بسم الله الرحمن الرحیم

دوره آموزشی نرم افزار MATLAB (مبانی)

مدرس: گلناز جوزانی کهن

دکترای مهندسی اکتشاف معدن

MATLAB®

پاییز ۱۳۹۴

فهرست موضوعات مورد بحث

آشنایی با بخشهای مختلف نرم افزار، انواع متغیرها، ثوابت، توابع، علائم، فرمان‌ها، ساختار تکرار، آرایه ها، ماتریس‌ها و عملگرهای منطقی و ریاضی بر روی آنها

آشنایی با نحوه ترسیم نمودار در نرم افزار به صورت دو بعدی و سه بعدی، آشنایی با دستورات اولیه، تغییر رنگ نمودار، انداختن دو نمودار روی هم و ...

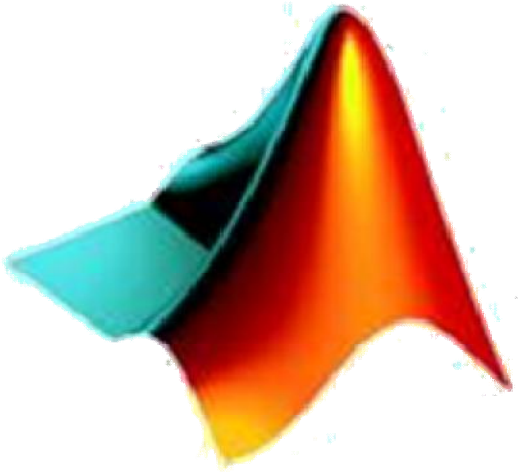
آشنایی با برنامه نویسی (mfile)در محیط MATLAB -توابع پایه و اشکال زدایی از برنامه‌ها

آشنایی با حلقه‌ها درMATLAB شامل: حلقه While و For، دستورات Continue وBreak، حلقه‌های تودرتو و آرایه‌های منطقی

معرفی تولباکس شبکه عصبی نرم افزار MATLAB و حل چند مثال عملی

آشنایی با برنامه نویسی در محیط

MATLAB



MATLAB

m فایل چیست؟

- به جای نوشتن دستورات در پنجره command و اجرا شدن به صورت تک تک، مجموعه ای از دستورات را می توان در یک فایل قرار داد تا MATLAB با هر بار اجرای آن، دستورات موجود در این فایل را اجرا نماید. به این فایل script یا m file گفته می شود.
- در حقیقت m file یک فایل ASCII شبیه به فایل های سورس کد در زبان های برنامه نویسی C یا فرترن است.
- نحوه نوشتن m file: مجموعه دستورات مورد نظر را در یک ویرایشگر می نویسیم. فایل را با پسوند .m ذخیره می کنیم.
- در محیط MATLAB یک ویرایشگر برای انجام این کار دارد که با انتخاب new script اجرا می شود و می توان دستورات را در آن تایپ نمود.
- بعد از نوشتن m file با انتخاب گزینه run، برنامه اجرا می شود. روش دیگر اجرای m file تایپ نام آن بدون پسوند (.m) در پنجره دستورات یعنی command window است.

دو فرمان مهم در برنامه نویسی

- دستور input برای گرفتن ورودی از کاربر به کار می رود.
- تابع disp برای چاپ پیغام یا مقدار متغیر به کار می رود.
- با استفاده از علامت , می توان چند دستور را با هم در پنجره command اجرا کرد.

```
>> x=input ('Please enter the value of velocity:')
Please enter the value of velocity:89

x =

    89

>> disp ('This is the end of program.')
This is the end of program.
>> disp (x)
    89
>> disp ('The value of speed is:'), disp (x)
The value of speed is:
    89
fx >>
```

نکته: اضافه نمودن '\n' در دستور fprintf سبب ایجاد یک خط فاصله در خروجی می شود.

(تکرار دستور برای ایجاد فواصل بیشتر).

- برای نشان دادن خروجی عددی در یک خط از فرمان همراه با متن، ابتدا با تابع num2str باید متغیر عددی را به رشته کاراکتری تبدیل کرد.

انواع نحوه نمایش خروجی

- دو متغیر به دلخواه تعریف کنید و آن ها را با آنچه تاکنون آموخته اید، به صورت های گوناگون نمایش دهید.

```
>> x= 21;
>> y= 34;
>> disp (x), disp (y)
    21

    34

>> disp (x), fprintf('\n'), disp (y)
    21



    34

>> disp (['The X is ', num2str(x), '. The Y is ', num2str(y), '.'])
The X is 21. The Y is 34.
fx >>
```

یک مثال ساده

- برنامه ای بنویسید که ماتریسی با اندازه دلخواه را از کاربر بگیرد و مقادیر آن را نمایش دهد.

