:4) = []
c = [b  b; b+4 b-1]
c(1:2:16) = []
a = rand(3)
b = [a, zeros(3,2); zeros(2,3), eye(2)]
```



رسم نمودارهای دو بعدی

نرم‌افزار Matlab برای رسم نمودارها امکانات خوبی دارد

دستورات رسم منحنی به صورت زیر می‌باشند.

plot

xlabel

ylabel

title

grid

axis

stem

subplot

hold on

hold off

برای رسم یک نمودار دو بعدی دو حالت ممکن است داشته باشیم

الف) ممکن است نقاط (x,y) زیادی در دسترس باشد و بخواهیم برای این نقاط نموداری را رسم کنیم .

ب) ممکن است تابع y=f(x) مشخص باشد و بخواهیم این تابع را در بازه [a,b] رسم کنیم .

در هر دو حالت نقاط x ها را در یک آرایه و y ها را در آرایه ای دیگر ذخیره می نماییم و با دستور plot نمودار مورد نظر رسم می شود .

```
t = -pi:0.1:pi;
y = cos(t);
plot(t, y)
```

- اگر ضابطه تابع مشخص باشد می توان بدون در نظر گرفتن بازه با دستور ezplot تابع مورد نظر را رسم کرد. بایستی عبارت f داخل کوتیشن قرار گیر: ezplot('f')

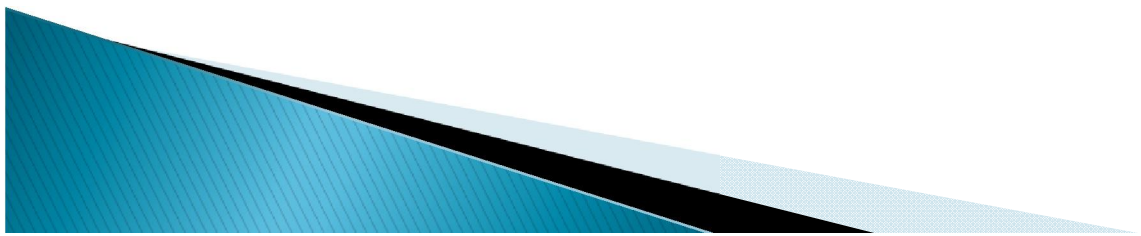
مثال:

```
ezplot('sin(x)')
k='sin(x)'
ezplot(k)
```

- اگر بخواهیم نمودار در بازه مشخصی رسم شود می نویسیم: ezplot('f',[a,b])

مثال:

```
ezplot('sin(x)',[-3,3])
k='sin(x)'
ezplot(k,[-3,3])
```



- وقتی از دستور plot استفاده می کنید می توان نمودار منحنی ها و توابع را با علامت جدول زیر به صورتهای مختلف رسم نمود.

کاراکتر	علامت
o	دایره ای
+	به شکل +
*	به شکل *
s	به صورت مربع
d	به صورت لوزی
^	به صورت مثلثی
>	به صورت مثلث رو به چپ
<	به صورت مثلث روبه راست
p	ستاره پنج پر
h	ستاره شش پر
-	به صورت خط توپر
:	نقطه چین
-.	خط نقطه
--	خط چین

کاراکتر	رنگ
b	آبی
g	سبز
r	قرمز
c	فیروزه ای
m	ارغوانی
y	زرد
k	سیاه
w	سفید

```
t = -pi:0.1:pi;
y = cos(t);
plot(t, y)
z = sin(t);
plot(t, y, t, z)
plot(t, y, '--')
plot (t, y, '+.')
```

```
plot(t, y,'s')
plot(t, y,'.-g')
```

- چند منحنی را باهم می توان رسم کرد.
- نوع رسم و رنگ منحنی را می توان تنظیم نمود.
- با استفاده از دستور زیر، می توان مشخصات کامل این دستور را بررسی کرد.

```
help plot
```

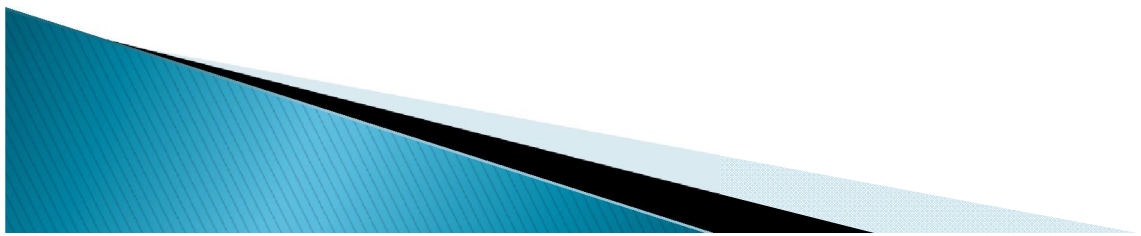
- بعد از رسم منحنی می توان برای محور افقی و عمودی منحنی برچسب زد و برای منحنی یک عنوان انتخاب کرد.

```
xlabel({'first line';'second line'})
ylabel('George''s
  Popularity','fontsize',12,'fontweight','b')
```

- اگر بخواهیم در بالای یک نمودار عنوان دلخواهی را درج نماییم می توان با دستور title این کار را انجام داد.

title(‘عبارت مورد نظر’)

- بایستی دقت شود که دستور فوق بعد از اجرای دستور رسم نمودار باشد.



- بعداز رسم منحنی می توان منحنی را شبکه بندی کرد.

```
grid
```

- بعد از رسم منحنی با دستور axis می توان رنج تغییرات منحنی را تنظیم نمود.

```
axis([-2,2,1,10])
```

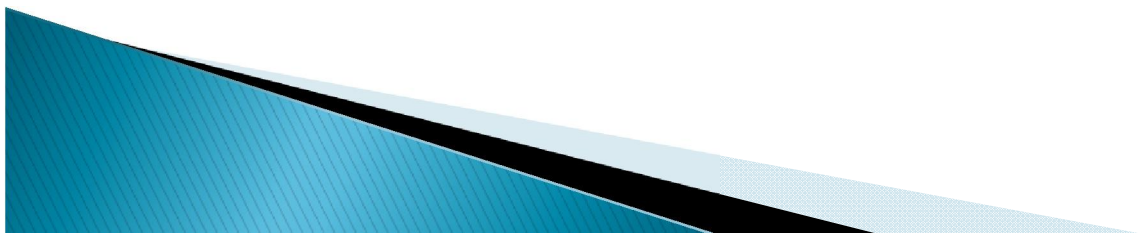
- از فرمان stem برای رسم نمودارهای گسسته استفاده می شود.

```
y=1:10
stem(y)
```

- از فرمان subplot برای رسم همزمان چند منحنی می توان استفاده کرد.

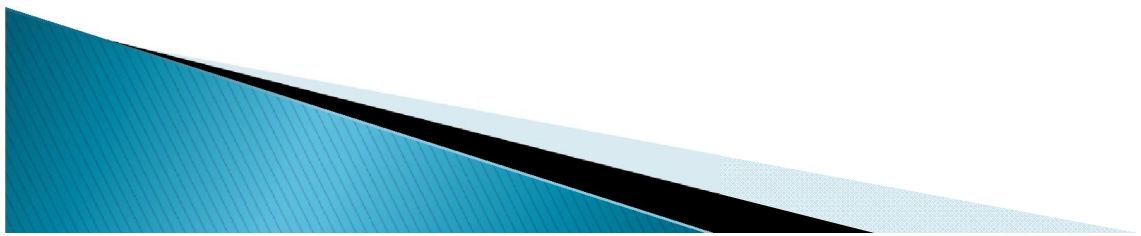
Subplot(n,m,k) : پنجره figure به n سطر و m ستون تقسیم می شود. و ناحیه k ام فعال است.

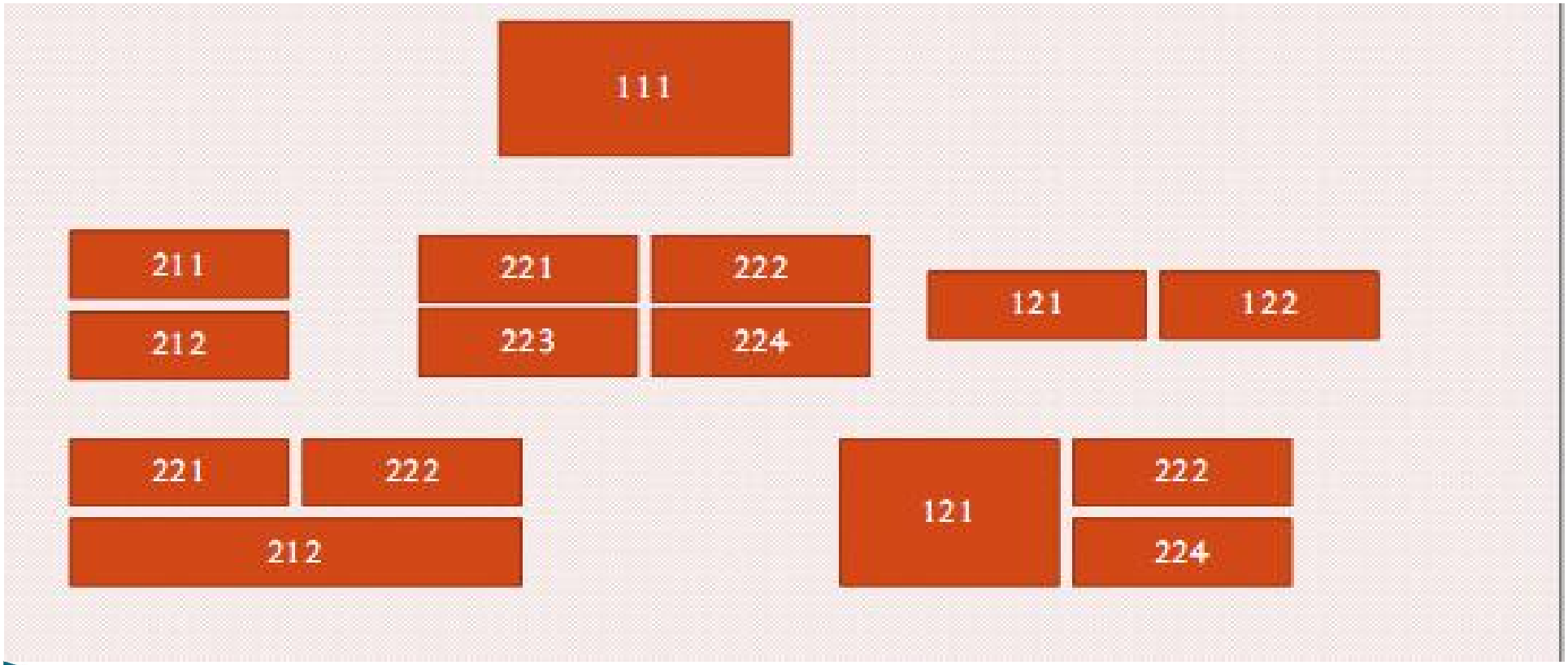
```
t=1:10;
z=cos(t)
y=t.^2;
subplot(2,1,1)
plot(t,z)
subplot(2,1,2)
plot(t,y)
```



- می توان صفحه را به چهار قسمت تبدیل کرد ودر هر قسمت منحنی را رسم کرد.

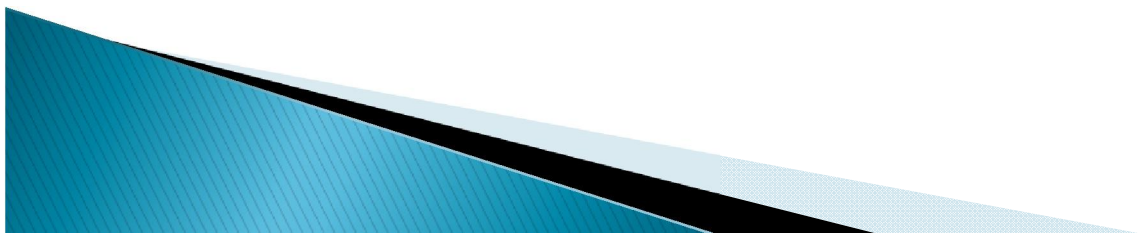
```
subplot(221),……, subplot(222),….
subplot(223),……., subplot(224)
```




111
211
212
221
222
223
224
121
122
221
222
212
121
222
224

- بعد از رسم یک منحنی اکر بخواهیم منحنی بعدی روی همین محورها رسم شود کافیست بعدا از رسم اولین منحنی از فرمان hold on استفاده کرد.

- هرگاه بخواهیم از این ویژگی دیگر استفاده نکنیم باید از فرمان hold off استفاده کرد.



رسم نمودارهای سه بعدی

- معادله منحنی که به صورت پارامتری است با دستور plot3 رسم می کنیم.

```
t=linspace(0,4*pi,100);
x=sin(t);
y=cos(t);
plot3(x,y,t)
```

برای رسم رویه z=f(x,y) (که در آن x وy متغیرهای مستقل و z متغیر وابسته می باشد) بایستی x و y را مقداردهی نمود (دستور meshgrid) و به ازای آن مقادیر مختلف z را به دست آورد. (دستور mesh یا دستور surf)

```
>> x=linspace(-3,3,50);
>> y=linspace(-3,3,50);
>> [a,b]=meshgrid(x,y);
>> z=a.^2-b.^2;
>> mesh(x,y,z)
```

با دستور sphere می توان کره به صورت سه بعدی رسم کرد. برای این کار می نویسیم sphere(n) لذا کره ای به مرکز مبدا و متشکل از n صفحه در هر ردیف تشکیل می شود.

