

## کاربرد MATLAB در سیگنالها و سیستمها و کنترل

### مقدمه

بسته نرم‌افزاری MATLAB (Matrix Laboratory) یک سیستم ماتریس محور برای محاسبات ریاضی و مهندسی است. MATLAB در طول سالهای اخیر به ابزار بسیار قدرتمندی برای انجام پردازشهای پرزحمت و سنگین مهندسی مبدل شده است؛ به طوری که رسم برخی نمودارها یا انجام بعضی محاسبات را جز به کمک MATLAB نمی‌توان تصور کرد.

MATLAB امروزه دارای جعبه‌ابزارهای (TOOLBOX) متنوعی برای انجام محاسبات مختلف است که از جمله آنها می‌توان به جعبه‌ابزارهای پردازش سیگنال (Signal Processing)، پردازشهای آماری (Statistics)، شبکه‌های عصبی (Neural Network)، دریافت تصویر (Image Acquisition)، پردازش تصویر (Image Processing)، سیستمهای کنترلی (Control Systems)، طراحی فیلتر (Filter Design)، منطق فازی (Fuzzy Logic)، الگوریتمهای ژنتیک (Genetic Algorithms) و ... اشاره کرد. بعضی از این جعبه‌ابزارها به همراه نرم‌افزار MATLAB ارائه می‌شوند و بعضی دیگر را می‌توان در کارگروههای مختلف و نیز سایت [www.Mathworks.com](http://www.Mathworks.com) یافت. هدف این نوشتار، تشریح کامل ساختار MATLAB و جعبه‌ابزارهای آن نیست. هدف ما ارائه مطالب ضروری برای کاربرد مؤثر MATLAB در حل مسایل پردازش سیگنال و سیستمهای کنترل است.

در بخش ابتدایی این مطلب به ساختارهای اساسی MATLAB و اصول استفاده از آن خواهیم پرداخت.

بخش دوم به کاربرد MATLAB در پردازش تصویر به عنوان یکی از کاربردهای اصلی سیگنالها و سیستمها اختصاص دارد.

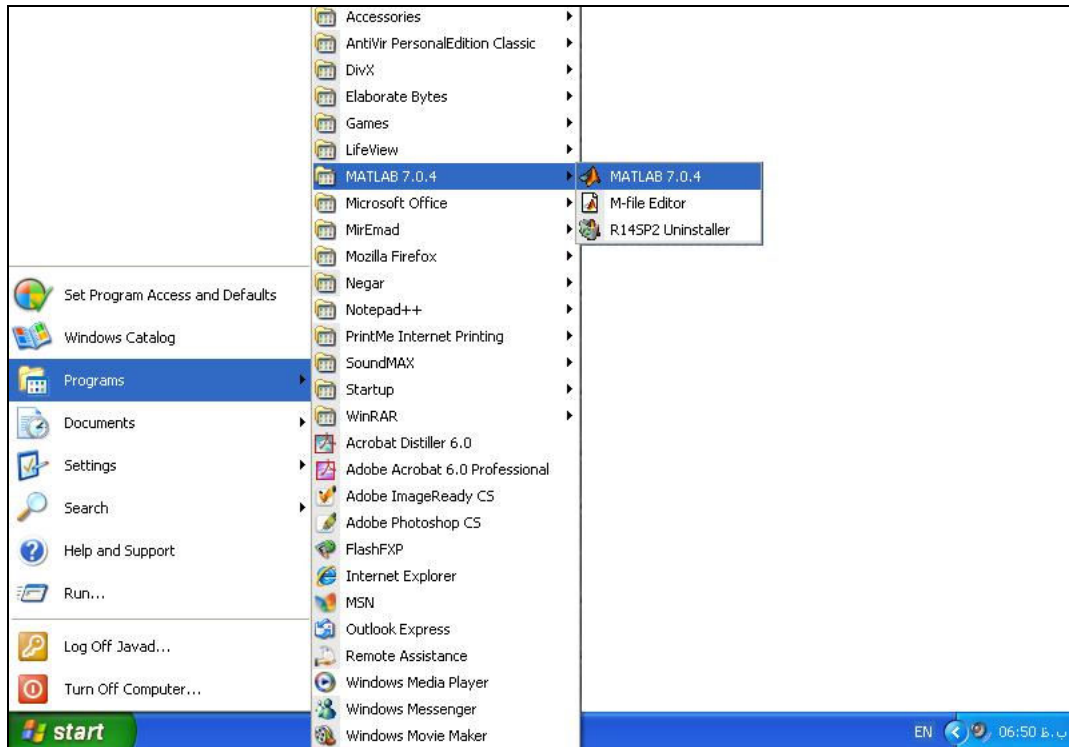
در بخش سوم کاربردهای MATLAB در تحلیل و طراحی سیستمهای کنترلی را مرور خواهیم کرد. نحوه استفاده از ابزارهای قدرتمند LTIVIEW، SISOTOOL و SIMULINK در این بخش تشریح خواهد شد.