New/Script/**نوشتن برنامه**/Save/Run

matrixdisp.m

```
1       % This is a very simple program which displays a matrix that user has
2       % entered.
3 -     x= input('Please enter your matrix:');
4 -     disp ('The value of your matrix is:'), disp (x)
```

```
>> matrixdisp
Please enter your matrix:[8 9 25; 4 -19 78; 7 19 42]
The value of your matrix is:
     8     9    25
     4   -19    78
     7    19    42

>>
```

نکته: هر آنچه که بعد از %

نوشته می شود، کامپایل نمی شود!

(توضیحات تکمیلی مربوط به هر خط از برنامه معمولاً پس

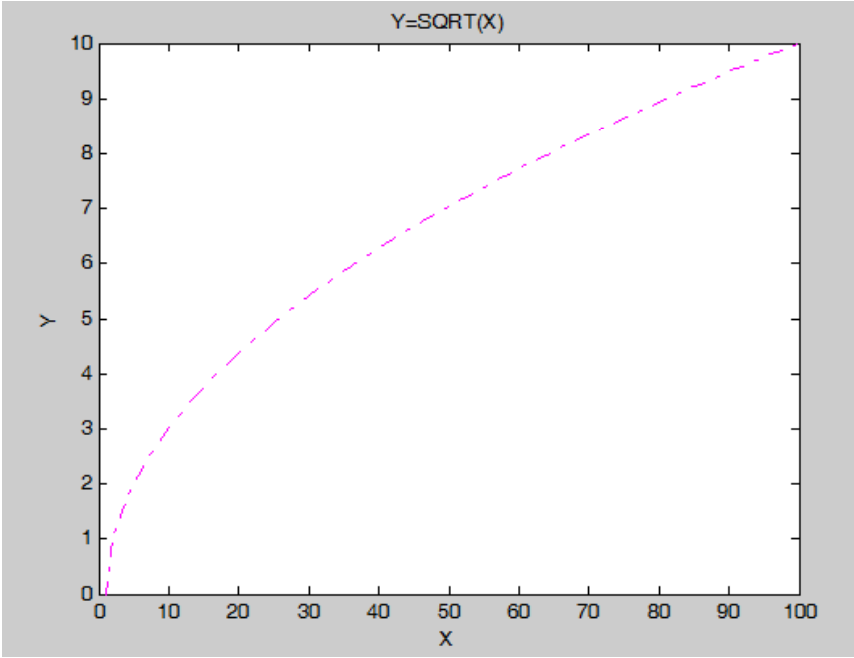
از % نوشته می شود.)

مثال (برنامه های ساده)

- برنامه ای بنویسید که جذر اعداد صفر تا صد را نمایش دهد.

squaredroot.m

```
1       % This program plots the squared root of numbers between 10 and 100.
2 -     k=linspace(0,100);
3 -     plot (sqrt(k), '-. m'), title ('Y=SQRT(X)'), xlabel ('X'), ylabel ('Y')
```



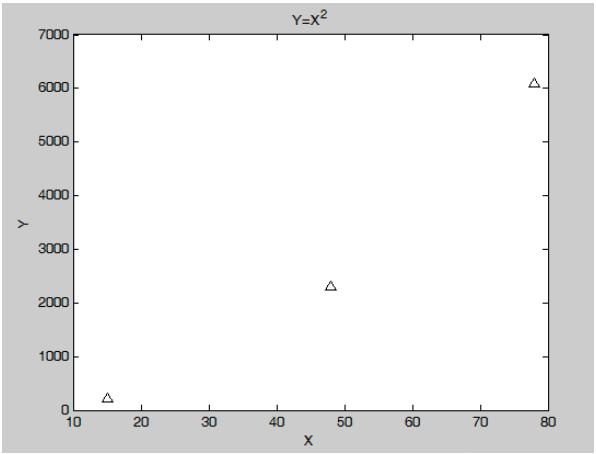
مثال (برنامه های ساده)

- برنامه ای بنویسید که برداری را از کاربر دریافت کند و مولفه های بردار را به توان دو برساند و نمودار Y برحسب X را رسم کند.

plotssquared.m

```
1      % This scrip plots the squared value of a given vector.
2 -    x=input('Please enter your vector:');
3 -    y=x.^2;
4 -    plot (x,y, '^ k'), title ('Y=X^2'), xlabel ('X'), ylabel ('Y')
```

```
>> plotssquared
Please enter your vector:[15 48 78]
>>
```



مزایای تابع نویسی در MATLAB

مزایای استفاده از توابع به جای فایلهای اسکریپت

- **سرعت بالاتر**
- **صرفه‌جویی در حافظه کامپیوتر**
- **توسعه توانایی‌های متلب**

توابع بر خلاف فایلهای اسکریپت در هنگام اجرا یکبار کامپایل شده و اجرا می‌شوند. در حالیکه فایلهای اسکریپت سطر به سطر کامپایل و اجرا می‌گردند. این امر باعث افزایش سرعت اجرای توابع در مقایسه با فایلهای اسکریپت می‌شود.