```
>> sphere(50);
```

البته می توان با دستور زیر کره ای به شعاع و مرکز دلخواه رسم کرد.

```
>>[x,y,z]=sphere(40)
>>x=3*x+5
>>y=3*y+6
>>z=3*x+2
>>surf(x,y,z)
```

با این دستورات کره ای به مرکز (5,6,2) و شعاع 3 با 40 صفحه تولید و رسم می شود.

- با دستور cylinder می توان یک استوانه به مرکز مبدا رسم نمود. برای این کار می نویسیم Cylinder(n) که استوانه ای به مرکز مبدا و متشکل از n صفحه و طول 1 رسم خواهد شد.

تمرین :

۱- مکعبی به ضلع ۵ و مرکز مبداء مختصات به صورت سه بعدی رسم کنید . برای این کار شش صفحه را با هم در یک محور مختصات (به صورت یک گراف) به گونه ای رسم کنید که تشکیل یک مکعب بدهد .

۲- استوانه ای رسم کنید که در منحنی فضایی (t , sin(6t) , cos(6t)) محاط شده است .

- تابع زیر را تایپ کنید توابع مهم داده لیست می شوند.

help datafun

تابع	توضیح
max	ماکزیمم
min	مینیمم
mean	متوسط
median	میانه
std	انحراف معیار
var	واریانس
sort	مرتب کردن

تابع	توضیح
sum	جمع عناصر
prod	ضرب عناصر
hist	هیستوگرام
cumsum	توزیع تجمعی
cumprod	توزیع تجمعی ضربی
diff	مشتق
corrcoef	ضریب همبستگی
cov	کوواریانس

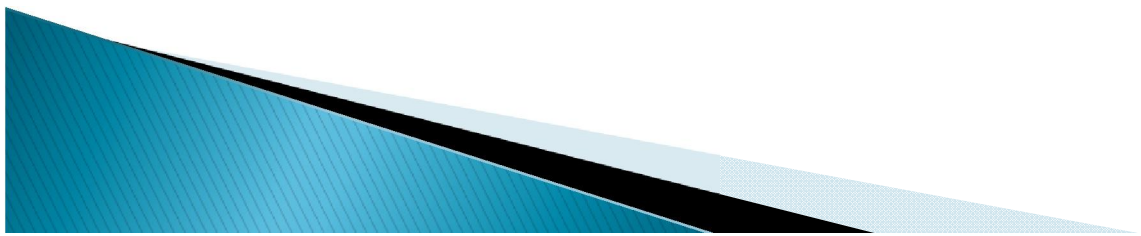
مثال

```
b = [5 1 2;3 9 4;7 6 8]
index = find(b == 6)
[r,c] = find(b == 6)
m = max(b)
m = max(max(b))
[v,r] = max(b)
min(b)
s = size(b)
d = b(2,:)
s = size(d)
l = length(d)
max(size(d))
```

```
n = ndims(b)
length(size(b))
p = numel(b)
```

- در متلب می توان بردارهای سه بعدی نیز تعریف کرد.

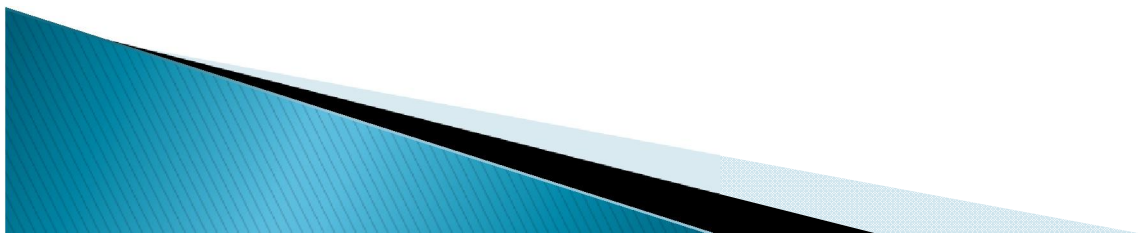
```
a = [5 7 8;0 1 9; 4 3 6]
a(:,:,2) = [1 0 4;3 5 6; 9 8 7]
a(:, :, 3) = 5
r = rand(4, 3, 2)
r(4, 1, 2)
r([1 3 4], 2 ,1)
r(3, :, 2)
s = size(r)
```



- برای چسباندن دو ماتریس به هم و تشکیل ماتریس جدید از دستور cat استفاده می شود.

```
a=[1 2;3 4];
b=[5 6;7 8];
c=cat(1,a,b);
   [a;b]
```

```
c = cat(2, a, b)
[a, b]
```



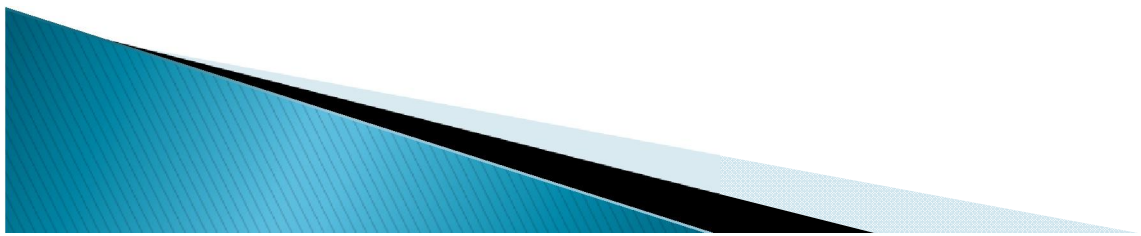
- در این قسمت چندین مثال جهت یادآوری مطالب گذشته ارائه می شود.

```
a = 1+2i
b = Real(a)
c = imag(a)

abs(a)
angle(a)
conj(a)
complex(2,3)
pow2(5)
nextpow2(13)
```

```
log10(10)
x = rand(1,10)
x(4:-1:2)
x = (0:0.1:1)*pi
linspace(0, pi, 11)
logspace(0, 2, 11)
linspace(0,10,11)
```

- دستور linspace با سه پارامتر استفاده میشود. بین پارامتر اول و دوم به اندازه پارامتر سوم نقطه ایجاد میشود.
- دستور logspace با سه پارامتر استفاده میشود. بین 10 به توان پارامتر اول و 10 به توان پارامتر دوم به اندازه پارامتر سوم نقطه ایجاد میکند.

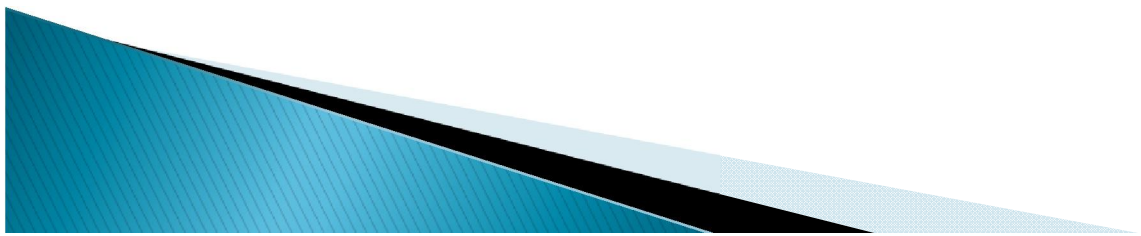
- پیش فرض تعداد نقاط linspace عدد 100 و logspace عدد 50 است.



```
b = [5:-1:1 3 8]
c = [b, 0]
d = [a(1:2:5) 1 0]
c, d
who
clear b c
 2^4
 a = [1 2 3]
 a.^2
 a.^3
 a = 2:3:8
 size(a)
 b = [a' a' a']
size(b)
c = b(1:2:3, 1:2:3)
size(c)
d = a+b(2, :)
size(d)
w = [zeros(1, 3) ones(3,1)' 3:5']
size(w)
b([1, 3], 2) = b([3, 1], 2]
Size(b)
e = 1:-1:5
```

m – فایلها

- به جای نوشتن دستورات در پنجره command و اجرا شدن به صورت تک تک، می توان مجموعه ای از دستورات را در یک فایل قرار داد. به این فایل script یا m file می گوییم.
- نحوه نوشتن m file : مجموعه دستورات مورد نظر را در یک ویرایشگر می نویسیم. فایل را با پسوند m. ذخیره میکنیم.
- Matlab خود یک ویرایشگر برای انجام این کار دارد که با انتخاب file/new/script اجرا میشود.



دستور input

n=input(‘عبارت ’);

مثال:

```
>> a=input('enter a number:')
enter a number:6

a =

     6
```

دستور disp

disp(نام متغیر)

disp('متن')

```
>> x=[1 2 3 4 5];
>> disp(x)
     1     2     3     4     5

>> disp('ave_points');
ave_points
>>
```

تابع بی مشخصه

Name=@(arglist)expr;

منظور از expr یک عبارت ریاضی ساده است.

```
>> f=@(x)x^2

f =

    @(x)x^2

>> f(4)

ans =

    16
```

```
>> f=@(x)x.^2

f =

    @(x)x.^2

>> f([1 2 3 4])

ans =

     1     4     9    16

>>
```

```
>> f=@(x,y)x.^2+y.^2

f =

    @(x,y)x.^2+y.^2

>> x=[1 2 3 4];
>> y=[3 4 5 6];
>> f(x,y)

ans =

    10    20    34    52

>>
```

در Matlab یک سری ساختارهایی مشابه سایر زبان‌های برنامه‌نویسی وجود دارد. لیست کامل این توابع در تابع زیر می‌باشد.

```
help lang
```

تابع	توضیحات
if	تابع شرطی
else	تابع شرطی
elseif	تابع شرطی
end	پایان شرط و حلقه
for	حلقه
while	حلقه
break	خروج از حلقه

دستورات حلقه و شرط

- دستور for

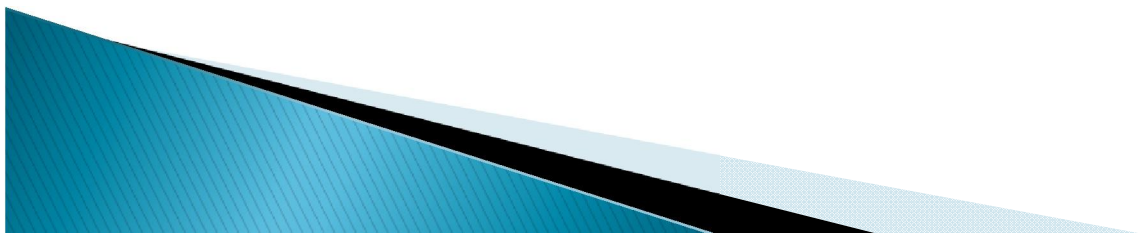
for نام متغیر=ابتدای حلقه: گام حرکت: انتهای حلقه

< یک یا چند دستور >

end

```
for  x=1:0.1:2
< یک یا چند دستور >
end
```

X از 1 تا 2 با گام 0.1 تغییر میکند.

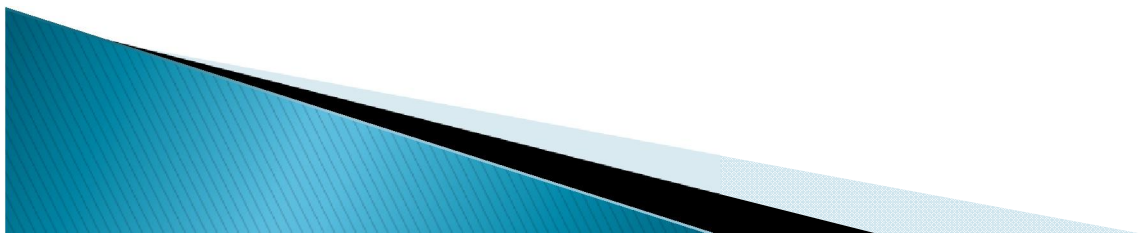


- دستور while

while شرط
< یک یا چند دستور >
end

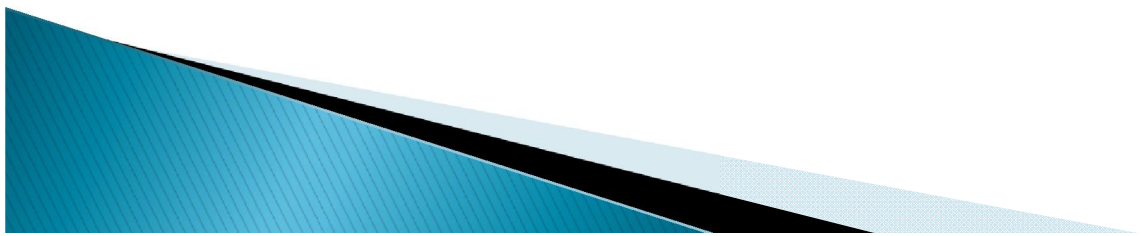
- دستور if

if شرط
< یک یا چند دستور >
end



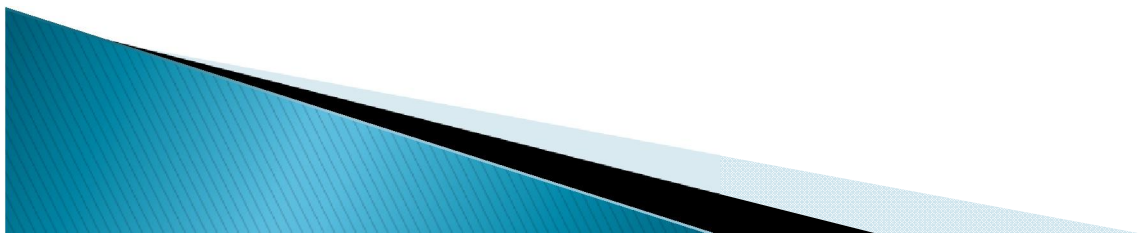
- دستور if-else