## بخش اول - MATLAB، یک آزمایشگاه ماتریسی

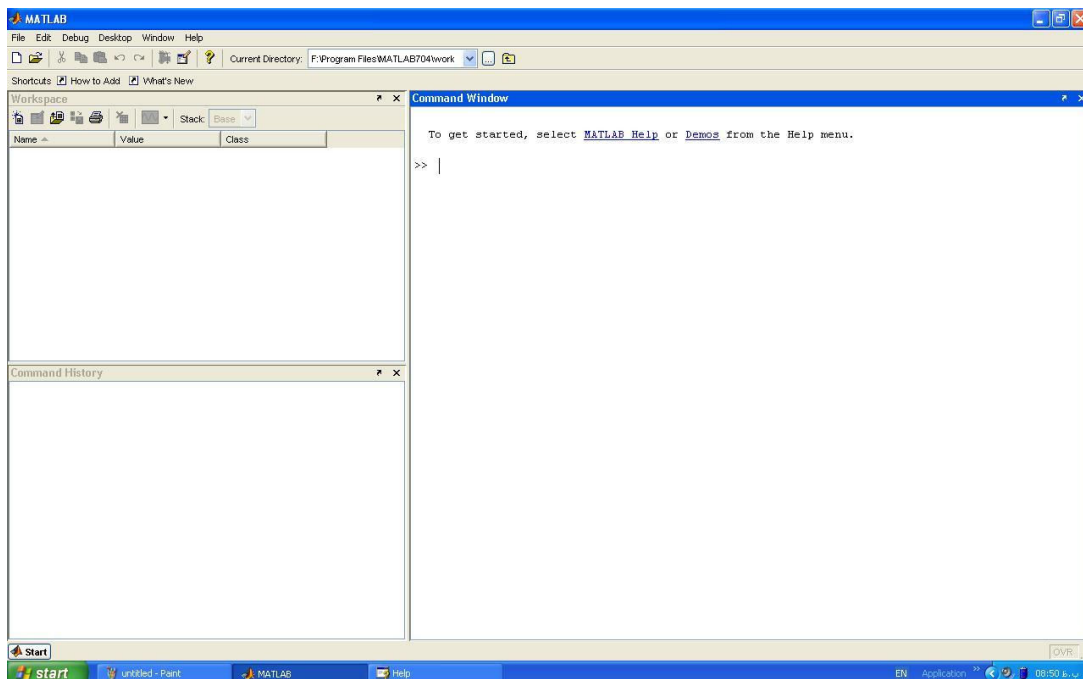
### شروع کار با MATLAB

نرم‌افزار کامل MATLAB شامل دو یا سه لوح فشرده است. نیازی به نصب تمام جعبه‌ابزارهای همراه با MATLAB نیست. این کار تنها باعث هدر رفتن فضای حافظه کامپیوتر شما می‌شود. بنابراین در هنگام نصب MATLAB، با انتخاب گزینه Custom Installation، تنها جعبه‌ابزارهای مورد نیاز خود را انتخاب کنید. نکته مهم در هنگام نصب MATLAB، لزوم داشتن یک شماره سریال تحت عنوان PLP است. چنانچه نرم‌افزار MATLAB شما نسخه اصلی (original) نیست، باید برنامه crack که به همراه آن ارائه شده را اجرا کنید و PLP که توسط آن تولید می‌شود را در هنگام نصب MATLAB به کار برید.

پس از نصب MATLAB، از طریق منوی Start باید به محیط MATLAB وارد شوید:



پس از ورود به MATLAB باید منتظر بمانید تا بارگذاری محیط آن به پایان رسیده و آماده دریافت دستورات شما شود:



محیط اصلی کاری MATLAB، محیط فرمانپذیر (Command) است. در این محیط برای اجرای یک دستور یا تابع باید عنوان آن را در تایپ کرده و کلید Enter را فشار دهید. به عنوان مثال با تایپ دستور quit یا exit و فشار کلید Enter از MATLAB خارج می‌شوید. با فشار کلید ↑ می‌توانید دستورات قبلی را مجدداً احضار کنید. به یاد داشته باشید که MATLAB نسبت به بزرگی و کوچکی حروف حساس است و باید دستورات و متغیرها را به همان صورتی که تعریف شده‌اند به کار ببرید.

روش دیگر استفاده از MATLAB، ذخیره یک مجموعه دستور در یک فایل با پسوند `.m`، تایپ نام فایل در محیط MATLAB و فشار کلید `Enter` است. به کمک این روش می‌توانید تعداد زیادی از دستورات را به صورت متوالی اجرا کنید. فایل فوق باید در زیرشاخه `work` در مسیری که MATLAB در آن مسیر نصب شده است قرار داشته باشد (مثلاً `F:\Program Files\MATLAB704\work`). در بخشی از محیط MATLAB که `Command History` نام دارد، آخرین دستوراتی که اجرا نموده‌اید نگهداری می‌شوند (با انتخاب گزینه `Command History` از منوی `desktop` می‌توانید این بخش را ببینید). با کلیک کردن روی هر دستور می‌توانید آن را مجدداً اجرا کنید.

### متغیرها در MATLAB

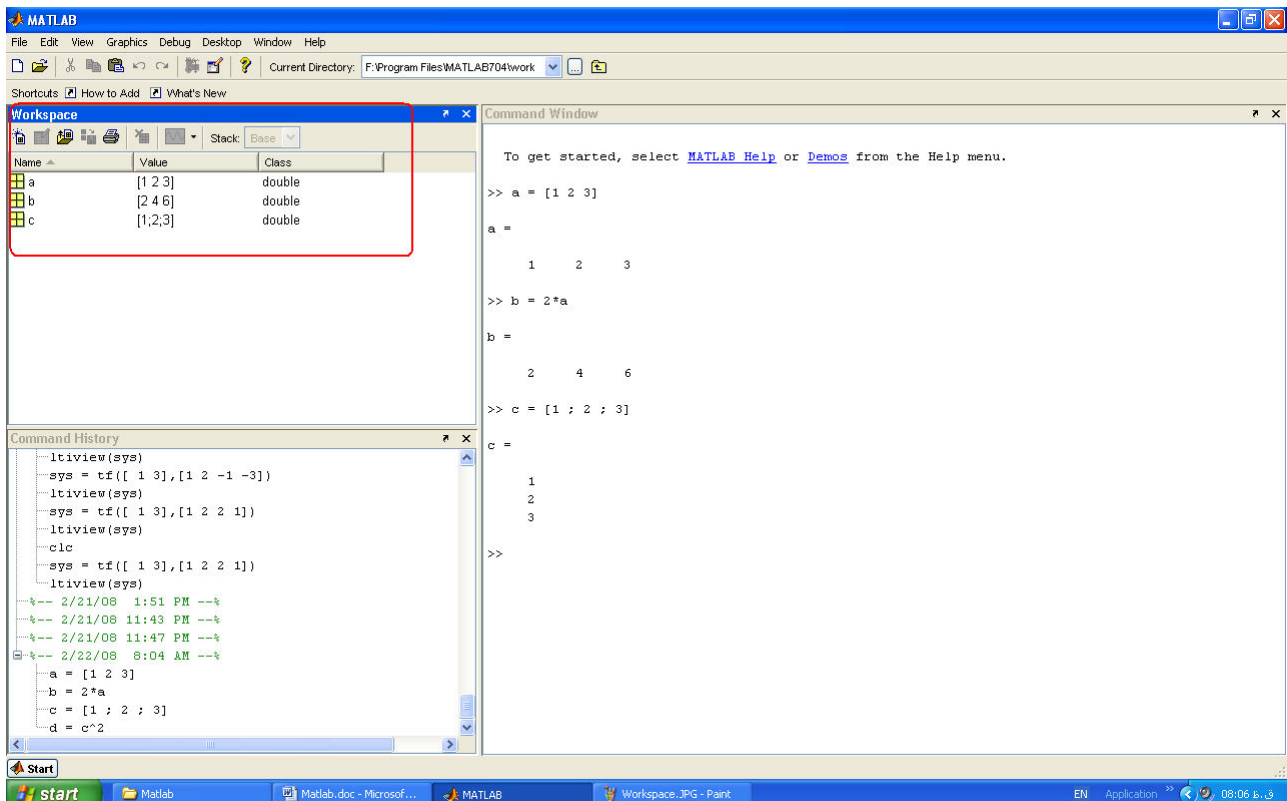
MATLAB زبانی است که تنها برای کار با ماتریسها طراحی شده است. تمام متغیرهای MATLAB از نوع ماتریس هستند. حتی متغیری که باید یک عدد را درون خود نگهداری کند به صورت یک ماتریس  $1 \times 1$  تعریف می‌شود! بنابراین تمام توابع MATLAB روی ماتریسها اجرا می‌شوند.

