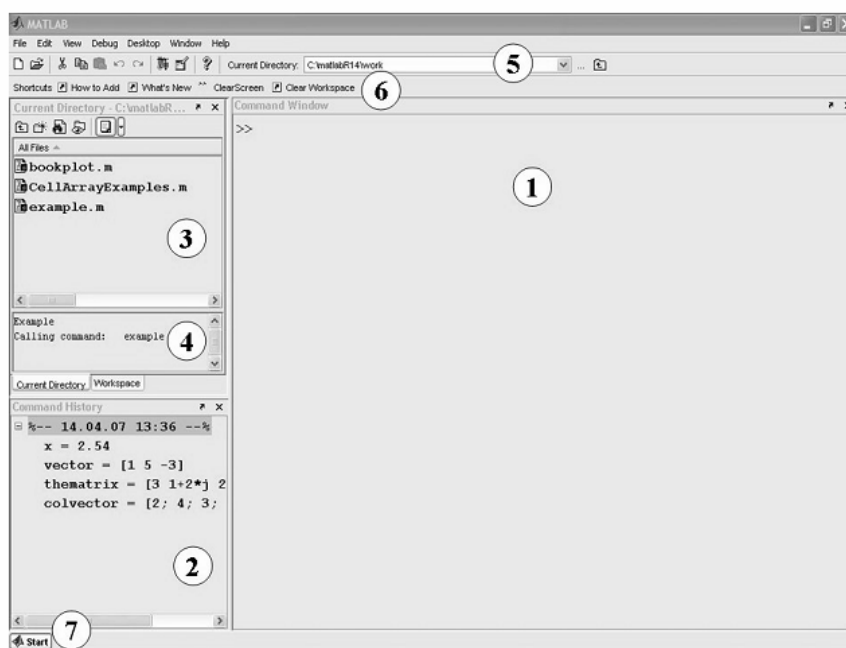


آشنایی با MATLAB

۱-۱- معرفی

نرم‌افزار MATLAB یک نرم‌افزار تجاری و یک نشان تجاری از شرکت MathWork می‌باشد. این نرم‌افزار یک سیستم مجتمع برنامه‌نویسی، شامل واسطه‌های گرافیکی و تعداد زیادی جعبه‌ابزار برای کاربردهای مختلف است و استفاده از آن در تمام زمینه‌های علمی و مهندسی به طور فزاینده‌ای در حال گسترش است.

با شروع به کار نرم‌افزار، صفحه‌ای مشابه شکل (۱-۱) ایجاد می‌شود که از بخش‌های زیر تشکیل شده است:



شکل (۱-۱): نمای ظاهری نرم‌افزار MATLAB (نسخه ۲۰۰۸)

- ۱- پنجره‌ی دستورها: در این پنجره، دستورهایی که نیاز به اجرای آنی آن‌ها داریم را وارد می‌کنیم. در این حالت دستورها بعد از وارد شدن توسط صفحه کلید با فشردن کلید Enter اجرا می‌شوند.
- ۲- پنجره‌ی دستورهای قبلی: در این بخش، دستورهایی که قبلاً در پنجره‌ی دستورها وارد و

اجرا شده، به ترتیب زمانی مرتب شده‌اند که با دوبار کلیک بر روی آنها می‌توانیم آنها را دوباره اجرا کنیم.

۳- پنجره‌ی مسیر ذخیره‌ی فعلی یا فضای کار (که در این نما ناپیداست): این پنجره، مسیر ذخیره‌ی فعلی نرم‌افزار را نشان می‌دهد و فهرستی از فایل‌ها و برنامه‌های موجود در این مسیر را فراهم می‌آورد. با رفتن به برگه‌ی دیگر، تمام متغیرهایی که اکنون در فضای کار امکان دسترسی به آنها وجود دارد، فهرست می‌شوند.

۴- پنجره‌ی اطلاعات فایل: با انتخاب یکی از فایل‌های مسیر ذخیره، مشخصات آن در این بخش نمایش داده می‌شود.

۵- نوار ابزار آیکنی با منوی انتخاب مسیر فعلی: در این نوار، تعدادی از ابزارهای پرکاربرد به صورت آیکنی قرار گرفته‌اند. همچنین می‌توان با استفاده از منوی موجود، مسیر فعلی را تغییر داد.

۶- نوار ابزار میان‌بر: در این نوار تعدادی میان‌بر برای دسترسی به ابزارها موجود می‌باشند.

۷- دکمه‌ی Start: این بخش از نرم‌افزار مشابه ویندوز، امکان دسترسی به سایر بخش‌های نرم‌افزار از جمله جعبه‌ابزارهای کاربردی، بخش Help نرم‌افزار و محیط شبیه‌سازی Simulink را فراهم می‌کند.