```
if     شرط
< یک یا چند دستور >
else
< یک یا چند دستور >
end
```



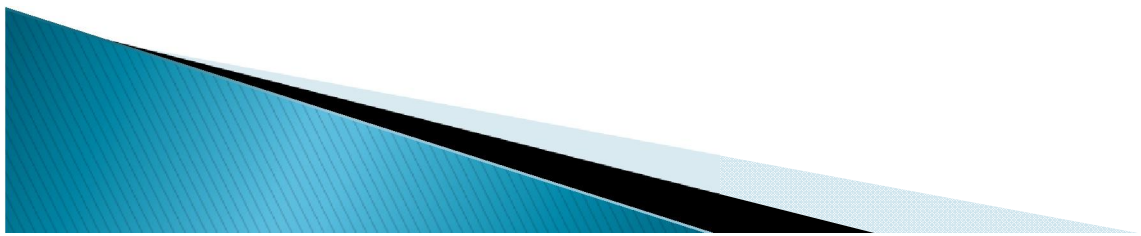
- دستور switch

Switch متغیر
case 1 مقدار
< یک یا چند دستور >
Case 2 مقدار
< یک یا چند دستور >
Otherwise
< یک یا چند دستور >
end



در مثال زیر تابع مقدار قدرمطلق ورودی x را محاسبه ودر متغیر y برمی گرداند.

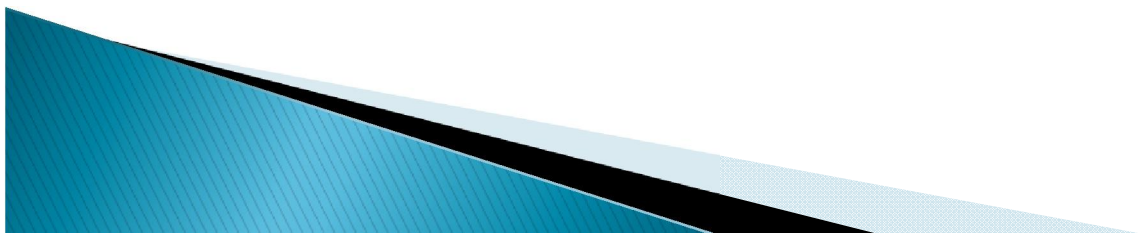
```
function y= abs(x)
if(x>0)
   y=x;
else
   y=-x;
end
```



```
function y = S(x)
%UNTITLED3 Summary of this function goes he
%    Detailed explanation goes here
for i=1:length(x)
if x(i)>0
    y(i)=x(i);
else
    y(i)=-x(i);
end
end
disp(y);
end
```

توسط یک حلقه مقادیر 2,4,8,...,2^13 را به عنوان عناصر یک آرایه تولید کند.

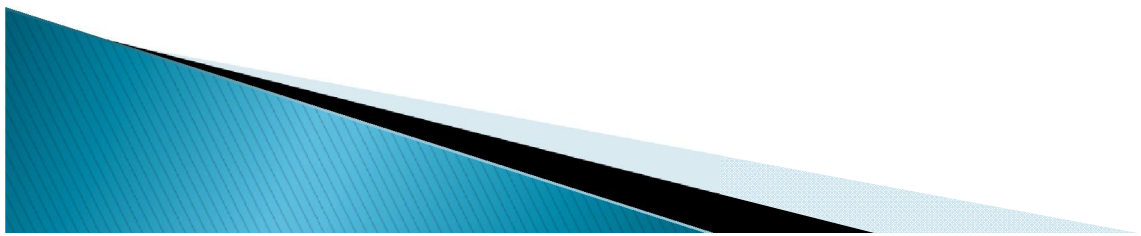
```
for i=1:13
    a(i)=2^i;
end
a
```

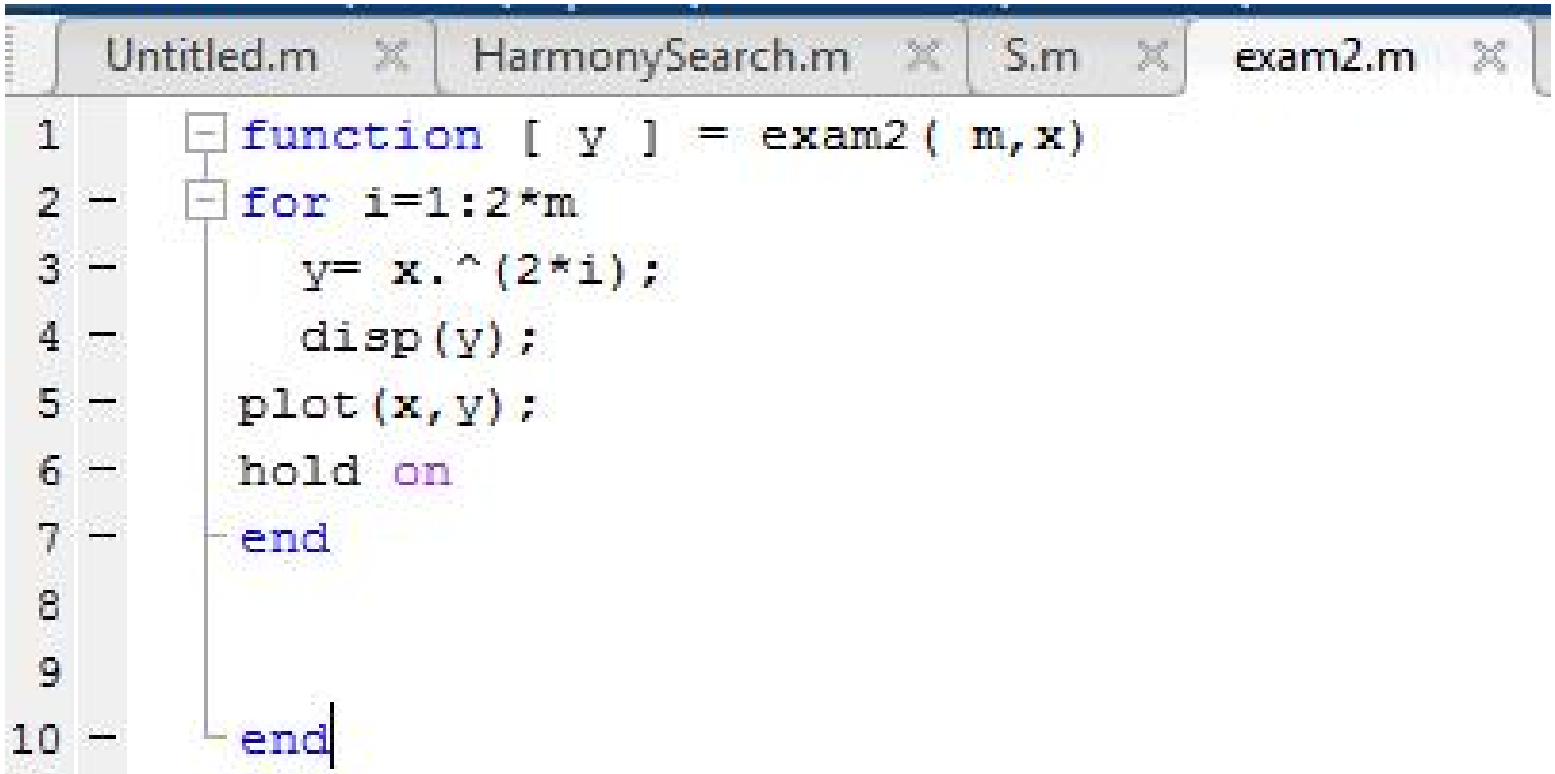


```
a =

  Columns 1 through 9

           2           4           8          16          32          64         128         256         512

  Columns 10 through 13

        1024        2048        4096        8192

>> a'

ans =

           2
           4
           8
```

- برنامه ای بنویسید که به ازای m دلخواه و صحیح و مثبت که کاربر وارد میکند توابع x^2, x^4, ...,x^2m در یک محور مختصات رسم شود.

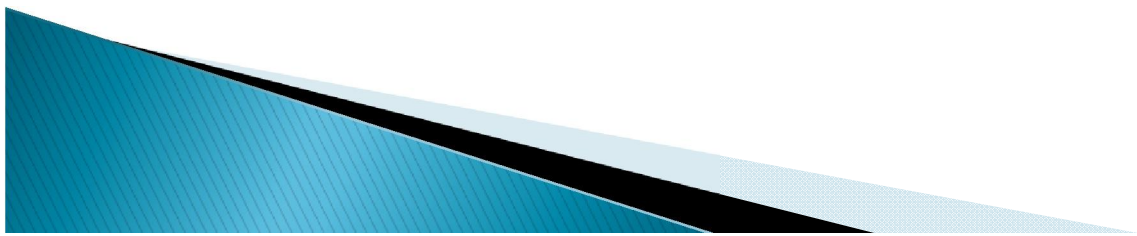
X=linspace(-3,3,100)




Untitled.m
HarmonySearch.m
S.m
exam2.m
function [y] = exam2(m,x)
for i=1:2*m
y= x.^(2*i);
disp(y);
plot(x,y);
hold on
end
end

☺ نرم‌افزار Matlab دستوراتی درباره عملیات رابطه‌ای دارد. لیست این توابع در دستور زیر است که در ادامه این توابع بررسی می‌شوند.

```
help relop
```



عملگرهای مقایسه ای

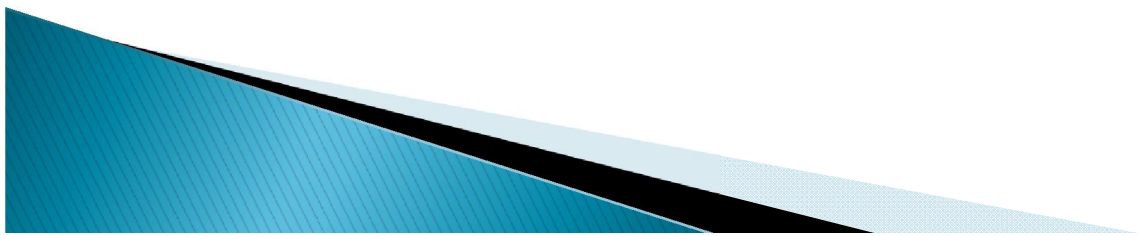
- > < >= <= == (مساوی) =~ (نامساوی)

&(ومنطقی) |(یا منطقی) ~(نقیض)

مثال:

```
X=4;
Y=5;
if(x==4 & y==2)
Z=1;
else
Z=4;
end
```

Z برابر 4 خواهد شد.



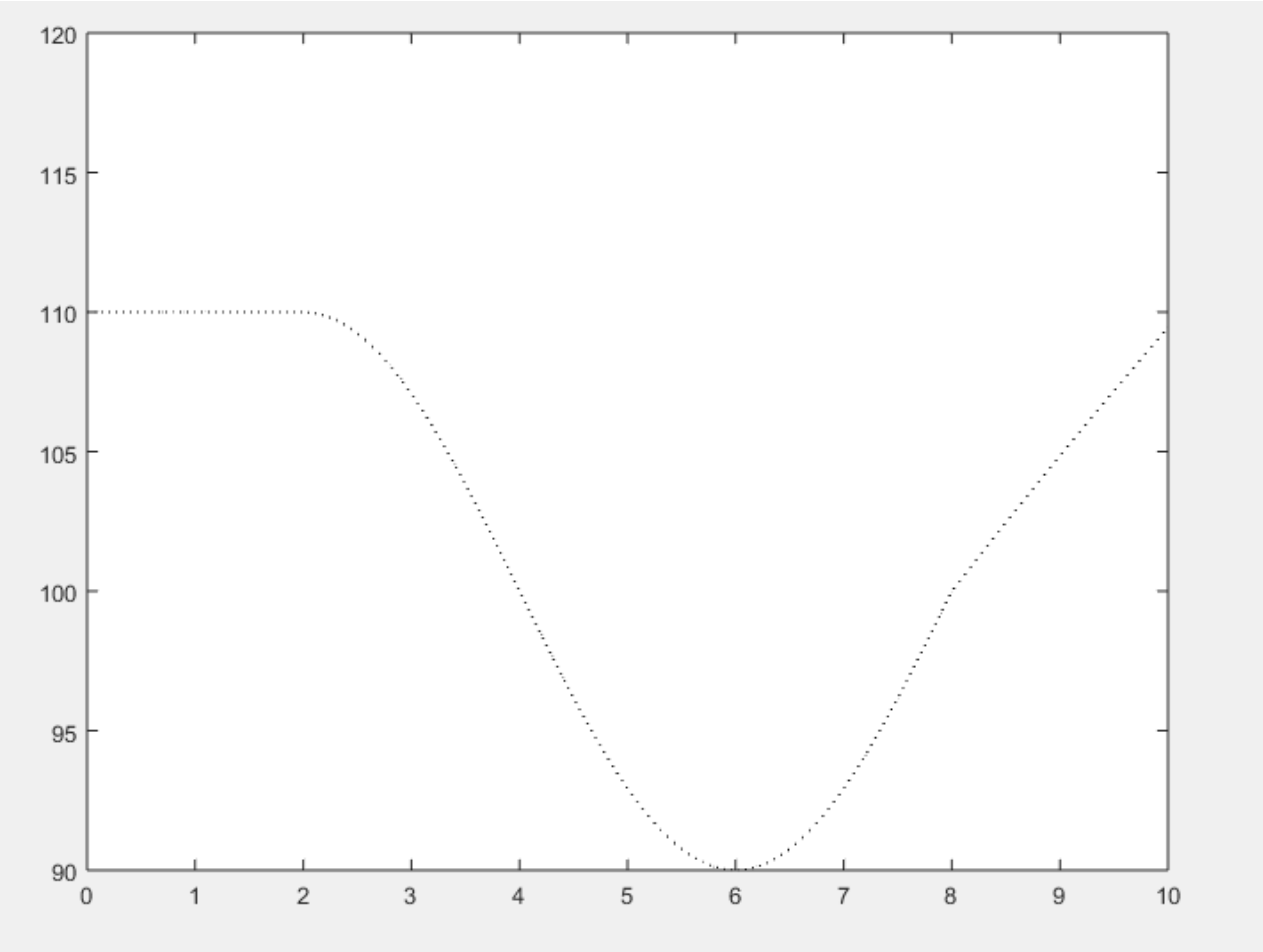
تابع f را در بازه [0,10] را رسم کنید.

$$f(x)=\begin{cases}20\sqrt{2x}+20 & x\geq 8\\ 10\sin\left(\frac{\pi x}{4}\right)+100 & 2<x<8\\ 110 & x\leq 2\end{cases}$$

Editor - C:\Users\top co\Documents\MATLAB\Untitled.m

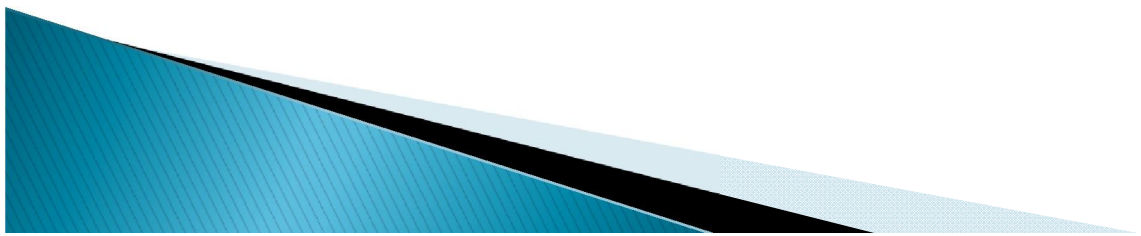
Untitled.m | HarmonySearch.m | S.m | Untitled2 | exam2.m

```
x1=linspace(0,2,21);
x2=linspace(2,8,61);
x3=linspace(8,18,21);
X=[x1,x2,x3];
for i=1:length(X)
    if X(i)<=2
        Y(i)=110;
    elseif X(i)>=8
            Y(i)=20*sqrt(2*X(i))+20;
        else Y(i)=10*sin(pi*X(i)/4)+100;
        end
end
plot(X,Y,':k')
axis([0 10 90 120])
```

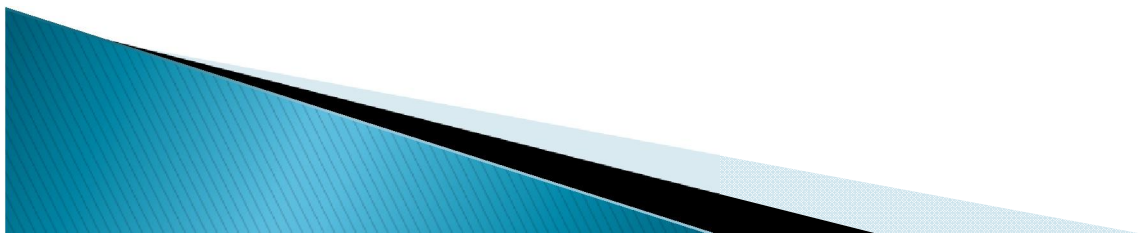

120
115
110
105
100
95
90
0
1
2
3
4
5
6
7
8
9
10

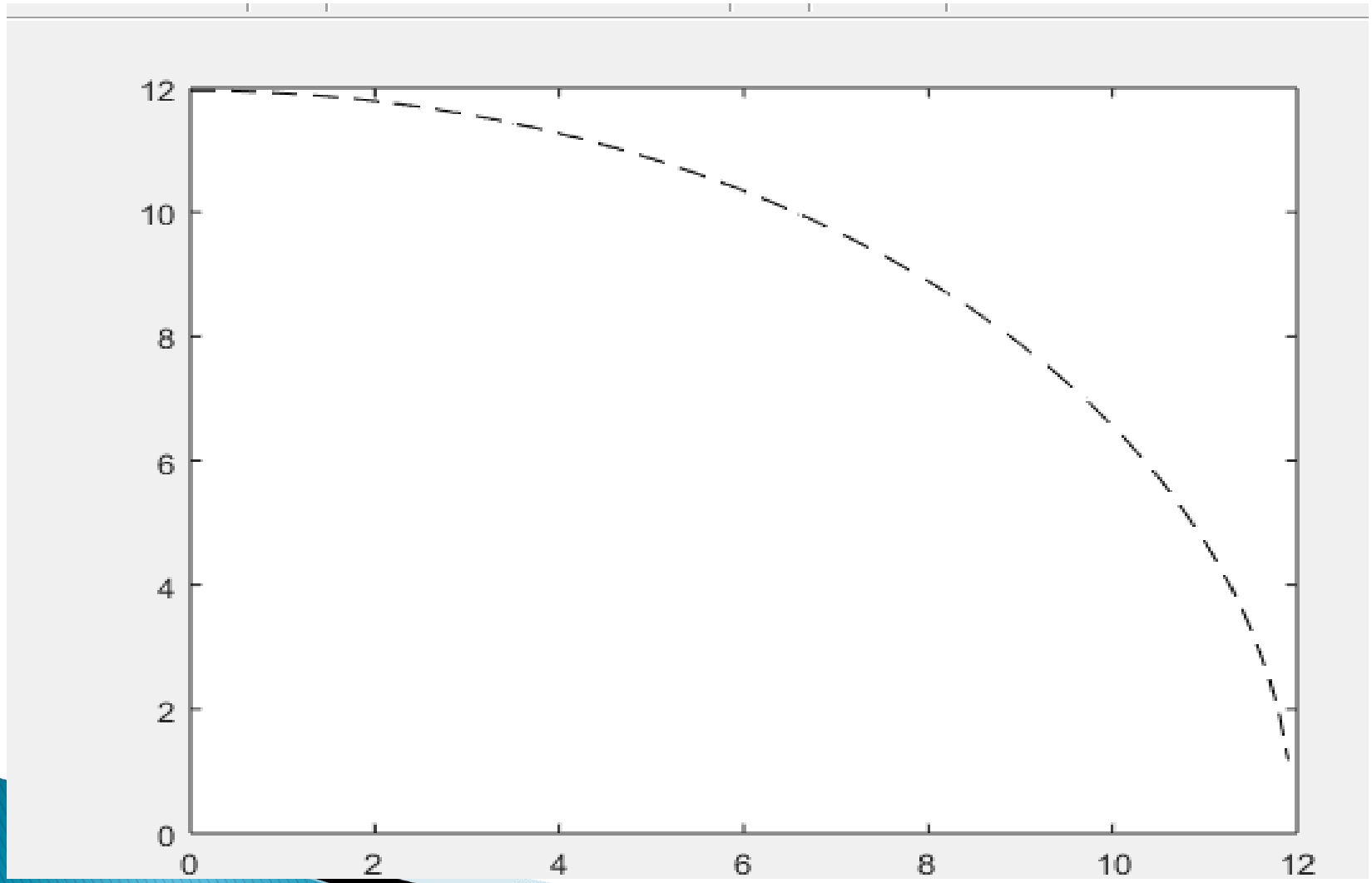
مثال. تابع f را در بازه $[0,\infty]$ بر روی نمودار اعداد حقیقی رسم کنید.

$$f(x) = \sqrt{143 - x^2}$$

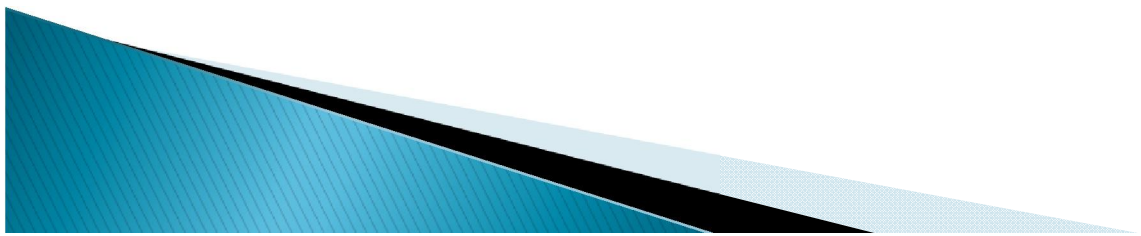


```
n=0;
for x=0:0.1:20
    if (143-x^2)>0
        n=n+1;
        Y(n)=sqrt(143-x^2);
        X(n)=x;
    else
        break
    end
end
plot(X,Y,'--k')
```

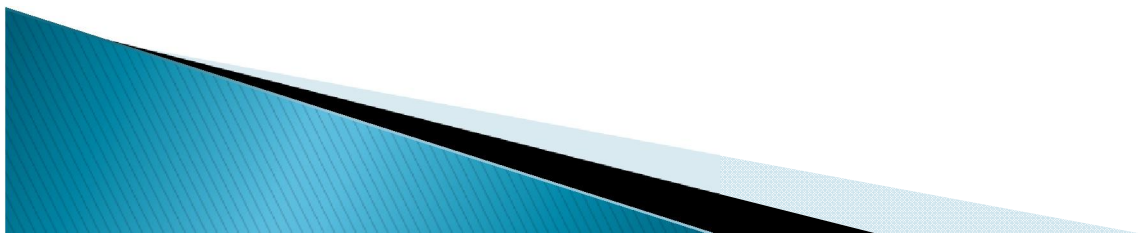



12
10
8
6
4
2
0
0
2
4
6
8
10
12

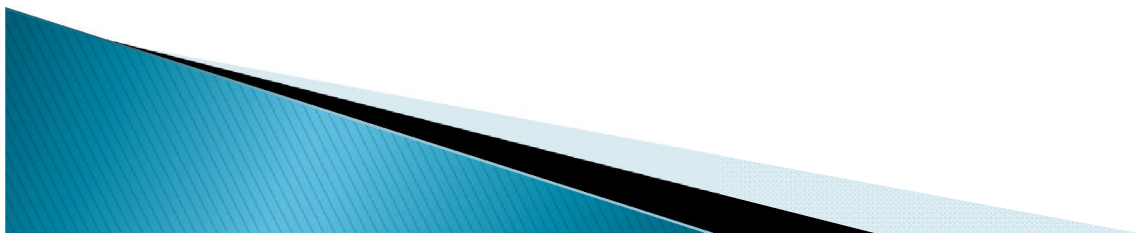
مثال. تعیین کنید چه تعداد جمله نیاز است تا مجموع سری $\{k^2-7k, k=1,2,3,....\}$ از 100000 بیشتر شود.



