به نام خدا

آموزش MATLAB به زبان ساده

تهیه و تنظیم:محمد صادق نظری
دانشجوی مهندسی برق دانشگاه شهیدچمران

Nazari.sadegh@gmail.com

کاربرد های MATLAB:

MATLAB برنامه قدرتمند در تمام زمینه های علوم پایه ومهندسی در اکثر دانشگاههای کشور به پرکاربرد ترین نرم افزار جهت محاسبات ریاضی الگوریتم ها ، آنالیز، سمیولوشن ،دیتا،رسم نمودارو کارهای فنی مهندسی تبدیل شده است.

MATLAB مخفف MATRIX LABORATORY به معنی کتابخانه عملیات ماتریسی می باشد.

نرم افزارMATLAB از 5 قسمت تشکیل شده است:

1)رابط اصلی کاربر و نرم افزار

2)کتابخانه توابع ریاضی

3)زبان برنامه نویسی

4)گراف های MATLAB

5)رابط های خارجی(API)

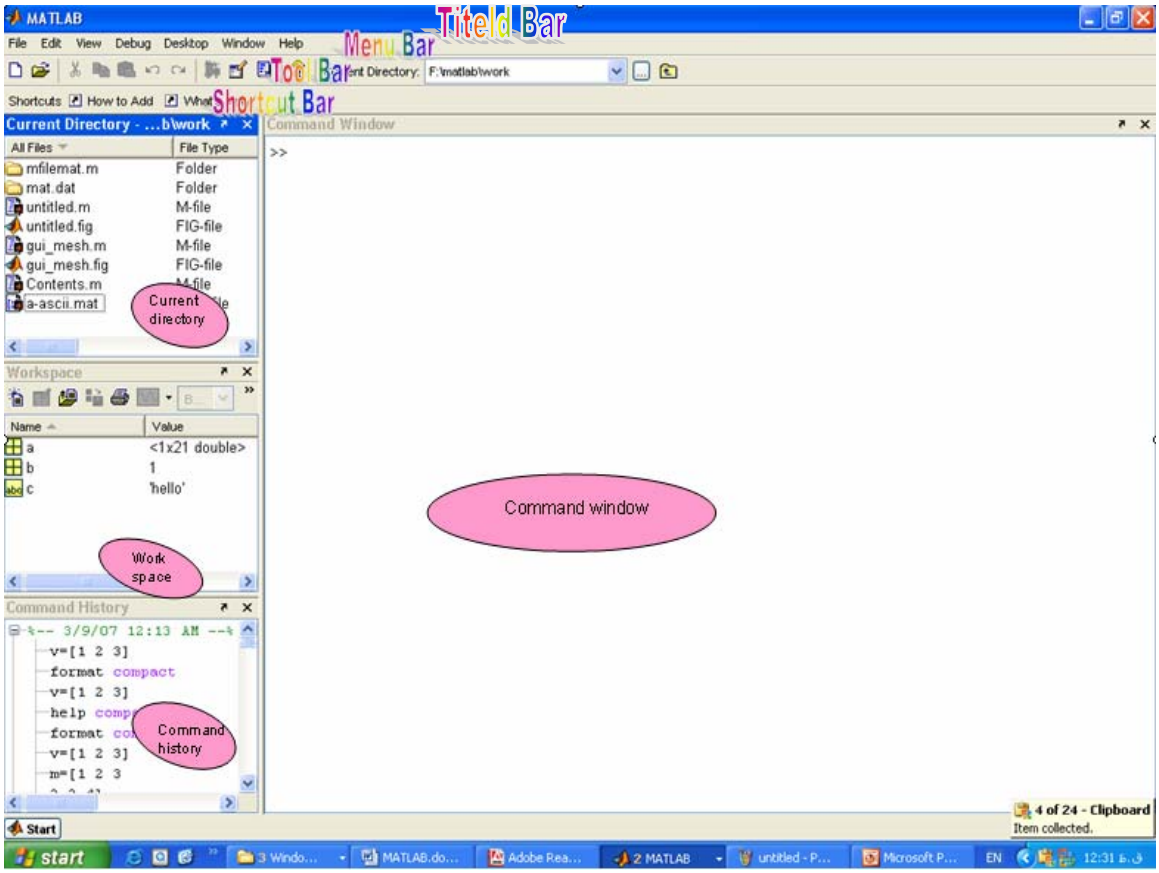
در اینجا قصد داریم شما را با قسمت های مختلف نرم افزار بجز قسمت آخر آشنا کنیم.البته بطور مختصر به ساخت سند های گرافیک GUIخواهیم پرداخت.

با این امید که مجموعه پیش رو در ارتقا سطح کیفی آموزش موثر واقع شود.

شرو ع به کار با MATLAB:

روی آیکون MATLAB در دسکتاپ دبل کلیک کنید تا نرم افزار لود شود و

صفحه زیر ظاهر شود:



رابط کاربر از 7 قسمت تشکیل شده است :

1)نوار عنوان (Titeld Bar)

2)نوارمنو(Menu Bar)

3)نوارابزار(Tool Bar)

4)نوارSHORT CUT (ShortCut Bar)

5)FILE MANAGER: که از 2 قسمت تشکیل شده است:

1)WORKSPACE:لیست متغیر ها و نوع آنها را نشان می دهد.

2)CURRENT DIRECTORY:شاخه ای از هارد که در آن قرار دارید

Nazari.Sadegh@Gmail.com

نمایش می دهد و بصورت DEFAULT شا خه ای که نرم افزار را درآن نصب کرده ایم می باشد که می توان در این قسمت متغیر جدید ایجاد کرد.

COMMAND WINDOW(6

COMMAND HISTORY(7

:DESKTOP

DESKTOP را می توان از طریق گزینه DESKTOPدر نوار منو تغییر داد که هر کدام از قسمت های DESKTOP را میتوان مخفی یا شناور کرد.تغییرات زیادی می توان در شمایل MATLAB ایجاد کرد.

همین طور میتوان متغیر ها را تغییر محل داد یا SIZE آنها را عوض کرد.

:FILE

بطور مثال:

جهت تغییررنگ FONT از گزینه :
FILE-PREFERENCE-FONTS-COLORS

:START

از طریق این گزینه میتوان به DEMO،HELP، FILEوکلیه امکانات نرم افزار دسترسی پیدا کرد.

:COMMAND WINDOW

در این پنجره باید کلیه دستورات را وارد کرد.دستورات را باید در جلوی علامت << وارد کرد که بصورت DEFAULT پاسخ را به شکل یک متغیر با نام ans

در command window بیان می کند.البته میتوان با انتخاب اسم پاسخ را در آن ذخیره کرد.

:COMMAND HISTORY

لیست کاملی از دستورهایی را که وارد کرده ایم بر اساس روز و زمان استفاده نشان می دهد که با دبل کلیک روی هر کدام می توان دستورات را از HISTORY اجرا کرد.

HELP: (راهنمای دستورات و نرم افزار)

جهت گرفتن اطلاعات در مورد مطلب خاص بکار می رود.

بطور مثال: HELP MAGIC

جهت دستیابی به HELP اصلی F1 را فشار می دهیم.

HELPاز 4 قسمت تشکیل شده است:

CONTENTS(1

INDEX(2

SEARCH(3

DEMO(4

که می توان از مثال های موجود در DEMO استفاده کرد و مطالب زیادی یاد گرفت.

CURRENT DIRECTORY(مسیر جاری)

در MATLAB می توان دستورات یا توابع را درفایل های خاصی ذخیره کرد که با پسوند M. ذخیره می شوند و این فایل ها در قسمت CURRENT DIRECTORY قرار می گیرند وبطور پیش فرض :

C:\PROGRAM FILES\MATLAB7.1\WORK

می باشد که می توان آن را تغییر داد.

WORKSPACE(پنجره فضای کاری)

این قسمت هر متغیری که ایجاد شده باشد نوع و اندازه آن را نمایش میدهد که به کمک این پنل می توان یک متغیر جدید ایجاد کرد یا متغیر های فعلی را ویرایش یا چاپ یا ذخیره کرد .

برای مشاهده محتویات یک متغیر روی آن دبل کلیک روی کنیم که می توان تک تک المان ها را مورد ویرایش قرار داد.

تعریف کردن آرایه ها

در MATLAB چهار نوع آرایه می توان تعریف کرد:

۱. اعداد اسکالر که تک عضوی هستند.

۲. بردارها که شامل یک سطر یا ستون می باشند (یک بعدی).

۳. ماتریسها که از اعضای چیده شده در یک آرایش مربعی تشکیل می گردند (دو بعدی).

۴. آرایه های با ابعاد بیش از دو.

اعضای یک آرایه می توانند عدد و یا حرف باشند و تفاوتی بین اعداد صحیح و اعشاری وجود نـدارد. در صورت جایگزینی یک عـدد و یـا حـرف در یـک متغـیر، MATLAB مقـدار جـایگزین شـده را بلافاصله نشان می دهد مگر آنکه عبارت تعریف متغیر با semicolon خاتمه یابد.

```
» a=2.5

a =

   2.5000

» a=3.2;
» a


a =

   3.2000

» p='hello'

p =

hello
```

ماتریس ها:

ماتریس ها درایه هایی از اعداد به صورت مستطیل هستند.در واقع معنای ماتریس تا حدی گسترده است که هر عدد یک ماتریس 1×1 در نظر گرفته می شود.هر بردار در حقیقت یک

ماتریس سطری یا ستونی است.

