بسم الله الرحمن الرحیم

درس نرم افزارهای ریاضی، آشنایی با نرم‌افزار متلب و لاتک

مدرس: نجمه حسینی منجزی

دانشگاه اصفهان، دانشکده ریاضی و آمار، گروه ریاضیات کاربردی و علوم کامپیوتر

بخش ۳

بهمن ۱۴۰۰

فهرست مطالب

۱ توابع روی ماتریس‌ها

بردارها و ماتریس‌ها مهمترین متغیرهایی هستند که نرم افزار با آن‌ها کار می‌کند. در واقع MATLAB بر اساس ماتریس‌ها طراحی شده است و بهترین ورودی ست که توابع MATLAB می‌تواند با آن‌ها کند. حتی اعداد را به صورت یک ماتریس یک در یک ذخیره می‌کند. بردارها که می‌توانند به صورت سطری یا ستونی تعریف شوند نیز به صورت یا ماتریس ذخیره می‌شوند.

اکثر توابعی از MATLAB که برای متغیر حقیقی یا اعداد تعریف شده‌اند برای ماتریس‌ها و بردارها نیز قابل استفاده هستند. پس در این صورت خروجی آن‌ها نیز به صورت ماتریس یا بردار است با همان ابعاد ورودی.

$$A=\begin{bmatrix} ۱ & ۲ & ۰ & ۴ \\ ۸ & ۰ & ۱ & -۱ \\ ۲ & ۱ & ۳ & ۰ \end{bmatrix} \quad \Longrightarrow \quad B=f(A)=\begin{bmatrix} f(۱) & f(۲) & f(۰) & f(۴) \\ f(۸) & f(۰) & f(۱) & f(-۱) \\ f(۲) & f(۱) & f(۳) & f(۰) \end{bmatrix}$$

که تابع f هر تابعی می‌تواند باشد.

با استفاده از عملگر . می‌توان عملیات مختلف ریاضی روی ماتریس‌ها را با هم ترکیب کرد. اگر a و b و c و d ماتریس‌های هم بعد باشند.

به طور مثال اگر همگی ۳ × ۲ باشند آنگاه عبارت

$$z=(tana-(\frac{b}{c})^{d}))^{۲}$$

به صورت زیر نوشته می‌شود

$$>> z=(tan(a)-(b./c).\hat{}d).\hat{}۲$$

قبلا درباره دستور find صحبت کردیم اگر این دستور به صورت [l،u]=find(A) به کار برده شود آنگاه اندیس درایه‌های غیرصفر ماتریس A را برمی‌گرداند. و اگر تنها بنویسم find(A) آنگاه متناظر با هر درایه غیرصفر یک عدد که شماره درایه است برگردانده می‌شود. می‌توان به صورت

$$>> [l۱, u۱] = find(A)$$

$$>> [l۲, u۲] = find(A, ۳,' first')$$

$$>> [l۳, u۳] = find(A, ۲,' last')$$

مثال - مجموع سری‌های فوریه

نمایش سری فوریه یک پالس مستطیلی در مدت d و دوره T با معادله زیر داده می‌شود

$$f(\tau) = \frac{d}{T}(۱ + ۲\sum_{n=۱}^{\infty} \frac{\sin(\pi nd/T)}{n\pi d/T} \cos(۲\pi n\tau)).$$

می‌خواهیم ۱۵۰ جمله از $f(\tau)$ را با هم جمع کنیم وآن را در $-۰/۵ \leq \tau \leq ۰/۵$ هنگامی که $\frac{d}{T} = ۰/۲۵$ می‌باشد را ترسیم کنیم.

