

**بِسْمِ تَعَالَى**

# **آموزش نرم افزار MATLAB (بخش اول)**

**مدرس: مهندس محتشمی**

**[shmohtashami.blog.ir](http://shmohtashami.blog.ir)**

---

▶ MATLAB نرم افزاری برای انجام کارهای ریاضی، آماری، مهندسی و... می باشد.

▶ MATLAB=MATrix LABoratory (آزمایشگاه ماتریسی)

▶ در اوایل دهه ۱۹۷۰ توسط Cleve Moler به وجود آمد.

▶ در اوایل دهه ۱۹۸۰ با زبان C بازنویسی شد.

▶ در ۱۹۸۴ شرکت Mathwork تأسیس شد.

# MATLAB در نقش یک ماشین حساب

---

▶ ساده ترین کارهایی که با MATLAB می توان انجام داد همان  
اعمالی است که یک ماشین حساب معمولی انجام می دهد. جمع،  
تفریق، ضرب و تقسیم

# MATLAB در نقش یک ماشین حساب (ادامه)

---

- ▶ بعد از اجرای نرم افزار چند window (پنجره) ظاهر می شود.
- ▶ یکی از این پنجره ها command window نام دارد.
- ▶ در این محیط دستورات MATLAB نوشته و اجرا می شوند.

Workspace

Stack: Base

Name	Size	Bytes	Class
------	------	-------	-------

Workspace Current Directory

Command Window

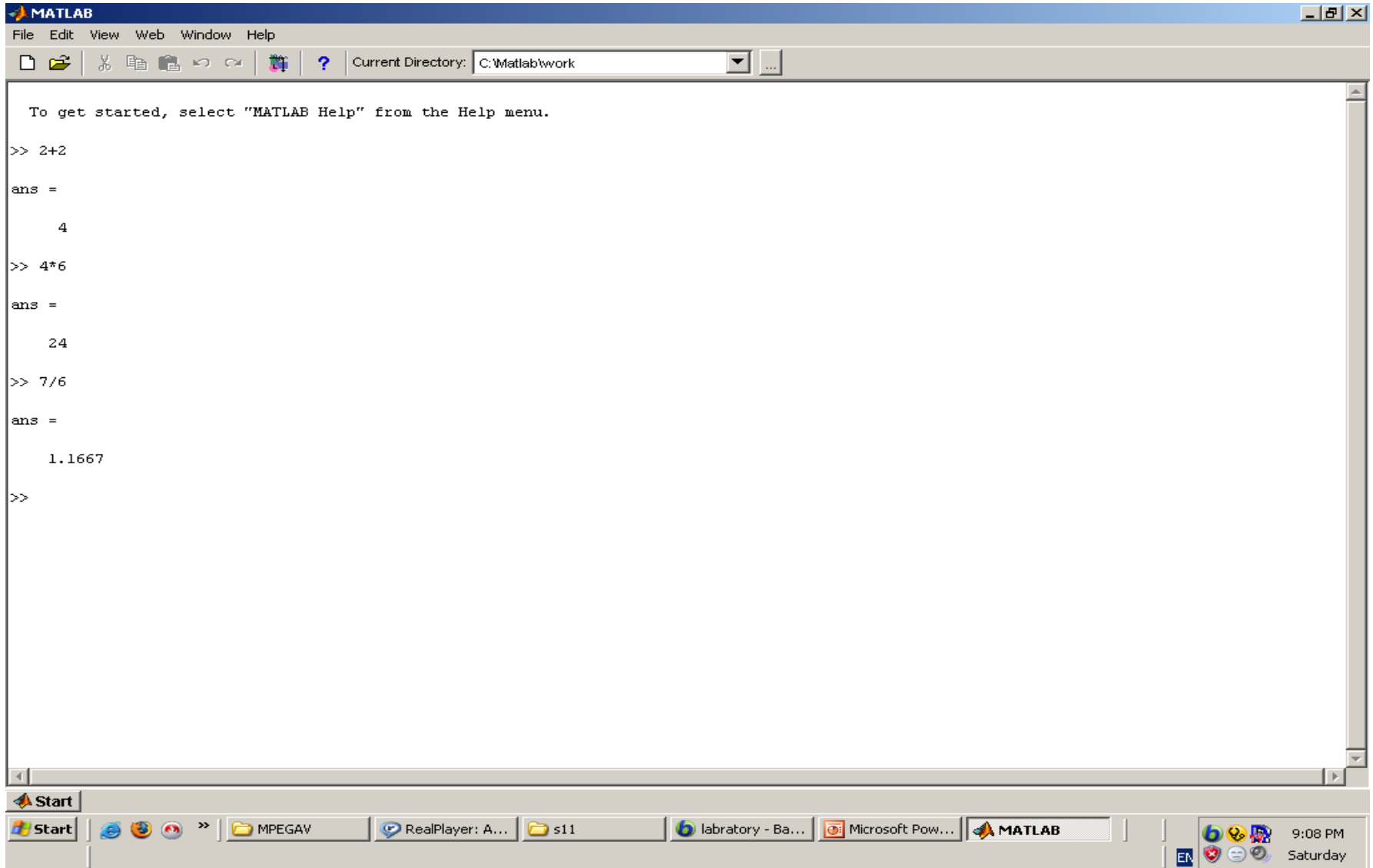
To get started, select "MATLAB Help" from the Help menu.

```
>> |
```

Command History

```
%-- 12/17/05 9:03 PM --%
```

# انجام کارهای ساده محاسباتی



The image shows a MATLAB window with a blue title bar and a menu bar (File, Edit, View, Web, Window, Help). The current directory is set to C:\Matlab\work. The command window contains the following text:

```
To get started, select "MATLAB Help" from the Help menu.  
>> 2+2  
ans =  
     4  
>> 4*6  
ans =  
    24  
>> 7/6  
ans =  
    1.1667  
>>
```

The Windows taskbar at the bottom shows the Start button, several icons, and open applications including MPEGAV, RealPlayer, s11, laboratory - Ba..., Microsoft Pow..., and MATLAB. The system tray shows the time as 9:08 PM on Saturday.

