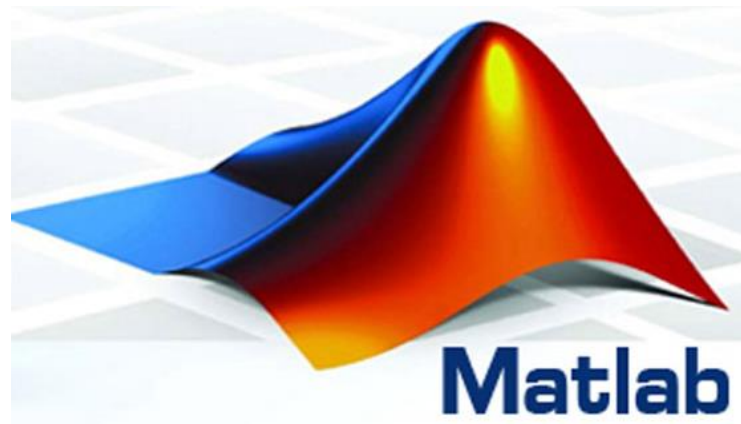


آموزش نرم افزار

متلب



R2014a

تالیف : مهندس مسعود حمیدی

# جلسه اول

# تاریخچه

نرم افزار MATLAB یک زبان برنامه نویسی توانمند برای محاسبات فنی به شمار می رود. از آنجا که المان اصلی داده در این نرم افزار ماتریس می باشد نام این نرم افزار از روی معادل انگلیسی آن یعنی **matrix labrotary** استخراج شده است. این نرم افزار در اواخر دهه 1970 میلادی توسط Cleve Moler رئیس دانشکده علوم کامپیوتر نیو مکزیکو اختراع شد. او این نرم افزار را با هدف در دسترس ساختن LINPACK و EISPACK برای دانشجویان بدون نیاز به یادگیری زبان فرترن داشته باشند طراحی کرد، نرم افزار او به زودی در دانشگاههای دیگر منتشر شد و این نرم افزار به زبان برنامه نویسی C بازنویسی شد. نرم افزار متلب به علت وسیع بودن کاربرد آن نسبت به نرم افزارهای محاسبات عددی دیگر دارای مزیت های زیر است:

- ۱- دارای امکان کد نویسی سریع و آسان با یک زبان سطح بالا
- ۲- محیط آن طوری طراحی شده است که به راحتی می توان از آن استفاده کرد و در صورت نیاز به آسانی اشکال زدائی نمود.
- ۳- امکانات مختلف نمایش گرافیکی نتایج و رسم نمودارهای ۲ بعدی و ۳ بعدی
- ۴- m-file های MATLAB به راحتی در نسخه های مختلف آن قابل استفاده می باشند.
- ۵- جعبه ابزارها می توانند به راحتی به متلب اضافه شوند که هر کدام از آنها به صورت تخصصی به موضوع خاصی می پردازد.

## کاربردهای گسترده نرم افزار MATLAB :

- . هوافضا : سیستم های پیشران، شبیه سازی آیرودینامیک سازه های مختلف
- . خودرو : شبیه سازی سیستم های ایمنی و حرارتی ، سیستم تعلیق خودرو ، سیستم محرکه خودرو
- . پزشکی و زیست شناسی : پردازش سیگنال های زیستی ، تحقیقات پزشکی و ابزارهای جراحی
- . مخابرات : دستگاه های بیسیم ، پردازش سیگنال ، پردازش صوت و تصویر ، شبکه های نوری
- . کامپیوتر : دستگاه های ارتباطی مانند مودم ، دیسک سخت
- . الکترونیک : دستگاه های صوتی و تصویری
- . خدمات مالی و اقتصاد : شبیه سازی مدل های اقتصادی ، مدیریت سرمایه گذاری
- . اتوماسیون صنعتی : ربات ها و شبیه سازی آن ها ، سیستم توزیع انرژی ، موتورها

شرکت توسعه دهنده نرم افزار متلب، شرکت MathWorks می باشد.