در matlab اعداد بصورت ماتریسی در نظر گرفته می شوند.

برای ساخت ماتریس ها باید ابتدا و انتهای ماتریس کروشه قرار دهیم و آرایه ها را با space یا کاما از هم جدا کرد.در انتهای هر سطر ; قرار می دهیم یا اینتر می زنیم..

بطور مثال:

```
» a=[1 2 3];
» b=[2 -1 0];
```

```
» m=[1 2 3
4 5 6]
m =
    1    2    3
    4    5    6
```

```
» q=[1, 2, 3
4 5 6; 7 8 9]
q =
    1    2    3
    4    5    6
    7    8    9
```

```
» n=['abcd'
'1234']
n =
abcd
1234
```

در matlab می توان ماتریس های از پیش ساخته را به کمک توابع خاص ایجاد کرد.

این توابع خاص به صورت زیرند:

zeros(1

ones(2

rand(3 (عدد رندم در بازه [1 ,0])

randn(4 (عدد رندم در بازه [1, 1-])

بطور مثال:

```
>> z=zeros(2,4)

z =

     0     0     0     0
     0     0     0     0

>> s=ones(2,2)

s =

     1     1
     1     1

>> n=randn(2,3)

n =

   -0.4326    0.1253   -1.1465
   -1.6656    0.2877    1.19

>> n=fix(10*rand(2,2))

n =

     9     6
     2     4
```

همچنین از فایا های data خارجی میتوان برای نمایش داده خاص که می تواند ماتریس باشد استفاده می کنیم که در ادامه بررسی خواهد شد.

اعداد در matlab

در رابطه های ریاضی از چند واسطه استفاده می شود:

1) variable ها(متغیر ها)

2)numbers ها(اعداد)

3) operator ها

4) function ها

در matlab لازم نیست متغیرها را تعریف کرد یا نوع آنها را مشخص کرد و با ایجاد متغیر نوع آن هم مشخص می شود.در ضمن matlab به کوچک و بزرگ بودن حروف حساس است.نام یک متغیر را می توان تا 63 کاراکتر در نظر گرفت.

تابع عددی مثل i,j, pi(Π) بطور پیش فرض در matlab وجود دارد.عدد دیگری که می توان استفاده کرد eps است که می توان یک مقدار جدید به مقدار پیش فرض نسبت داد البته می توان از دستور clear آن را به مقدار orginal برگرداند.

```
>>eps

ans=

2.2204  e-016

>>eps=1.e-6

eps=

1.0000  e-006

>>clear eps
```

```
>> eps

ans =

  2.2204e-016
```

ذخیره کردن و بازیابی داده ها

اگر بخواهید نام متغیرهای ایجاد شده را ببینید می توانید از دستور who استفاده نمایید:

```
» who

Your variables are:

a       n       q       w
m       p       v
```

برای مشاهده نام متغیرهای موجود به همراه اطلاعات اضافه تر در مورد آنها دستور whos را بکار ببرید:

```
» whos
  Name      Size          Bytes  Class

  a         1x1               8  double array
  m         2x3              48  double array
  n         2x4              16  char array
  p         1x5              10  char array
  q         3x3              72  double array
  v         1x3              24  double array
  w         1x8              16  char array

Grand total is 31 elements using 122 bytes
```

در صورتی که بخواهید کلیه متغیرهای موجود در محیط کار (workspace) را ذخیره کنید از دستور save استفاده کنید:

```
» save

Saving to: matlab.mat
```

این دستور، داده ها را در پرونده matlab.mat ذخیره می نماید. داده های موجود در این پرونـــده را می توان به طریق زیر بازیابی نمود:

```
» load

Loading from: matlab.mat
```

درصورتی که لازم باشد می توانید نام پرونده ذخیره را خودتان تعیین کنید:

```
» save myfile
```

و آن را با دستور زیر بازیابی نمایید:

```
» load myfile
```

اگر می خواهید که فقط بعضی از متغیرها را ذخیره کنید، نام آنها را بعد از نام پرونده بیاورید:

```
» save myfile t f
```

در صورتی که بخواهید تعدادی از متغیرها را از حافظه پاک کنیـــد کـافی اســت نـام آنـها را پـس از دستور clear بیاورید:

```
» who

Your variables are:

a       f       n       t       w
ans     m       p       v

» clear a f
» who

Your variables are:

ans     n       t       w
m       p       v
```

در صورت استفاده از دستور clear بدون ذکر نام متغیری پس از آن، کلیه متغیرهـا از حافظـه پـاک می شوند.

بازیابی ماتریس ها از m-file ها

در پنل current directory گزینه new m-file را انتخاب می کنیم.نام آنرا mfilemat.m انتخاب می کنیم .m-file را باز کرده وقرار می دهیم:

```
a=[...
    16 3 4
    5 7 8
    4 2 11];
```

از m-file خارج می شویم و در command window مینویسیم :

```
>>mfilemat
>> a

a=

4    3    16
8    7    5
11   2    4
```

Operator ها

اپراتور های موجود در matlab بصورت زیرند:

ضرب * ، اسلش / ،بک اسلش \ ،پاور ^ ،ترانهاده ' ،پرانتز () برای اولویت بخشیدن به عملیات ریاضی و کالن : که در ادامه مثال های متنوعی مطرح می شود.

```
>> 1:10

ans =

     1     2     3     4     5     6     7     8     9
10
```

ایجاد step با operator کالن:

```
>> 0:pi/4:pi

ans =

         0    0.7854    1.5708    2.3562    3.1416
```

جمع تمام المان های ستون 4

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

a =

     1     2     3
     3     4     2
     8     7     6

>> sum(a(1:3,3))

ans =

    11
```

اگر بخواهیم تمام المان های ستون آخر ماتریس را با هم جمع بزنیم:

```
>> sum(a(:,end))

ans =

    11
```

کاربرد دیگر operator :

اگر بخواهیم ماتریس b را در a قرار دهیم و جای ستون 2و3 را عوض کنیم.

```
>> a=b(:,[1 3 2 ])
```

```
a =

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

کاربرد دیگر end

سطر آخر، ستون 2 تا آخر:

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

A =

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

>> B=A(end,2:end)

B =

     8     7     6
```

Function ها

برای دیدن لیست کامل از توابع در matlab می توان help elfun را وارد کرد

برای دیدن لیست توابع خاص مثل Besselو gamma می توان

help spec fun را وارد کرد.

برخی از توابعی که در ساختن آرایه ها بکار می روند عبارتند از:

یک ماتریس ۲×۲ با مولفه های ۱ ایجاد می کند	ones(2)
یک ماتریس ۳×۲ با مولفه های ۱ ایجاد می کند	ones(2,3)
یک ماتریس ۳×۲ با مولفه های صفر ایجاد می کند	zeros(2)
یک ماتریس یکه ۳×۳ ایجاد می کند	eye(3)
برداری با ۷ مولفه با فواصل مساوی بین ۱- و ۵ ایجاد می کند	linspace(-1,5,7)
برداری با ۸ مولفه با فواصل لگاریتمی مساوی بین 10^{-1} و 10^{2} ایجاد می کند	linspace(-1,2,8)

تعدادی از توابعی که روی آرایه ها عمل می کنند عبارتند از:

حاصل جمع مولفه های x	sum(x)
حاصل جمع مولفه های x از اول تا هر مولفه	cumsum(x)
حاصلضرب مولفه های x	prod(x)
حاصلضرب مولفه های x از اول تا هر مولفه	cumprod(x)
بزرگترین مولفه x را پیدا می کند	max(x)
کوچکترین مولفه x را پیدا می کند	max(x)
مولفه های x را مرتب می کند	sort(x)
میانگین حسابی مولفه های x	mean(x)
انحراف معیار مولفه های x	std(x)

عملیات ماتریسی روی آرایه ها

در MATLAB می توان دو نوع عملیات روی آرایه ها انجـام داد کـه بـه آنـها عملیـات ماتریسـی و عملیات عضو به عضو می گویند. عملیات ماتریسی شامل محاسبه ترانهاده، ضرب ماتریسی، جمـع و تفریق آرایـه های هم اندازه و غیره می شود. ترانهاده یک ماتـریـس با کمک علامـت پریـم بدسـت می آید:

```
» r=rand(2,4)
r =
   0.9501   0.6068   0.8913   0.4565
   0.2311   0.4860   0.7621   0.0185
» r'
ans =
   0.9501   0.2311
   0.6068   0.4860
   0.8913   0.7621
   0.4565   0.0185
```

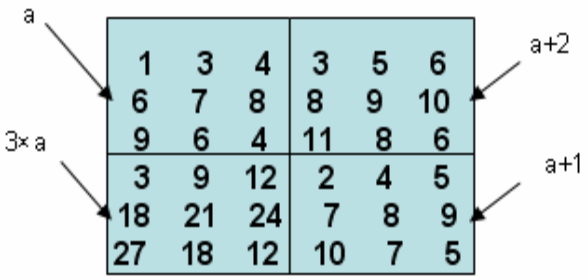
ضرب ماتریسی با استفاده از علامت * و جمع و تفریق ماتریسـها بـا اسـتفاده از علامتـهای مربوطـه انجام می گیرند:

```
» v=[1:4];
» r*v'
ans =
   6.6636
   3.5634
```

```
» s=[0:3; 2:-.5:.5];
» s+r
ans =
   0.9501   1.6068   2.8913   3.4565
   2.2311   1.9860   1.7621   0.5185
```

چسباندن ماتریس ها به هم:

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

a =

     1     3     4
     6     7     8
     9     6     4

>> b=[a a+2;3*a  a+1 ]

b =
```



پاک کردن سطر یا ستون در ماتریسها

اگر : را در مورد سطر یا ستون بکار ببریم تمام عناصر سطر یا یتون شورده می شود.پس برای پاک کردن سطر یا ستون آن سطر یا یتون را برابر یک مقدار پوچ قرار می دهیم.