این فصل بخشی از مفاهیم پایه‌ی این نرم‌افزار را که برای فهم مطالب کتاب مورد نیاز است، در بر می‌گیرد. همچنین می‌توان با استفاده از سایر کتاب‌های مرتبط یا بخش help به درک عمیق‌تری از نرم‌افزار MATLAB دست یافت.

۲-۱- ماتریس‌ها

ماتریس‌ها، عناصر اصلی در نرم‌افزار MATLAB هستند و به خصوص در این کتاب بسیار مورد توجه قرار می‌گیرند. ماتریس‌ها می‌توانند در نرم‌افزار MATLAB به طرق مختلفی ایجاد شوند. ساده‌ترین راه، استفاده از عبارت زیر است:

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

توجه کنید که علامت نقطه‌ویرگول «;» در پایان هر سطر ماتریس آمده است و بیانگر اتمام

عناصر آن سطر می‌باشد. همچنین می‌توانیم ماتریس‌ها را با استفاده از توابع از پیش تعریف شده، نظیر ماتریس‌های تصادفی ایجاد کنیم:

```
>> rand(3)
ans =
0.8147 0.9134 0.2785
0.9058 0.6324 0.5469
0.1270 0.0975 0.9575
```

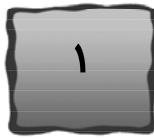
در این جا تابع rand، دستوری برای ایجاد ماتریس‌های تصادفی است. در صورت استفاده از یک آرگومان به عنوان ورودی تابع، خروجی یک ماتریس مربعی شامل مقادیر تصادفی با تعداد سطر و ستون برابر آرگومان ورودی است. ماتریس‌های غیرمربعی نیز می‌توانند با استفاده از تعیین تعداد ستون‌ها و سطرها به این ترتیب ایجاد شوند:

```
>> rand(2,3)
ans =
0.9649 0.9706 0.4854
0.1576 0.9572 0.8003
```

۳-۱- عملیات ماتریس‌ها

با استفاده از این نرم‌افزار می‌توانیم عملیات جمع، تفریق، ضرب و ترانزپوز را بر روی ماتریس‌ها انجام دهیم. برای این کار تنها لازم است که ماتریس‌هایی را که می‌خواهیم این عملیات را بر روی آن‌ها انجام دهیم، از نظر اندازه با یکدیگر سازگار باشند. به عنوان مثال می‌توانیم ماتریس c را که جمع نظیر به نظیر ماتریس‌های a و b است از طریق عبارت‌های زیر ایجاد کنیم:

```
>> a=rand(4)
a =
0.2769 0.6948 0.4387 0.1869
0.0462 0.3171 0.3816 0.4898
0.0971 0.9502 0.7655 0.4456
0.8235 0.0344 0.7952 0.6463
>> b=rand(4)
b =
0.7094 0.6551 0.9597 0.7513
0.7547 0.1626 0.3404 0.2551
0.2760 0.1190 0.5853 0.5060
0.6797 0.4984 0.2238 0.6991
>> c=a+b
c =
0.9863 1.3499 1.3985 0.9381
0.8009 0.4797 0.7219 0.7449
0.3732 1.0692 1.3508 0.9515
1.5032 0.5328 1.0190 1.3454
```



با استفاده از دستورهای موجود در MATLAB، ماتریس‌ها می‌توانند در هم ضرب شوند. البته باید توجه داشت که برای جلوگیری از بروز خطا باید تعداد ستون‌های ماتریس اول با تعداد سطرهاى ماتریس دوم برابر باشد. خروجی این عملیات، ماتریسی با تعداد سطریهایی برابر تعداد سطرهاى ماتریس اول و تعداد ستونیهایی برابر تعداد ستون‌های ماتریس دوم می‌باشد. به عنوان مثال، عبارت $e=a \times d$ به صورت زیر محاسبه می‌شود:

```
>> d=rand(4,1)
d =
0.8909
0.9593
0.5472
0.1386
>> e=a*d
e =
1.1792
0.6220
1.4787
1.2914
```

عملیات ترانهاده یک ماتریس نیز می‌تواند در MATLAB توسط علامت «'» یا آپوستروف

انجام شود:

```
>> a=rand(3,2)
a =
0.1493 0.2543
0.2575 0.8143
0.8407 0.2435
>> a'
ans =
0.1493 0.2575 0.8407
0.2543 0.8143 0.2435
```

۴-۱- عبارتها

عبارتها، مجموعه‌ای از عملگرها، توابع و متغیرهایی هستند که همیشه یک ماتریس را به عنوان خروجی، برای استفاده‌های بعدی ایجاد می‌کنند. تعدادی از این عبارتها به صورت زیر است:

```
>> a=3
a =
3
>> b=a*3
b =
9
>> eye(3)
ans =
1 0 0
```

```
0 1 0
0 0 1
```

که در آن تابع `eye()` برای ایجاد یک ماتریس واحد با تعداد سطر و ستون برابر با آرگومان ورودی استفاده می‌شود. اگر بخواهیم از نمایش خروجی دستورهای ورودی جلوگیری کنیم، کافی است که یک نقطه‌ویرگول در انتهای دستور قرار دهیم.