## قابلیت های این نرم افزار :

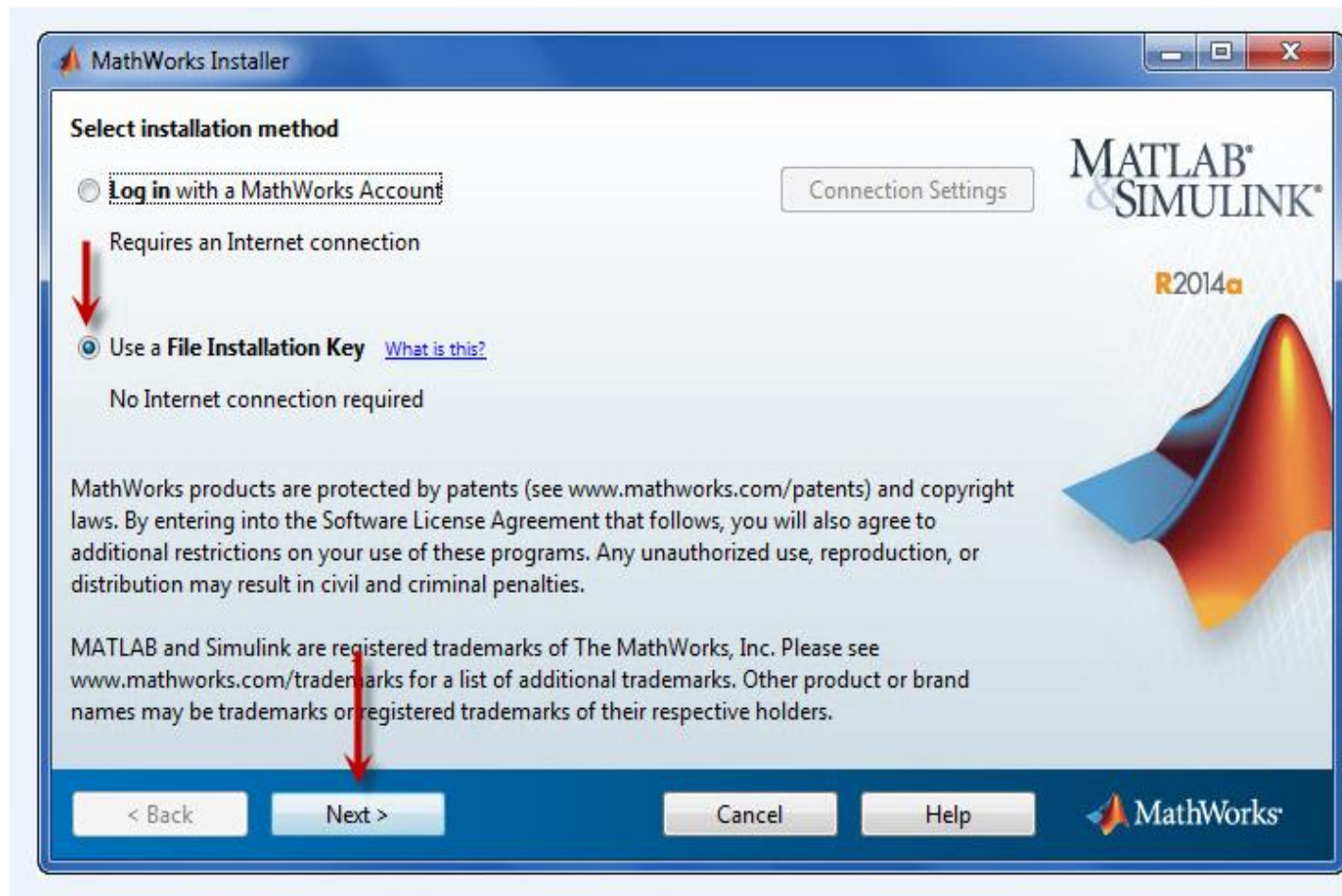
- ✓ عملیات ریاضی مثل مشتق و انتگرال و . . .
- ✓ محاسبات پیشرفته ماتریس
- ✓ برنامه نویسی
- ✓ رسم انواع نمودارهای ریاضی ( دو بعدی و سه بعدی )  
و آماری
- ✓ جعبه ابزارهای مختلف و متنوع
- ✓ محیط Simulink برای رشته های مختلف مهندسی