مثلا برای پاک کردن ستون 2 به این صورت عمل می کنیم:

```
>> x=a

x =

     1     3     4
     6     7     8
     9     6     4

>> x(:,2)=[]
```

```
x =

     1     4
     6     8
     9     4
```

نمی توان یک المان خاص را از ماتریس حذف کرد:

```
>> x(1,2)=[]

???  Indexed empty matrix assignment is not allowed.
```

عملیات عضو به عضو روی آرایه ها

انجام عملیات جبری روی آرایه ها در MATLAB نیازمند دقت است. بطور کلی دو نوع عملیات می توان روی آرایه ها انجام داد: ۱-عملیات عضو به عضو، ۲-عملیات برداری-ماتریسی. اشتباه گرفتن این دو نوع عملیات باعث بروز مشکل در محاسبات می گردد. دو بردار زیر را در نظر بگیرید:

```
» a=[1 2 3];
» b=[2 -1 0];
```

فرض کنید که می خواهید این دو را در هم ضرب کنید:

```
» a*b
??? Error using ==> *
Inner matrix dimensions must agree.
```

دلیل گرفتن پیام خطا از عمل فوق این است که در MATLAB استفاده از علامت ضرب به تنهایی به معنای ضرب ماتریسی است. بنابراین عمل بالا را می توان با ترانهاده بردار دوم به انجام رسانید:

```
» a*b'
ans =
    0
```

این عمل در حقیقت ضرب اسکالر دو ماتریس است، یعنی: ۱×۲+۲×(−۱)+۳×۰=۰

حال اگر بخواهید ضرب عضو به عضو این دو بردار را به دست آورید باید یک نقطه قبل از علامت ضرب بگذارید:

```
» a.*b
ans =
    2   -2    0
```

همین دستورالعمل را می توان برای تقسیم و به توان رساندن آرایه ها بکار بست:

```
» a.^2
ans =
    1     4     9
```

در صورت فراموش کردن نقطه قبل از علامت توان:

```
» a^2
??? Error using ==> ^
Matrix must be square.
```

تنظیم خروجیها روی صفحه نمایش با دستورات disp و format

اگر مقدار یک متغیر را بخواهید بدانید می توانید آن را با نوشتن نام متغیر مشاهده کنید. در این صورت MATLAB نام متغیر و به دنبال آن علامت تساوی را نشان داده و سپس مقدار را در سطر یا سطور بعد می نویسد. برای دیدن مقدار متغیر بدون آنکه لازم باشد دوباره نام آن و علامت تساوی را مشاهده کنید می توانید دستور disp را بکار ببرید.

```
» x=[2 4 5];
» disp(x)
    2     4     5
» y='That is better';
» disp(y)
That is better
```

پنجره MATLAB را می توانید با دستور clc پاک کنید:

```
» clc
```

همانطور که قبلاً دیدید دستور format compact باعث می شود که خطوط اضافی هنگام ارائه نتایج حذف گردند. دستور format دارای کاربردهای فراوان دیگری نیز هست. فرض کنید که می خواهید مولفه های بردار زیر را روی صفحه نمایش ببینید:

```
» v=exp(-10*(1:5))
v =
  1.0e-004 *
    0.4540    0.0000    0.0000    0.0000    0.0000
```

واضح است که در حالت فعلی نمی توانید مقادیر مولفه ها را بخوانید. در این وضعیت می توانید با کمک دستور format نحوه نمایش اعداد را تغییر دهید:

```
» format long
» v
v =
  1.0e-004 *
  Columns 1 through 4
   0.45399929762485   0.00002061153622   0.00000000093576   0.00000000000004
  Column 5
   0.00000000000000
```

مشاهده می کنید با وجود اینکه این دستور تعداد اعداد نشان داده شده بعد از ممیز را افزایش می دهد ولی هنوز قادر نیست که همه مولفه های بردار مورد نظر را بطور مناسبی نمایش دهد. در چنین حالتی بهتر است اعداد را با استفاده از نماد علمی به نمایش بگذارید:

```
» format short e
» v
v =
  4.5400e-005  2.0612e-009  9.3576e-014  4.2484e-018  1.9287e-022
```

برای اطلاع بیشتر از امکانات دستور format توصیه می شود که توضیحات مربوط به این دستور را در help مطالعه کنید.

چند جمله ایها

یک چند جمله ای در MATLAB به صورت یک بردار سطری که مولفه های آن ضرایب چند جمله ای به ترتیب نزولی هستند معرفی می شود. برای مثال چند جمله ای $p(x) = x^3 - 2x + 5$ در MATLAB به شکل زیر معرفی می گردد:

```
» p=[1  0  -2  5];
```

ریشه های یک چند جمله ای

ریشه های یک چند جمله ای را می توانید به صورت زیر بدست اورد:

```
» r=roots(p)
r =
  -2.0946
   1.0473 + 1.1359i
   1.0473 - 1.1359i
```

با دانستن ریشه های معادله می توانید ضرایب چند جمله ای مربوطه را محاسبه نمائید:

```
» p2=poly(r)
p2 =
   1.0000    0.0000   -2.0000    5.0000
```

محاسبه مقدار یک چند جمله ای

تابع polyval مقدار چند جمله ای را در هر نقطه محاسبه می نماید. برای مثال مقدار p(۵) به طریق زیر محاسبه می گردد:

```
» polyval(p,5)
ans =
   120
```

ضرب و تقسیم چند جمله ایها

برای ضرب و تقسیم چند جمله ایها می توانید توابع conv و deconv را بکار ببرید. چند جمله ایهای $a(x)=x^2+x+1$ و $b(x)=x-1$ را در نظر بگیرید. حاصلضرب این دو چند جمله ای به طریق زیر بدست می آید:

```
» a=[1  1  1]; b=[1  -1];
» c=conv(a,b)
c =
    1    0    0   -1
```

و تقسیم a/b نیز به صورت زیر قابل محاسبه است:

```
» [q,r]=deconv(a,b)
q =
    1    2
r =
    0    0    3
```

مشتق چند جمله ای

مشتق چند جمله ای را می توانید با بکار بردن تابع polyder محاسبه کنید.

```
» c=polyder(a)
c =
   2    1
```

مشتق حاصلضرب دو چند جمله ای (a×b) را می توانید به صورت زیر بدست آورید:

```
» d=polyder(a,b)
d =
   3    0    0
```

در صورتی که تعداد آرگومانهای خروجی تابع polyder برابر ۲ باشد، تابع مشتق تقسیم دو چند جمله ای (a/b) را تعیین می نماید:

```
» [q,d]=polyder(a,b)
q =
    1   -2   -2
d =
    1   -2    1
```

حل معادلات دیفرانسیل به کمک matlab

معادله بر حسب متغیر x حل می شود

```
>> dsolve('Dx=-a*x')

ans =

C1*exp(-a*t)
```

معادله بر حسب متغیر f حل می شود با مقدار اولیه:

```
>> y=dsolve('Df=f+sin(t)','f(pi/2)=0')

y =
-1/2*cos(t)-1/2*sin(t)+1/2*exp(t)/(cosh(pi)+sinh(pi))^(1/2)
```

محاسبه انتگرال به کمک matlab

محاسبه انتگرال نا معین:

ابتدا باید x را بصورت سمبولیک تعریف کنیم: بطور مثال:

(سمبولی کردن (x) syms)

```
>> syms x;
>> int(atan(x))
ans =
x*atan(x)-1/2*log(x^2+1)
```

محاسبه انتگرال معین:

$$\int_0^{\pi/2} \sin(2x)dx = 1$$

```
>> syms x;
>> int(sin(2*x),0,pi/2)
ans =
1
```

برازش منحنی چند جمله ای

تابع polyfit ضرایب بهترین چند جمله ای را پیدا می کند که از میان مجموعه نقاط داده شده عبور می نماید. به عنوان مثال مجموعه نقاط زیر را در نظر بگیرید:

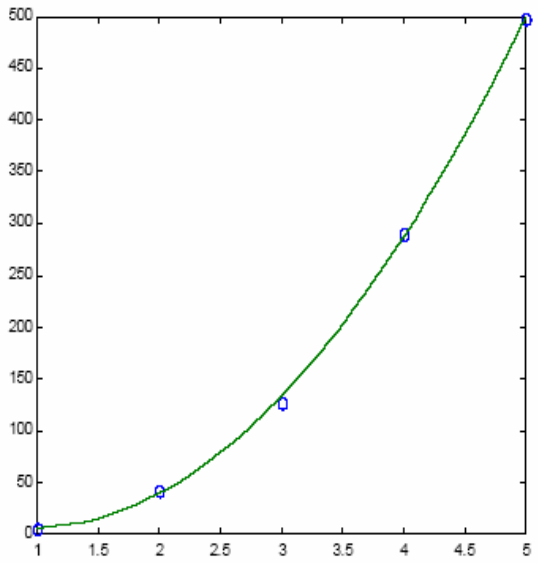
```
» x=[1  2  3  4  5];
» y=[5.5  43.1  128  290.7  498.4];
```

دستور زیر ضرایب بهترین چند جمله ای درجه سوم را محاسبه می کند که از بین نقاط فوق می گذرد:

```
» p=polyfit(x,y,3)
p =
   -0.1917   31.5821  -60.3262   35.3400
```

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

```
» x2=1:.1:5;
» y2=polyval(p,x2);
» plot(x,y,'o',x2,y2)
```



عملیات و توابع منطقی

عملگرهای منطقی

در MATLAB علامتهای زیر برای مقایسه مقادیر عددی و حرفی بکار می روند.