```
1 -    k=0;
2 -    S=0;
3 -    while S<100000
4 -        k=k+1;
5 -        S=S+(k^2+7*k);
6 -    end
7 -    disp('The requires sentences to exceed 100000 is===>')
8 -    disp(k)
9 -    disp('The summation is ===>')
10 -   disp(S)
```



```
The requires sentences to exceed 100000 is===>
    64

The summation is ===>
      104000
```



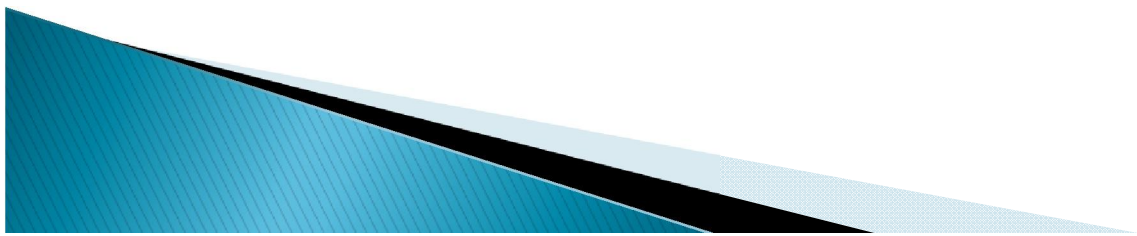
مثال

اعداد اول بین 2 تا 10000 را پیدا کنید.

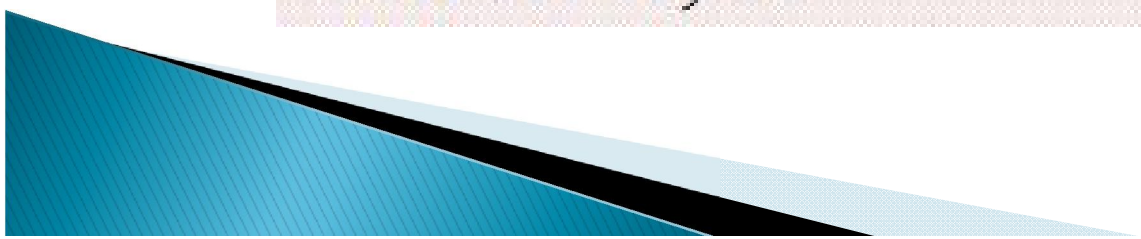
```
1  -    N=[];
2  -  for n=2:10000
3  -        k=0;
4  -        for i=2:fix(sqrt(n))
5  -            if rem(n,i)==0
6  -                k=k+1;
7  -                break
8  -            end
9  -        end
10 -        if k==1
11 -            continue
12 -        else
13 -            N=[N n];
14 -        end
15 -    end
16 -    disp(N);
17
18
```

مثال

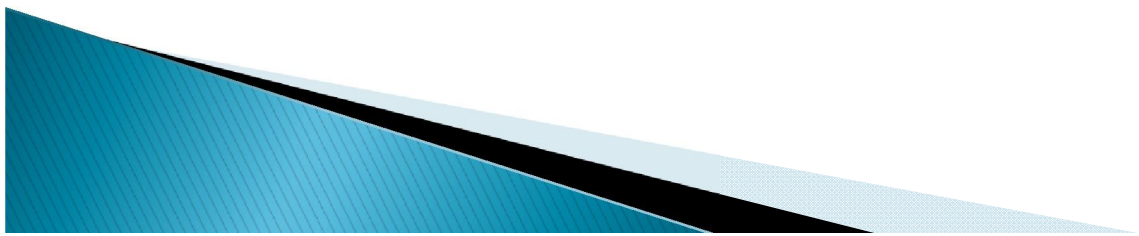
```
tf = [30 40 50 60 70]>40
a = [2.5 6.7 9.2 inf 4.8 NaN];
b = isfinite(a)
c = islogical(a)
d = islogical(b)
```



```
x = -3:3
tf = logical(x)
x = randperm(12)
x = reshape(x,3,4)
tf = (x<=5)
whos tf, x
```



```
c = [true false]
a = [1 2 3;4 5 6;7 8 9];
b = rem(a,2) ~= 1
a(b)
clear
a = 0;
c = a&b
```



دستورات ورودی و خروجی

- دستور input برای گرفتن ورودی از کاربر به کار می رود:

```
X=input('enter a number') ;
```

در اجرا، ابتدا پیغام enter a number چاپ میشود. سپس منتظر می ماند تا کاربر عدد یا ماتریسی وارد کند. این عدد یا ماتریس در X قرار میگیرد.

- تابع disp برای چاپ پیغام یا مقدار متغیر به کار می رود.

```
disp('hello')
```

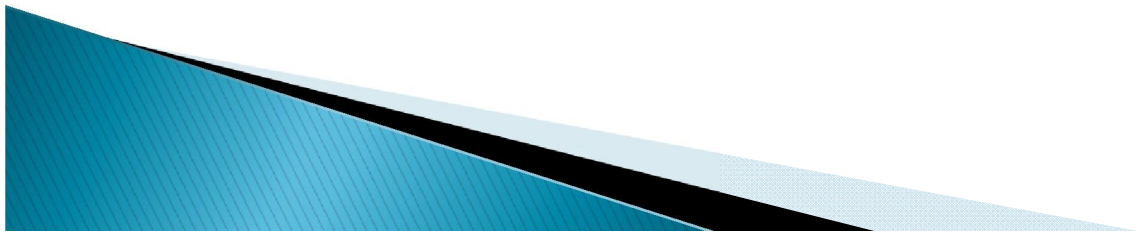
پیغام hello را چاپ می کند.

```
x=[3  4;5  6]
disp(x)
```

مقدار متغیر x را چاپ می کند.

```
disp('the value=') , disp(x)
```

با استفاده از علامت , می توان چند دستور را باهم در پنجره command اجرا کرد.

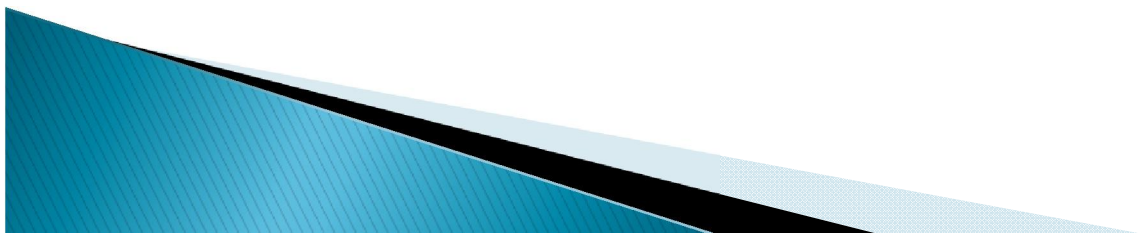


- برای اندازه گیری مدت زمان اجرای یک برنامه از دو دستور tic و toc استفاده می شود.

tic

برنامه مورد نظر

toc



چندجمله ایها

- چندجمله ایها، توابعی به صورت زیر میباشند:

$$f(x)=a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0$$

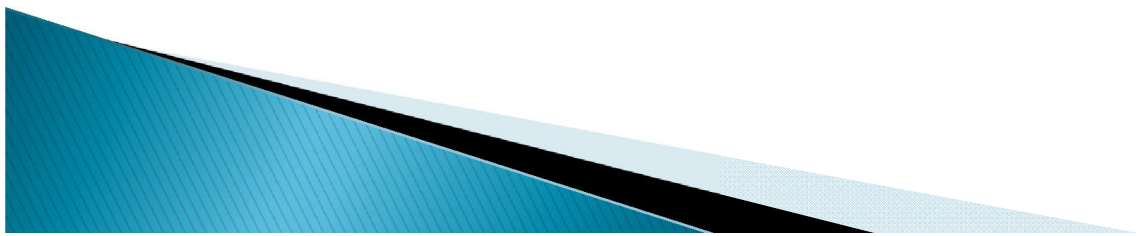
که در متلب با استفاده از بردار سطری بیان میشوندکه عناصر آنها ضرایب زیر میباشد.

$$a_n\ ,\ a_{n-1}\ ,\ldots,a_1\ ,\ a_0$$

مثال:

$$2x^2 - 4x + 10$$

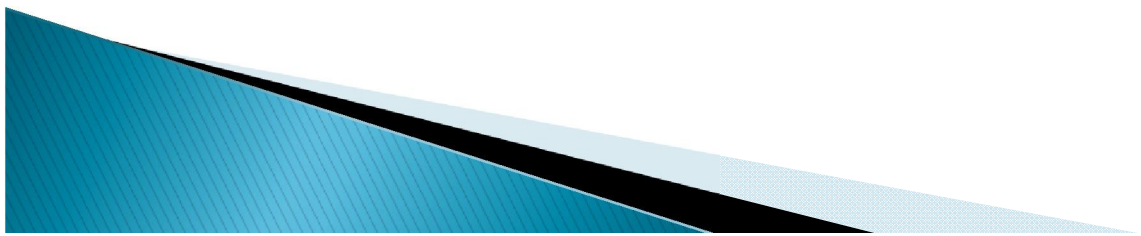
[2 -4 10]



ریشه های چندجمله ای

- از دستور roots استفاده میکنیم.

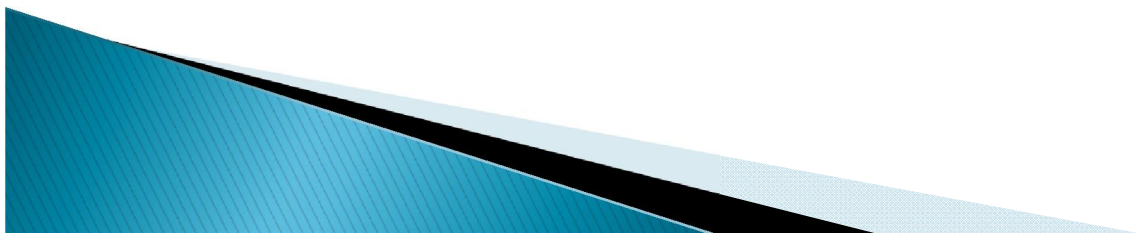
```
>> p=[4 10 -8];
>> r=roots(p)

r =

   -3.1375
    0.6375
```



```
>> r=roots([4 10 -8])

r =

   -3.1375
    0.6375

>>
```

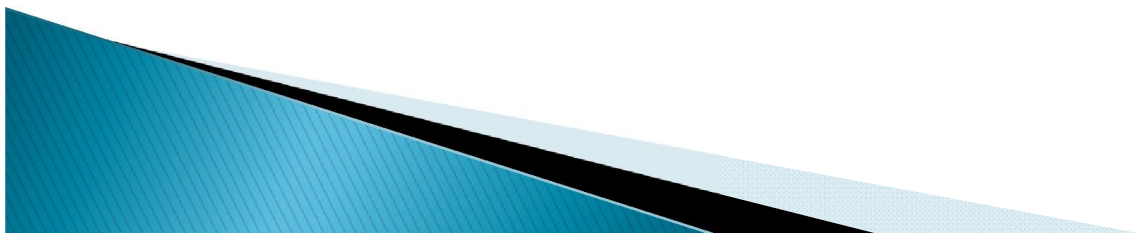
دو ریشه مختلط دارد:

```
>> a=[1 2 3];
>> r=roots(a)

r =

   -1.0000 + 1.4142i
   -1.0000 - 1.4142i
```



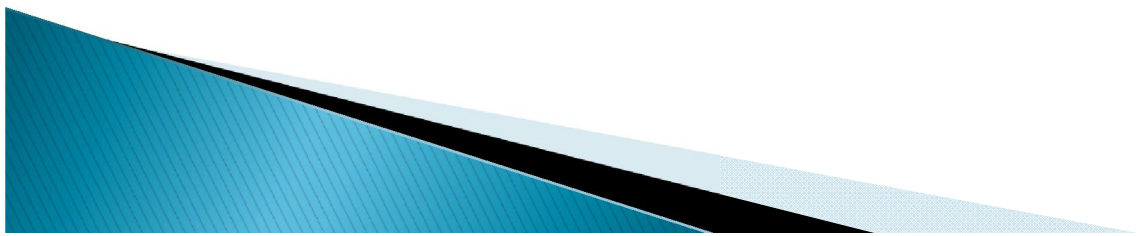
با دانستن ریشه های معادله میتوان ضرایب چندجمله ای مربوطه را محاسبه نمایید.

```
r =

  -1.0000 + 1.4142i
  -1.0000 - 1.4142i

>> poly(r)

ans =

    1.0000    2.0000    3.0000
```



محاسبه مقدار یک چندجمله ای

```
>> p=[1  0  -2  4];
>> polyval(p,5)

ans =

   119
```



ضرب و تقسیم چندجمله ایها

برای ضرب چندجمله ای a و b از دستور c=conv و برای a/b از دستور[q,r]=deconv(a,b) استفاده میکنیم.

```
>> a=[1 1 1];
>> b=[1 -1];
>> c=conv(a,b)

c =

     1     0     0    -1

>> [q r]=deconv(a,b)

q =

     1     2


r =

     0     0     3

>>
```

مشتق چندجمله ای

```
>> a=[1 1 1];
>> c=polyder(a)

c =

     2     1
```

مشتق حاصلضرب دو چندجمله ای

```
>> a=[1 1 1];
>> b=[1 -1];
>> c=polyder(a,b)

c =

     3     0     0
```

در صورتیکه تعداد آرگومانهای خروجی تابع polyder بربر 2 باشد، تابع مشتق تقسیم دو چندجمله ای (a/b) را تعیین می نماید:

```
>> a=[1 1 1];
>> b=[1 -1];
>> [q,p]=polyder(a,b)

q =

     1    -2    -2


p =

     1    -2     1
```

برازش منحنی چندجمله ای

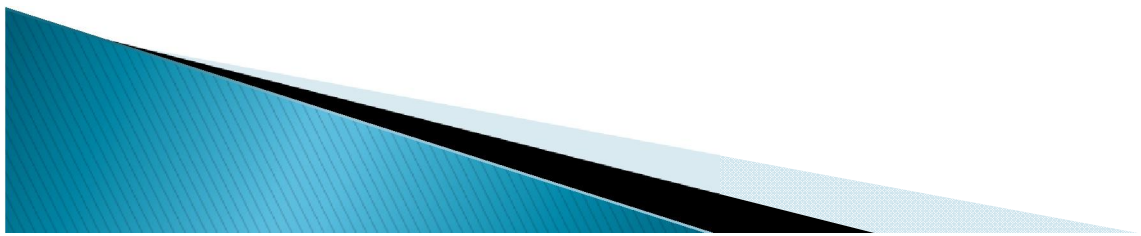
```
>> x=[1 2 3 4 5];
>> y=[5.5 43.1 128 290.7 498.4];
```

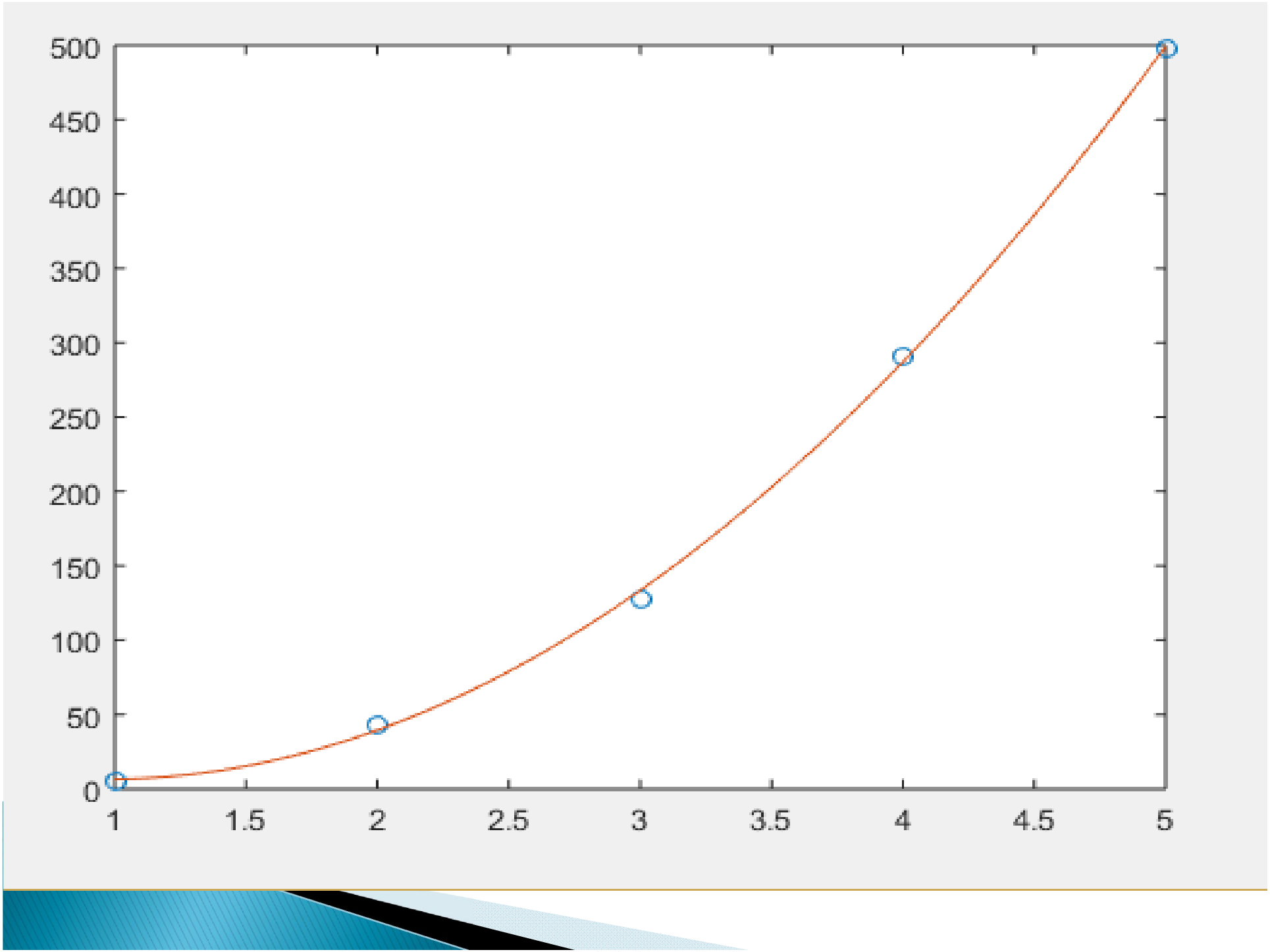
دستور زیر ضرایب بهترین چندجمله ای درجه 3 را محاسبه میکند که از بین نقاط فوق میگذرد.

```
>> p=polyfit(x,y,3)

p =

   -0.1917   31.5821  -60.3262   35.3400
```

- حال میتوانید برای مقایسه منحنی محاسبه شده و داده های اولیه را در یک نمودار رسم کنید:

```
>> x1=1:.1:5;
>> y1=polyval(p,x1);
>> plot(x,y,'o',x1,y1)
>>
```



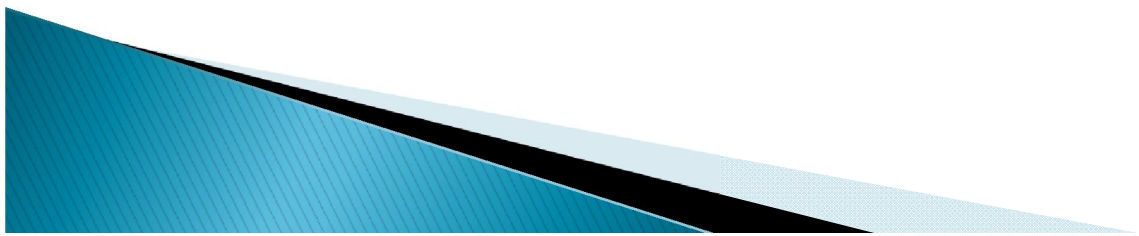

500
450
400
350
300
250
200
150
100
50
0
1
1.5
2
2.5
3
3.5
4
4.5
5

مثال

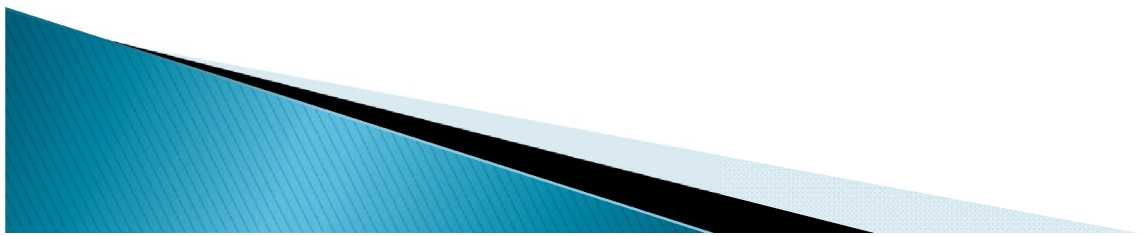
برای چندجمله ای $f(x)=x^5-12.1x^4+40.59x^3-17.015x^2-71.95x+35.88$

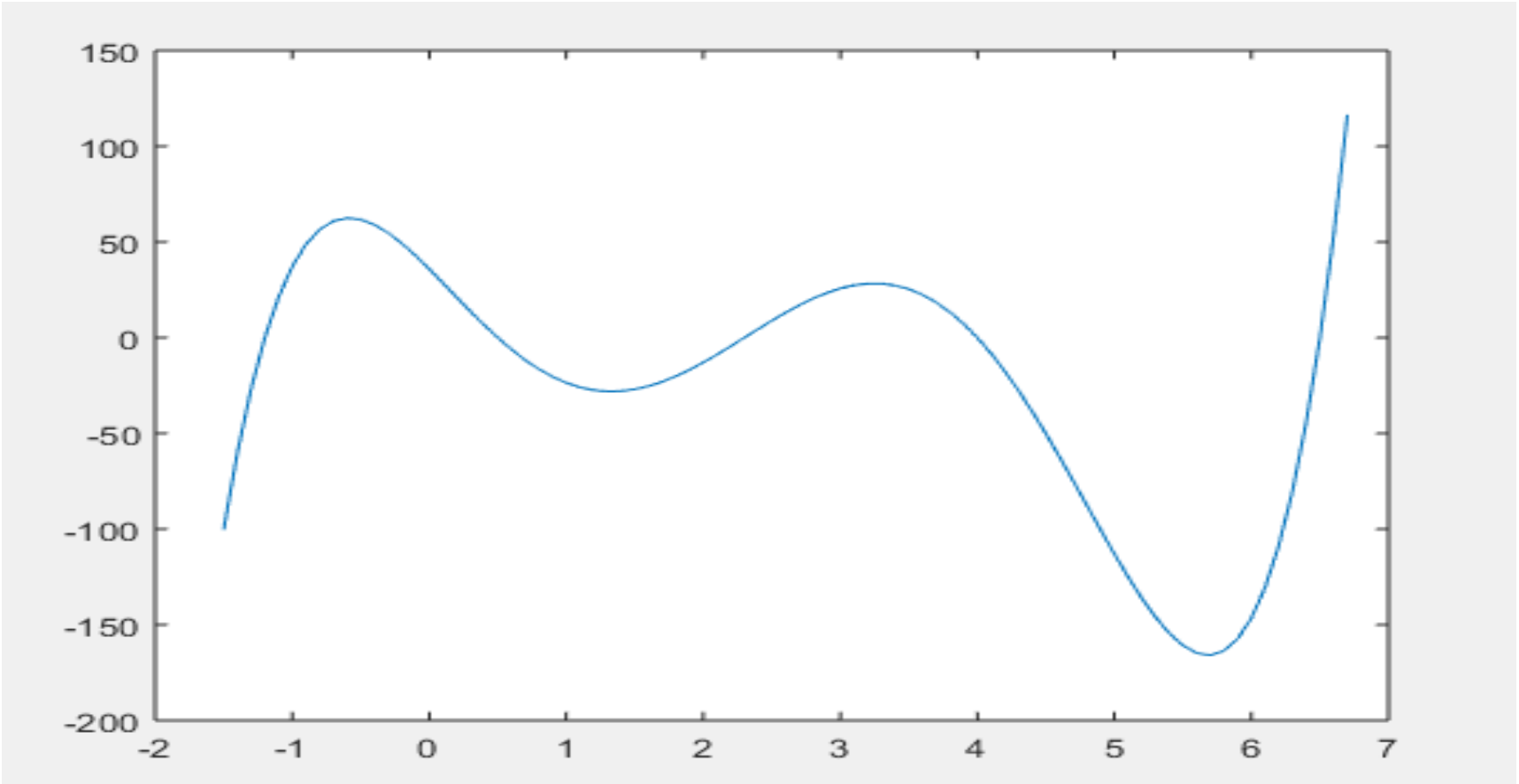
الف) f(9) را محاسبه کند.

ب) چندجمله ای را به ازای $-1.5\leq x\leq 6.7$ رسم کنید.



```
>> p=[1 -12.1 40.59 -17.015 -71.95 35.88];
>> polyval(p,9)

ans =

   7.2611e+03

>> x=-1.5:.1:6.7;
>> y=polyval(p,x);
>> plot(x,y)
```




150
100
50
0
-50
-100
-150
-200
-2
-1
0
1
2
3
4
5
6
7

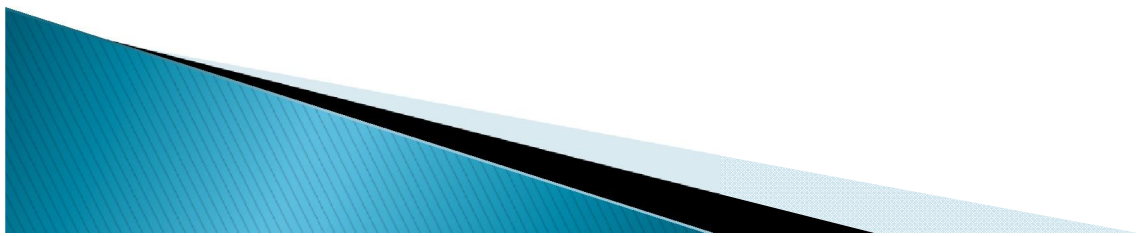
توابع سمبولیک

- به وسیله دستور زیر پارامترهای مورد استفاده در توابع را تعریف میکنیم.

```
syms x y z
```

حال میتوانیم تابع مورد نظر را تعریف کنیم.

```
Y=x^2+sin(z)
```

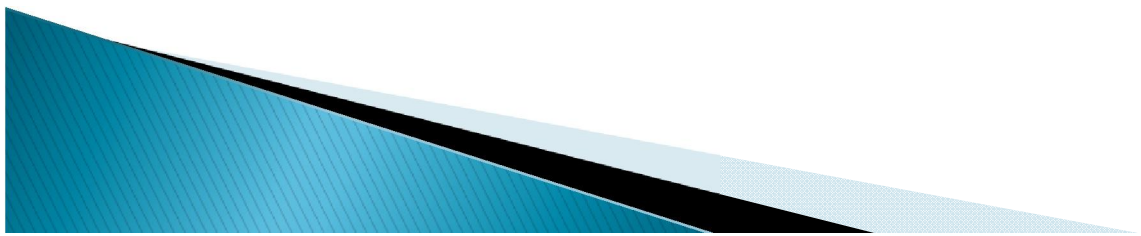


```
>> syms x y z
>> y=x^2+sin(z)

y =

x^2 + sin(z)

>> x=2; z=pi/2;
>> eval(y)
```

```
>>  clear
>>  syms  x
>>  y=x^2

y  =

x^2

>>  limit(y,2)

ans  =

4
```

```
>> diff(y)

ans =

2*x
```

```
>> diff(y,2,'x')

ans =

2
```



```
>> syms x y z
>> y=x^2+z^2

y =

x^2 + z^2

>> diff(y,2,z)

ans =

2

>> diff(y,z)

ans =

2*z

>> diff(y,'x')

ans =

2*x

>>
```

```
>> int(y,'x')

ans =

x^3/3 + x*z^2

>> int(y,'z')

ans =

x^2*z + z^3/3

>> int(int(y,'z'),'x')

ans =

(x*z*(x^2 + z^2))/3
```

```
>> int(y,'x',1,6)

ans =

5*z^2 + 215/3

>>
```

```
>> syms x y f g
>> f=x^2

f =

x^2

>> g=2*x+5

g =

2*x + 5

>> compose(f,g)

ans =

(2*x + 5)^2

>> compose(g,f)

ans =

2*x^2 + 5

>>
```

```
>>  y=sin(x)

y  =

sin(x)

>>  finverse(y)

ans  =

asin(x)

>>
```

```
>> y=x^2

y =

x^2

>> finverse(y)

ans =

x^(1/2)

>>
```

مثال

عبارت z=x^10-y^10 به صورت سیمبولیک تعریف کنید و آنرا به صورت حاصلضرب عبارات جبری بنویسید.