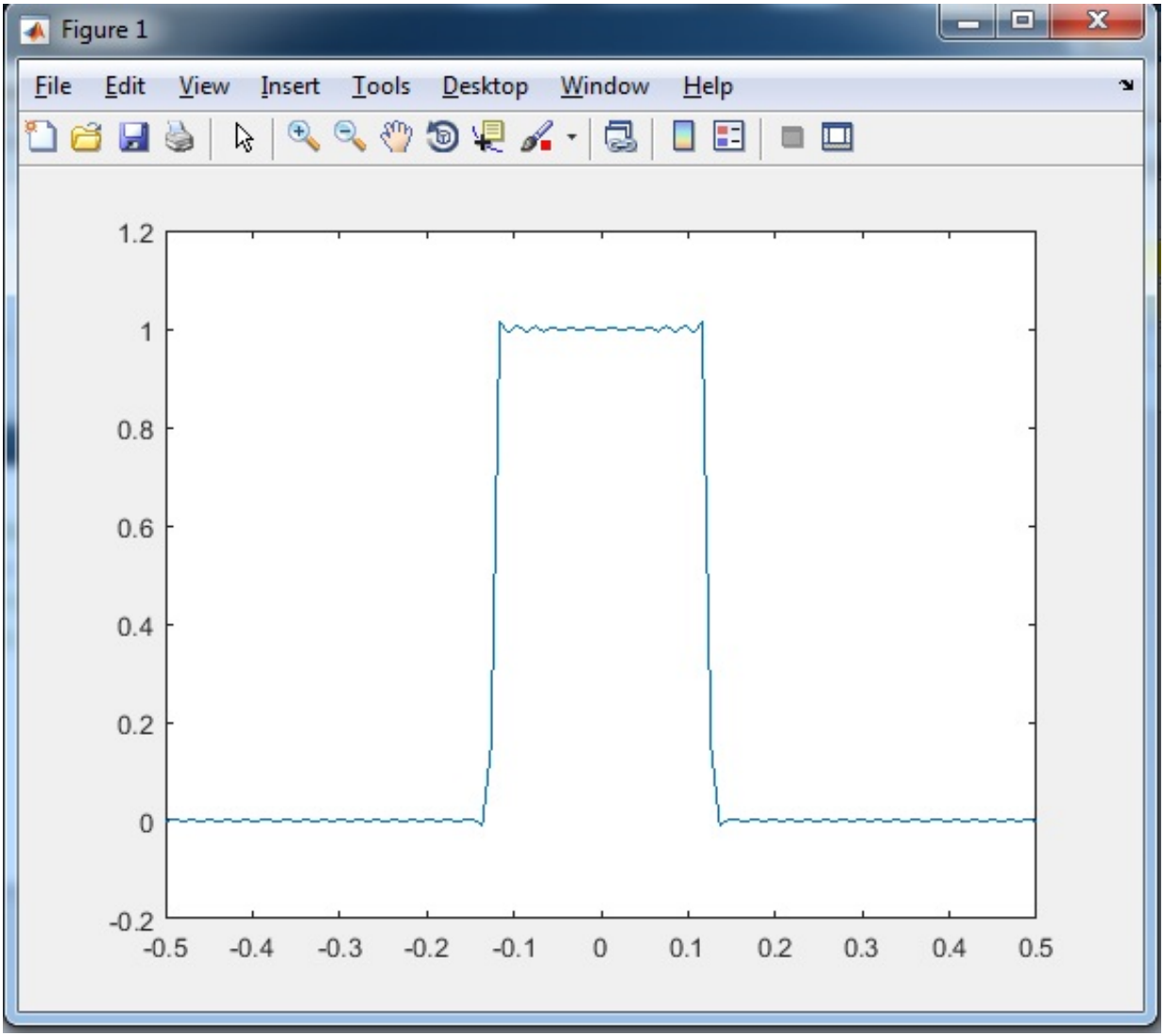
ادامه مثال – مجموع سری‌های فوریه

که دستورات متناظر برای رسم به صورت زیر می‌باشد

Command Window

```
>> n=1:1:150;       %1*150
>> tau=linspace(-0.5,0.5,100);    %1*100
>> sn=sin(pi*n*0.25)./(pi*n*0.25);   %1*150
>> cn=cos(2*pi*n'*tau);       %150*100
>> f=0.25*(1+2*sn*cn);       %1*100
>> plot(tau,f)
```

و شکل متناظر به صورت زیر است



مثال

مقدار sechx را با فرمول زیر به ازای $N = ۳۰۵$ و $۰ \le x \le ۲$ در پنج نقطه در فواصل یکسان تقریب بزنید و با مقادیر یافته شده توسط MATLAB مقایسه کنید.