کوچکتر از	<
کوچکتر از یا مساوی با	<=
بزرگتر از	>
بزرگتر از یا مساوی با	>=
مساوی با	==
مخالف با	~=

چنین مقایسه ای را می توان بین دو اسکالر، دو آرایه یا اسکالر و اعضای آرایه انجام داد. مثالهایی برای نحوه عمل این عملگرها در زیر آورده می شوند. توجه کنید که حاصل همه عملیات منطقی می تواند ۰ به معنی نادرست یا ۱ به معنی درست باشد.

```
» 3<5
ans =
    1

» [1 2]>=[0 3]
ans =
    1    0

» a=[1 2 3
2 3 4];
» b=[-1 2 1
0 2 4];
» a~=b
ans =
    1    0    1
    1    1    0
```

بردار زیر را در نظر بگیرید:

```
» x=[1  2  -1  0  -5  4  -1.5  3  2.5  -.5];
```

عبارت زیر مولفه های مثبت این بردار را نمایش می دهد:

```
» x(x>0)
ans =
   1.0000    2.0000    4.0000    3.0000    2.5000
```

روابط منطقی را می توان با استفاده از عملگرهای منطقی با هم ترکیب کرد. این عملگرها عبـــارتند از:

و (ترکیب عطفی)	&
یا (ترکیب فصلی)	|
یا (مانع جمع)	xor
نقیض	~

مثالهایی از طرز عمل این عملگرها در زیر آورده شده اند:

```
» m=[1  2  4;  -2  3  -1];
» ~(m>0)
ans =
    0    0    0
    1    0    1
» (m>0)|(m<=2)
ans =
    1    1    1
    1    1    1

» (m>0)&(m<=2)
ans =
    1    1    0
    0    0    0

» xor([0 0 1 1],[0 1 0 1])
ans =
    0    1    1    0
```

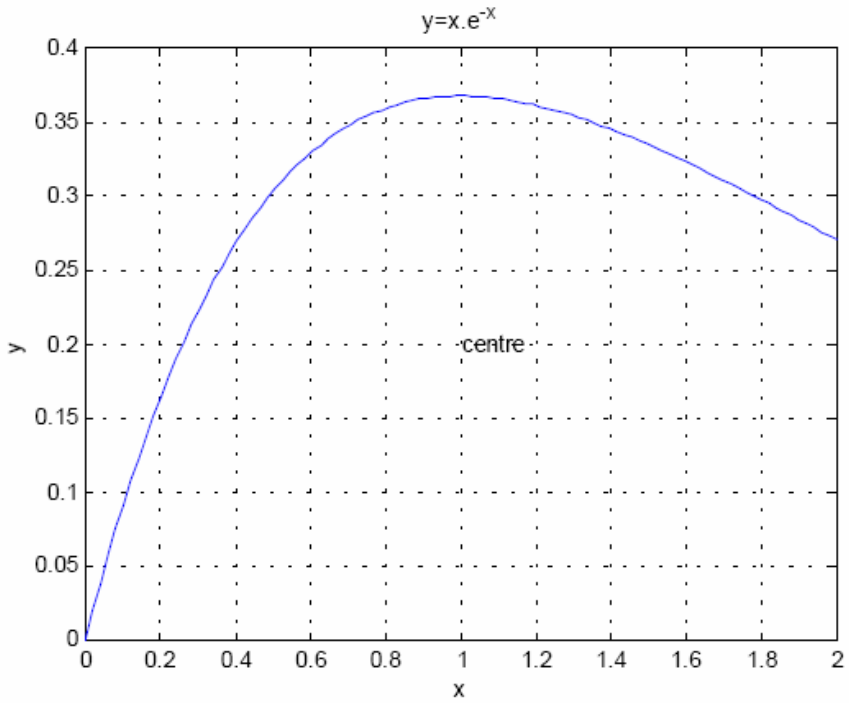
توجه کنید که xor یک تابع است و دو بردار ورودی به آن باید هم اندازه باشند.

ترسیم داده ها

نمودارهای ۲ بعدی

مجموعه دستورات زیر نحوه ترسیم یک تابع بر حسب یک متغیر مستقل را نشان می دهد:

```
x=linspace(0,2); y=x.*exp(-x);
plot(x,y)
grid
xlabel('x')
ylabel('y')
title('y=x.e^{-x}')
text(1,.2,'centre')
```



هفت خط فوق به ترتیب اعمال زیر را انجام می دهند:

۱- بردار متغیرهای مستقل (x) و تابع (y) را ایجاد می کند.

۲- مقادیر y را بر حسب x رسم می نماید.

۳- شبکه را به نمودار می افزاید.

۴- توضیح محور افقی را می نویسد.

۵- توضیح محور عمودی را می نویسد.

۶- تیتر نمودار را در بالای آن می نویسد.

۷- در نقطه مورد نظر (در این مثال نقطه (۰/۲ و ۱)) متغیر حرفی مشخص شده (در این مثال centre) را می نویسد.

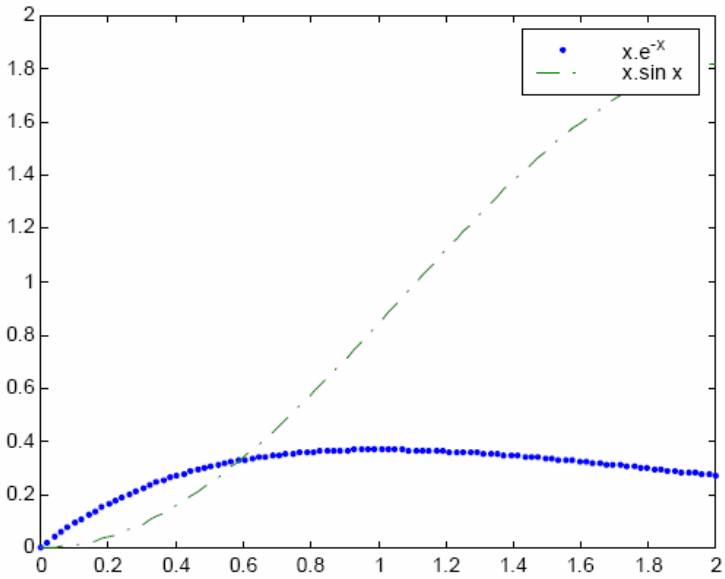
می توانید نمودار ایجاد شده را به کمک دستور Save As در منوی File پنجره نمودار، ذخیره نمایید. این دستور نمودار را در یک پرونده که نام آن را خودتان وارد خواهید کرد و دنباله آن ffig. می باشد ذخیره می کند. شما می توانید این نمودار را در دفعات بعدی کار با MATLAB با استفاده از دستور open بازیابی نمایید.

در هنگام رسم نمودارها می توانید از علامتهای مختلف (بجای خط) برای رسم توابع استفاده کنید. همچنین می توانید بیش از یک تابع را در یک نمودار نمایش دهید.

```
» plot(x,y,'.',x,x.*sin(x),'-.')
```

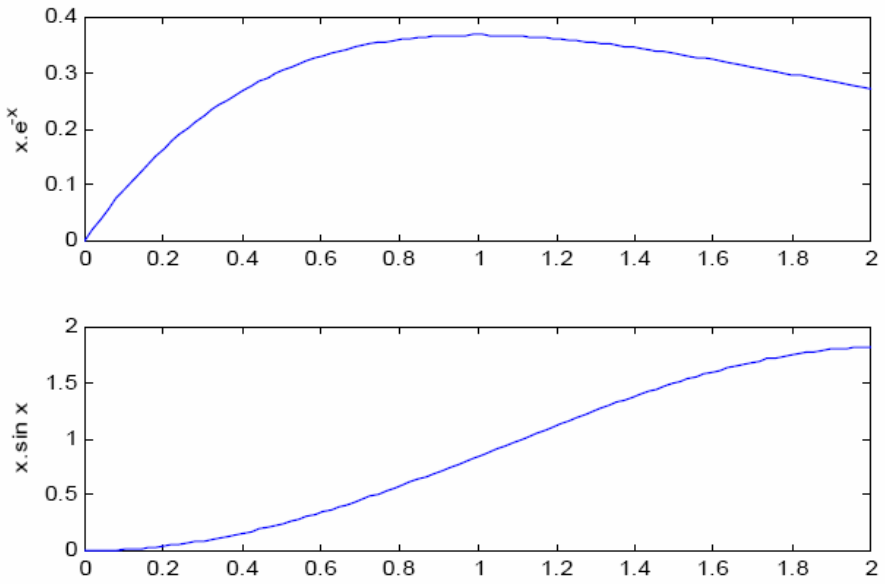
و در صورت لزوم نام توابع را نیز در همان نمودار نشان دهید.

```
» legend('x.e^{-x}','x.sin x')
```