# آشنایی با بعضی قابلیت های MATLAB

---

▶ تعریف و استفاده از متغیرها

▶ محاسبات ماتریسی

▶ نوشتن M-file (Script نویسی)

▶ دستورات حلقه و شرط

▶ دستورات ورودی-خروجی

▶ ساخت و احضار توابع

▶ رسم نمودار

# تعریف و استفاده از متغیرها

---

- ▶ با نوشتن دستور  $x=23$  متغیری به نام  $x$  در حافظه ایجاد می شود و مقدار 23 در آن ریخته می شود.
- ▶ اگر دستور به صورت  $x=23$  نوشته شود MATLAB مقدار آن را مجدداً نمایش می دهد و اگر به صورت  $x=23$  نوشته شود MATLAB آن مقدار را نشان نمی دهد.
- ▶ در MATLAB نیاز به تعریف متغیر وجود ندارد.
- ▶ همه متغیرها از نوع double (۸ بایت) فرض می شود.
- ▶ MATLAB زبانی حساس به حالت حروف (Case sensitive) است.



---

▶ با استفاده از دستور `whos` می توان فهمید چه متغیرهایی تاکنون تعریف شده است.

▶ با استفاده از دستور `clear` می توان یک متغیر را از حافظه پاک کرد.

▶ متغیرها تا زمانی که `clear` نشوند در حافظه باقی می مانند.

▶ دستور `clc` برای پاک کردن پنجره `command` به کار می رود.

>> x=23

To get started, select "MATLAB Help" from the Help menu.

x =

23

>> y=24;

>> zzz=x+y;

>> whos

Name	Size	Bytes	Class
x	1x1	8	double array
y	1x1	8	double array
zzz	1x1	8	double array

Grand total is 3 elements using 24 bytes

>> clear x

>> whos

Name	Size	Bytes	Class
y	1x1	8	double array
zzz	1x1	8	double array

Grand total is 2 elements using 16 bytes

>> |

## متغیرهای توکار

---

$\pi$  عدد: pi

realmax: بزرگترین عدد حقیقی مثبت قابل نمایش

realmin: کوچکترین عدد حقیقی مثبت قابل نمایش

inf: بی نهایت مثبت

# تعریف ماتریس و محاسبات ماتریسی

---

نحوه تعریف ماتریس ▶

$$\begin{bmatrix} 3 & -9 & 4 \\ -5 & 3 & 6 \end{bmatrix}$$

برای تعریف ماتریس فوق در MATLAB به صورت زیر عمل می کنیم:

$$AA=[3 \ -9 \ 4; \ -5 \ 3 \ 6]$$

# محاسبات ماتریسی

---

- ▶ جمع و تفریق با + و -
- ▶ ضرب دو ماتریس (با ابعاد  $m \times n$  و  $n \times p$ ): با \*
- ▶ ترانزاده ماتریس: با '،
- ▶ نکته: اگر بخواهیم عناصر دو ماتریس نظیر بنظیر در هم ضرب شوند، بجای \* از \* استفاده می کنیم. به همین شکل از / برای تقسیم نظیر به نظیر استفاده می شود.
- ▶  $A^2$ ،  $A \cdot A$ ،  $A \cdot A$  را محاسبه می کند.
- ▶  $A \cdot B$ : به توان رسانی نظیر به نظیر  $A \cdot B$

# مثال

$$A=[1 \ 2 \ 3; 4 \ 5 \ 6; 3 \ 4 \ 7];$$

$$B=[2 \ 3; 1 \ 1; 4 \ 5];$$

در اینجا یک ماتریس  $3 \times 3$  به نام A و یک ماتریس  $3 \times 2$  به نام B تعریف کرده ایم.

$$C=A*B$$

این دستور ضرب ماتریسی A در B می باشد و حاصل که ماتریسی  $3 \times 2$  است در C قرار می گیرد.

$$D=B*A$$

این دستور اشتباه است چون نمی توان یک ماتریس  $3 \times 2$  را در ماتریس  $3 \times 3$  ضرب کرد.

$$E=A^2$$

ماتریس  $3 \times 3$  A را در خودش ضرب می کند. حاصل ماتریس  $3 \times 3$  E است.

$$F=B^2$$

خطا اعلام می شود. چون  $B^2$  یعنی  $B*B$  . اما نمی توان یک ماتریس  $3 \times 2$  را در  $3 \times 2$  ضرب کرد.

$$G=B.^2$$

این دستور درست است و معادل  $B.*B$  یعنی ضرب نظیر به نظیر عناصر B است.



```
>> A=[1 2 3;4 5 6;3 4 7];  
B=[2 3; 1 1;4 5];  
>> C=A*B
```

```
C =  
  
16    20  
37    47  
38    48
```

```
>> D=B*A  
??? Error using ==> *  
Inner matrix dimensions must agree.
```

```
>> E=A^2  
  
E =  
  
18    24    36  
42    57    84  
40    54    82
```

```
>> F=B^2  
??? Error using ==> ^  
Matrix must be square.
```

```
>> G=B.^2  
  
G =  
  
4     9  
1     1  
16    25
```

```
>> |
```

# چند دستور پر کاربرد در ایجاد ماتریس ها و بردارها

---

$A=[ ]$ ; یک ماتریس تهی ایجاد می کند

$x=0:0.1:1$

یک بردار ایجاد می کند که عنصر اول آن 0 است. عناصر بعدی با اضافه کردن 0.1 (گام افزایش) به دست می آیند تا زمانی که به 1 برسیم. یعنی:

$x=[0 \ 0.1 \ 0.2 \ 0.3 \ 0.4 \ 0.5 \ 0.6 \ 0.7 \ 0.8 \ 0.9 \ 1]$

$x=0:4$

اگر مانند این مثال، گام افزایش ذکر نشود 1 در نظر گرفته می شود:

$x=[0 \ 1 \ 2 \ 3 \ 4]$ ;

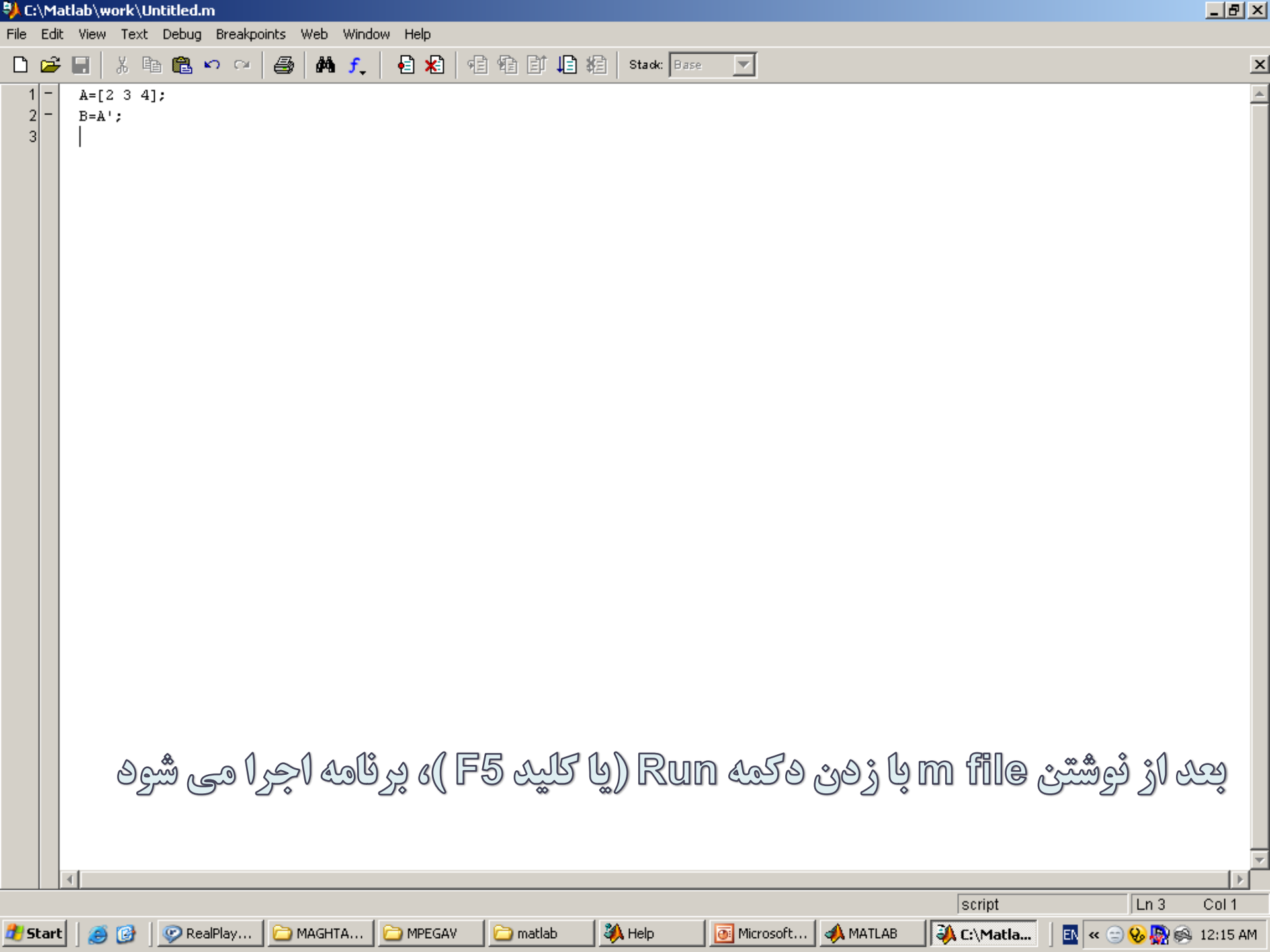




# script نویسی m- فایلها

---

- ▶ 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. MATLAB دستورات موجود در این فایل را اجرا می کند. به این فایل script متلب یا m file می گوئیم.
- ▶ نحوه نوشتن m file: مجموعه دستورات موردنظر را در ویرایشگر می نویسیم. فایل را با پسوند m ذخیره می کنیم.
- ▶ MATLAB یک ویرایشگر برای انجام این کار دارد که با Home/New Script اجرا می شود.



بعد از نوشتن m file با زدن دکمه Run (یا کلید F5)، برنامه اجرا می شود

# دستورات حلقه و شرط

---

▶ در MATLAB مشابه زبان های برنامه نویسی دستورات حلقه و شرط وجود دارد.

▶ فرمت دستور for

```
for x=1:0.1:2
```

«یک یا چند دستور»

```
end
```

متغیر x از ۱ تا ۲ با گام ۰٫۱ تغییر می کند.

---

فرمت دستور while ▶

while شرط

«دستور یا دستورات»

end

فرمت دستور if ▶

if شرط

«دستور یا دستورات»

end

---

## دستور if-else ▶

if شرط

«دستور یا دستورات»

else

«دستور یا دستورات»

end

---

## فرمت دستور switch

switch متغیر

case مقدار ۱

«دستور یا دستورات»

case مقدار ۲

«دستور یا دستورات»

otherwise

«دستور یا دستورات»

end

اگر هیچ یک از case ها برقرار نباشند، بخش otherwise اجرا می شود. نوشتن این بخش اختیاری است.

