

آشنایی با متلب

متلب یک محیط نرم‌افزاری برای انجام محاسبات عددی و یک زبان برنامه‌نویسی نسل چهارم است. واژه متلب هم به معنی محیط محاسبات رقمی و هم به معنی خود زبان برنامه‌نویسی مربوطه است که از ترکیب دو واژه Matrix ماتریس و Laboratory آزمایشگاه ایجاد شده است. این نام حاکی از رویکرد ماتریس محور برنامه است، که در آن حتی اعداد منفرد هم به عنوان ماتریس در نظر گرفته می‌شوند. کار کردن با ماتریس‌ها در متلب بسیار ساده است. در حقیقت تمام داده‌ها در متلب به شکل یک ماتریس ذخیره می‌شوند. حتی یک تصویر به شکل یک ماتریس سه بعدی ذخیره می‌گردد که بُعد اول و دوم آن برای تعیین مختصات نقاط و بُعد سوم آن برای تعیین رنگ نقاط استفاده می‌شود. بنابراین جای تعجب نیست که متلب مخفف عبارت آزمایشگاه ماتریس باشد.

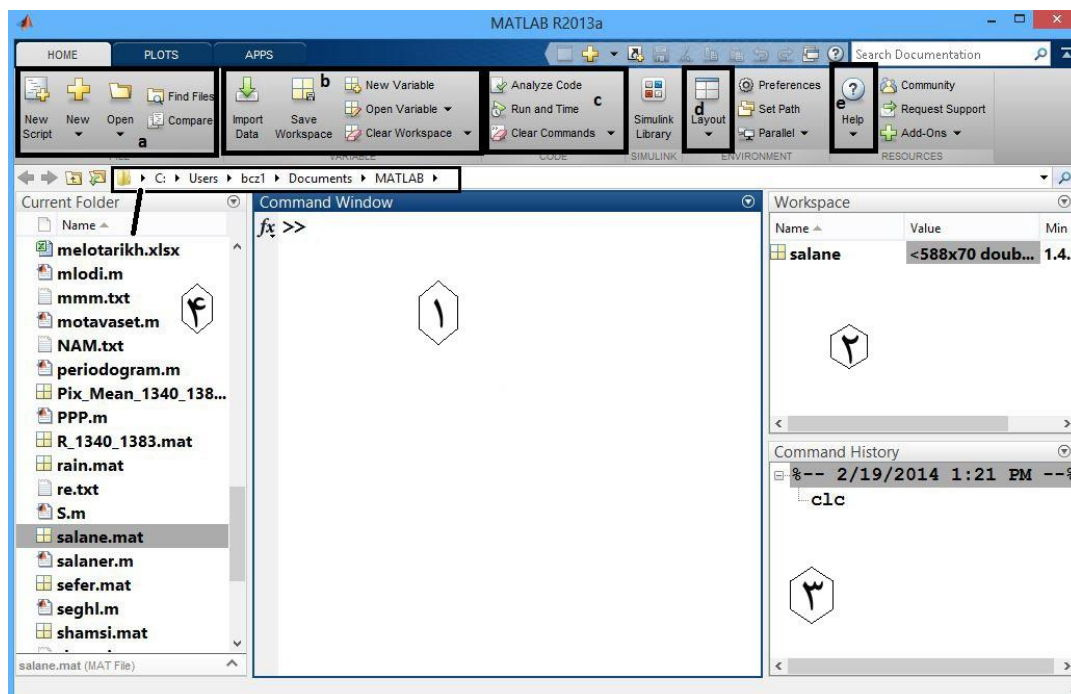
علاوه بر توابع فراوانی که خود متلب دارد، برنامه‌نویس نیز می‌تواند توابع جدید تعریف کند. ساخت رابط گرافیکی کاربر مانند دیالوگ‌هایی که در محیط‌های ویژوال مانند بیسیک و C وجود دارند، در متلب امکان‌پذیر است. این قابلیت، ارتباط بهتری را میان برنامه‌های کاربردی نوشته‌شده با متلب و کاربران برقرار می‌کند. با این وجود متلب زبانی است که کاربرد کامپیوتر در مهندسی را با کارایی بالا تضمین کرده و امکانات محاسباتی، تصویری، و برنامه نویسی را در محیطی آسان و آشنا فراهم می‌کند. کارایی متلب در مقوله هائی نظیر: محاسبات ریاضی، آنالیز داده‌ها، مدل سازی و شبیه سازی، گرافیک، و تولید نرم افزار (حتی برای محیط ویندوز) به اثبات رسیده است.