می توانید بیش از یک نمودار را در یک پنجره نشان دهید:

```
» subplot(2,1,1), plot(x,y)
» ylabel('x.e^{-x}')
» subplot(2,1,2), plot(x,x.*sin(x))
» ylabel('x.sin x')
```



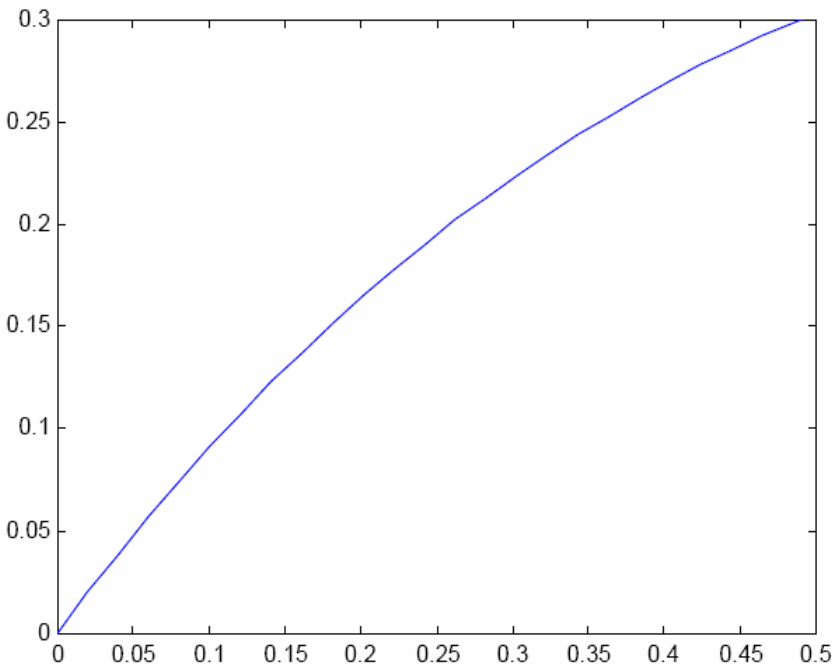
دو عدد اول در دستور subplot تعداد تقسیمات صفحه را معین می کنند (سطری و ستونی) و عـدد سوم مکان رسم نمودار (یا تغییر روی نمودار موجود) را مشخص می نماید.

نمودار را می توانید با استفاده از دستور clf پاک کنید.

```
» clf
```

با استفاده از دستور figure می توانید پنجره جدیدی برای رسـم نمـودار بـاز نمـائید. دسـتور axis حدود بالا وپایین محورهای مختصات را به صورت یک بردار ارائه می نماید.

```
» figure(2)
» plot(x,y)
» axis
ans =
      0    2.0000         0    0.4000
```



در تمامی مثالهای بالا مقادیر متغیر مستقل و متغیر وابسته به صورت دو بردار بر حسب هم رسم شده اند. در صورتی که تابعیت متغیر وابسته بر حسب متغیر مستقل مشخص باشد می توانید از دستور fplot برای رسم آن استفاده کنید:

```
» fplot('x*exp(-x)',[0  2])
```

آرگومان اول این دستور یک بردار حرفی است که مشخص کننده رابطه تابع (در صورت ساده بودن رابطه تحلیلی تابع، همانند مثال فوق) یا نام m-file حاوی تابع (که جداگانه باید ایجاد شده باشد) است. آرگومان دوم fplot یک بردار دو عضوی است که حد پائین و بالای متغیر مستقل را مشخص می کند.

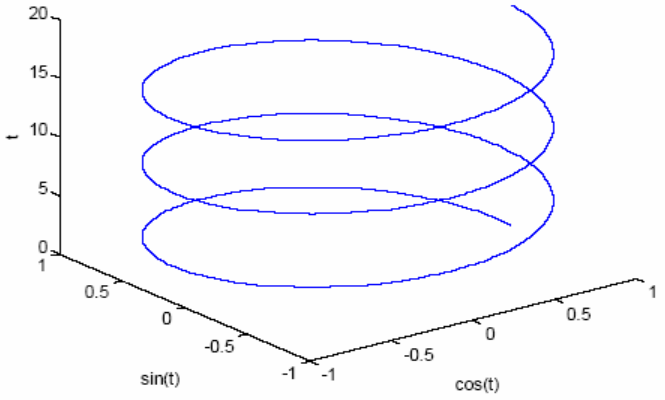
تعدادی از دستورهای ترسیم دو بعدی در زیر آورده شده اند:

نمودار نیمه لگاریتمی (محور x لگاریتمی)	semilogx(x,y)
نمودار نیمه لگاریتمی (محور y لگاریتمی)	semilogy(x,y)
نمودار تمام لگاریتمی	loglog(x,y)
رسم در دستگاه مختصات قطبی	polar(r,theta)
نمودار میله ای	bar(x,y)
نمودار مساحت	area(x,y)

نمودارهای ۳ بعدی

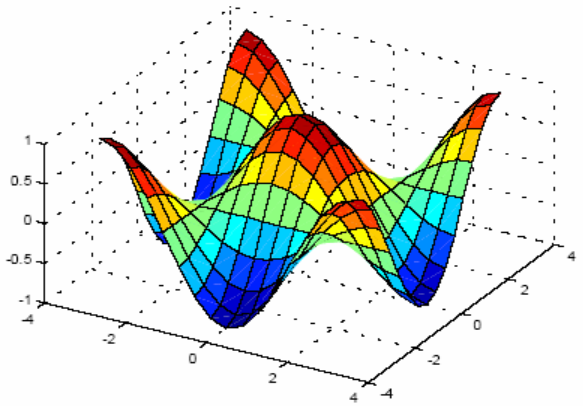
دستورهای زیادی در MATLAB برای ترسیم نمودارهای سه بعدی وجود دارند. یک منحنی سه بعدی را می توانید به کمک دستور plot3 ببینید:

```
» t=0:.01:6*pi;
» plot3(cos(t),sin(t),t)
» xlabel('cos(t)')
» ylabel('sin(t)')
» zlabel('t')
```



سطوح سه بعدی را می توانید با استفاده از دستور surf ترسیم کنید:

```
» [x,y]=meshgrid(-pi:pi/8:pi,-pi:pi/8:pi);
» z=cos(x).*cos(y);
» surf(x,y,z)
» view(30,45)
```



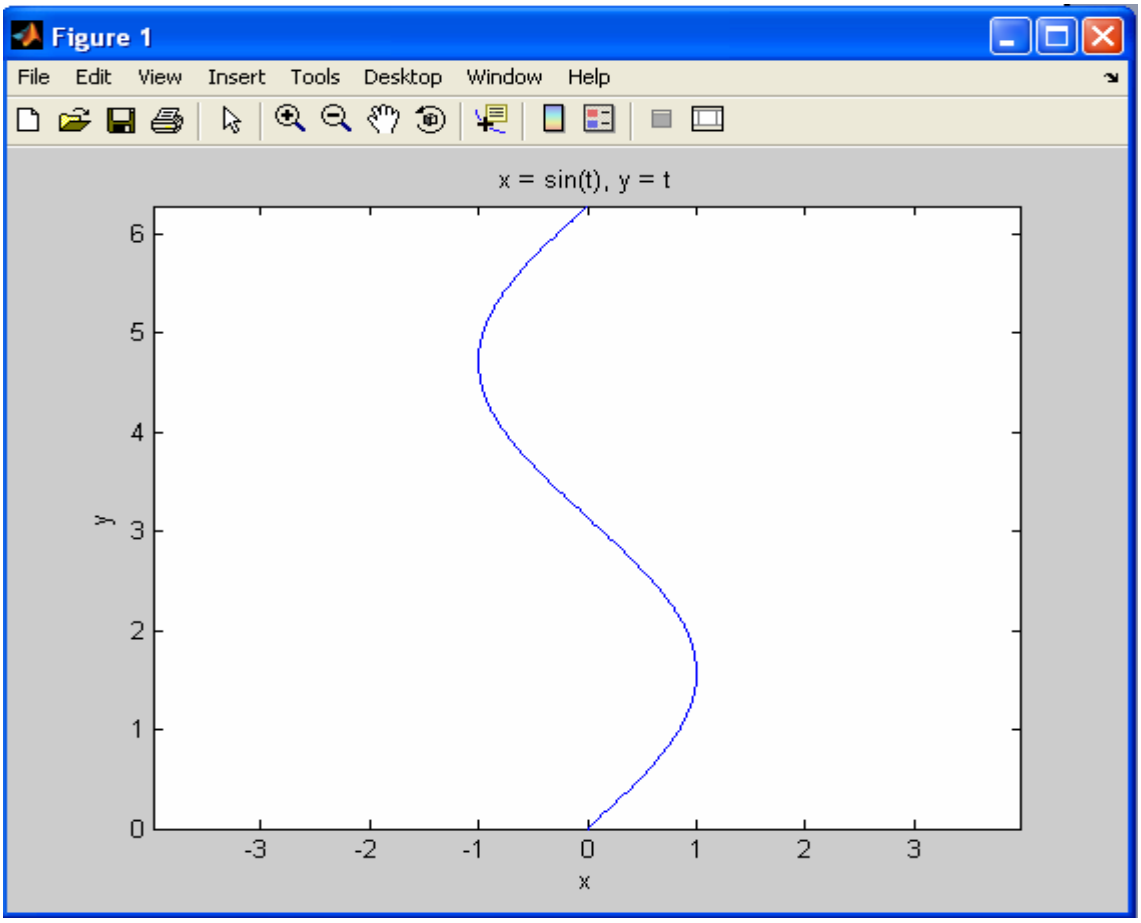
دستور meshgrid شبکه دو بعدی روی صفحه xy را ایجاد می کند. بردارهای ورودی به این دستور مشخص کننده تقسیمات در جهات x و y هستند. سطح ایجاد شده را می توانید با کمک دستور shading هموار کنید. همچنین برای تطابق رنگها با اعداد محور z می توانید از دستور colorbar استفاده کنید.

```
» shading interp
» colorbar
```

برای رسم سطوح سه بعدی از دستورات دیگری مانند mesh، meshc، meshz و waterfall نیز می توانید کمک بگیرید.