نکته: عملگرهای مقایسه ای در MATLAB:

> < >= <= == (مساوی) ~= (نامساوی)

& (و منطقی) | (یا منطقی) ~ (نقیض)

&& (و) || (یا)

مثال

```
x=4;  
y=5;  
if x==4 && y==2  
    z=1;  
else  
    z=4;  
end
```

z برابر ۴ خواهد شد.

# دستورات ورودی و خروجی در MATLAB

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

```
x=input('Enter a number or matrix:');
```

در اجرا، ابتدا پیام Enter a number or matrix: چاپ می شود. سپس منتظر می ماند تا کاربر عدد یا ماتریسی وارد کند. این عدد یا ماتریس در x قرار می گیرد (صفحه بعد).

▶ تابع disp برای چاپ پیام یا مقدار متغیر در پنجره Command 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 با هم در پنجره command اجرا کرد (صفحه بعد)



```
>> x=input('Enter a number or matrix:');
```

```
Enter a number or matrix:[2 3 4;6 7 8]
```

```
>> x
```

```
x =
```

```
     2     3     4  
     6     7     8
```

```
>> disp('the value')
```

```
the value
```

```
>> disp(x)
```

```
     2     3     4  
     6     7     8
```

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

```
the value
```

```
     2     3     4  
     6     7     8
```

```
>> |
```

# نوشتن تابع در MATLAB

---

▶ در MATLAB مشابه زبان های برنامه نویسی می توان برای برنامه تابع نوشت .

# نوشتن تابع توسط کاربر

▶ یک file m به صورت زیر ایجاد می کنیم (کلماتی که با رنگ قرمز مشخص شده اند باید عینا نوشته شوند):

**function** y=afun(a,b)

«یک سری دستورات»

afun اسم تابع است.

y برگشتی تابع است.

a و b ورودی تابع.

▶ بعد از نوشتن m file آن را با نام تابع (در اینجا afun) و پسوند m ذخیره می کنیم.

▶ سپس می توان در پنجره command، تابع را اجرا کرد. مثلا اگر بنویسیم afunc(3,4)، 3 به جای a و 4 به جای b قرار می گیرد. تابع اجرا می شود و مقدار y به دست آمده نشان داده می شود.

▶ می توان تابعی با چند خروجی داشت

**function [y,t,x]=afunc(m)**

```
1 function y=abs(x)
2     if (x>0)
3         y=x;
4     else
5         y=-x;
6     end
7
8
```

## نکاتی در مورد نوشتن تابع

---

- ▶ نام تابع باید هم نام با `m file` باشد.
- ▶ می توان در یک `m file` چندین تابع داشت. اولین تابع در آن، تابع اصلی بوده و بقیه زیر تابع گفته می شوند. اسم `m file` باید هم نام با این تابع اصلی باشد.
- ▶ متغیرهای موجود در یک تابع، محلی هستند و توسط توابع دیگر قابل دسترسی نمی باشند.
- ▶ در صورت وجود چند تابع در یک `m-file` تنها احضار تابع اصلی در پنجره `command` امکان پذیر است. البته توابع داخل `m-file` می توانند یکدیگر را احضار کنند.

## روشی دیگر برای احضار تابع

---

▶ می توان در یک تابع تابعی دیگر را که در یک m-file دیگر است

احضار کرد. مشروط بر اینکه این m-file در همان دایرکتوری تابع

قرار داشته باشد.

# توابع کتابخانه ای

---

▶ **توابع مثلثاتی:** `atan`، `acos`، `asin`، `tan`، `cos`، `sin`

▶ `exp` (نمایی)، `log` (لگاریتم طبیعی)، `log10` (لگاریتم در مبنای ۱۰).

`factorial` (محاسبه فاکتوریل)

▶ `floor` (جزء صحیح)، `fix` (قسمت صحیح)، `ceil` (سقف)، `sqrt` (جذر)،

`abs` (قدر مطلق).

▶ `eye` (ایجاد ماتریس واحد)، `zeros` (ایجاد ماتریس با عناصر فقط صفر)،

`ones` (ایجاد ماتریسی با عناصر فقط یک)

## مثال

---

▶  $\text{floor}(-0.3) = -1$     $\text{floor}(0.3) = 0$     $\text{floor}(-0.8) = -1$     $\text{floor}(0.8) = 0$

▶  $\text{fix}(-0.3) = 0$     $\text{fix}(0.3) = 0$     $\text{fix}(-0.8) = 0$     $\text{fix}(0.8) = 0$

▶  $\text{ceil}(-0.3) = 0$     $\text{ceil}(0.3) = 1$     $\text{ceil}(-0.8) = 0$     $\text{ceil}(0.8) = 1$

▶  $\text{round}(-0.3) = 0$     $\text{round}(0.3) = 0$     $\text{round}(-0.8) = -1$     $\text{round}(0.8) = 1$





# مثال

---

$$x=[0 \ 0.1 \ 0.2 \ 0.5]$$

$$y=\sin(x)$$

**y، برداری است حاوی سینوس عناصر x (صفحه بعد):**

$$y=\text{eye}(4)$$

**ماتریس واحد با ابعاد ۴\*۴ ایجاد می کند.**



```
>> x=[0 0.1 0.2 0.5]
```

```
y=sin(x)
```

```
x =  
  
    0    0.1000    0.2000    0.5000
```

```
Y =  
  
    0    0.0998    0.1987    0.4794
```

```
>> x=[1 2 ;3 4];
```

```
y=inv(x)
```

```
Y =  
  
 -2.0000    1.0000  
  1.5000   -0.5000
```

```
>> y=eye(4)
```

```
Y =  
  
  1    0    0    0  
  0    1    0    0  
  0    0    1    0  
  0    0    0    1
```

```
>>
```

---

`b=zeros(4,3)`

**یک ماتریس  $4 \times 3$  با عناصر صفر ایجاد می کند.**

`x=ones(4,3)`

**ماتریس  $4 \times 3$  با عناصر یک ایجاد می کند.**



# رسم نمودار

---

▶ دستورات متعددی برای رسم نمودار در MATLAB وجود دارند.