یکی از جنبه‌های مثبت MATLAB این است که متغیرهای آن نیازی به تعریف ندارند و در هر نقطه از برنامه می‌توانید متغیر جدیدی تعریف کنید. ویژگی جالب دیگر MATLAB آن است که بُعد (اندازه) متغیرها می‌تواند در طول برنامه عوض شود. به بیان دیگر، با اولین استفاده از یک متغیر بُعد آن مشخص می‌شود و در استفاده‌های بعدی قابل تغییر است. به عنوان مثال اگر خط فرمان زیر را به کار برید:

`a = [1 2]`

یک متغیر ماتریسی  $1 \times 2$  به نام `a` تعریف کرده‌اید. چنانچه در ادامه برنامه از دستور `a = 5` استفاده شود، بُعد متغیر `a` به یک ماتریس  $1 \times 1$  تغییر خواهد کرد.

MATLAB دارای بخشی به نام `Workspace` است که متغیرهای تعریف شده و بُعد و مقدار فعلی آنها را در آنجا نگهداری می‌کند:



چنانچه این بخش را در محیط MATLAB نمی‌بینید، از منوی `Desktop` گزینه `Workspace` را انتخاب کنید. به کمک فرمان `who` می‌توانید تمام متغیرهایی که در فضای فعلی MATLAB تعریف شده‌اند را ببینید.

برای پاک کردن تمام متغیرها از فرمان clear استفاده می‌کنیم. چنانچه می‌خواهید فقط یک متغیر را پاک کنید، باید نام آن متغیر را مقابل دستور clear ذکر کنید؛ مثلاً دستور clear c در شکل قبلی، متغیر c را پاک می‌کند. با خروج از MATLAB تمام متغیرها پاک می‌شوند. چنانچه تمایل دارید متغیرها برای اجرای بعدی MATLAB باقی بمانند، قبل از خروج از دستور save استفاده کنید. با این کار، متغیرها در فایل به نام matlab.mat ذخیره شده و با ورود مجدد به MATLAB به کمک دستور load می‌توانند بازیابی شوند.

### تعریف آرایه‌ها و ماتریسها

همانگونه که پیشتر گفته شد، MATLAB یک نرم‌افزار ریاضی ماتریس محور است و تمام متغیرهای آن باید از نوع ماتریسی تعریف شوند.

چنانچه «یک» قلم اطلاعات (مثلاً یک عدد یا یک کاراکتر) را به متغیری نسبت دهید، یک ماتریس  $1 \times 1$  برای آن متغیر در نظر گرفته می‌شود. مثلاً دستور  $X = 0.56$  یا  $C = 'a'$  متغیرهای یک بعدی X و C را با مقادیر اولیه معلوم (0.56 و کاراکتر 'a') تعریف می‌کند. برای تعریف یک آرایه سطری، باید مقادیر اولیه مورد نظرتان (که با فاصله خالی یا کاما باید از هم جدا شوند) را مابین دو علامت [] قرار داده و به متغیر نسبت دهید. مثلاً دستور زیر را ببینید:

```
A = [1 2 3 4 5]
```

این دستور یک ماتریس  $1 \times 5$  (یک آرایه سطری  $1 \times 5$ ) به نام a تعریف کرده و مقادیر 1, 2, 3, 4, 5 را به عناصر آن نسبت می‌دهد.

با اجرای دستورات MATLAB، نتیجه اجرای آن نیز در محیط MATLAB نمایش داده می‌شود؛ مثلاً با اجرای دستورات بالا، نتیجه اجرا به صورت زیر نشان داده می‌شود:

```
A =
    1    2    3    4    5
```

اگر مایل نیستید نتیجه اجرای دستورات نمایش داده شود (مثلاً وقتی با ماتریسهای مفصل مانند یک تصویر کار می‌کنید) می‌توانید در انتهای دستور از علامت ; (سمی کالن) استفاده کنید.

چنانچه مایل باشید آرایه را به صورت ستونی تعریف کنید، باید بین عناصر آرایه به جای فاصله خالی یا کاما از علامت ; استفاده کنید. این علامت نشانه ایجاد یک سطر جدید در ماتریس است. مثلاً دستور

```
A = [1 ; 2 ; 3 ; 4 ; 5]
```

باعث می‌شود یک ماتریس  $5 \times 1$  (یک آرایه ستونی  $5 \times 1$ ) به صورت زیر ایجاد شود:

```
A =
    1
    2
    3
    4
    5
```

برای تعریف یک ماتریس، باید عناصر آن ماتریس را بین علامتهای [] با شروع از سطر اول ماتریس وارد کرده و برای ایجاد سطر جدید از علامت ; استفاده کنید. مثلاً دستور

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

یک ماتریس  $3 \times 3$  به صورت زیر ایجاد می‌کند:

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

برای دسترسی به یکی از خانه‌های ماتریس، باید شماره سطر و ستون آن را بین دو علامت ( ) قرار دهید. مثلاً اگر بعد از تعریف ماتریس بالا، مقدار ذخیره شده در سطر ۲ و ستون ۳ ماتریس را بخواهید (در MATLAB شماره سطرها و ستونها از یک آغاز می‌شود)، کافی است دستور  $a(2,3)$  را تایپ کنید تا MATLAB به صورت زیر مقدار آن عنصر را به شما نمایش دهد:

```
ans =
    6
```

MATLAB یک متغیر پیش فرض به نام ans (مخفف answer) دارد که هرگاه خروجی یک دستور به متغیری نسبت داده نشده باشد، به صورت پیش فرض مقدار خروجی آن دستور را در این متغیر قرار می دهد.

چنانچه آرایه شما از عناصری با فواصل معلوم و مساوی تشکیل شده باشد، به کمک عملگر : می توانید آن آرایه یا ماتریس را تعریف کنید و دیگر نیازی به تعریف تمام عناصر نیست. شکل زیر را ببینید:

The screenshot shows the MATLAB environment. The Command Window displays the following commands and their outputs:

```

>> k = 1 : 10
k =
    1     2     3     4     5     6     7     8     9    10

>> k(2:5)
ans =
     2     3     4     5

>> k(4:end)
ans =
     4     5     6     7     8     9    10

>> |
  
```

The Workspace window shows variables 'ans' and 'k' with their values and classes. The Command History window shows the sequence of commands entered.