$$sechx = ۴\pi \sum_{n=۱,۳,۵}^{\infty} \frac{n(-۱)^{\frac{n-۱}{۲}}}{(n\pi)^{۲} + ۴x^{۲}}$$

دستورات MATLAB به صورت زیر است

Command Window

```
>> nn=1:2:305;       %1*153
>> xx=linspace(0,2,5);     %1*5
>> [x,n]=meshgrid(xx,nn);    %153*5
>> s=4*pi*sum(n.*(-1).^((n-1)/2)./((pi*n).^2+4*x.^2));
>> se=sech(xx);
>> compare=[s' se']

compare =

    1.0021    1.0000
    0.8889    0.8868
    0.6501    0.6481
    0.4272    0.4251
    0.2679    0.2658

fx >>
```

دستورات فوق را می‌توان به صورت خلاصه زیر هم نوشت

Command Window

```
>> [x,n]=meshgrid(linspace(0,2,5),1:2:305);
>> s=4*pi*sum(n.*(-1).^((n-1)/2)./((pi*n).^2+4*x.^2));
>> compare=[s' sech(linspace(0,2,5))']

compare =

    1.0021    1.0000
    0.8889    0.8868
    0.6501    0.6481
    0.4272    0.4251
    0.2679    0.2658

fx >>
```

تمرین

سری‌های داده شده در زیر را در بازه نشان داده شده رسم کنید. برای هر سری ۲۰۰ جمله را در نظر بگیرید. دستورات را بدون sum بنویسید.

$$f(\tau) = \frac{۴}{\pi} \sum_{n=۱,۲,۳,\ldots} \frac{۱}{n} \sin(۲n\pi\tau) \quad -۰/۵ \leq \tau \leq ۰/۵$$

$$f(\tau) = \frac{۱}{۲} + \frac{۱}{\pi} \sum_{n=۱}^{\infty} \frac{۱}{n} \sin(۲n\pi\tau) \quad -۱ \leq \tau \leq ۱$$

$$f(\tau) = \frac{\pi}{۲} - \frac{۴}{\pi} \sum_{n=۱}^{\infty} \frac{۱}{(۲n-۱)^{۲}} \cos((۲n-۱)\pi\tau) \quad -۱ \leq \tau \leq ۱$$

$$t(\tau) = \frac{e^{۲*\pi} - ۱}{\pi} (\frac{۱}{۲} + \sum_{n=۱}^{۲۵۰} \frac{۱}{۱+n^{۲}} (\cos(n\pi) - n\sin(n\pi))), \quad ۰ \leq \tau \leq ۴\pi$$

۲ عملگرهای منطقی و رابطه‌ای

برنامه MATLAB علاوه بر عملگرهای محاسباتی از عملگرهای منطقی و رابطه‌ای نیز استفاده می‌کند. پاسخ این نوع عملگرها بصورت True و یا False می‌باشد. که درست را با ۱ و مقدار غلط را با ۰ نشان می‌دهد. قبلا در قسمتی که توابع روی ماتریس‌ها را معرفی کردیم درباره‌ی این عملگرها نیز صحبت کردیم اکنون می‌خواهیم با جزئیات بیشتری به آنها بپردازیم. این عمگرها می‌توانند به ساده‌تر شدن برنامه کمک کنند به علاوه با کمک این عملگرها کنترل روند اجرای برنامه ساده‌تر می‌شود.

عملگرهای رابطه‌ای:

Operator	Description
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
==	Equal to
~=	Not equal to

برای مثال داریم

Function	Description
eq(a, b)	Tests whether a is equal to b
ge(a, b)	Tests whether a is greater than or equal to b
gt(a, b)	Tests whether a is greater than b
le(a, b)	Tests whether a is less than or equal to b
lt(a, b)	Tests whether a is less than b
ne(a, b)	Tests whether a is not equal to b
isequal	Tests arrays for equality

برای این علگرها MATLAB توابعی نیز دارد از جمله

Function	Description
eq(a, b)	Tests whether a is equal to b
ge(a, b)	Tests whether a is greater than or equal to b
gt(a, b)	Tests whether a is greater than b
le(a, b)	Tests whether a is less than or equal to b
lt(a, b)	Tests whether a is less than b
ne(a, b)	Tests whether a is not equal to b
isequal	Tests arrays for equality