تابع `plot(x,y)` نمودار  $y$  را بر حسب  $x$  رسم می کند.

```
x=0:0.01:pi;
```

```
y=sin(x);
```

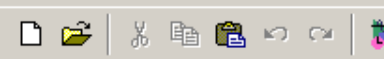
```
plot(x,y);
```

مجموعه دستورات بالا نمودار  $y=\sin(x)$  را در بازه  $[0, 3.14]$  رسم می کند.

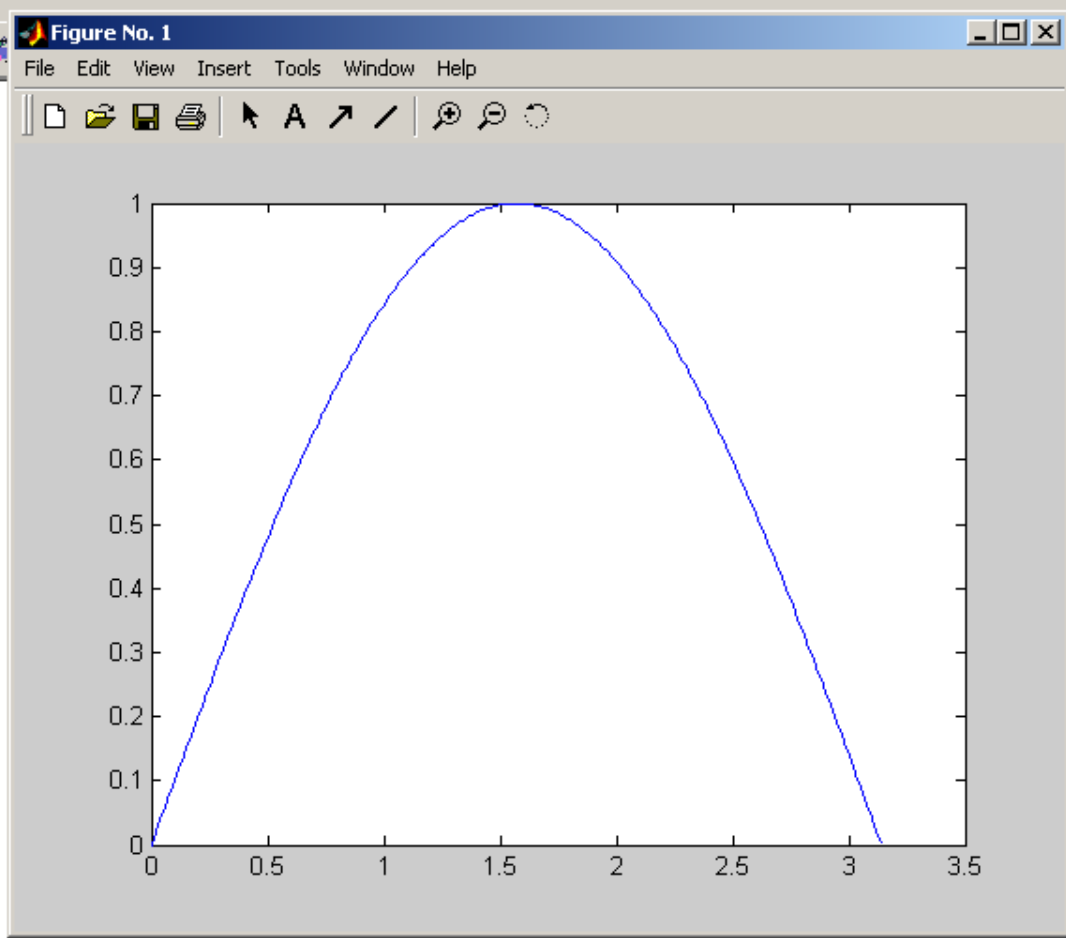
نکته: نقاط داده شده گسسته هستند. اما تابع `plot` با اتصال نقاط رسم شده به یکدیگر یک

نمودار پیوسته را نشان می دهد. تابع `stem` همان کار `plot` را انجام می دهد منتها نقاط را

به هم وصل نمی کند و نمودار گسسته ایجاد می کند.



```
>> x=0:0.01:3.14;  
y=sin(x);  
plot(x,y);  
>>
```