## روش نصب :

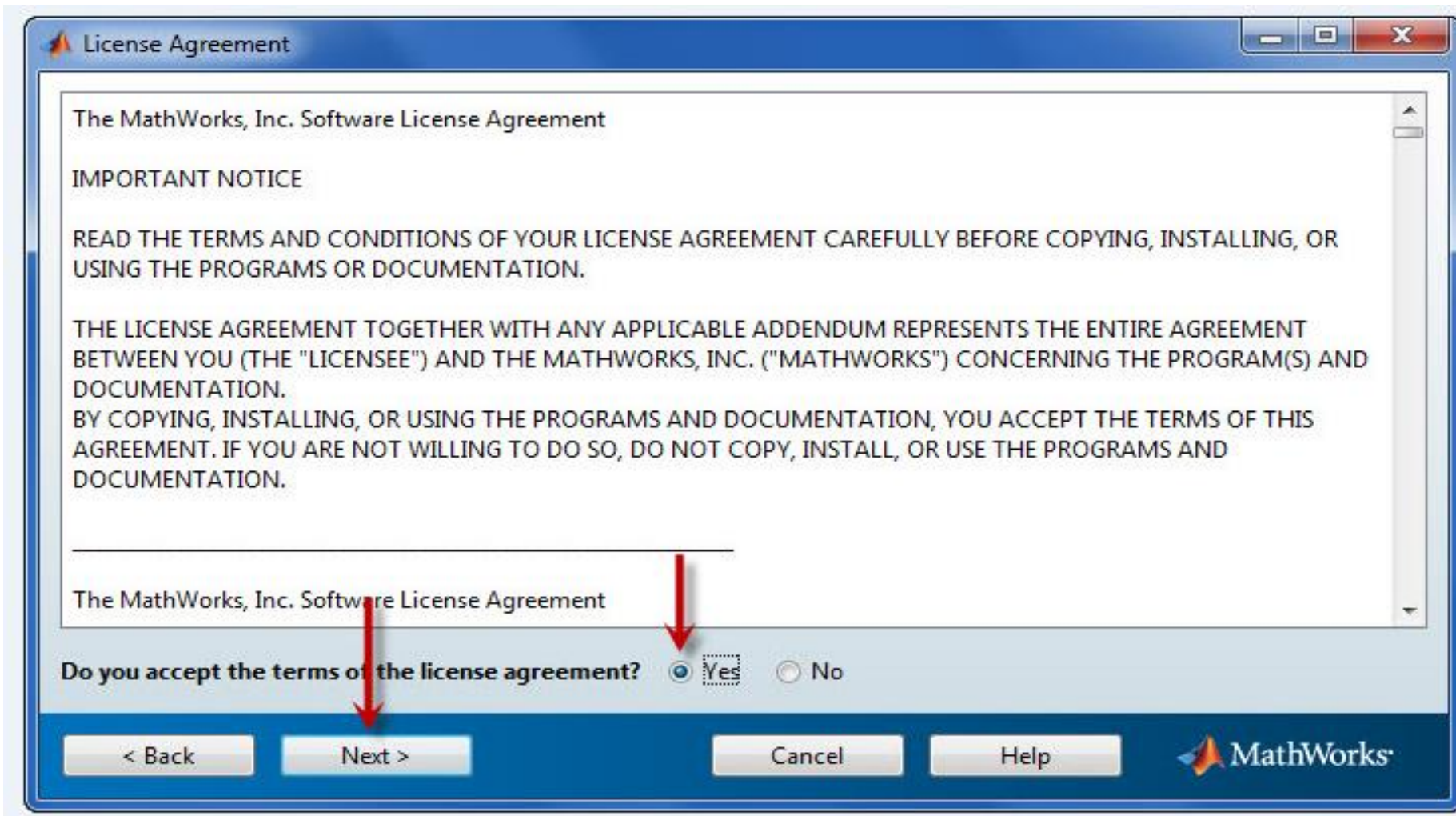
توجه : در زمان نصب این برنامه ارتباط خود را با اینترنت قطع کنید.