متغیرهای تعریف شده در توابع پس از پایان اجرای آن از حافظه پاک می‌شوند و بطور کلی فضای کاری توابع مستقل از فضای کاری متلب است. خصوصا در مواقعی که برنامه با ماتریسهای بزرگ(مانند تصاویر) کار می‌کند بهتر است از توابع استفاده شود.

تابع نویسی در MATLAB

- در MATLAB مشابه زبان های برنامه نویسی می توان برای برنامه تابع نوشت .
- یک m file به صورت زیر ایجاد می کنیم (کلماتی که با رنگ قرمز مشخص شده اند باید عینا نوشته شوند):

```
function [outputs]=name(inputs)
> یک سری دستورات <
end
```

- بعد از نوشتن m file آن را با نام تابع و پسوند m. ذخیره می کنیم.
- سپس می توان در پنجره command، تابع را اجرا کرد.
- اگر تابعی یک خروجی دارد به کروشه در فرمان فوق نیازی نیست.

مثال (تعریف تابع)

- تابعی بنویسید که طول وتر یک مثلث قائم الزاویه را محاسبه می کند.

فراخوانی تابع در پنجره دستور/Save/نوشتن تابع/New/Function

Editor - D:\MATLAB\hypoten.m

hypoten.m

```
1   function [l] = hypoten(a,b)
2   %This function calculates the hypotenuse of a right-angled triangle.
3   %  The input arguments (i.e., a, b) are the sides of the triangle.
4   [l]=sqrt(a.^2+b.^2);
5   end
```

```
>> hypoten (3,4)

ans =

     5

>> help hypoten
 This function calculates the hypotenuse of a right-angled triangle.
   The input arguments (i.e., a, b) are the sides of the triangle.
```

مثال استفاده از تابع قبل در برنامه

- برنامه ای بنویسید که طول اضلاع یک مثلث قائم الزاویه را از کاربر می گیرد و طول وتر آن را با استفاده از تابع تعریف شده در اسلاید قبل، محاسبه می کند.

hyps.m

```
1 -    a=input('Please enter the length of first side in your triangle:');
2 -    b=input ('Please enter the length of second side in your triangle:');
3 -    y=hypoten (a,b);
4 -    disp (['The hypotenuse of your triangle is: ', num2str(y)])
```

```
>> hyps
Please enter the length of first side in your triangle:80
Please enter the length of second side in your triangle:100
The hypotenuse of your triangle is: 128.0625
>>
```

مثال تابع نویسی

- تابعی بنویسید که مسافت طی شده، سرعت و شتاب جسم در حال سقوط آزاد را در هر زمان دلخواه محاسبه کند.

free_fall.m

```
1   function [dist,vel,acc] = free_fall( t )
2   %free_fall returns values for distance, velocity, and accelerating g
3   g=9.8;
4   dist=0.5*g.*(t.^2);
5   vel= g.*t;
6   acc=g;
7   end
```

اجرای تابع در پنجره دستور:

```
>> [x,y,z]=free_fall (0.5);
>> disp (['The Distance is ', num2str(x)]), fprintf ('\n'),disp (['The velocity is ', num2str(y)]), fprintf ('\n'), disp (['The acceleration is ', num2str(z)])
The Distance is 1.225

The velocity is 4.9

The acceleration is 9.8
>>
```

مثال استفاده از توابع در برنامه نویسی

- برنامه ای بنویسید که با استفاده از دو تابع از پیش نوشته شده، درجه را به رادیان و بالعکس تبدیل نماید.