دستور  $k = 1:10$  یک آرایه ۱۰ عنصری با مقادیر ۱ تا ۱۰ تعریف می کند. اگر بخواهیم فاصله عناصر آرایه  $0.5$  باشد باید از دستور زیر استفاده کنیم:

$k = 1 : 0.5 : 10$

کاربرد کلی عملگر : در تعریف آرایه ها به شکل زیر است:

مقدار آخرین عنصر آرایه : فاصله عناصر آرایه : مقدار اولین عنصر آرایه = نام آرایه

چنانچه بخواهیم با بخشی از یک آرایه یا ماتریس کار کنیم، می توان از عملگر : استفاده کرد. در شکل بالا دستور  $k(2:5)$  عناصر ۲ تا ۵ آرایه  $k$  و دستور  $k(4:end)$  عناصر ۴ تا انتهای آرایه  $k$  را در محیط MATLAB به نمایش درمی آورد.

چند نمونه از کاربردهای عملگر : در نمایش زیرماتریسها را در شکل زیر می بینید:

The screenshot shows the MATLAB interface with the following content:

**Workspace:**

Name	Value	Class
a	[1 2 3; 4 5 6; 7 8 9]	double
ans	[3;6;9]	double

**Command Window:**

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

>> a(1,2)
ans =
     2

>> a(1:2,2:3)
ans =
     2     3
     5     6

>> a(1,:)
ans =
     1     2     3

>> a(:,3)
ans =
     3
     6
     9

>> |
```

**Command History:**

```
subplot(2,2,4), imshow(I3), title('cell')
2/16/08 12:12 AM -->
clear
clc
k = 1 : 10;
clc
k = 1 : 10
k(2:5)
k(4:end)
a = [1 2 3 ; 4 5 6 ; 7 8 9]
clear
clc
a = [1 2 3 ; 4 5 6 ; 7 8 9]
a(1,2)
a(3,2)
a(1,:)
a(:,3)
```

دستور  $a(1,:)$  تمام عناصر موجود در سطر ۱ و دستور  $a(:,3)$  تمام عناصر موجود در ستون ۳ را نمایش می دهد.

عملگر  $'$ : این عملگر ترانپاده (Transpose) یک ماتریس را ایجاد می کند:

The screenshot shows the MATLAB interface with the following content:

**Workspace:**

Name	Value	Class
a	[1 2 3; 4 5 6; 7 8 9]	double
b	[1 4 7; 2 5 8; 3 6 9]	double

**Command Window:**

```
To get started, select MATLAB Help or Demos from the Help menu.

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

>> b = a'
b =
     1     4     7
     2     5     8
     3     6     9

>> |
```

**Command History:**

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

دستور **reshape**: به کمک این دستور می‌توانید به ماتریسی که قبلاً تعریف شده، بعدی جدید بدهید:

```

MATLAB
File Edit Debug Desktop Window Help
Current Directory: F:\Program Files\MATLAB704\work
Shortcuts How to Add What's New

Workspace
Name Value Class
M [1 2 3; 4 5 6; 7 8 9] double
r1 [1 4 7 2 5 8 3 6 9] double
r2 [1 2 3 4 5 6 7 8 9] double

Command Window
>> M = [1 2 3 ; 4 5 6 ; 7 8 9]

M =

     1     2     3
     4     5     6
     7     8     9

>> r1 = reshape(M,1,9)

r1 =

     1     4     7     2     5     8     3     6     9

>> r2 = reshape(M',1,9)

r2 =

     1     2     3     4     5     6     7     8     9

>>

Command History
2/15/08 8:50 PM --
v = rand(1,50);
figure,stem(v)
clc
clear
clc
M = [1 2 3 ; 4 5 6 ; 7 8 9]
r1 = reshape(M,1,9)
r2 = reshape(M,9,1)
r3 = (reshape(M,9,1))'
clc
M = [1 2 3 ; 4 5 6 ; 7 8 9]
r1 = reshape(M,1,9)
r2 = reshape(M',1,9)