برای مثال داریم

Command Window

```
>> a=[1 2 3 0 -5];
>> b=[0 2 3 7 8];
>> eq(a,b)

ans =

     0     1     1     0     0

>> ge(a,b)

ans =

     1     1     1     0     0

>> gt(a,b)

ans =

     1     0     0     0     0
```


```
Command Window
>> le(a,b)

ans =

     0     1     1     1     1

>> lt(a,b)

ans =

     0     0     0     1     1

>> ne(a,b(
 ne(a,b(
        ↑
Error: Expression or statement is incorrect--possibly unbalanced (, {, or [.

Did you mean:
>> ne(a,b)

ans =

     1     0     0     1     1
```

عملگرهای دیگری در MATLAB داریم تحت عنوان عملگرهای منطقی از جمله & و | و~ که به ترتیب اعمال and و or و not را انجام می‌دهند. که این عملگرها همانند قبل می‌توانند روی اعدادو بردارها و ماتریس‌ها عمل کنند. برای مثال

Command Window

```
>> A=1:4;
>> B=A+4;
>> C=A>3

C =

     0     0     0     1

>> D=~(A>3)

D =

     1     1     1     0

>> E=(A>2)&(A<=5)

E =

     0     0     1     1
```

عملکرد این عملگرها به صورت زیر می‌باشد

p	q	p & q	p | q	p ^ q
0	0	0	0	0
0	1	0	1	1
1	1	1	1	0
1	0	0	1	1

برای مثال


```
A = 0011 1100

B = 0000 1101

----------------

A&B = 0000 1100

A|B = 0011 1101

A^B = 0011 0001

~A  = 1100 0011
```

برای عملگرهای منطقی توابعی نیز داریم که به صورت زیر عمل می‌کنند

Function	Description
and(A, B)	Finds logical AND of array or scalar inputs; performs a logical AND of all input arrays A, B, etc. and returns an array containing elements set to either logical 1 (true) or logical 0 (false). An element of the output array is set to 1 if all input arrays contain a nonzero element at that same array location. Otherwise, that element is set to 0.

not(A)	Finds logical NOT of array or scalar input; performs a logical NOT of input array A and returns an array containing elements set to either logical 1 (true) or logical 0 (false). An element of the output array is set to 1 if the input array contains a zero value element at that same array location. Otherwise, that element is set to 0.
or(A, B)	Finds logical OR of array or scalar inputs; performs a logical OR of all input arrays A, B, etc. and returns an array containing elements set to either logical 1 (true) or logical 0 (false). An element of the output array is set to 1 if any input arrays contain a nonzero element at that same array location. Otherwise, that element is set to 0.

xor(A, B)	Logical exclusive-OR; performs an exclusive OR operation on the corresponding elements of arrays A and B. The resulting element C(i,j,...) is logical true (1) if A(i,j,...) or B(i,j,...), but not both, is nonzero.
all(A)	Determine if all array elements of array A are nonzero or true. If A is a vector, all(A) returns logical 1 (true) if all the elements are nonzero and returns logical 0 (false) if one or more elements are zero. If A is a nonempty matrix, all(A) treats the columns of A as vectors, returning a row vector of logical 1's and 0's. If A is an empty 0-by-0 matrix, all(A) returns logical 1 (true).