✓ از اتوران روی کلید نصب برنامه کلیک کنید.

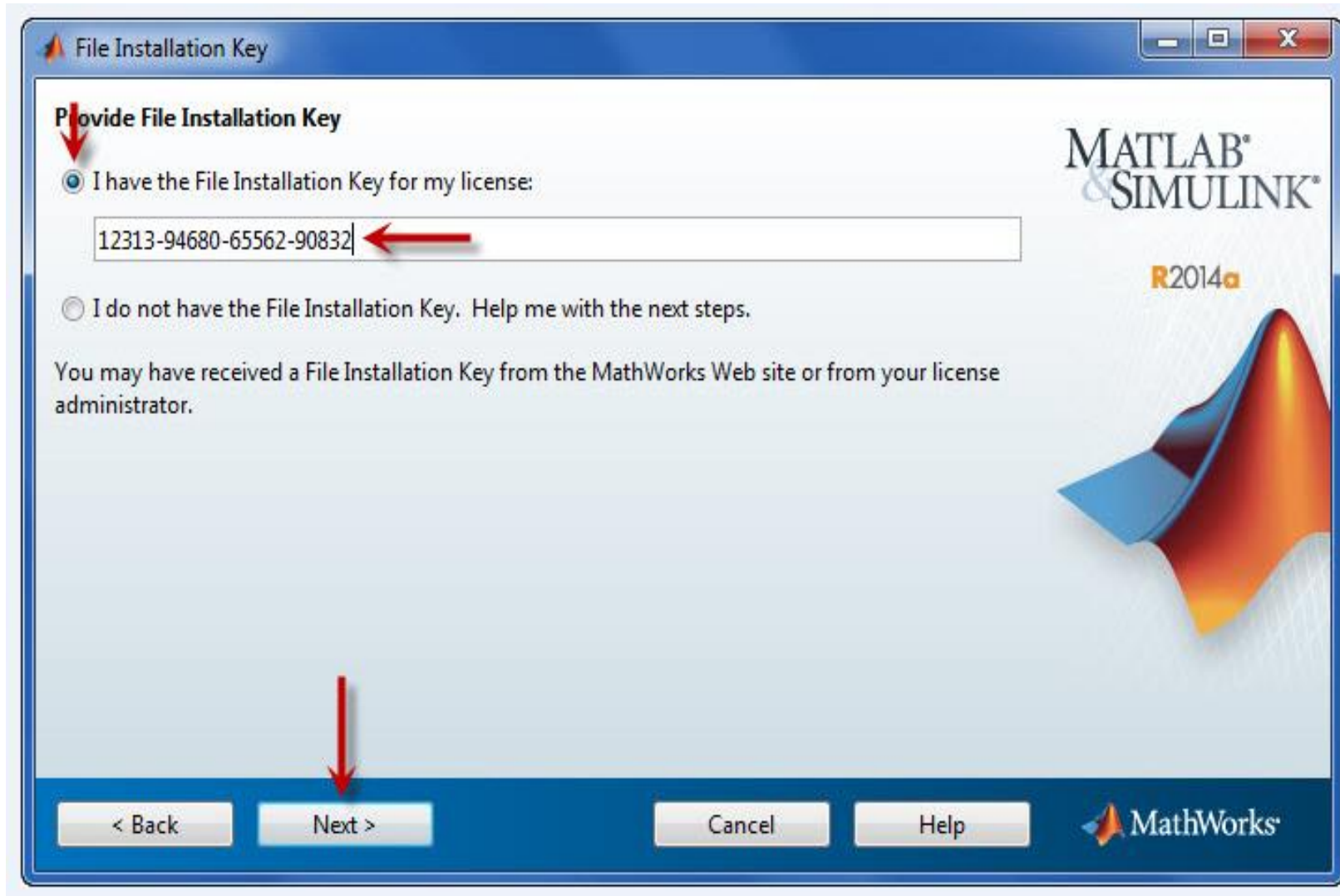
✓ گزینه Use a File Installation Key را انتخاب کرده و بر روی Next کلیک کنید.