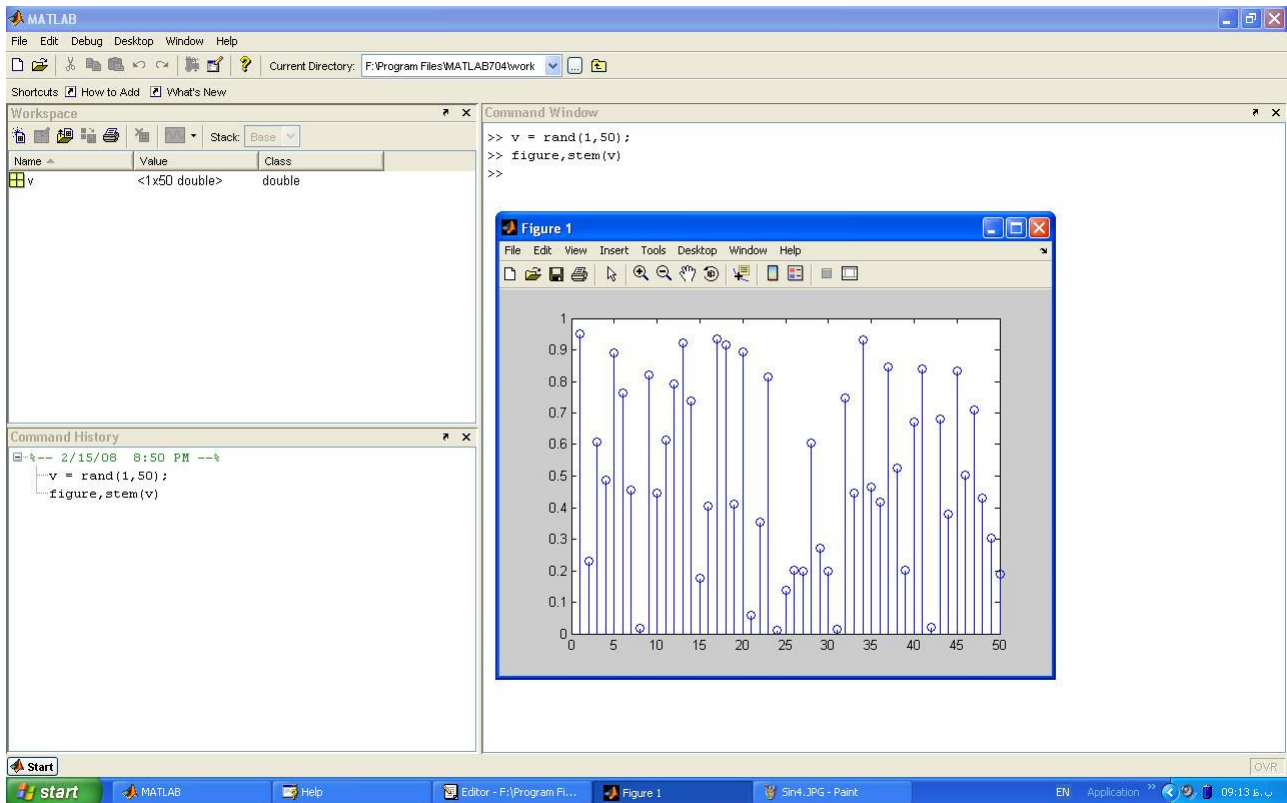
```

دستور **reshape** اول، ماتریس  $M$  که  $3 \times 3$  است را به ماتریس (آرایه)  $r1$  با بُعد  $1 \times 9$  تبدیل می‌کند. با این کار شما می‌توانید با کنار هم گذاشتن ستونهای ماتریس یک آرایه ایجاد کنید. دستور **reshape** دوم، ستونهای ترانزاده ماتریس  $M$  (که در واقع سطرهای ماتریس  $M$  هستند) را کنار هم گذاشته و ماتریس (آرایه)  $r2$  با بُعد  $1 \times 9$  را می‌سازد.

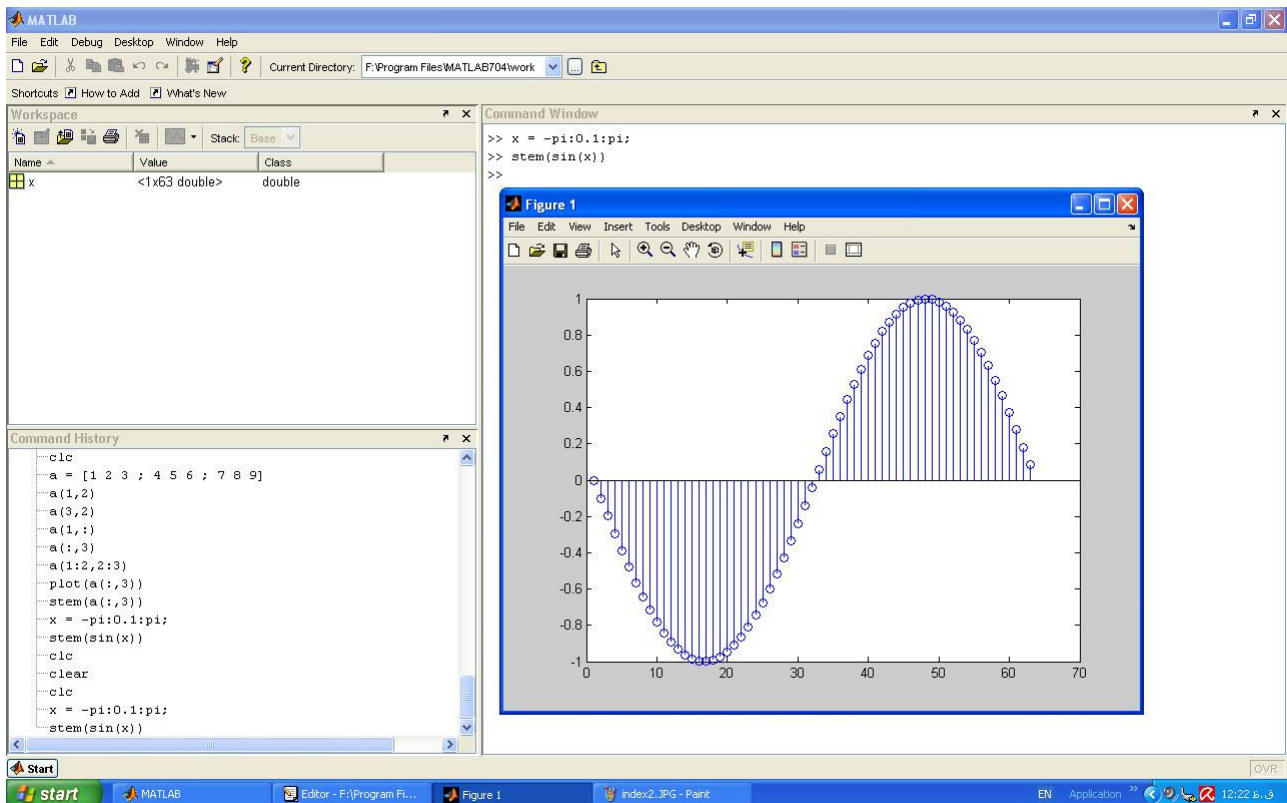
### دستورهای ترسیمی

یکی از نقاط قوت مهم MATLAB، توانایی ترسیم توابع و نتایج محاسبات با ساده‌ترین دستورات است. در زیر به مهمترین دستورات ترسیمی MATLAB می‌پردازیم.

دستور **stem**: به کمک این دستور می‌توانید مقادیر یک آرایه را به صورت گسسته نمایش دهید:



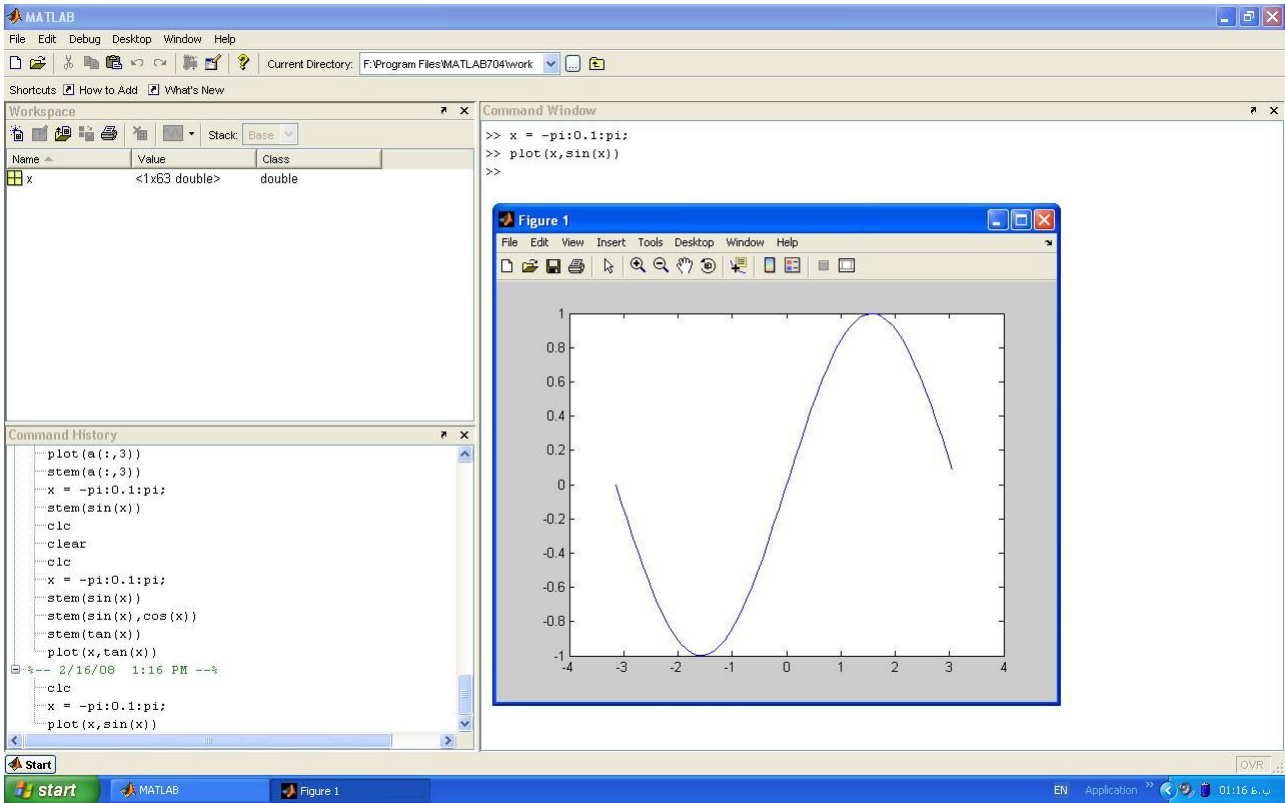
دستور `v = rand(1,50)` یک ماتریس  $1 \times 50$  با مقادیر تصادفی ایجاد کرده و در متغیر `v` ذخیره می کند. دستور `figure` یک شکل خالی ایجاد می کند که می توان در آن هر ترسیمی انجام داد. دستور `stem(v)` مقادیر بردار `v` را به صورت گسسته نمایش می دهد.



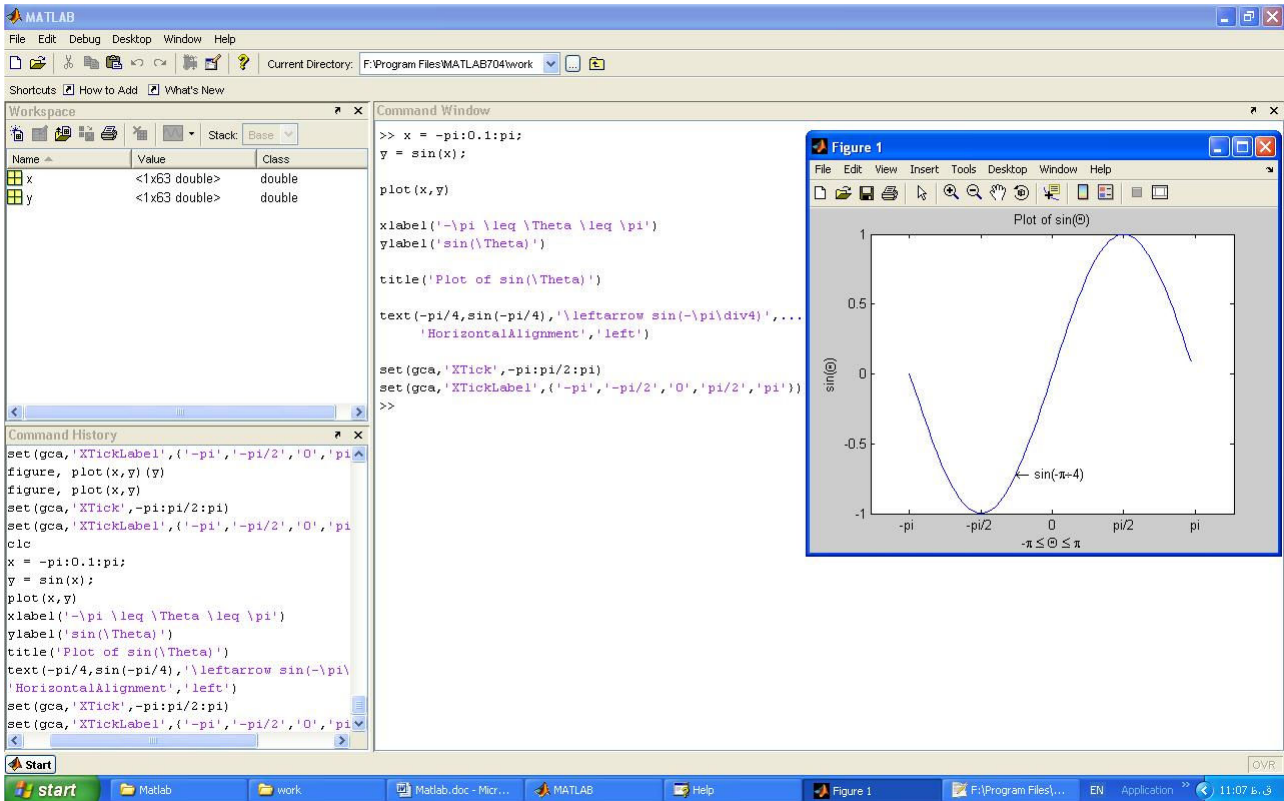


دستور اول یک بردار از  $-\pi$  تا  $+\pi$  با فواصل  $0.1$  ایجاد کرده و مقادیر آن (که ۶۳ عدد هستند) را در آرایه  $x$  ذخیره می کند. نتیجه اجرای دستور  $stem(\sin(x))$  را در شکل بالا مشاهده می کنید.