```
>> syms x y z
>> z=x^10-y^10

z =

x^10 - y^10

>> u=factor(z)

u =

[ x - y, x + y, x^4 + x^3*y + x^2*y^2 + x*y^3 + y^4, x^4 - x^3*y + x^2*y^2 - x*y^3 + y^4]
```

مثال

عبارات جبری z=(x-y)^3 و w=sin(x-y) را به صورت سمبولیک تعریف کنید و آنها را بسط دهید.

```
>> syms x y z
>> z=(x-y)^3;
>> w=sin(x-y);
>> z1=expand(z)

z1 =

x^3 - 3*x^2*y + 3*x*y^2 - y^3

>> w1=expand(w)

w1 =

cos(y)*sin(x) - cos(x)*sin(y)

>>
```

مشتق عبارت سمبولیک زیر را نسبت به x وy وa به دست آورید. $z = x^5 y^3 + x^3 y^5 + axy$

```
>> syms x y z a
>> z=x^5*y^3+x^3*y^5+a*x*y

z =

x^5*y^3 + x^3*y^5 + a*x*y

>> zx=diff(z,x);
>> disp(zx)
5*x^4*y^3 + 3*x^2*y^5 + a*y

>> zy=diff(z,y)

zy =

3*x^5*y^2 + 5*x^3*y^4 + a*x

>> za=diff(z,a)

za =
```

تمرین

- مشتق دوم و سوم عبارت قبل رانسبت به x به دست آورید.

```
>> syms x y z a
>> z=x^5*y^3+x^3*y^5+a*x*y;
>> zx1=diff(z,x,2)

zx1 =

20*x^3*y^3 + 6*x*y^5

>> zx2=diff(z,x,3)

zx2 =

60*x^2*y^3 + 6*y^5
```

- عبارت جبری زیر را در نظر بگیرید.مقدار x را به گونه ای پیدا کنید که f برابر 0 شود.مقدار c را به گونه ای پیدا کنید که f برابر 0 شود.

```
>> syms a b x y c
>> f=a*x^2+b*x+c;
>> X=solve(f,x)

X =

 -(b + (b^2 - 4*a*c)^(1/2))/(2*a)
 -(b - (b^2 - 4*a*c)^(1/2))/(2*a)

>> X=solve(f,c)

X =

- a*x^2 - b*x
```

ریشه های حقیقی معادله را تعیین کنید. $x^3 + x^2 + x + 1 = 0$

```
>> syms x
>> f=x^3+x^2+x+1

f =

x^3 + x^2 + x + 1

>> c=solve(f,x)

c =

   -1
  -1i
   1i
```

مثال

دستگاه معادلات جبری زیر را حل کنید.

$$\begin{cases} x^2 y^2 - xy = 6 \\ x^2 y + xy^2 = 12 \end{cases}$$

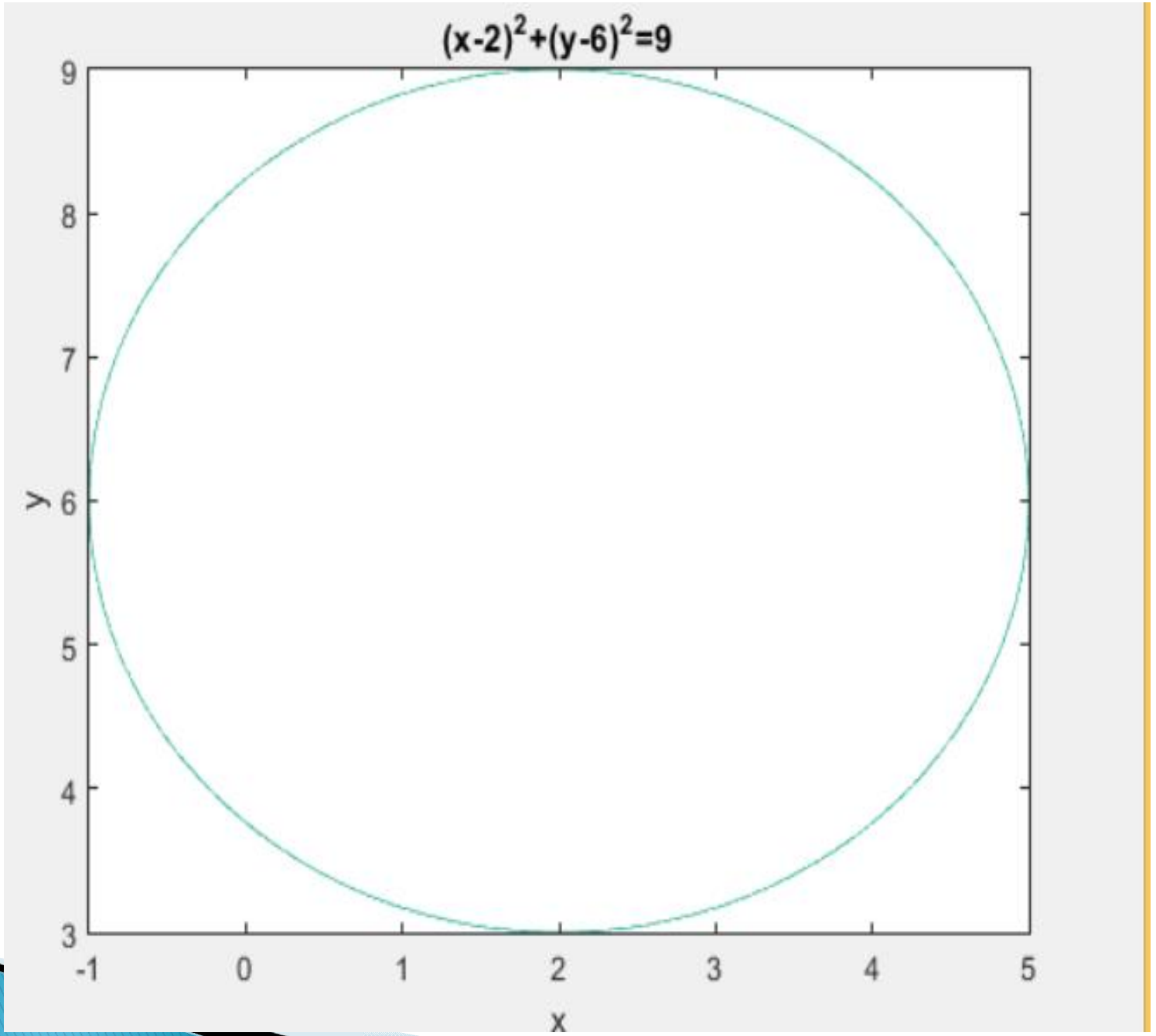
```
>> syms x y
>> f1=x^2*y+x*y^2-12;
>> f2=x^2*y^2-x*y-6;
>> root=solve(f1,f2);
>> disp(root)
    x: [4x1 sym]
    y: [4x1 sym]

>> X=root.x;
>> Y=root.y;
>> disp([X Y])
[   11^(1/2) - 3, - 11^(1/2) - 3]
[ - 11^(1/2) - 3,   11^(1/2) - 3]
[              3,              1]
[              1,              3]

>>
```

ترسیم عبارات سیمبولیک y=f(x)

ezplot(f, $[x_{\min}, x_{\max}, y_{\min}, y_{\max}]$)

دایره ای به شعاع 3 در نقطه [2,6] به عنان مرکز ایجاد کنید.

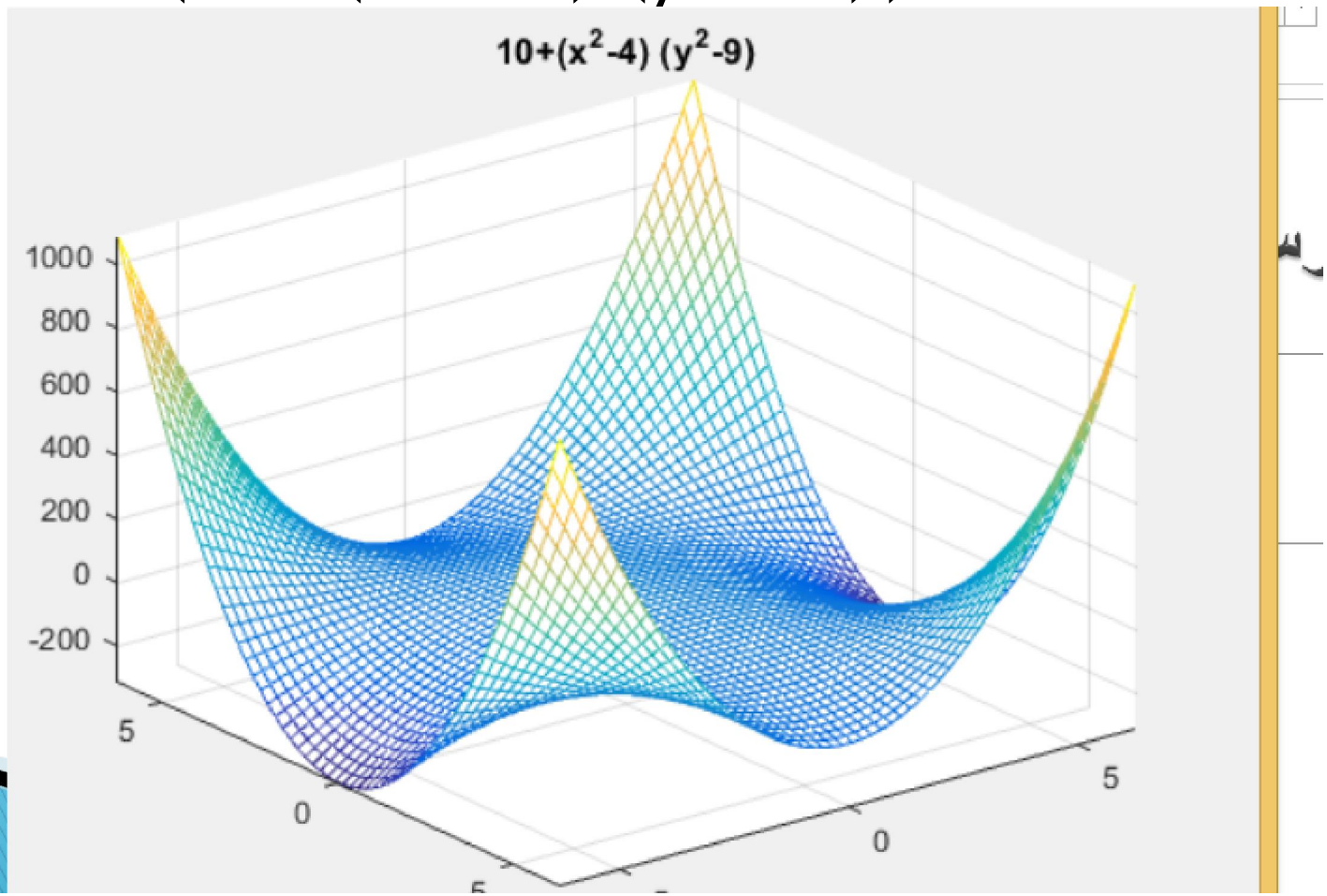
```
>> ezplot('(x-2)^2+(y-6)^2=9',[-1,5,3,9])
>>
```


$(x-2)^2+(y-6)^2=9$
9
8
7
6
5
4
3
y
-1
0
1
2
3
4
5
x

ترسیم عبارات سیمبولیک z=f(x,y)

ezmesh(f, $[x_{\min}, x_{\max}, y_{\min}, y_{\max}]$

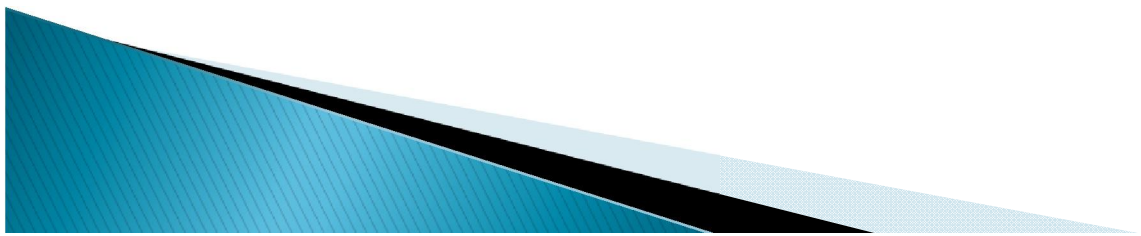
>> ezmesh('10+(x^2-4)*(y^2-9)')



محاسبه سری های سمبولیک

- مجموع سری E(i)=1/i را از 10 تا 1000 محاسبه کنید.

```
>> syms k
>> S=symsum(1/k,10,1000);
>> disp(double(S))
      4.656502606582091e+00

>> format short
>> disp(double(S))
     4.6565
```



- مجموع سری F(K)=K^3 را از m تا n محاسبه کنید. سپس مجموع سری را در حالتیکه m=5 و n=101 به دست آورید.