✓ در این پنجره گزینه Yes را فعال کرده و بر روی Next کلیک کنید.

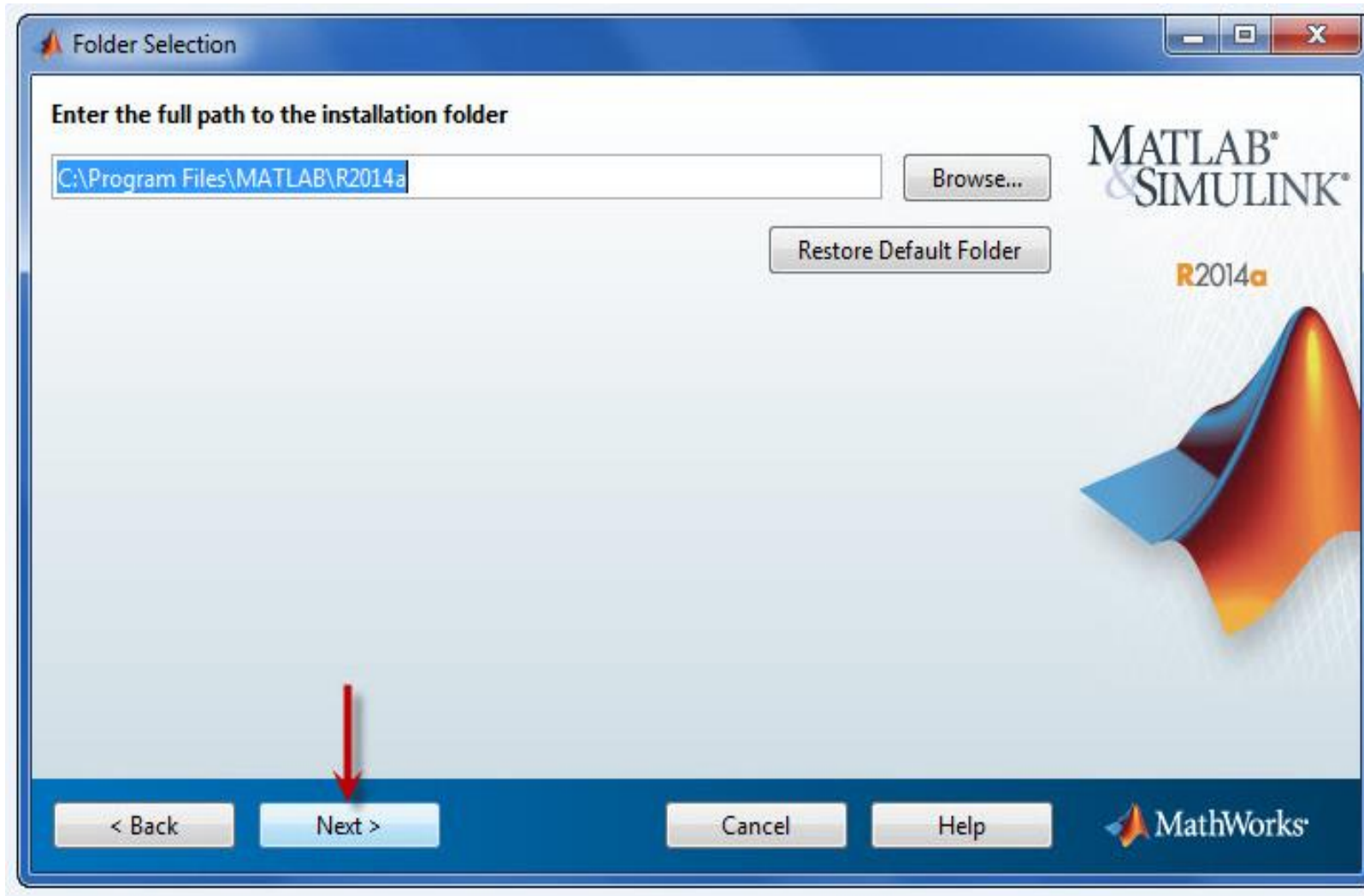


✓ در این پنجره گزینه I have the File Installation Key for my license را انتخاب کرده و سپس از اتوران بر روی کلید شماره سریال کلیک کنید. سریال را در محل مربوطه وارد کرده و بر روی Next کلیک کنید.