DR.m

```
function y = DR(x)
%This function converts the degree to radian.
y=x*pi/180;
end
```

RD.m

```
function y = RD(x)
%This function converts the radian to degree.
y=x*180/pi;
end
```

degconvert.m

```
%This program convert degree to radian and vice versa.
a=input('Please enter the value in degree format that you want to be converted to radian:');
b=DR (a);
disp (['The ', num2str(a),' degree in radian is: ',num2str(b)])
c=input('Please enter the value in radian format that you want to be converted to degree:');
d=RD (c);
disp (['The ', num2str(c),' radian in degree is: ',num2str(d),])
```

```
>> degconvert
Please enter the value in degree format that you want to be converted to radian:60
The 60 degree in radian is: 1.0472
Please enter the value in radian format that you want to be converted to degree:1.0472
The 1.0472 radian in degree is: 60.0001
>>
```

مثال برنامه نویسی

- برنامه ای بنویسید که مختصات کارتزین را از کاربر دریافت می کند و به مختصات قطبی تبدیل نماید.

cart2plr.m

```
function [r,theta] = cart2plr(x,y)
    %   cart2plr  Convert Cartesian coordinates to polar coordinates
    %
    %   [r,theta] = cart2plr(x,y) computes r and theta with
    %
    %       r = sqrt(x^2 + y^2);
    %       theta = atan2(y,x);

    r = sqrt(x^2 + y^2);
    theta = atan2(y,x);
end
```

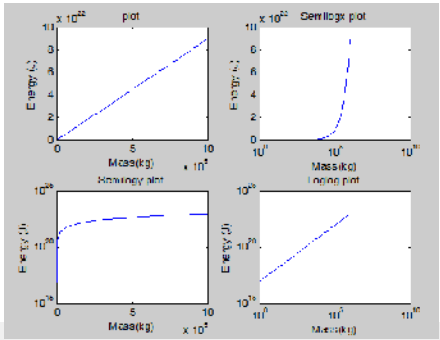
cart2plr_exercise.m

```
% This program converts a given Cartesian coordinates to polar coordinates.
m=input('Please enter the X in Cartesian coordinates: ');
n=input('Please enter the Y in Cartesian coordinates: ');
[k,l]=cart2plr(m,n);
disp (['The r is: ' num2str(k), ' The theta is: ' num2str(l)])
```

```
>> cart2plr_exercise
Please enter the X in Cartesian coordinates: 40
Please enter the Y in Cartesian coordinates: 80
The r is: 89.4427 The theta is: 1.1071
fx >>
```

مثال استفاده از توابع در برنامه نویسی

- برنامه ای بنویسید که انرژی محاسبه شده از قانون انیشتین مربوط به اجرام از ۱ تا $۱۰^{۶}$ کیلوگرم را با چهار پلات (محورها عادی، محور x لگاریتمی، محور y لگاریتمی و هر دو محور لگاریتمی) رسم کند.



energy.m

```
function [en] = energy(mass)
%energy defines the Einestain law
c=2.9979e8;
en=mass.*(c^2);
end
```

einestain_law.m

```
%This programs gives 4 plots (Normal, semilogx, semilogy, and loglog) for
%energy which is calculated from Einestan law
mass=logspace(0,6);
en=energy(mass);
subplot(2,2,1); plot (mass, en); title('plot'); xlabel('Mass(kg)'); ylabel('Energy (J)');
subplot(2,2,2); semilogx(mass, en); title('Semilogx plot'); xlabel('Mass(kg)'); ylabel('Energy (J)');
subplot(2,2,3); semilogy(mass, en); title('Semilogy plot'); xlabel('Mass(kg)'); ylabel('Energy (J)');
subplot(2,2,4); loglog(mass, en); title('Loglog plot'); xlabel('Mass(kg)'); ylabel('Energy (J)');
```

چند نکته

- با تایپ دستور what می توان تمام m file های موجود در دایرکتوری حاضر را دید.
- با تایپ دستور help قبل از نام توابعی که خودمان تعریف کرده ایم، می توان توضیحاتی را که در زمان تعریف تابع نوشته ایم را ببینیم.
- اگر قبل از یک مجموعه دستورات، tic و پس از پایان آن ها toc را بنویسیم، علاوه بر اجرای دستورات، زمان کلی اجرای دستورات را نیز نمایش می دهد.
- توابع با run اجرا نمی شوند، فقط m file ها با run اجرا می شوند.