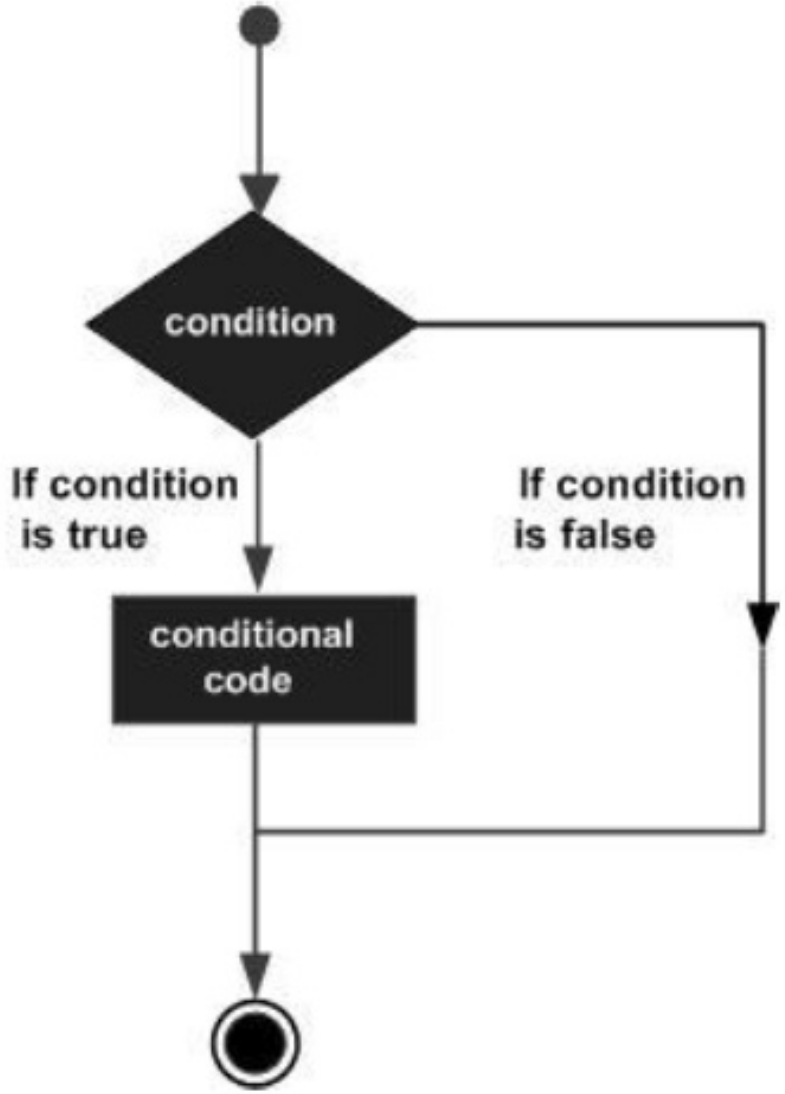
any(A)	Determine if any array elements are nonzero; tests whether any of the elements along various dimensions of an array is a nonzero number or is logical 1 (true). The *any* function ignores entries that are NaN (Not a Number). If A is a vector, any(A) returns logical 1 (true) if any of the elements of A is a nonzero number or is logical 1 (true), and returns logical 0 (false) if all the elements are zero. If A is a nonempty matrix, any(A) treats the columns of A as vectors, returning a row vector of logical 1's and 0's. If A is an empty 0-by-0 matrix, any(A) returns logical 0 (false).

۳ کنترل اجرای برنامه

تا اینجا توابع زیادی رو معرفی کردیم و درباره اعمال روی ماتریس‌ها و بردارها و عملگرهای مختلف صحبت کردیم. دقت کنید توابع در MATLAB بسیار گسترده می‌باشند و به توابعی که در اینجا مطرح شده است محدود نمی‌شود. به علاوه با گسترش و به روز رسانی این نرم افزار هرسال ممکن است توابع جدیدی نیز به نرم افزار اضافه شود. بنابراین تسلط روی کل توابع امکان پذیر نمی‌باشد و با توجه به مسئله‌ای که داریم و می‌خواهیم حل کنیم لازم است جستجو کنیم و تابع مورد نظرمان را شناسایی کنیم. ممکن است تابع مورد نظر برای مسئله ما در نرم افزار موجود نباشد در این صورت لازم است خودمان برنامه و تابع مورد نیاز مان را بنویسم و آن را ذخیره کینم و همانند خود توابع MATLAB هر زمان آن را احتیاج داشتیم آن را بکار ببریم. برای این منظور باید از محیط Editor استفاده کنیم و توابع و برنامه‌ها را در محیط m-file بنویسیم و ذخیره کنیم. پس لازم است با این محیط آشنا شویم قبل از آن با دستورات لازم جهت کنترل اجرای برنامه آشنا می‌شویم.