## رسم نمودارهای سه بعدی

---

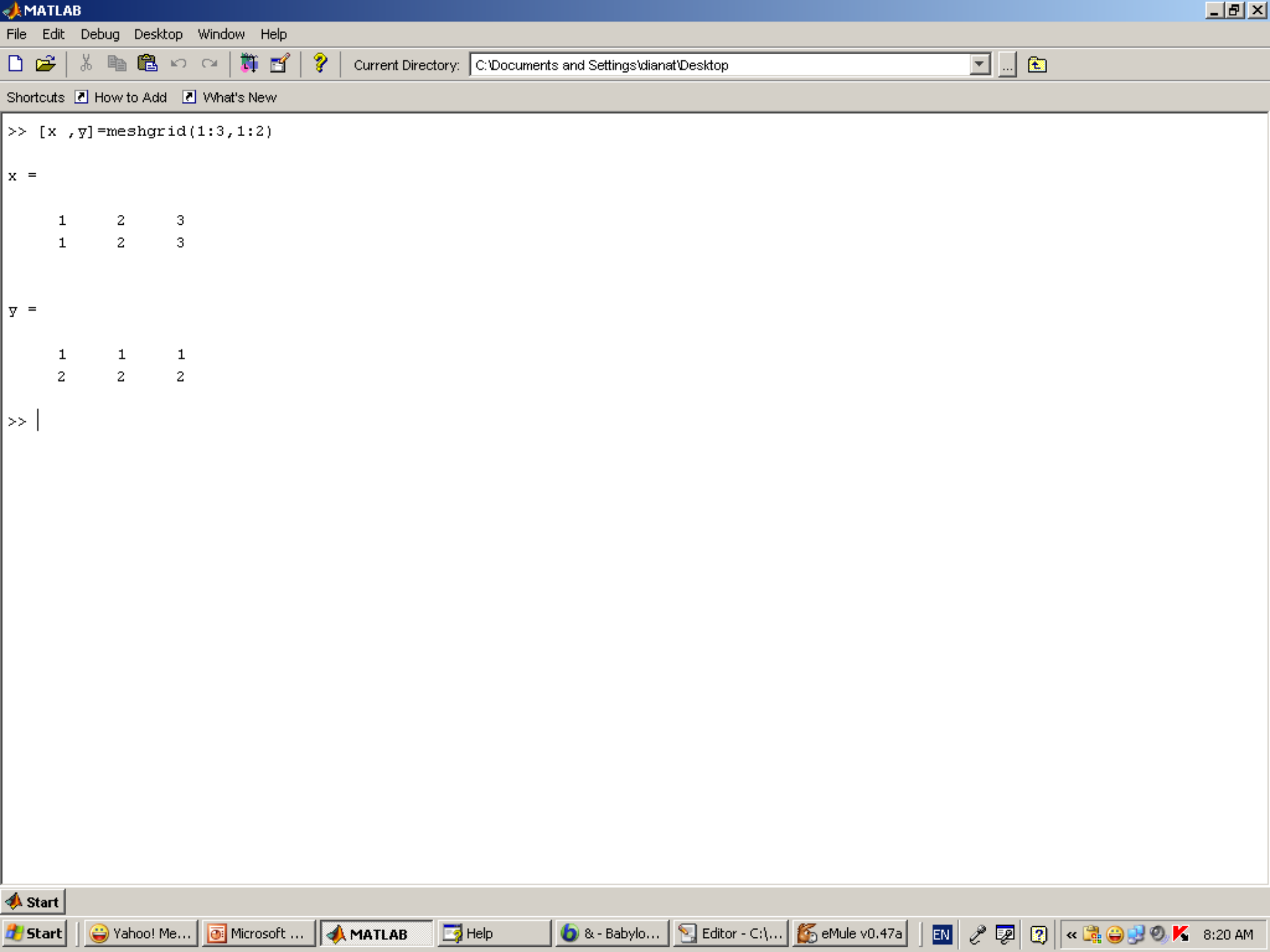
▶ هدف : رسم تابع  $z=f(x,y)$

▶ نکته مهم: اگر مثلا  $x=1:3$  و  $y=1:2$  ،  $z$  به ازای همه  $(x,y)$  ها مقدار

باید داشته باشد.  $(1,1)$ ،  $(2,1)$  ،  $(3,1)$  و....

دستور `meshgrid` برای ساخت این نقاط به کار می رود:

```
[x ,y]=meshgrid(1:3,1:2);
```



>> [x ,y]=meshgrid(1:3,1:2)

x =  
  
 1 2 3  
 1 2 3

y =  
  
 1 1 1  
 2 2 2

>> |

---

**تابع  $\text{plot3}(x,y,z)$  برای رسم  $z=f(x,y)$  به کار می رود:**

```
[x,y]=meshgrid(-3:0.01:3,-3:0.01,3);
```

```
z=exp(-x.^2-y.^2);
```

```
plot3(x,y,z);
```

**نمودار  $z = \exp(-x^2 - y^2)$  را رسم می کند. این تابع منحنی را به صورت یک سری خطوط رسم می کند.**



### Workspace

Stack: Base

Name	Value	Class
x	<61x61 double>	double
y	<61x61 double>	double
z	<61x61 double>	double