اشکال زدایی از برنامه ها

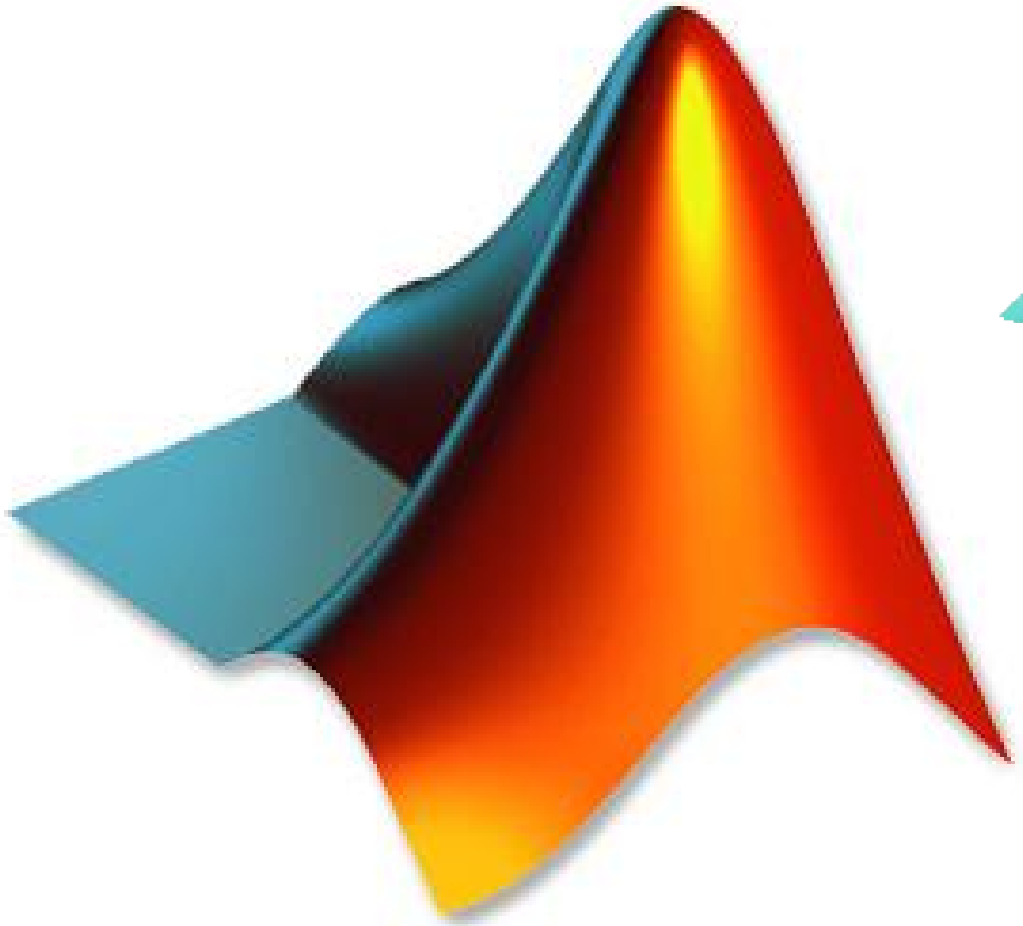
- با کلیک روی پیغام خطا، خط مشکل ساز برنامه مشخص می شود. با کمی دقت، استفاده از فرمان help می توان به راحتی خطا را برطرف نمود و دوباره برنامه را اجرا کرد. مثلاً در برنامه زیر یک کاما بعد از y جا افتاده است.

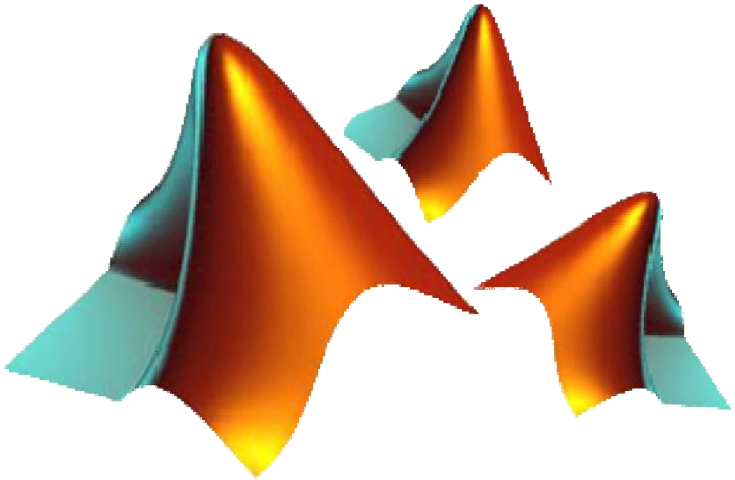
plotssquared.m

```
1     % This scrip plots the squared value of a given vector.
2 -   x=input('Please enter your vector:');
3 -   y=x.^2;
4 -   plot (x,y '^ k'), title ('Y=X^2'), xlabel ('X'), ylabel ('Y')
```

```
>> plotssquared
Error: File: plotssquared.m Line: 4 Column: 11
Unexpected MATLAB expression.

fx >>
```





MATLAB

The Language of Technical Computing

با تشکر

از

حسن توجه شما