تصمیم‌گیری یا Decision Making

یکی از جاهایی که لازم است تصمیم‌گیری کنیم در موقعیتی شبیه به موقعیت زیر می‌باشد



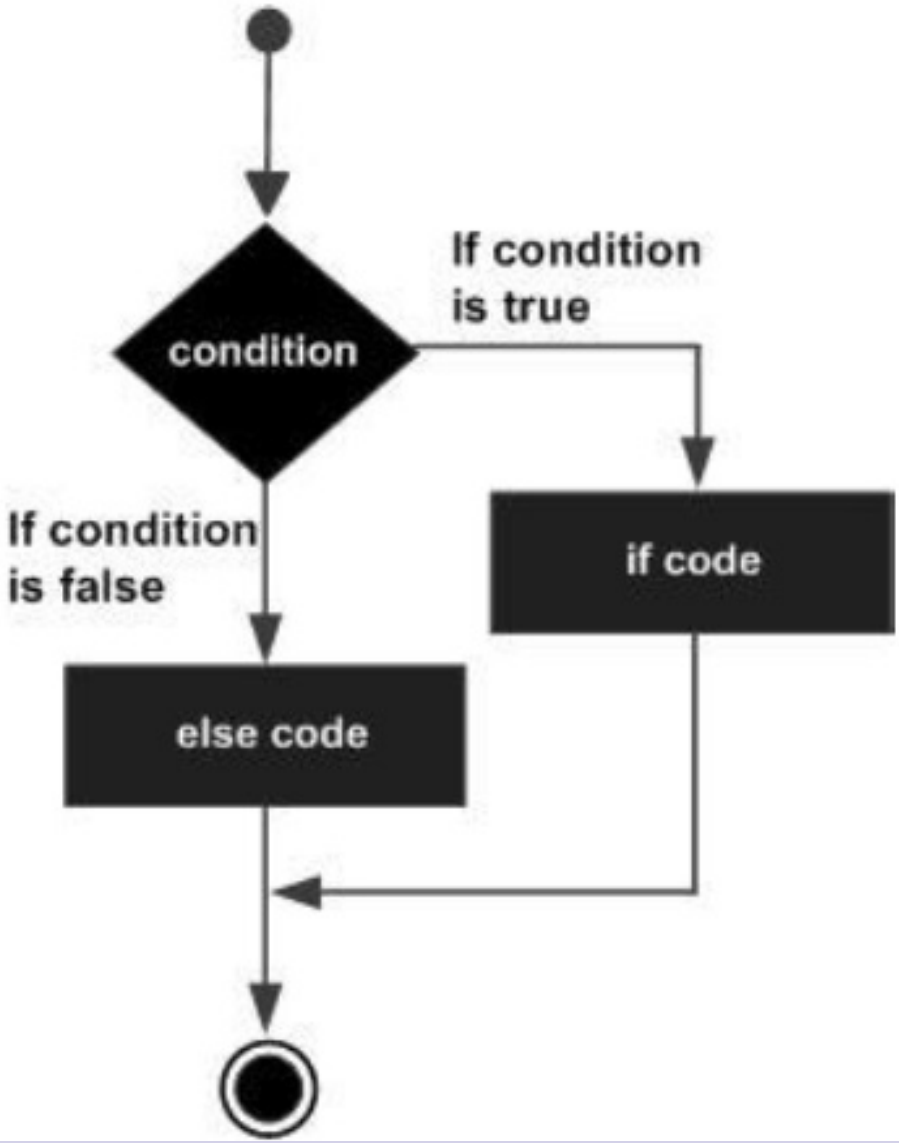
برای این منظور از دستور if استفاده می‌کنیم که به صورت زیر به کار گرفته می‌شود.

Statement	Description
if ... end statement	An if ... end statement consists of a boolean expression followed by one or more statements.
if...else...end statement	An if statement can be followed by an optional else statement, which executes when the boolean expression is false.
If... elseif...elseif...else...end statements	An if statement can be followed by one (or more) optional elseif... and an else statement, which is very useful to test various conditions.