پس از اینکه متلب را نصب کردید با کلیک بر روی آیکون آن پنجره‌های متلب به شکل ۱ ظاهر می‌گردد. این صفحه از قسمت های مختلفی نظیر

- [1] Command window
- [2] Workspace
- [3] Command history
- [4] Current folder

تشکیل شده است. همچنین دارای سه سر برگ به شرح زیر می‌باشد:

- [1] Home
- [2] Plot
- [3] Apps



شکل ۱: نمای از پنجره‌های متلب

در اینجا هر یک از این قسمت ها به طور خلاصه شرح داده می‌شوند:

الف- command window

اولین و در واقع مهمترین پنجره متلب صفحه command window می‌باشد. تمام محاسبات و عملیات در این پنجره اجرا می‌گردد.

ب - workspace

تمام محاسباتی که در صفحه command window اجرا می‌گردد نتایج آن در این پنجره ظاهر می‌گردد. به عبارتی دیگر لیستی از نتایج تمام متغیرهای که اخیرا اجرا شده و یا محاسبه گردیده است را نشان می‌دهد.

ج- Command history

در این پنجره تمامی دستوراتی که در صفحه command window اجرا گردیده در این پنجره ذخیره می‌گردد. برای مثال در یک روز خاص توابع مربوط به یک عملیات در command window اجرا گردیده است، لذا با مراجعه به این پنجره می‌توان روز و ماه و سال این دستور را در این پنجره مشاهده کرد. به عبارتی دیگر این پنجره تاریخچه دستورات را نشان می‌دهد.

د- Current folder

چنانچه بخواهیم با فایل های موجود در یک فولدر خاص، درون ویندوز، کار کنیم، ابتدا باید آدرس آن را به صورت یک مسیر، به مسیرهای تعریف شده در متلب، اضافه کنید. برای افزودن یک مسیر به متلب، می‌توانیم از دستور cd استفاده کنیم.

در صورتی که در دستورات save و load، هیچ آدرسی را مشخص نکنیم، save و یا load کردن فایل ها، در

فولدر مشخص شده در پنجره Current Folder ، صورت می گیرد و فایل ها در آن محل، ذخیره و یا بازخوانی می شوند. همان طوری که اشاره شد روش ساده برای تغییر این فولدر، کلیک بر روی بالای پنجره و تغییر مسیر به صورت دستی می باشد، اما ممکن است در برنامه ای مجبور شویم که فایل هایی را در فولدرهای مختلف save و یا load کنیم، بنابراین باید این عمل را با دستوراتی در وسط برنامه اجرا کنیم. دستور لازم برای تغییر فولدر مربوط به Current Folder ، دستور cd می باشد. به مثال زیر توجه کنید :

فرض کنید بخواهیم آدرس فولدر مربوط به Current Folder را به e:\Project تغییر دهیم، می نویسیم:

```
cd e:\ Project
```

نتیجه:

چنانچه به آدرس بالای پنجره Current Folder نگاه کنید، مشاهده خواهید کرد که آدرس آن به آدرس e:\ Project تغییر یافته است.

سربرگ های متلب

Home-۱

این یکی از مهمترین سربرگ های متلب می باشد که خود از زیر منوهای زیر تشکیل گردیده است:

۱(a)- این بخش بیشتر مربوط به فراخوانی داده ها (open)، باز کردن پنجره جدید (new)، پیدا کردن عبارت یا کلمه مورد نظر (find file)، و بالاخره باز کردن پنجره اسکریپت^۱ (New script) در این بخش انجام می گیرد.

۲ (b)- در این سربرگ ها برای فراخوانی داده ها (import data)، ذخیره کردن فایل های که در صفحه workspace وجود دارند با فرمت متلب (save workspace)، پاک کردن متغیرهای که در workspace قرار دارند این سربرگ همانند دستور clear عمل می کند (clear workspace).