**دستور plot:** متغیرهای MATLAB همگی ماتریسی و گسسته هستند. به کمک دستور  $plot$  می توان این مقادیر گسسته را به شکل پیوسته نمایش داد. دستور  $plot(x,y)$  مقادیر بردار  $y$  را بر حسب  $x$  به صورت پیوسته نمایش می دهد:

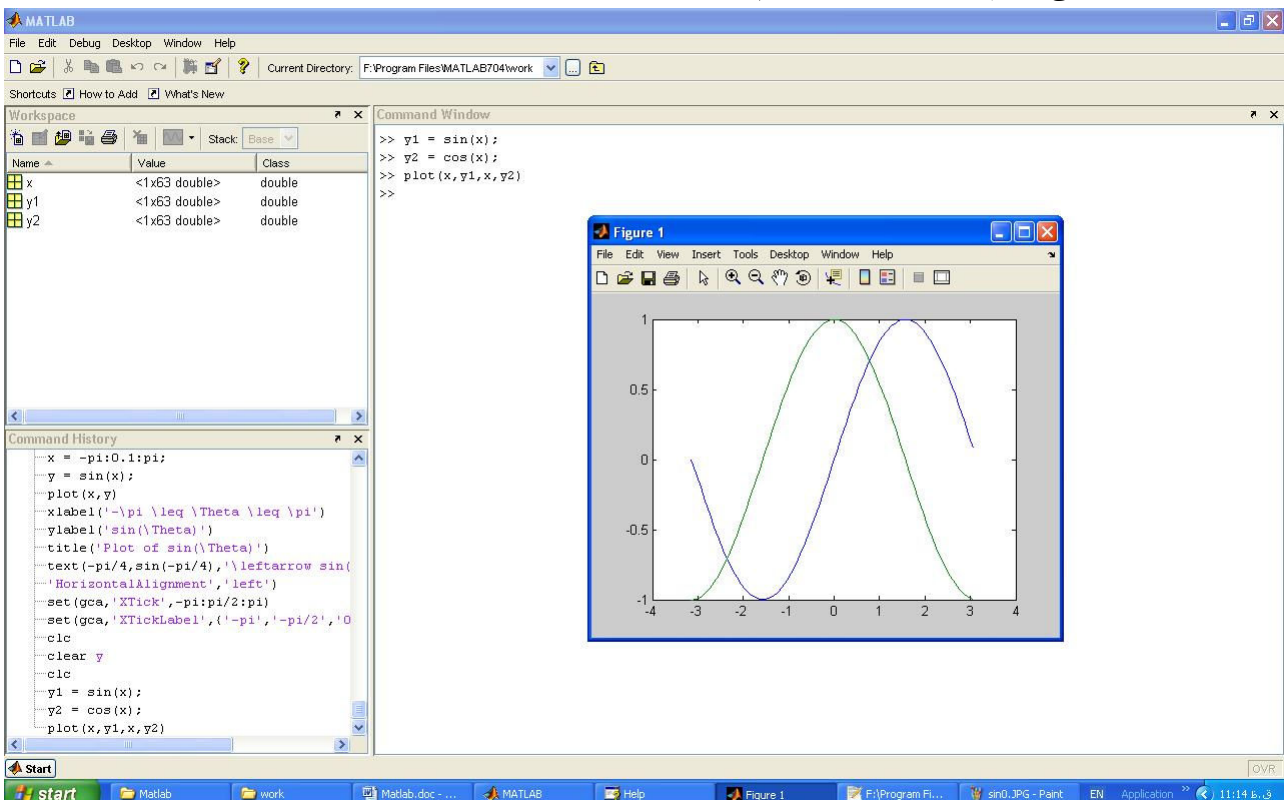


با اجرای دستورات ساده می توان جزئیات بیشتری در شکل رسم شده را نشان داد. دستور ('نام شکل' title نامی که در دستور آمده را بالای شکل نمایش می دهد. دستورات ('نام محور x' xlabel) و ('نام محور y' ylabel) نامهای ذکر شده را در کنار محورهای مربوطه نمایش می دهد. دستور (بردار فواصل set(gca,'XTick', نقاطی که باید در محور x تیک زده شوند را مشخص می کند. برای تیک زدن محور y باید از برچسب YTick استفاده کنیم. دستور (نام تیکها set(gca,'XTickLabel', به تیکهایی که به کمک دستور قبل مشخص شده، نامهای مورد نظر کاربر را اختصاص می دهد. برای نامگذاری تیکهای محور y از برچسب YTickLabel استفاده می کنیم. دستور ('متن' text(x,y) متن مشخص شده را در مختصات (x,y) شکل نمایش می دهد. دستور axis([x-min x\_max y\_min y\_max]) مقیاس بندی خودکار محورها را به دلخواه کاربر تغییر می دهد. اثر اجرای این دستورات اضافه را در شکل زیر می بینید:

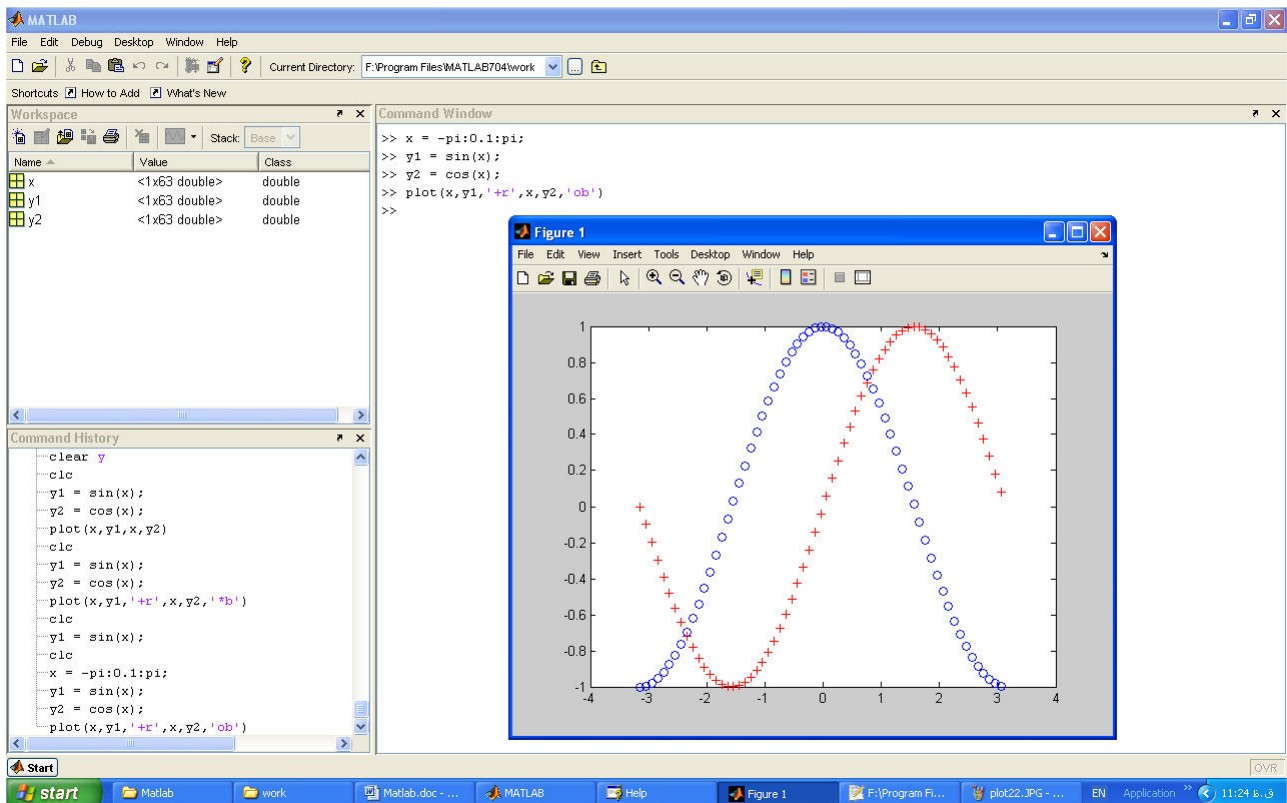