که بیان این دستور در MATLAB به صورت زیر است

```
if <expression>
% statement(s) will execute if the boolean expression is true
<statements>
end
```

در حالتی که if و else را بخواهیم با هم به کار ببریم مانند حالت زیر



که دستورات MATLAB آن به صورت زیر است

```
if <expression>
% statement(s) will execute if the boolean expression is true
<statement(s)>
else
<statement(s)>
% statement(s) will execute if the boolean expression is false
end
```

بعلاوه می‌توان از دستور switch نیز استفاده کرد

switch statement	A switch statement allows a variable to be tested for equality against a list of values.

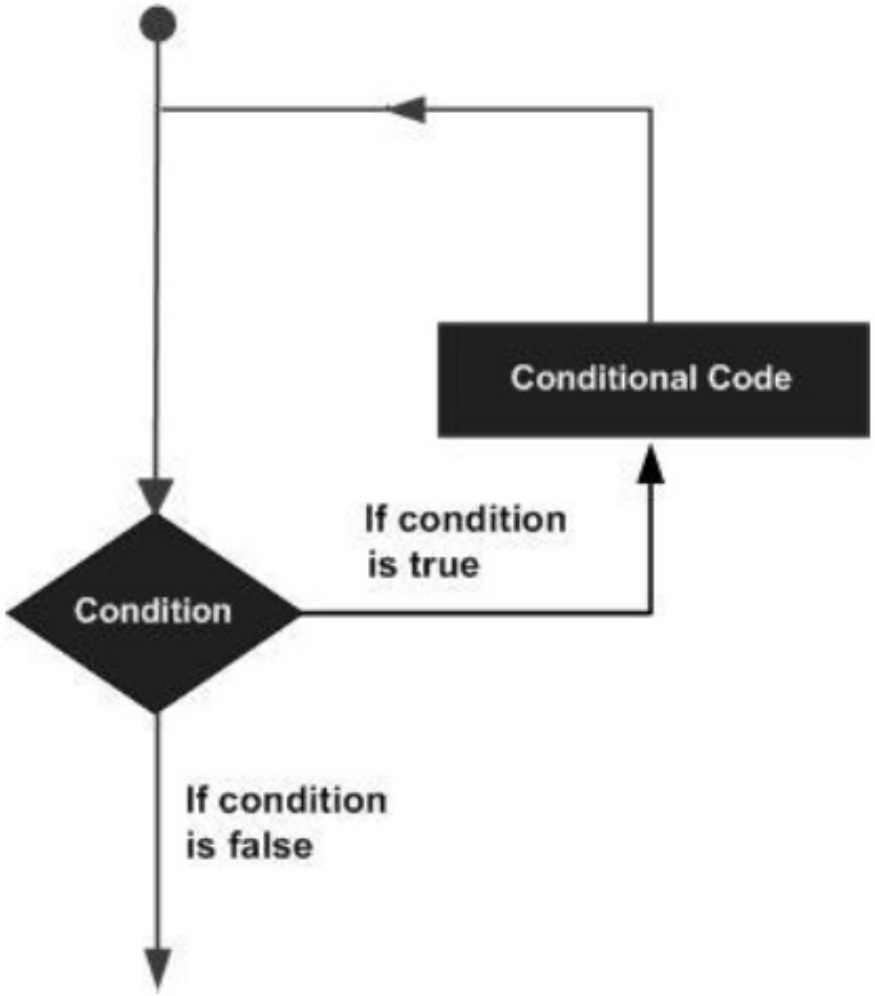
که دستورات MATLAB آن به صورت زیر است

```
switch <switch_expression>
   case <case_expression>
     <statements>
   case <case_expression>
     <statements>
     ...
     ...
   otherwise
       <statements>
end
```

حلقه‌های تکرار یا loop

گاهی لازم است دستورات خاصی را به صورت تکراری انجام دهیم. گاهی تعداد دفعاتی که لازم است این دستورات را انجام دهیم می‌دانیم و

گاهی لازم است دستورات را تا جایی انجام دهیم که شرایط خاصی برقرار شوند. در هر حالت از روش‌های مختلفی استفاده می‌کنیم. دیاگرام این شرایط به صورت زیر است



یکی از دستورات while می‌باشد که به صورت زیر عمل می‌کند

Loop Type	Description
while loop	Repeats a statement or group of statements while a given condition is true. It tests the condition before executing the loop body.