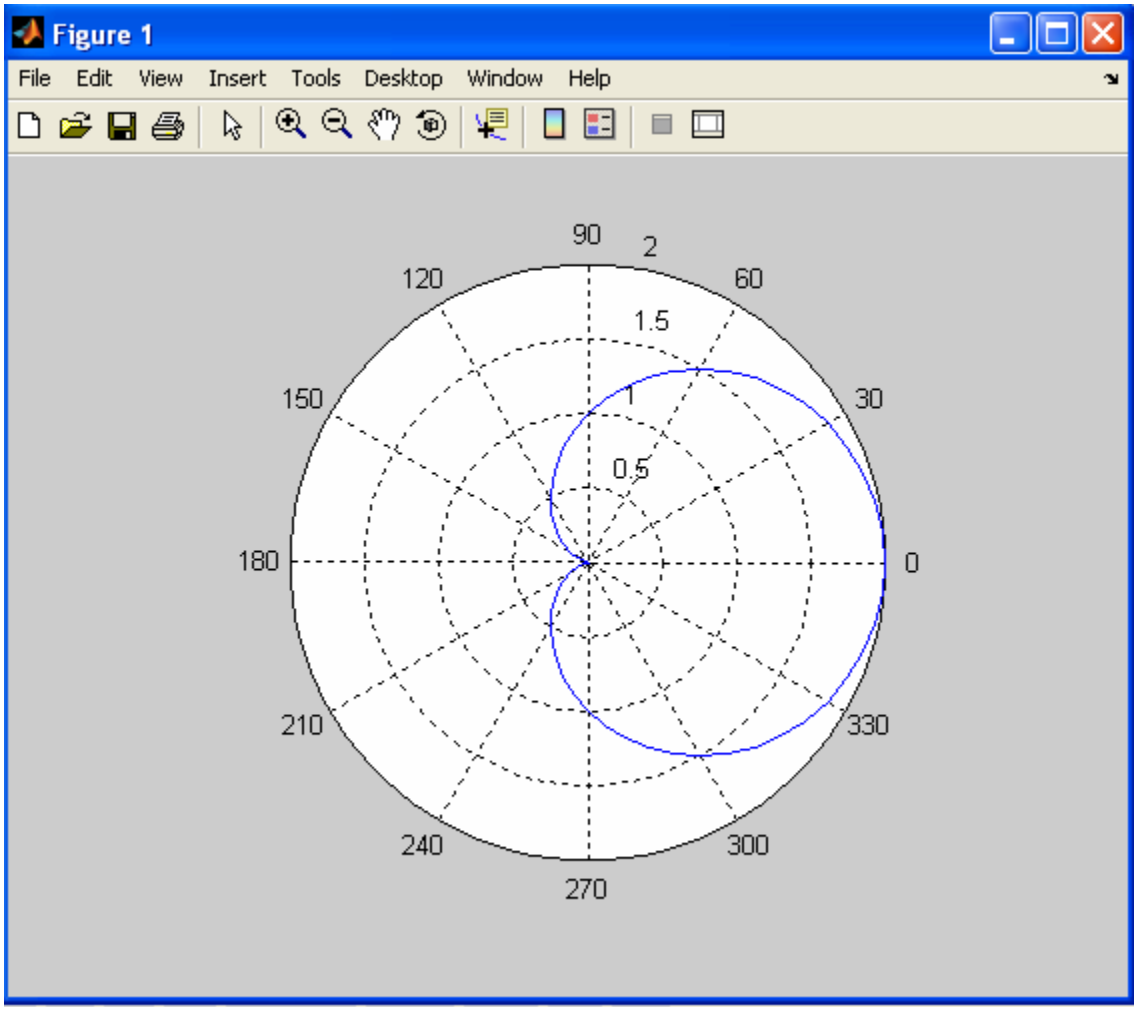
رسم نمودارها با معادلات پارامتری

```
>>ezplot('sin(t)','t',[0,2*pi])
```



رسم نمودارها با معادلات قطبی

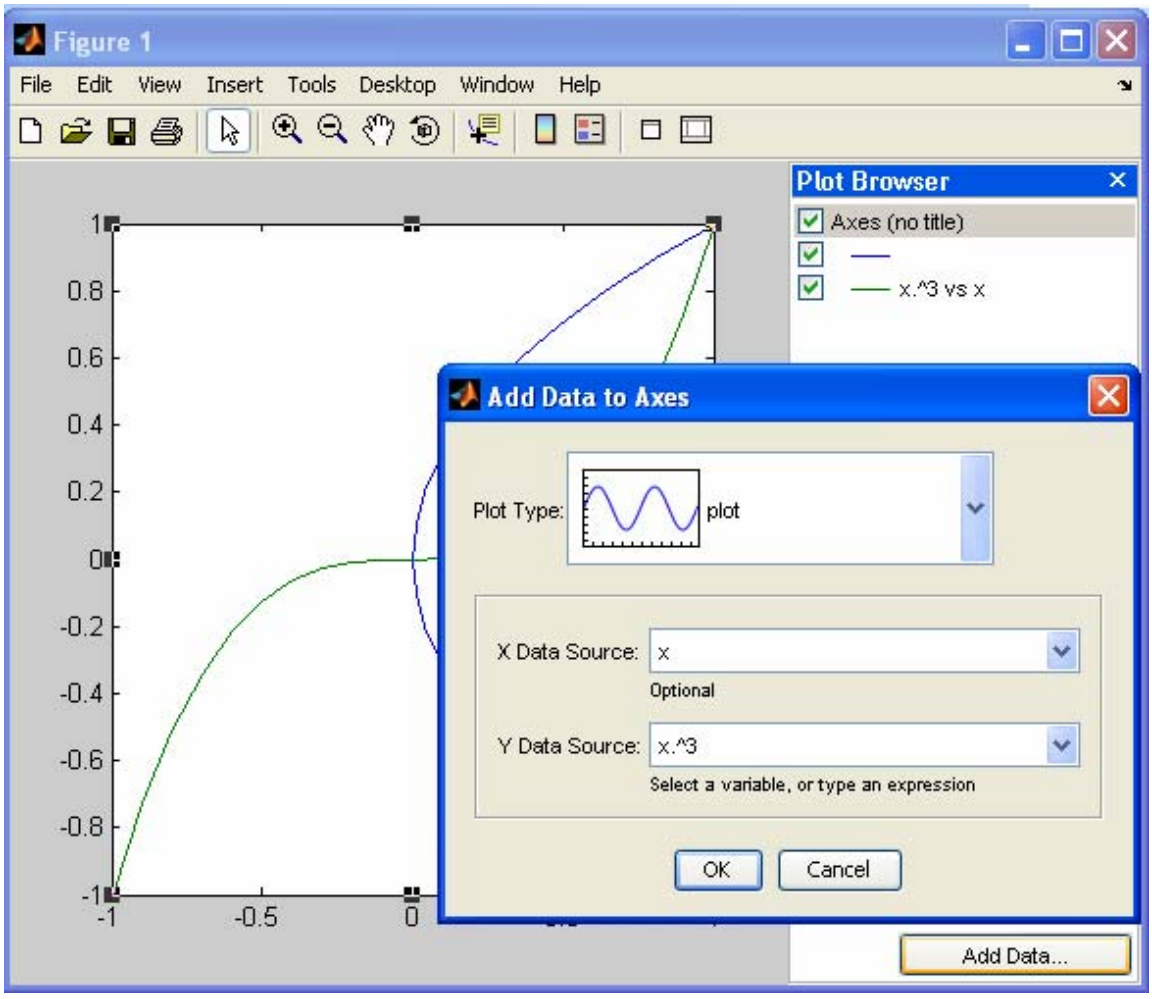
```
 >>t=0:pi/30:2*pi;
>>polar(t,1+cos(t))
          یا
>> ezpolar('1+cos(t)')
```



اضافه کردن یک گراف به گراف قبلی

ابتدا یک گراف رسم کرده و در figure ایجاد شده plot tools را انتخاب می کنیم وaxes را انتخاب می کنیم و در add data تابع جدید را وارد می کنیم.

```
>> x=-1:.1:1;
>> plot(x.^2,x)
```



اگر متغیر مربوط به data graph را تغییر دهیم از طریق گزینه refresh data اطلاعات مربوط به گراف جدید نمایش داده می شود.

با کمک  در plot tools میتوان از گراف مربوط خروجی گرفت.

برنامه نویسی (m-files)

مجموعه ای از دستورات MATLAB را می توانید در یک پرونده ذخیره کنید و سپس آنها را یکجا اجرا نمائید. چنین پرونده ای برای آنکه در محیط MATLAB قابل اجرا باشد باید حتماً دارای

دنباله ".m" باشد. در صورتی که از ویرایشگر MATLAB (MATLAB Editor) استفاده کنید، دنباله ".m" بطور خودکار در هنگام ذخیره پرونده به نام آن افزوده می گردد. در صورت استفاده از ویرایشگر دیگری بغیر از ویرایشگر MATLAB (نظیر Notepad) اطمینان حاصل کنید که پرونده حتماً به روش ascii و با دنباله ".m" ذخیره گردد.