اگر دستوری آن قدر بزرگ باشد که در یک خط جا نشود، با تایپ سه نقطه در انتهای خط، ادامه دستور را در خط بعد پی می گیریم (دستور text را در شکل بالا ببینید). به علاوه می توان چند دستور را در یک خط قرار داد. برای این کار دستورات را با نشانه ؛ یا ، از هم جدا کنید.

به کمک دستور plot می توان چند نمودار را در کنار هم روی یک شکل نمایش داد.



به کمک گزاره‌های ترسیمی می‌توان نحوه نمایش شکل را تغییر داد؛ مثلاً دستور  $\text{plot}(x,y,'+')$  از نشانه + برای رسم شکل استفاده می‌کند. دستور  $\text{plot}(x,y,'r')$  شکل را به رنگ قرمز رسم می‌کند. از این تکنیک می‌توان برای متمایز کردن چند نمودار که روی یک شکل رسم شده استفاده کرد (گزاره‌های رنگی عبارتند از: 'r' برای رنگ قرمز، 'g' برای رنگ سبز، 'b' برای رنگ آبی و 'w' برای رنگ سفید).



دستور  $\text{plot}(x,y1,'+r',x,y2,'ob')$  نمودار  $y1$  برحسب  $x$  را به رنگ قرمز و با علامت + و نمودار  $y2$  برحسب  $x$  را به رنگ آبی و با علامت o نمایش می‌دهد.

### ایجاد m-file

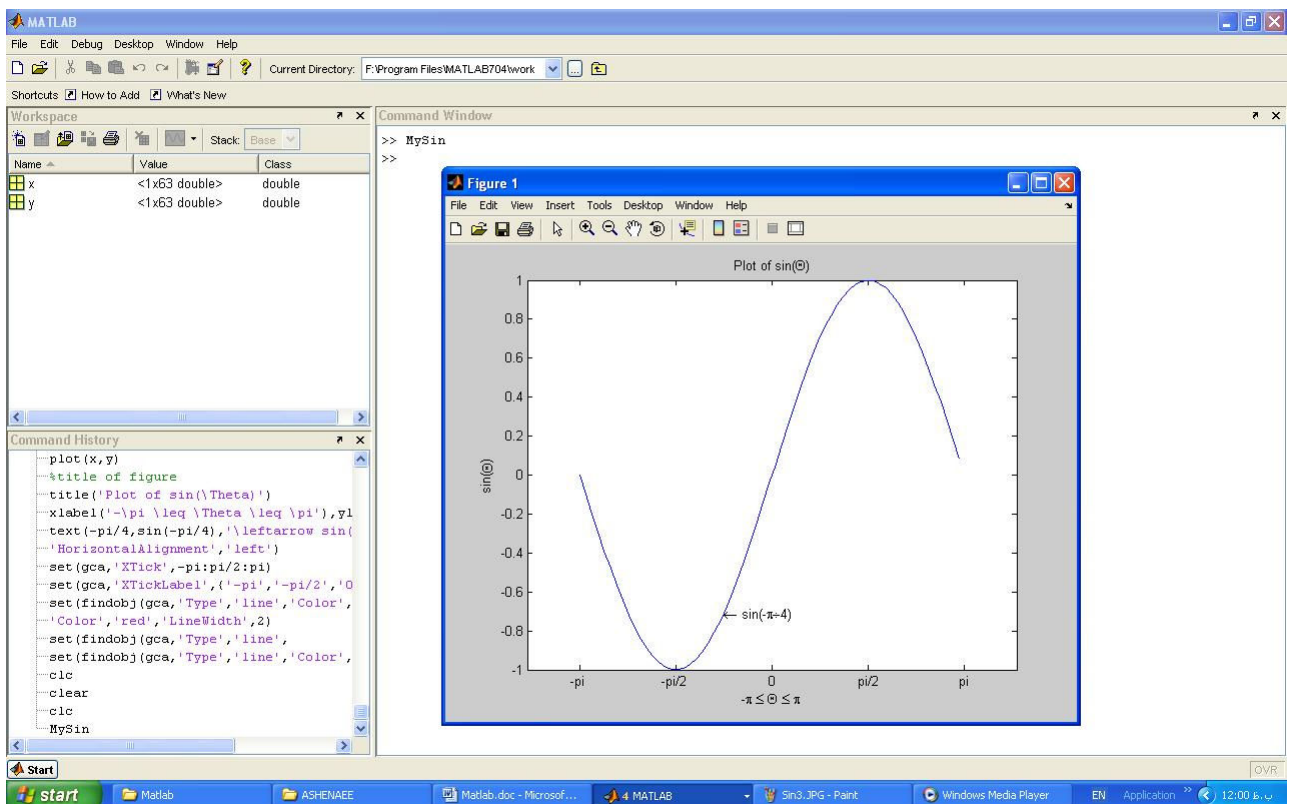
وقتی تعداد دستوراتی که کاربر می‌خواهد اجرا کند زیاد باشد، می‌تواند آنها را در یک فایل متنی بنویسد و با نام دلخواه (که مشابه نام توابع خود MATLAB نباشد) و پسوند m. در شاخه work مسیری که MATLAB در آن مسیر نصب شده ذخیره کند. برای فراخوانی این دستورات کافی است نام آن را در محیط MATLAB تایپ کنید و کلید Enter را فشار دهید:

```

Editor - F:\Program Files\MATLAB704\work\MySin.m
File Edit Text Cell Tools Debug Desktop Window Help
Stack: Base
1 x = -pi:0.1:pi;
2 y = sin(x);
3
4 plot(x,y)
5
6 %title of figure
7 title('Plot of sin(\Theta)')
8
9 xlabel('-\pi \leq \Theta \leq \pi'), ylabel('sin(\Theta)')
10
11 text(-pi/4, sin(-pi/4), '\leftarrow sin(-\pi/4)', ...
12     'HorizontalAlignment', 'left')
13
14 set(gca, 'XTick', -pi:pi/2:pi)
15 set(gca, 'XTickLabel', {'-pi', '-pi/2', '0', 'pi/2', 'pi'})

```

اگر در ابتدای خطی علامت % قرار داشته باشد، MATLAB آن خط را به عنوان دستور در نظر می گیرد. نحوه اجرای m-file بالا در محیط MATLAB به شکل زیر است:



### ماتریسهای خاص

در MATLAB توابعی وجود دارد که به کمک آنها می توان ماتریسهای خاص را به راحتی ایجاد کرد. مثلاً دستور  $a = \text{zeros}(m,n)$  یک ماتریس با  $m$  سطر و  $n$  ستون با عناصر صفر ایجاد می کند و نام آن را  $a$  می گذارد. اگر به جای دستور  $\text{zeros}$  از دستور  $\text{ones}$  استفاده کنیم، تمامی عناصر ماتریس مزبور ۱ خواهند بود. دستور  $\text{eye}(n)$  یک ماتریس مربعی واحد (عناصر روی قطر اصلی ۱ و بقیه صفر هستند) با اندازه  $n$  ایجاد می کند.