که دستورات MATLAB آن به صورت زیر است

```
while <expression>
   <statements>
end
```

گاهی لازم است عملیاتی را به تعداد متناهی بار انجام دهیم که انجام این عملیات به شرایط خاصی بستگی ندارد و از قبل می‌دانیم که این عملیات لازم است مثلا m بار انجام شود برای این منظور از دستور for استفاده می‌کنیم که بصورت زیر نوشته می‌شود.

```
for index = values
  <program statements>
        ...
end
```

در ادامه از تمام این موارد مثال‌هایی ارائه می‌کنیم که با نحوه‌ی کاربرد آن‌ها آشنا شویم. از محیط editor استفاده می‌کنیم تا کم کم با محیط برنامه‌نویسی نیز آشنا شویم. قبل از چند دستور ساده را معرفی می‌کنیم.

$$>> disp(\text{a message ... })$$

$$>> A = input(\text{ Please enter a number.})$$

دستور disp بیشتر در محیط برنامه‌نویسی به کار می‌رود و برای اطلاع رسانی به کاربر مورد استفاده قرار می‌گیرد و نقشی در محاسبات ندارد. بعلاوه اگر بخواهیم مقداری چاپ شود ولی نام متغیر چاپ نشود از دستور disp استفاده می‌کنیم. برای مثال

```
Command Window
>> A=[1 2 3; 4 5 6]

A =

     1     2     3
     4     5     6

>> A

A =

     1     2     3
     4     5     6

>> disp(A)
     1     2     3
     4     5     6

fx >>
```

و

Command Window

```
>> S='Hello word'

S =

Hello word

>> S

S =

Hello word

>> disp(S)
Hello word
>>
```

```
Command Window
>> x=rand(5,3);
>> disp('    x          y          z'),disp(x)
    x          y          z
    0.1419     0.6557     0.7577
    0.4218     0.0357     0.7431
    0.9157     0.8491     0.3922
    0.7922     0.9340     0.6555
    0.9595     0.6787     0.1712

fx >>
```

از طریق دستور input و با چاپ یک پیام می‌توانیم از کابر بخواهیم مقداری را وارد کند و این مقدار در متغیر A ذخیره می‌شود.

Command Window

```
>> a=input('Please enter a real number:')
Please enter a real number:25

a =

    25

>> A=input('Please enter a matrix 2*3:')
Please enter a matrix 2*3:[1 2; -1 0; 2 5]

A =

     1     2
    -1     0
     2     5

fx >>
```

در ادامه چند مثال از حلقه‌های تکرار می‌بینیم

Command Window

```
>> n=10;
>> sum=0;
>> for i=1:n
sum=sum+i;
end
>> sum

sum =

    55

>>
```

Command Window

```
>> for i=1:n
sum=sum+i
end

sum =

     1


sum =

     3


sum =

     6


sum =

    10
```

Command Window

```
sum =

    15


sum =

    21


sum =

    28


sum =

    36


sum =

    45
```

برنامه‌ای بنویسید که از کاربر عدد دریافت کند و این اعداد را در یک بردار ذخیره کند تا زمانی که یک عدد مثبت زوج وارد شود عملیات را متوقف کنید.

Command Window

```
>> i=0;
>> mod1=1;
>> while mod1~=0
b=input('Please enter a real number:')
i=i+1;
a(i,1)=b;
if b>0
mod1=mod(b,2);
end
end
Please enter a real number:25

b =

    25

Please enter a real number:13

b =

    13

Please enter a real number:-14

b =
```

```
Command Window
b =

    13

Please enter a real number:-14

b =

   -14

Please enter a real number:8

b =

     8

>> a

a =

    25
    13
   -14
     8
```

و مثال دیگر

Command Window

```
>> grade='B';
>> switch(grade)
case 'A'
fprintf('Excellent!\n');
case 'B'
fprintf('Well Done!\n')
case 'C'
fprintf('Good.\n')
case 'D'
fprintf('You passed.')
case 'E'
fprintf('Better try again.')
end
Well Done!
>>
```

از هر کدام از این دستورات می‌توان به صورت تو در تو نیز استفاده کرد.