لازم به ذکر است که نرم‌افزار MATLAB یک نرم‌افزار حساس به نوع است و متغیرهای `A` و `a` با یکدیگر متفاوت هستند.

می‌توانیم متغیرها را از فضای کار با استفاده از دستور `clear` یا `clear all` پاک کنیم. یک شیء خاص مانند `A` می‌تواند توسط دستور `clear A` پاک شود.

۱-۵- توابع ماتریسی

تعدادی از توابع ماتریسی مفید در جدول (۱-۱) آمده است.

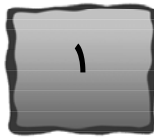
جدول (۱-۱): تعدادی از توابع مفید ماتریسی

عملکرد	تابع
ماتریس واحد	<code>eye</code>
یک ماتریس با عناصر صفر	<code>zeros</code>
یک ماتریس با عناصر یک	<code>ones</code>
ایجاد / استخراج یک ماتریس قطری	<code>diag</code>
ماتریس تصادفی	<code>rand</code>

تعدادی مثال از این توابع در دستورهایی زیر آمده است (در این جا ماتریس‌ها با استفاده از ویرگول به صورت بلوکی ساخته می‌شوند):

```
>> [eye(3),diag(eye(3)),rand(3)]
ans =
1.0000 0 0 1.0000 0.9293 0.2511 0.3517
0 1.0000 0 1.0000 0.3500 0.6160 0.8308
0 0 1.0000 1.0000 0.1966 0.4733 0.5853
```

در این مثال، به ترتیب یک ماتریس مربعی واحد (`eye(3)`)، عناصر قطر اصلی یک ماتریس مربعی واحد (`diag(eye(3))`) و یک ماتریس مربعی تصادفی (`rand(3)`) هر کدام به عنوان یک بلوک، با یکدیگر ترکیب شده و تشکیل یک ماتریس داده‌اند. یک مثال دیگر از ماتریسی که به صورت بلوکی ایجاد شده است در زیر آمده است:



```
>> A=rand(3)
A =
0.5497 0.7572 0.5678
0.9172 0.7537 0.0759
0.2858 0.3804 0.0540
>> B = [A, zeros(3,2); zeros(2,3), ones(2)]
B =
0.5497 0.7572 0.5678 0 0
0.9172 0.7537 0.0759 0 0
0.2858 0.3804 0.0540 0 0
0 0 0 1.0000 1.0000
0 0 0 1.0000 1.0000
```

در این مثال نیز ماتریس A با سه ماتریس دیگر ترکیب شده و ماتریس B را ایجاد کرده‌اند. باید توجه کرد که در انتهای بلوک‌هایی که با هم تشکیل یک بلوک افقی را می‌دهند. از علامت «;» استفاده شده است.

۱-۶- عبارتهای شرطی if و switch

معمولاً در یک تابع نیاز است که بر حسب شرایط در زمان اجرا، برنامه به چند شاخه تقسیم شود. نرم‌افزار MATLAB به این منظور ساختارهایی را مشابه تمام زبان‌های برنامه‌نویسی، پیشنهاد می‌کند. در زیر مثالی آورده شده است که عمده‌ی ویژگی‌های عبارتهای if را شامل می‌شود:

```
x= - 1
if x==0
    disp('Bad input!')
elseif max(x) > 0
    y = x+1;
else
    y = x^2;
end
```

در این مثال، ابتدا به متغیر یک مقدار اولیه داده شده است. سپس این مقدار اولیه برای این‌که مشخص شود که بزرگتر، کوچکتر یا برابر صفر است وارد یک عبارت شرطی می‌شود. قرارگرفتن