۳ (c)- در این کادر برنامه های را که به صورت m-file نوشته شد، گزارش کلی را در این خصوص با کلیک بر روی این عبارت به شما می دهد (analyze code)، برای ران کردن برنامه همراه با زمان اجرای آن (run and time) و بر پاک کردن پنجره command window بکار می رود این سر برگ مشابه با clc عمل می کند (clear command).

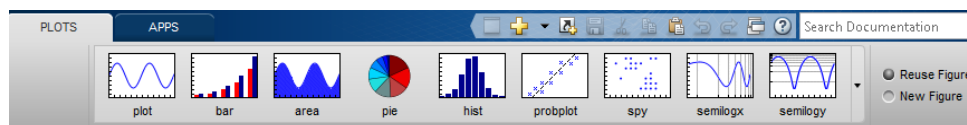
۳ (d)- این سربرگ بیشتر برای تنظیم پنجره های متلب به کار می رود. به عبارتی برای نمایش پنجره های متلب با فرمت مختلف کاربرد دارد.

۳ (e)- این سربرگ (help) یکی از مهمترین منوهای متلب می باشد. این منو بهترین استاد برای یاد دادن متلب می باشد به طوری که در بین نرم افزارهای مربوط به کد نویسی قوی ترین help را دارد.

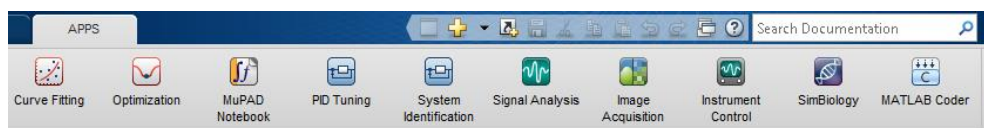
Plot-۲

در این بخش شما می توانید نمودهای مختلف را برای ترسیم انتخاب کنید.

^۱ - بخش مربوط به اسکریپت در فصل چهارم همین کتاب به طور مفصل شرح داده شده است.



Apps-۳



در نرم افزار متلب به سه شکل می توان متغیر ایجاد کرد:

نکته: تمام دستورات و محاسبات در پنجره command window انجام می گیرد.

– داده های آماده ای که در فایل های نظیر txt, Excel قرار دارند.

وارد کردن داده ها از اکسل و صادر کردن آنها

وارد کردن داده ها از اکسل با دستور xlsread انجام می شود. وقتی دستور اجرا شد، داده ها از صفحه گسترده به صورت یک آرایه به یک متغیر تخصیص داده می شود. ساده ترین شکل دستور xlsread عبارتست از:

`Xlsread('file name.xlsx')` = نام متغیر

در دستور بالا file name نام فایل اکسل مورد نظر می باشد. دایرکتوری فایل اکسل فراخوانی شده درای دو حالت است یا در دایرکتوری جاری قرار داشته باشد، یا در دایرکتوری فهرست شده در مسیر (search path) قرار گرفته باشد. این بدین معنی می باشد که ابتدا باید فایل اکسل را در مسیر current folder قرار دهید سپس از دستور بالا استفاده کنید (برای قرار گرفتن فایل مورد نظر از دستور cd استفاده کنید؛ برای مثال `.(cd:/drive/file name`

نکته: توجه کنید که xlsx پسوند فایل اکسل می باشد. این پسوند در ورژن های مختلف متفاوت است. بنابراین متناسب با اکسل های مختلف این پسوند فرق خواهد کرد. برای مثال پسوند فایل ها تا نسخه ۲۰۰۳، xls می باشد اما پسوند فایل ها تا نسخه ۲۰۰۷ با xlsx می باشد.

همان طوری که اشاره گفته شد دستور بالا زمانی استفاده می شود که فایل مورد نظر در مسیر current folder قرار گرفته باشد. اما وقتی بخواهیم فایل اکسلی را فراخوانی کنیم بدون اینکه نیاز باشد فایل را در مسیر current folder قرار داده باشیم از دستور دیگر که در ادامه اشاره خواهیم کرد استفاده می کنیم. برای مثال اگر فایل اکسل با نام able در درایو e و در فایل data باشد برای فراخوانی آن به متلب از دستور زیر استفاده می کنیم:

`Nam=xlsread ('e: /data/able.xlsx');`

	A	B	C	D	E	F	G	H
1	1912	45	1961	45	1937	12	1986	12
2	1913	27	1962	27	1938	16	1987	16
3	1914	23	1963	23	1939	23	1988	23
4	1915	34	1964	34	1940	45	1989	45
5	1916	12	1965	12	1941	27	1990	27
6	1917	16	1966	16	1942	23	1991	23
7	1918	23	1967	23	1943	34	1992	34
8	1919	45	1968	45	1944	12	1993	12
9	1920	27	1969	27	1945	16	1994	16
10	1921	23	1970	23	1946	23	1995	23
11	1922	34	1971	34	1947	45	1996	45
12	1923	12	1972	12	1948	27	1997	27
13								

شکل ۱-۳

حالا اگر فایل اکسل بیش از یک برگه داشته باشد، داده ها از اولین برگه وارد می شود. وقتی فایل اکسل دارای چندین برگه باشد، دستور `xlsread` می تواند برای وارد کردن داده ها از یک برگه معین استفاده شود. آنگاه شکل دستور عبارتست از:

```
>>Nam=xlsread('e:/able.xlsx','sheet2');
```

نام برگه به صورت یک رشته تایپ می شود. دستور بالا دارد از درایو e در فایل به اسم able از sheet2 یک سری داده را فراخوانی می کند.

گزینه دیگر برای وارد کردن فقط بخشی از داده های در صفحه گسترده می باشد. این کار با تایپ یک آرگومان دیگر انجام می شود.

```
>>Nam=xlsread('e:/able.xlsx','sheet2','A1:F4');
```

دستور بالا داده ای که در شکل - هایلایت (highlight) شده را فراخوانی می کند.

دستور فراخوانی داده از متلب به اکسل

فرض کنید داده های مشابه با شکل - در پنجره متلب داشته باشیم و بخواهیم آن را در فایل اکسل مورد نظر و در یک sheet خاصی ذخیره کنیم (در اینجا قصد داریم داده های فایل b در متلب را در فایل به اسم able و sheet3 در درایو e و فایل data ذخیره کنیم) با استفاده از دستور زیر داده ها به فایل اکسل منتقل داده خواهد شد:

```
>>Xlswrite('e:/data/able.xlsx', b,'sheet3','A1:E5');
```

The screenshot shows a MATLAB window titled 'Variables - b'. It displays a 5x5 matrix 'b' with the following values:

	1	2	3	4	5
1	17	24	1	8	15
2	23	5	7	14	16
3	4	6	13	20	22
4	10	12	19	21	3
5	11	18	25	2	9

شکل --

در این دستور داده های فایل b که در پنجره command window می باشد در فایل able که در درایو e و در فایل data قرار دارد در sheet3 و قسمت A1 تا E5 یعنی اندازه فایل b (۵ ردیف و ۵ ستون) ذخیره خواهد کرد.

The screenshot shows Microsoft Excel with the data from the MATLAB window. The data is located in the range A1:E5 of sheet3. The values are:

	A	B	C	D	E
1	17	24	1	8	15
2	23	5	7	14	16
3	4	6	13	20	22
4	10	12	19	21	3
5	11	18	25	2	9

شکل --

ذخیره یک فایل از متلب در قالب txt

```
>> dlmwrite('e:/nam.txt','a','delimiter','\t','precision','%.2f','newline','pc')
```

در این دستور نام فایل می باشد که به آن اسم فایل txt ساخته می شود. a اسم ماتریس متلب، delimiter این دستور نحوه جداسازی درایه های را مشخص، اگر کلمه بعد از آن t باشد درایه ها با tab از هم جدا می شود. در precision دقت ارایه شده در فایل txt را مشخص می کند که به کلمه بعد از خود یعنی 2f بستگی دارد این بدین معنی می باشد که با دو رقم اعشار چاپ می گردد. در newline تعریف سطر و ستون ها آورده شده است در صورتی که این دستور به همراه دستور بعدی آورده نشود فایل a در یک ردیف در قالب txt ذخیره خواهد شد. مثال:

M =

```
96 77 62 41 6 21 2 42
24 46 80 94 36 20 75 85
61 2 93 92 82 61 45 53
49 83 74 42 1 28 94 21
90 45 18 90 14 20 47 68
```

دستور ذخیره سازی:

```
>> dlmwrite('nam.txt',M,'delimiter','t','precision', '%.2f','newline','pc')
```