```
>> syms k n m
>> S=symsum(k^3,m,n)

S =

(n^2*(n + 1)^2)/4 - (m^2*(m - 1)^2)/4

>> pretty(S)
 2         2    2         2
n  (n + 1)     m  (m - 1)
---------- - ----------
     4             4

>> S=symsum(k^3,5,101)

S =

26532701
```

- عبارت زیررا به صورت سمبولیک تعریف کنید و عملیات زیر را بر روی آن انجام دهید

$$y = \frac{x^2 - 4x + 3}{x^2 - 6x + 8}$$

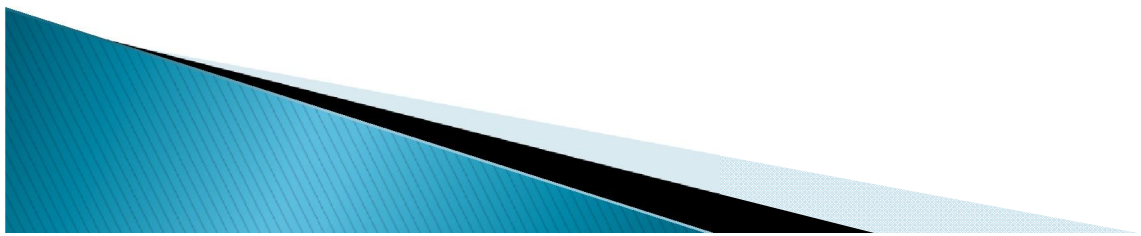
الف) تابع ایجاد شده را در بازه [-5,10] ترسیم کنید.

ب) ریشه های تابع را با نقاط دایره تو خالی مشخص کنید.

ج) نقاط max وmin را با علامت ضربدر مشخص کنید.

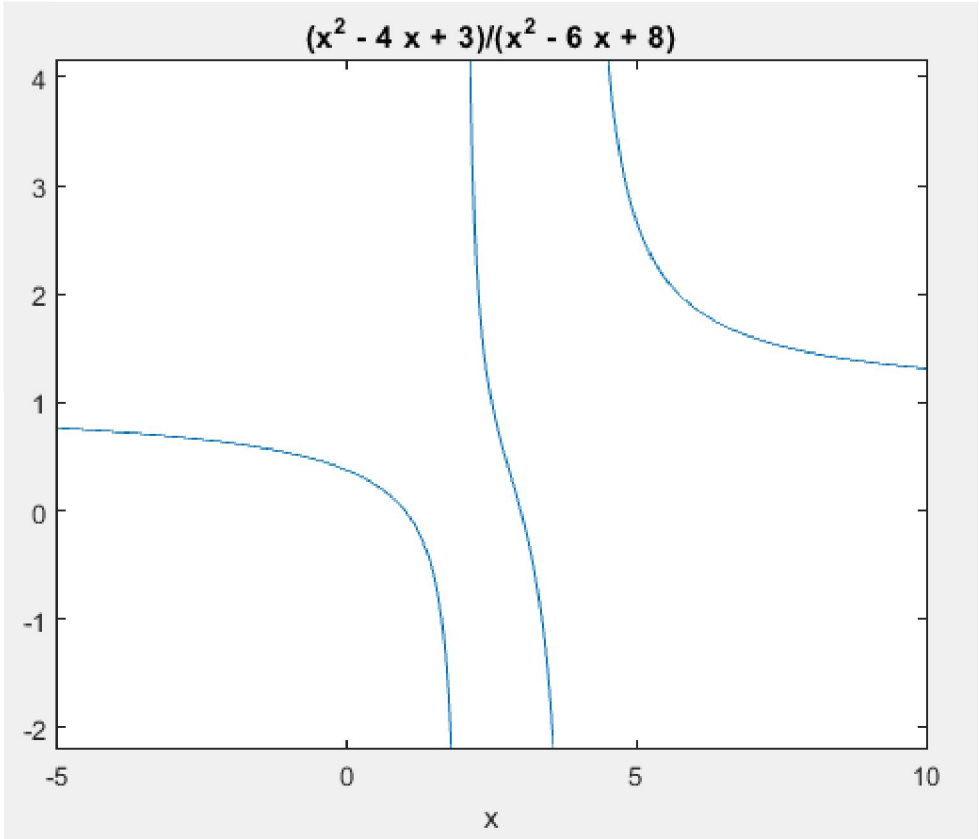
د) نقاط عطف را با علامت دایره تو پر مشخص کنید.

ه) مجانبهای افقی و عمودی آنرارسم کنید.



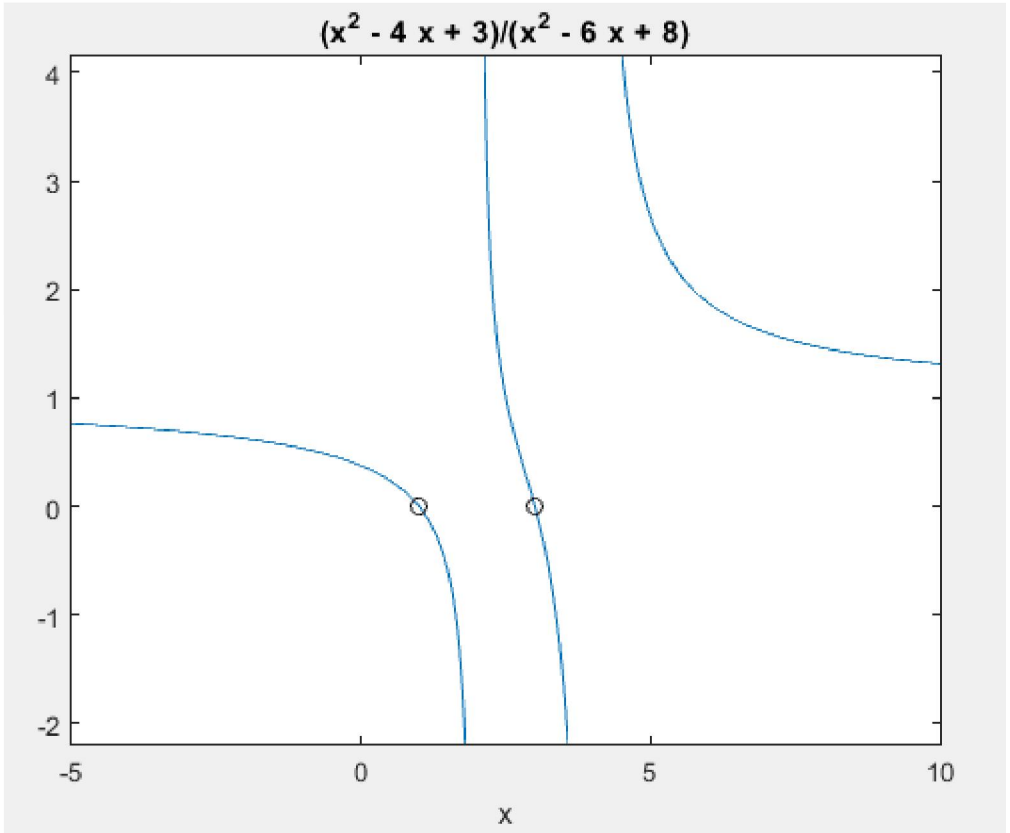
```
>> syms x
>> y1=x^2-4*x+3;
>> y2=x^2-6*x+8;
>> y=y1/y2;
>> pretty(y)
  2
x  - 4 x + 3
------------
  2
x  - 6 x + 8
```

```
>> ezplot(y,[-5,10])
>> hold on
```



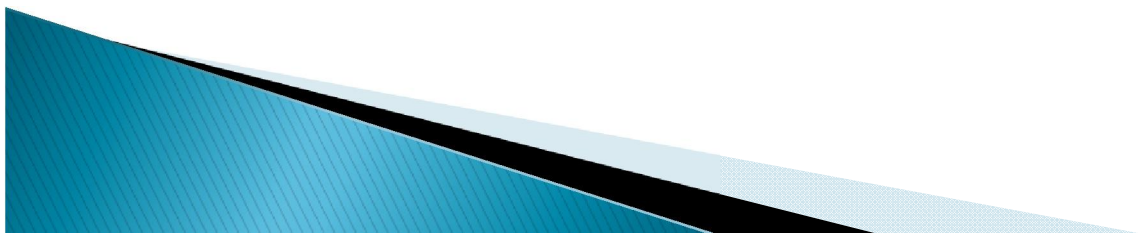
- ب) برای تعیین ریشه ها از دستور solve استفاده میکنیم.

```
>> root=solve(y,x);
>> root=double(root);
>> for i=1:length(root)
    if root(i)==real(root(i))
    plot(root(i),0,'ok')
    end
  end
```



- ج) برای تعیین max و min ریشه های مشتق اول تابع را محاسبه می کنیم

```
>> dy=diff(y,x,1);
>> root=solve(dy,x);
>> for i=1:length(root)
if root(i)==real(root(i))
Y=double(subs(y,x,root(i)));
plot(double(root(i)),Y,'xk')
end
end
```

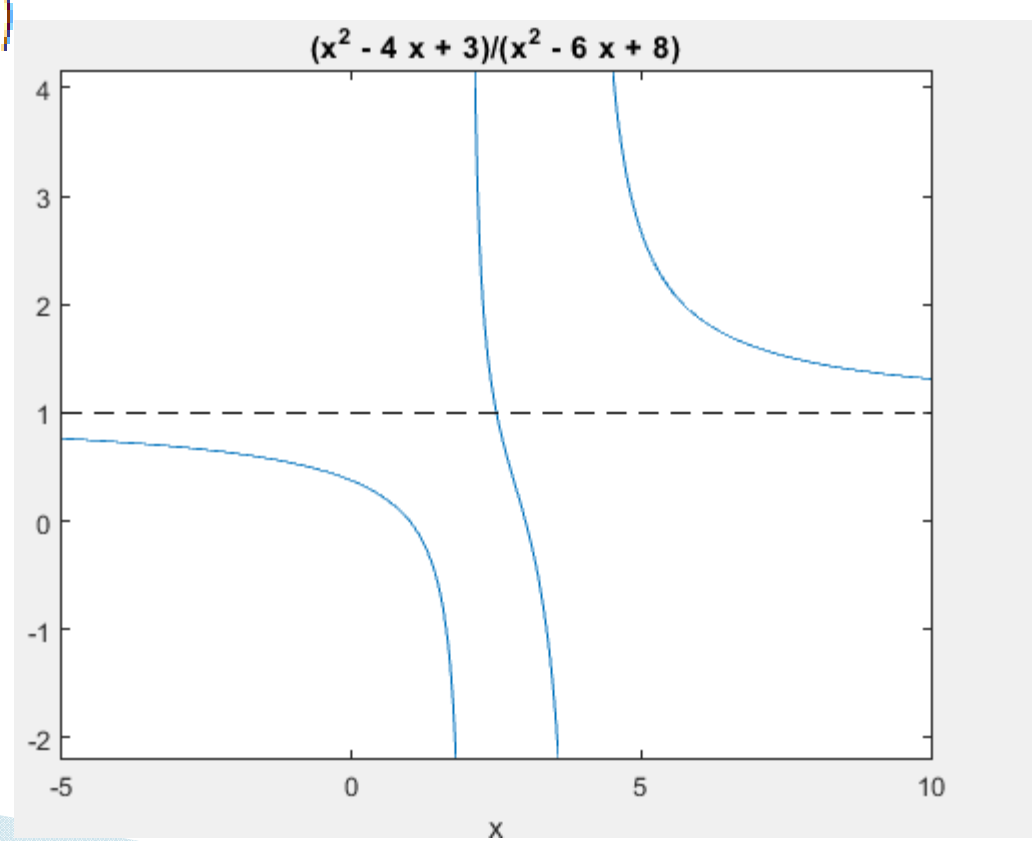


د) تعيين نقاط عطف:

```
>> d2y=diff(y,x,2);
>> root=solve(d2y,x);
>> for i=1:length(root)

if root(i)==real(root(i))
Y=double(subs(y,x,root(i)));
plot(double(root(i)),Y,'.k','markersize',20)
end
end
```

ترسیم مجانب افقی

```
L=double(limit(y,x,inf));
plot([-100 100],[L ,L],'--k')
```



newton.m*

```
function  newton(x0,tol,n )
k=0;
f=@(x)x^3-x^2-1;
df=@(x)3*x^2-2*x;
err=abs(f(x0)/df(x0));
disp('-------------------------------------------------');
disp('k     xn      f(xn)      df(xn)      |x(n+1)-xn|  ');
disp('-------------------------------------------------');
fprintf('%2.0f %12.6f %12.6f %12.6f\n',k,x0,f(x0),df(x0));
while(err>tol)&(k<=n)&(df(x0)~=0)
    x1=x0-f(x0)/df(x0);
    err=abs(x1-x0);
    x0=x1;
    k=k+1;
 fprintf('%2.0f %12.6f %12.6f %12.6f %12.6f\n',k,x0,f(x0),df(x0),err);
end;
end
```

```
>> newton(1,10^(-4),40)
-------------------------------------------------
k      xn        f(xn)        df(xn)        |x(n+1)-xn|
-------------------------------------------------
 0      1.000000    -1.000000      1.000000
 1      2.000000     3.000000      8.000000      1.000000
 2      1.625000     0.650391      4.671875      0.375000
 3      1.485786     0.072402      3.651108      0.139214
 4      1.465956     0.001352      3.515168      0.019830
 5      1.465571     0.000001      3.512556      0.000385
 6      1.465571     0.000000      3.512555      0.000000
>>
```