مقدار متغیر در هر کدام از محدوده‌های مورد بررسی، باعث عملکرد متفاوتی می‌شود. اگر تعداد گزینه‌های شرطی زیاد باشد، بهتر است که از عبارت switch به منظور ایجاد انشعاب در برنامه استفاده کنیم. به مثال زیر توجه کنید که در آن مقدار متغیر units با سه مقدار پیش‌فرض مقایسه شده و بسته به موقعیت، پیغام خاصی نمایش داده می‌شود. اگر مقدار متغیر با هیچ‌کدام از مقادیر مورد نظر مطابقت نداشته باشد نیز پیغامی برای کاربر نمایش داده می‌شود.

```
switch units
    case 'length'
        disp('meters')
    case 'volume'
        disp('cubic meters')
    case 'time'
        disp('hours')
    otherwise
        disp('not interested')
end
```

۷-۱- حلقه‌های for و while

برای بسیاری از برنامه‌ها نیاز است تا بخشی از دستورها تکرار شده و مجدداً اجرا شوند. در این جا نیز MATLAB مشابه سایر زبان‌های برنامه‌نویسی عمل می‌کند. کد زیر، نوع استفاده از عبارت‌های for/end را برای نمایش ده عدد اول از سری اعداد فیبوناچی نشان می‌دهد:

```
>> f=[1 2]
f=
1 2
>> for i=3:10;f(i)=f(i - 1)+f(i - 2);end;
>> f
f=
1 2 3 5 8 13 21 34 55 89
```

در این مثال که برای نمایش چند مقدار اول سری فیبوناچی نوشته شده است، ابتدا ماتریس f مقداردهی اولیه شده، سپس با استفاده از یک حلقه‌ی for این مقادیر محاسبه شده‌اند. نکته‌ای که در

این جا جلب توجه می کند، نوشته شدن دستورهای مربوط به حلقه در یک سطر، با استفاده از علامت ";" است. لازم به ذکر است که این کار تأثیری در روند اجرای حلقه ندارد. اعلان پایان حلقه ی for با کلمه ی کلیدی end انجام می شود و این اعلان، باعث اجرای دستورهای داخل حلقه به دفعات مورد نیاز می گردد. این مسئله در مورد حلقه ی while نیز صادق است. در بعضی موارد لازم است که عبارت ها، با توجه به شرط خاصی و نه به تعداد دفعات معینی، تکرار شوند. این کار با عبارت while انجام می شود.

```
>> x=10;while x > 1; x = x/2,end
x =
    5
x =
    2.5000
x =
    1.2500
x =
    0.6250
```

مثال دیگری برای کاربرد حلقه های for/end به صورت زیر است:

```
>> x = []; for i = 1:4, x=[x,i^2], end
x =
    1
x =
    1 4
x =
    1 4 9
x =
    1 4 9 16
```

و یا به طور معکوس:

```
>> x = []; for i = 4: - 1:1, x=[x,i^2], end
x =
    16
x =
```

```

16 9
x =
16 9 4
x =
16 9 4 1
    
```

نکته قابل ذکر در مورد مثال‌های اخیر، استفاده از عملگر دو نقطه ":" است که برای ایجاد یک دنباله از اعداد با فواصل مشخص استفاده می‌شود. در مثال اول، دستور `for i=3:10`، مقادیر ممکن برای i به ترتیب از مقدار صحیح ۳ تا مقدار صحیح ۱۰ با افزایش یک واحدی قرار داده می‌شوند و دستورهای داخل حلقه به ازای این مقادیر تکرار می‌شوند؛ یعنی بردار i به صورت دنباله‌ی ۱۰ و ... و ۴ و ۳ می‌باشد. ولی در مثال اخیر مشاهده می‌شود که یک پارامتر دیگر نیز اضافه شده است. در مثال اخیر داریم `for i = 4: - 1:1` به این معنا که مقدار i با شروع از مقدار ۴، با پله‌هایی برابر با -۱ کاهش یافته تا به مقدار ۱ برسد؛ یعنی برداری است شامل عناصر ۴، ۳، ۲ و ۱. البته لازم به ذکر است که مقدار مورد استفاده به عنوان پله‌ی کاهش‌ی یا افزایشی می‌تواند یک مقدار غیر صحیح باشد و گزاره‌هایی به صورت `i=3:-.01:10` یا `i=2:-.1:1` نیز صحیح می‌باشند.

۱-۱ عملگرهای مقایسه‌ای

عملگرهای مقایسه‌ای نرم‌افزار MATLAB در جدول (۱ - ۲) آمده است.

جدول (۱ - ۲): تعدادی از عبارتهای نسبی

عملکرد	عملگر مقایسه‌ای
کوچکتر از	<
بزرگتر از	>
کوچکتر یا مساوی با	<=
بزرگتر یا مساوی با	>=
برابر با	==
نامساوی با	~=

به تفاوت بین عملگرهای تساوی "=" و تساوی مقایسه‌ای "==" دقت کنید، عملگرهای منطقی موجود در MATLAB در جدول (۱-۳) داده شده‌اند.