برای فراخوانی فایل `txt` از دو روش استفاده خواهیم کرد:

در روش اول با استفاده از دستور زیر فراخوانی را انجام می دهیم:

```
cc=dlmread('nam.txt');
```

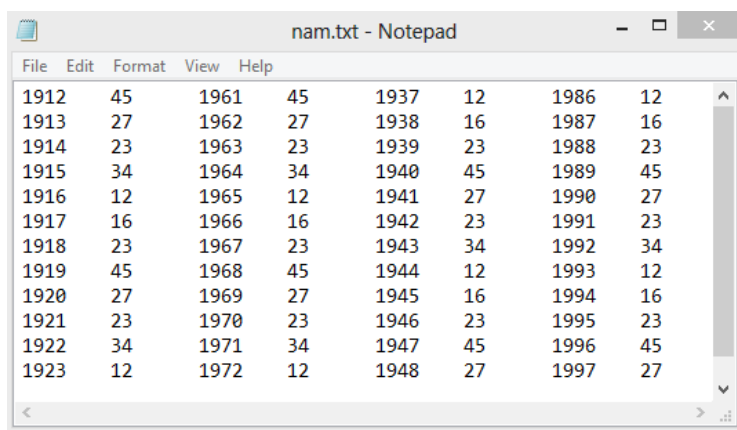
cc =

```
96 77 62 41 6 21 2 42
24 46 80 94 36 20 75 85
61 2 93 92 82 61 45 53
49 83 74 42 1 28 94 21
90 45 18 90 14 20 47 68
```

روش دوم

وارد کردن داده ها از یک فایل متنی

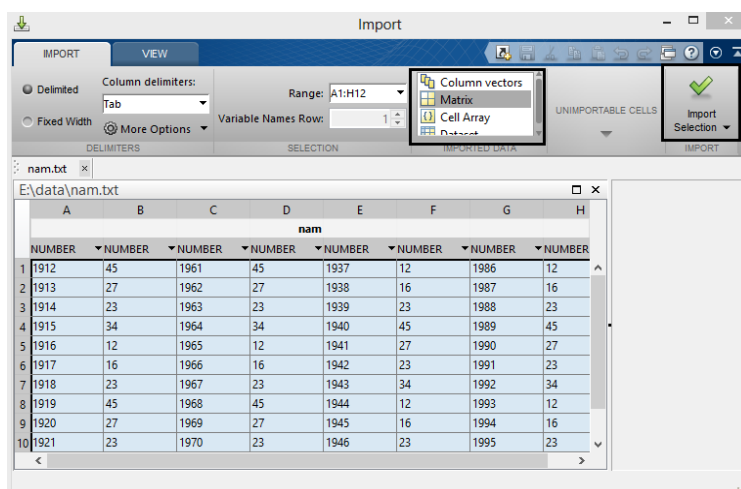
به عنوان نمونه، یک فایل متنی در قالب داده های (text) با نام `able.txt` در اختیار داریم که نمای از آن در تصویر -- نشان داده شده است.



شکل —

برای وارد کردن فایل، سربرگ `home` در متلب را باز کرده و در آنجا بر روی `import Data` کلیک می کنیم. پنجره عمومی باز کردن فایل باز می شود. در آن پنجره فایل را از روی دیسک کامپیوتر یافته و روی `Open` کلیک می کنیم تا پنجره تصویر — باز شود. بعد از اینکه پنجره مشابه با پنجره شکل — باز شد در کادر

مستطیلی شکل چند تا گزینه وجود دارد که متناسب با هدف یکی از آنها انتخاب می شود. برای مثال اگر قصد داشته باشید که داده‌های که به شکل وکتوری (ستونی) وارد command workspace شود یا به صورت ستونی انتخاب شود گزینه column vectors انتخاب می شود. بسته به نوع هدف، داده‌ها می‌تواند به شکل ماتریس وارد command workspace شود، برای این منظور گزینه Matrix را انتخاب کنید. بهترین نوع فراخوانی انتخاب گزینه Matrix می‌باشد. بعد از اینکه نوع فراخوانی را انتخاب کردید بر روی دکمه سبز رنگ import selection کلیک کرده تا داده‌ها به محیط command window انتقال داده شود.