در این بخش از یادداشت فقط بر نحوه برنامه نویسی و اجرای برنامه ها تاکید شده است و نتایج اجرای برنامه های مورد بحث نشان داده نشده اند. به خواننده توصیه می گردد که خود برنامه ها را اجرا کرده و نتایج آنها را مشاهده نماید.

برنامه اصلی

m-file ها می توانند به دو شکل برنامه اصلی و تابع باشند. برنامه اصلی عبارتست از مجموعه ای از دستورها که می توان آنها را بطور جداگانه در محیط کار MATLAB اجرا نمود. هنگامی که نام برنامه اصلی را در محیط کار MATLAB بنویسید این دستورها به ترتیب اجرا می گردند. به عنوان مثال برای محاسبه حجم گاز کامل، در دماهای مختلف و فشار معلوم، دستورات زیر را در ویرایشگر MATLAB بنویسید و سپس تحت عنوان pvt.m ذخیره کنید:

```
% A sample scritp file: pvt.m
disp(' Calculating the volume of an ideal gas.')
R = 8314;    % Gas constant (J/kmol.K)
t = ...
   input(' Vector of temperature (K) = ');
p = input(' Pressure (bar) = ')*1e5;
v = R*t/p;      % Ideal gas law
% Plotting the results
plot(t,v)
xlabel('T (K)')
ylabel('V (m^3/kmol)')
title('Ideal gas volume vs temperature')
```

علامت ٪ نشانگر وجود توضیحات در برنامه است. علامت ٪ و آنچه بدنبال آن در همان سطر می آید به هنگام اجرای برنامه نادیده گرفته می شود. همچنین علامت . . . بیانگر آن است که دستور مورد نظر در این سطر تمام نشده و در سطر بعدی ادامه می یابد. مورد استفاده این علامت بیشتر در مورد دستورهای محاسباتی طولانی است که برای مطالعه راحت تر این قسمت از برنامه بهتر است در دو یا سه خط نوشته شود.

پس از ایجاد پرونده pvt.m، برای اجرای آن کافی است که نام آن را در محیط کار MATLAB بنویسید و نتایج را مشاهده کنید (نمودار در زیر نشان داده نشده است).

```
» pvt
 Calculating the volume of an ideal gas.
 Vector of temperature (K) = 100:25:300
 Pressure (bar) = 10
```

MATLAB دارای چندین ترکیب کنترل جریان محاسبات است که به برنامه امکان می دهد که در حین اجرا تصمیمات لازم را اتخاذ کرده و ترتیب اجرای دستورات را کنترل کند. ایـن دسـتورها در زیر شرح داده می شوند.

if . . . (else . . .) end – دستور if برنامه را قادر می سازد که تصمیم بگیرد که چه دستورهایی باید اجرا گردند. مثال:

```
x = input(' x = ');
if x >= 0
   y=x^2
end
```

عبارتی که به دنبال کلمه if می آید باید یک عبارت منطقی باشـد. در صـورت درسـت بـودن ایـن عبارت منطقی، دستورهایی که در سطرهای بین if و end قرار دارند بترتیب اجـرا مـی گردنـد و در صورت نادرست بودن این عبارت منطقی، دستورهای گفته شده نادیده گرفته می شوند.

شما همچنین می توانید از دستور else استفاده کنید. مثال:

```
x = input(' x = ');
if x >= 0
   y=x^2
else
   y=-x^2
end
```

در این حـالت اگر عبارت منطقی مورد نظر درست باشد، مجموعه دسـتورهای بیـن if و else اجـرا می گردند و در غیر این صورت دستورهای بین else و end قابل اجرا می باشند.

for . . . end – دستور for به برنامه اجازه می دهد که دستورهای درج شـده بیـن for و end را بـه دفعات معینی تکرار نماید. مثال:

```
k = 0;
for x = 0:0.2:1
   k = k + 1
   y = exp(-x)
end
```

while . . . end – در مواردی که لازم باشد که در حین اجرای برنامه مجموعه ای از دستورات تکرار گردند ولی تعداد دفعات تکرار معلوم نباشد بلکه این عملیات تا ارضا شدن شرط یـا شـروط معینـی ادامه یابند، می توان از دستور while استفاده نمود. مثال:

```
x = 0;
while x < 1
   y = sin(x)
   x = x + 0.1;
end
```

همانند آنچه در مورد دستور if گفته شد، عبارتی که به دنبال کلمه while می آید باید یک عبـارت منطقی باشد که در واقع همان شرط مورد نظر است. در صورت صادق بودن ایـن عبـارت منطقـی، دستورهایی که در سطرهای بین while و end قرار دارند بترتیب اجرا می گردند تا آنجایی که شرط مورد نظر دیگر برقرار نباشد.

switch . . . case . . . (otherwise . . .) end – وقتی که لازم باشـد کـه برنامـه بـر حسـب مقـادیر مختلف یک متغیـر، متناظراً دستورهای متفـاوتی را اجـرا کنـد، بکـار بـردن ترکیـب switch-case راحت تر از بکار بردن چندین دستور if متداخل است. مثال:

```
a = input('a =');
switch a
case 1
   disp('One')
case 2
   disp('Two')
case 3
   disp('Three')
end
```

break و pause - دو دستور مفید دیگر که در برنامه نویسی می توانند مـورد اسـتفاده قـرار گـیرند عبارتند از break و pause. شما می توانید در صورت لزوم قبل از کامل شدن حلقه به کمک دستور break از آن خارج شوید. هنگامی که برنامه در حین اجرا به دستور pause برسد متوقف مـی مـاند تا اینکه شما کلیدی را روی صفحه کلید فشار دهید و سپس اجرای برنامه از دسـتور بعـد از pause ادامه می یابد. مثال:

```
k = 0;
for x = 0:0.2:1
   if k > 5
      disp('k > 5')
      break
   end
   k = k + 1;
   y = exp(-x);
   disp([' k = ',num2str(k),'  y = ',num2str(y)])
   pause
end
```

در مثال فوق، برنامه هر بار پس از نشان دادن مقادیر k و y متوقف می ماند تـا اینکـه کلیـدی روی صفحه کلید فشرده شود. سپس حلقه for بار دیگر تکرار می گردد و این عمل آنقدر ادامه می یـابد تا اینکه مقدار k از ۵ بیشتر شود. در این موقع دستور break باعث خروج برنامه از حلقـه for (و در این مثال پایان اجرای برنامه) می شود.

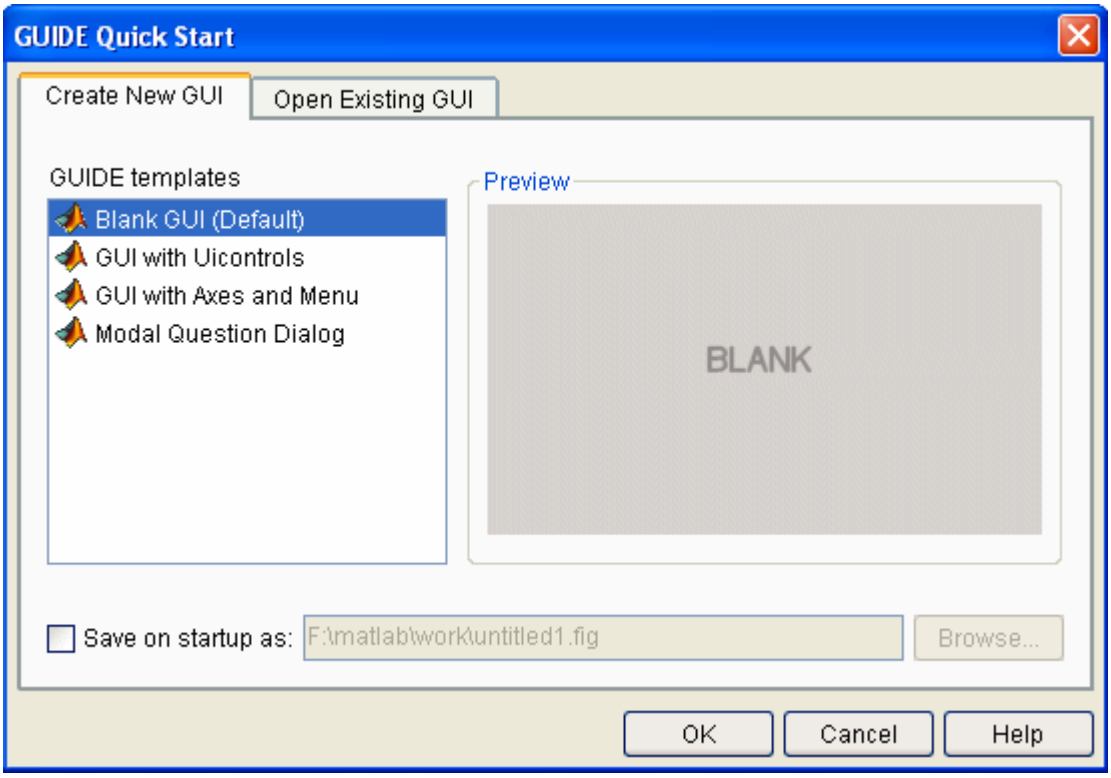
ساخت سند گرافیکی(GUI)

GUIDE چیست؟

GUIDE محیط توسعه یافته رابط گرافیکی است و مجموعه ابزارهای لازم برای ایجاد رابط گرافیکی کاربر تامین می کند. GUIDE اطور خود کار Mفایلی تولید می کند که از طریق آن می توان دستورات لازم را به آن داد.