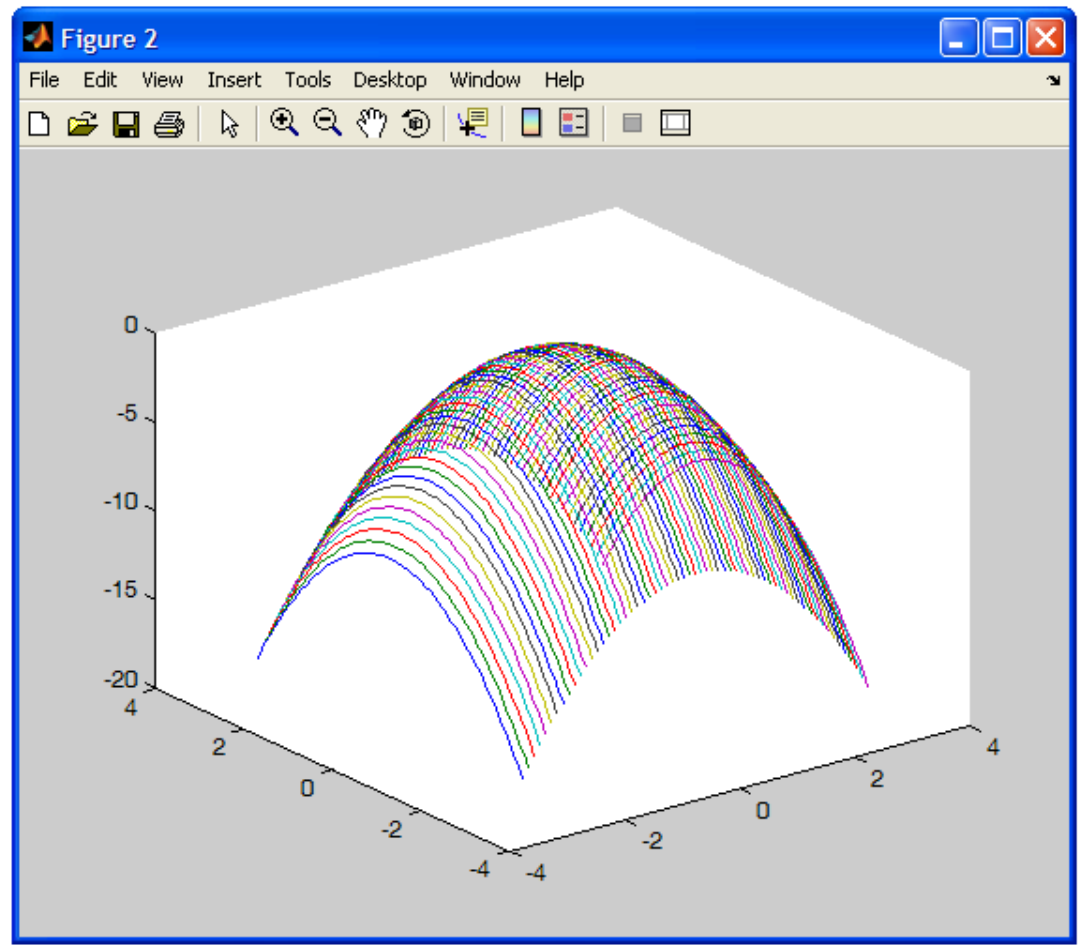
Current Directory Workspace

### Command History

```
plot3(x,y,z)
figure;
mesh(x,y,z)
surf(x,y,z)
clc
clear
clear all
clc
[x y]=meshgrid(-3:0.1:3,-3:0.1:3)
clc
clear
[x y]=meshgrid(-3:0.1:3,-3:0.1:3);
z=(-x.^2-y.^2);
plot3(x,y,z)
```

### Command Window

```
>> clear
>> [x y]=meshgrid(-3:0.1:3,-3:0.1:3);
>> z=(-x.^2-y.^2);
>> plot3(x,y,z)
>>
```



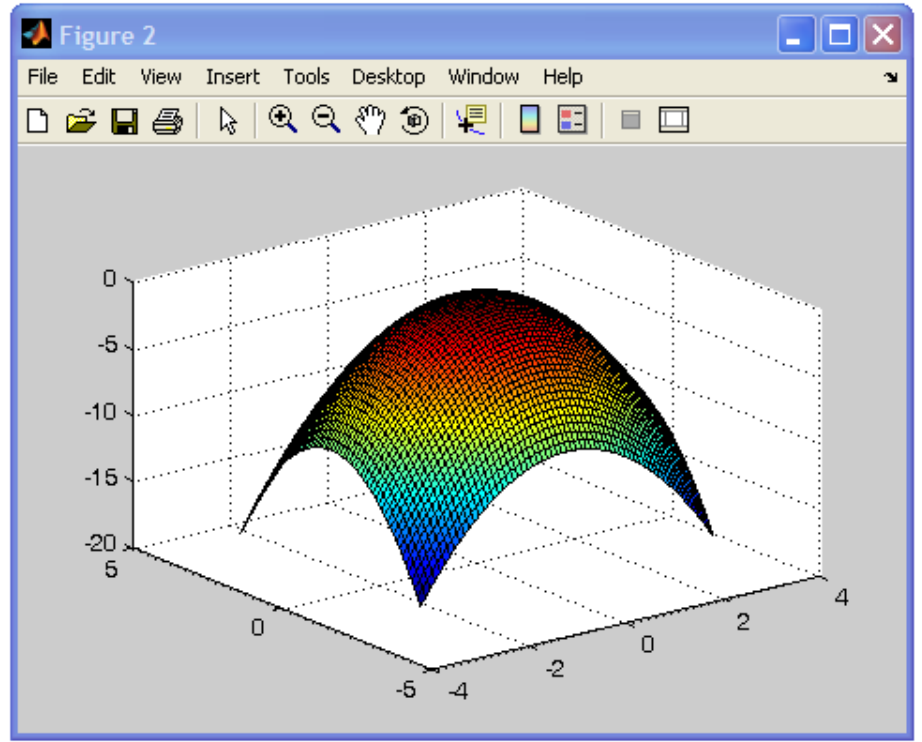
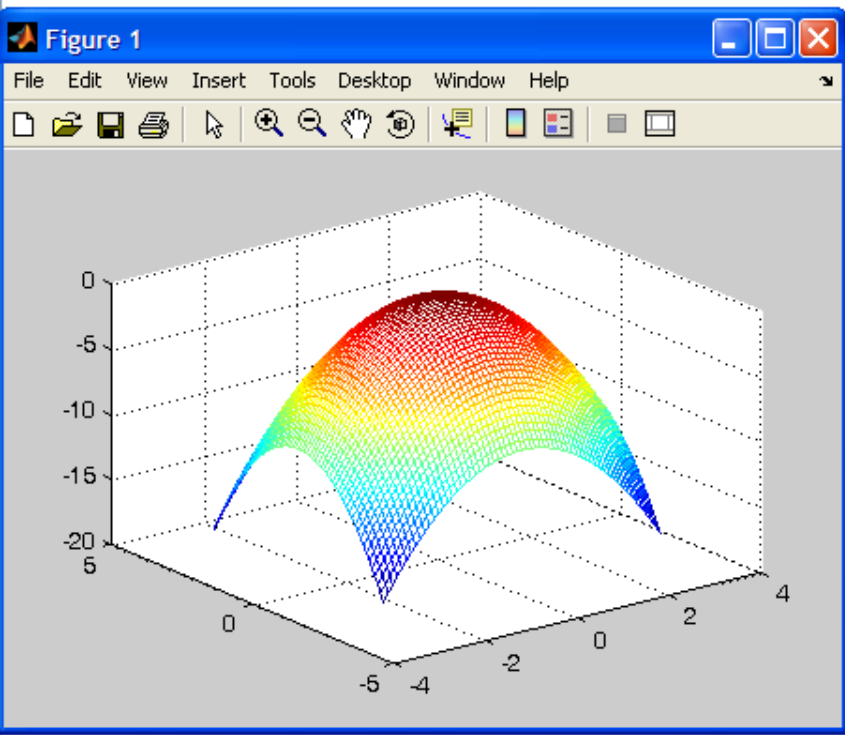
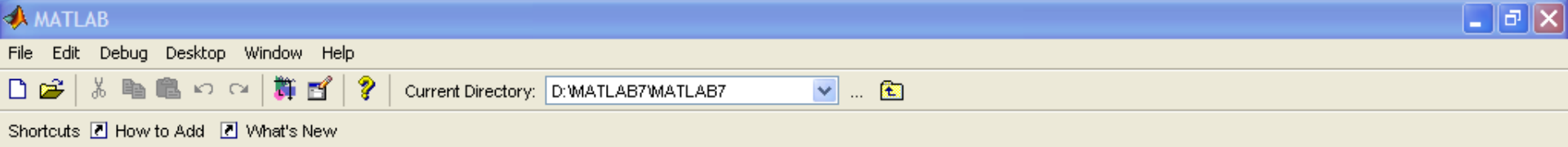
## دستورات mesh و surf