شکل --

ذخیره سازی و بازیابی داده‌ها

فرض کنید که بخواهید داده‌های موجود در workspace را با فرمت متلب در یکی از درایوهای دلخواه در اینجا مثلا e ذخیره و بازیابی بکنید. برای این کار به دو طریق می‌توانیم انجام داد.

۱- مسیر دایرکتور فایل در current folder مشخص باشد که برای این کار از طریق این دستور cd:/e انجام می‌گیرد و وقتی که مسیر مشخص باشد کافی است در صفحه command window عبارت زیر را تایپ کنید:

```
>> save a
```

با این دستور فایل a در مسیر که در دایرکتوری مشخص شده ذخیره خواهد کرد. برای فراخواهی کفایست عبارت زیر را در command window تایپ کنید:

```
>> load a
```

۲- زمانی که مسیر دایرکتور در current folder مشخص نباشد. در این مواقع برای ذخیره کردن فایل‌ها در به طریق زیر عمل خواهیم کرد:

```
>> save e:/data/a
```

این دستور ماتریس‌ها را در درایو e و در فولدر data ذخیره خواهد کرد. برای فراخوانی از دستور زیر استفاده

خواهیم کرد:

```
>> load e:/data/a
```

– داده‌های که خودمان می‌توانیم در متلب ایجاد کنیم.

در نرم افزار متلب ما می‌توانیم داده تصادفی تولید کنیم. برای اینکه درک مطلب ساده‌تر گردد به مثال زیر توجه کنید:

```
A= (1:2:10)
```

```
A =
```

```
1 3 5 7 9
```

```
A=
```

فرض کنید که ماتریسی به شکل زیر داشته باشید و بخواهید آن را در صفحه `command window` وارد کنید، پس به صورت زیر عمل می‌کنیم:

```
9 10 4
```

```
3 15 7
```

```
4 10 2
```

حالا اگر بخواهیم این ماتریس در متلب بنویسیم به شکل زیر خواهد بود:

```
A= [9 10 4; 3 15 7; 4 10 2]
```

یا

```
A= [9, 10, 4; 3, 15, 7; 4, 10, 2]
```

با تایپ هر کدام از دستورات بالا ماتریس مورد نظر ایجاد خواهد شد. بنابراین در دستور اولی برای فاصله بین اعداد از (space) استفاده شده است اما در دستور دومی برای ایجاد فاصله بین اعداد از کاما (,) استفاده شده است. با این وجود از هر کدام استفاده کنید نتایج مشابه خواهد بود.

– استفاده از فانکشن‌های (توابع) مخصوص

بعضی از توابع هستند که می‌توانند یک سری اعداد خاصی را تولید کنند، برای مثال `zeros` (اعدادی تولید می‌کند که همه درایه‌های آن صفر باشد)، `ones` (اعدادی تولید می‌کند که همه درایه‌های آن یک باشد)، `magic` (اعدادی جادویی تولید می‌کند که میانگین سطر و ستون آن یک عدد می‌باشد). در بین تابع‌های یاد شده توابع `zeros` از کاربرد فراوانی به ویژه در برنامه نویسی (برای صفرسازی) برخوردارند. برای مثال برای مطالعه گرد و غبار، توفان‌های تندری، بارش‌های سنگین و فوق سنگین، یخبندان، نمایه‌های حدی و همچنین کدهای خاص کاربرد زیادی دارد. مهمترین این توابع در جدول زیر نمایش داده شده است:

eye - ماتریس همانی ایجاد می کند

zeros - ماتریس صفر ایجاد می کند

ones - ماتریس یک‌ها ایجاد می کند

diag - ماتریس قطری ایجاد می کند

triu - ماتریس بالا مثلثی ایجاد می کند

tril - ماتریس پایین مثلثی ایجاد می کند

rand - ماتریس تصادفی ایجاد می کند

hilb - ماتریس هیلبرت ایجاد می کند

magic - ماتریس جادویی ایجاد می کند

gallery - لیستی از ماتریس‌های معروف را نشان می دهد

برای مثال:

```
>>a=magic (3)
```

```
a =
```

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

```
b=zeros (3)
```

```
b =
```

```
0 0 0
0 0 0
0 0 0
```

```
c=ones (3)
```

```
c =
```

```
1 1 1
1 1 1
1 1 1
```