راه اندازی GUIDE

با تایپ guide در command window میتوان GUIDE را راه اندازی کرد.این کار منجر به نمایش پنجره GUIDE Quick start مطابق شکل می شود:

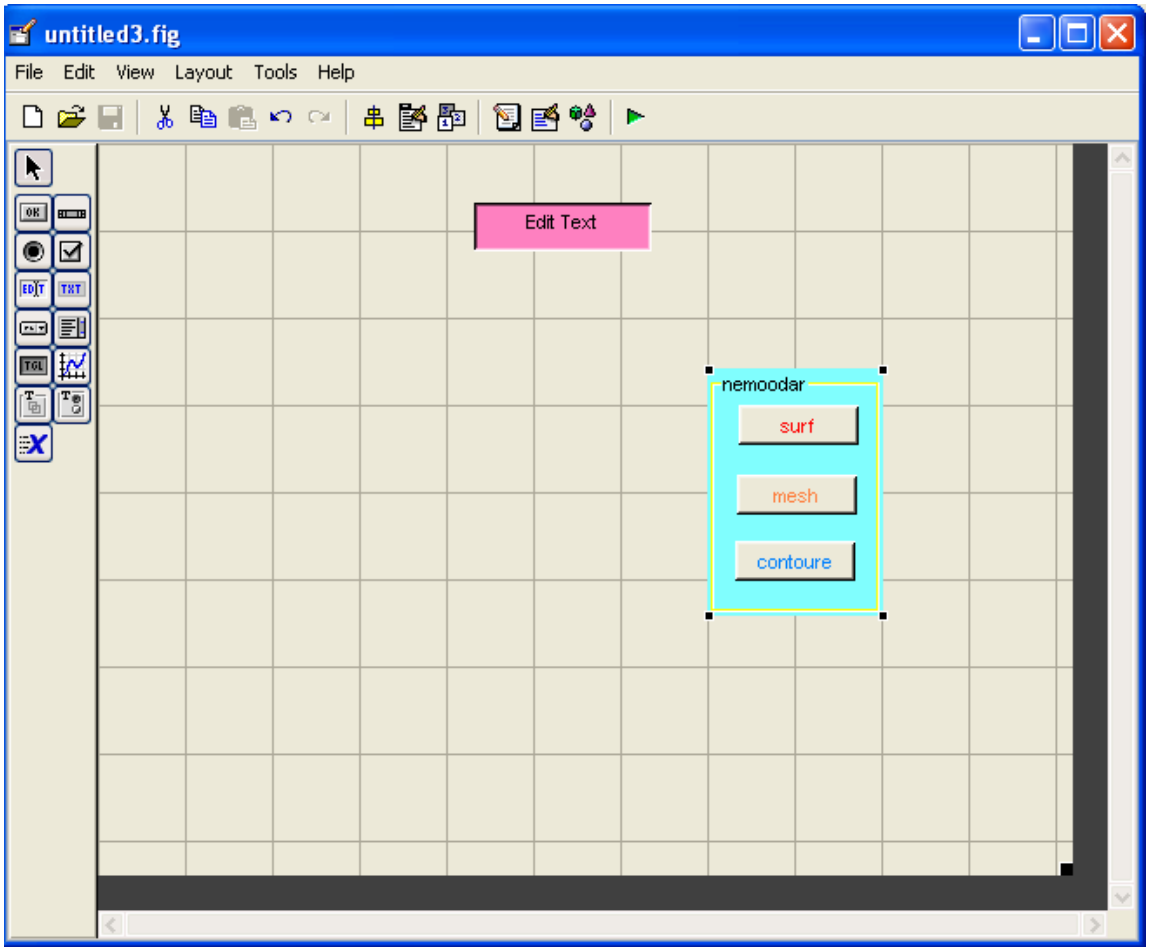


در این پنجره می توان یک GUI از یک GUIDE موقت ایجاد کرد یا یک GUI را باز کرد.با انتخاب Blank GUI یک GUI در GUIDE باز می شود.می توان با کشیدن وانداختن هرکدام از پنل ها آنها را روی صفحه ایجاد کرد.

برنامه نویسی یک GUI

باید کلیه دستورات مربوطه را در Mفایل مربوطه نوشت.بطور مثال:

شماتیک زیررا ایجاد کرده



ودر m-file editor در قسمت های مربوط به هر

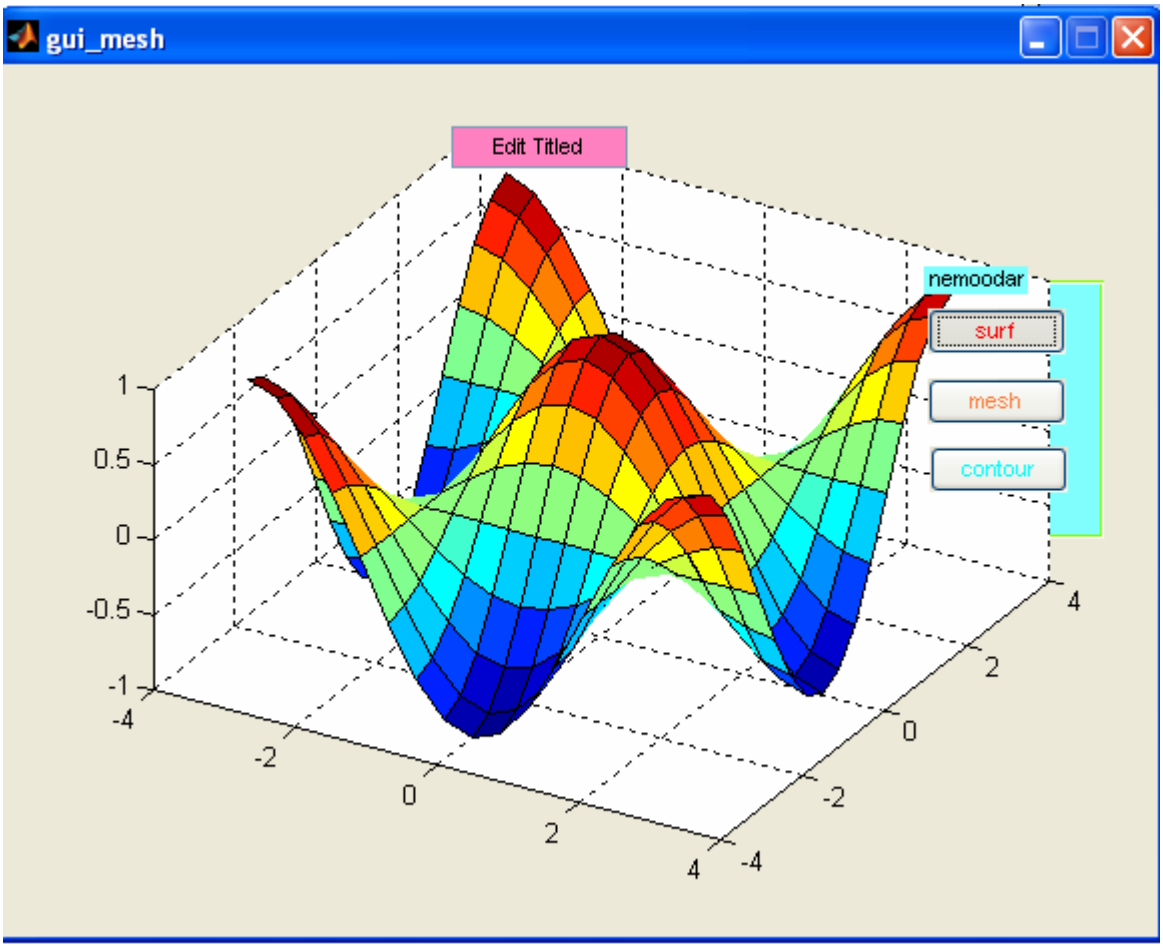
function کد های زیر را ایجاد کنید(بقیه کد ها را پاک نکنید)

```
function pushbutton1_Callback(hObject, eventdata, handles)

[x,y]=meshgrid(-pi:pi/8:pi,-pi:pi/8:pi);
 z=cos(x).*cos(y);
 surf(x,y,z)
 view(30,45)
function pushbutton2_Callback(hObject, eventdata, handles)

[x,y]=meshgrid(-pi:pi/8:pi,-pi:pi/8:pi);
 z=cos(x).*cos(y);
mesh(x,y,z)
 view(30,45)
function pushbutton3_Callback(hObject, eventdata, handles)
```

```
[x,y]=meshgrid(-pi:pi/8:pi,-pi:pi/8:pi);
 z=cos(x).*cos(y);
contour(x,y,z)
 view(30,45)
```

علامت  را برای اجرا کلیک کنید ;

روی mesh یا surf یا contour کلیک کنید:



منابع:

راهنمای استفاده از matlab نوید مستوفی،خود آموز matlab 7علی فکور یکتا
help matlab،mabna soft و...

نظرات و پیشنهادات خود را به آدرس زیرارسال کنید:
Nazari.sadegh@gmail.com