---

▶ mesh: منحنی را به صورت شبکه شبکه با خطوط رنگارنگ رسم می کند.

▶ surf: منحنی را به صورت شبکه شبکه با خطوط رنگارنگ رسم می کند و به

علاوه داخل شبکه ها را هم رنگ می کند.



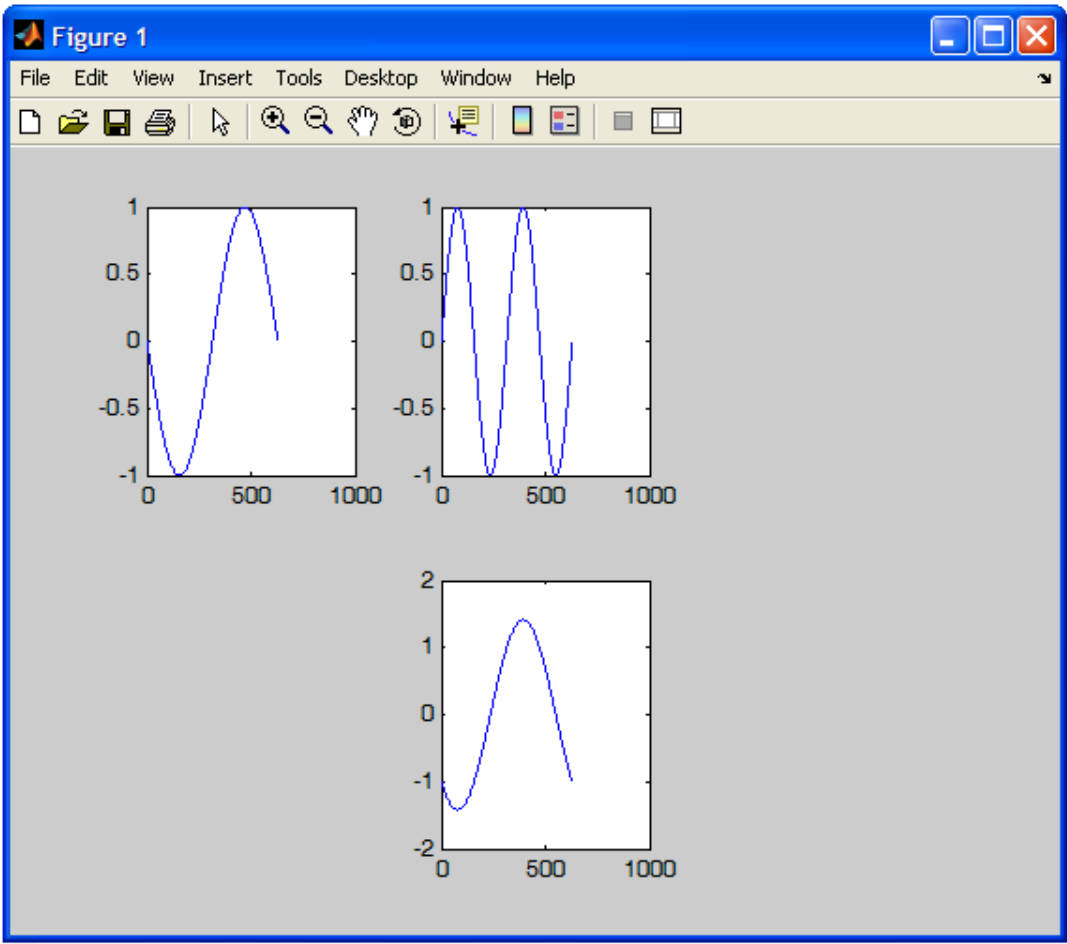
MATLAB

File Edit Debug Desktop Window Help

Current Directory: D:\MATLAB7\MATLAB7

Shortcuts How to Add What's New

```
>> x=-pi:0.01:pi;  
>> y1=sin(x);y2=sin(2*x);y3=sin(x)+cos(x);  
>> subplot(2,3,1); plot(y1); subplot(2,3,2);plot(y2); subplot(2,3,5) ;plot(y3);  
>>
```



رسم چند منحنی  
در یک پنجره