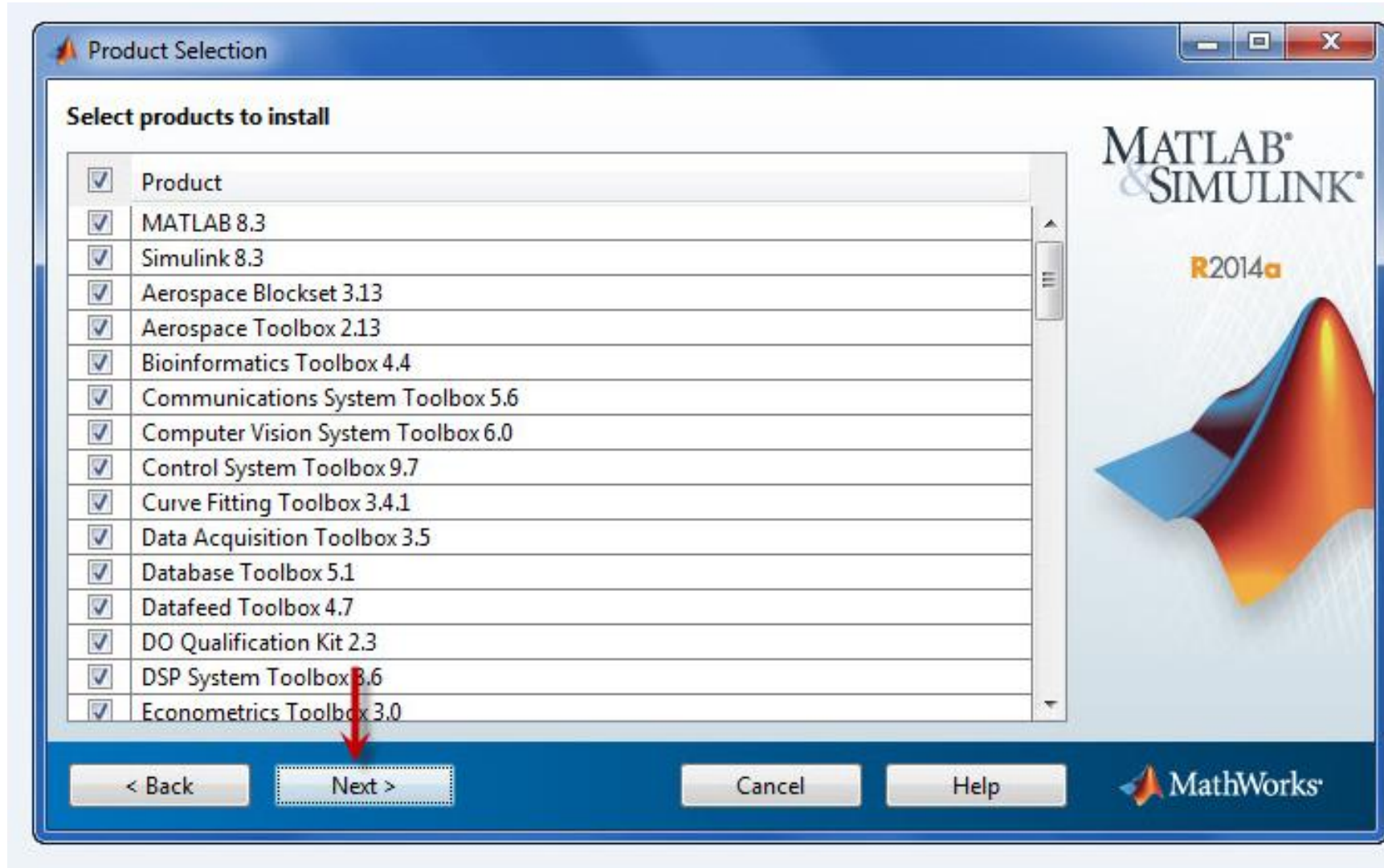




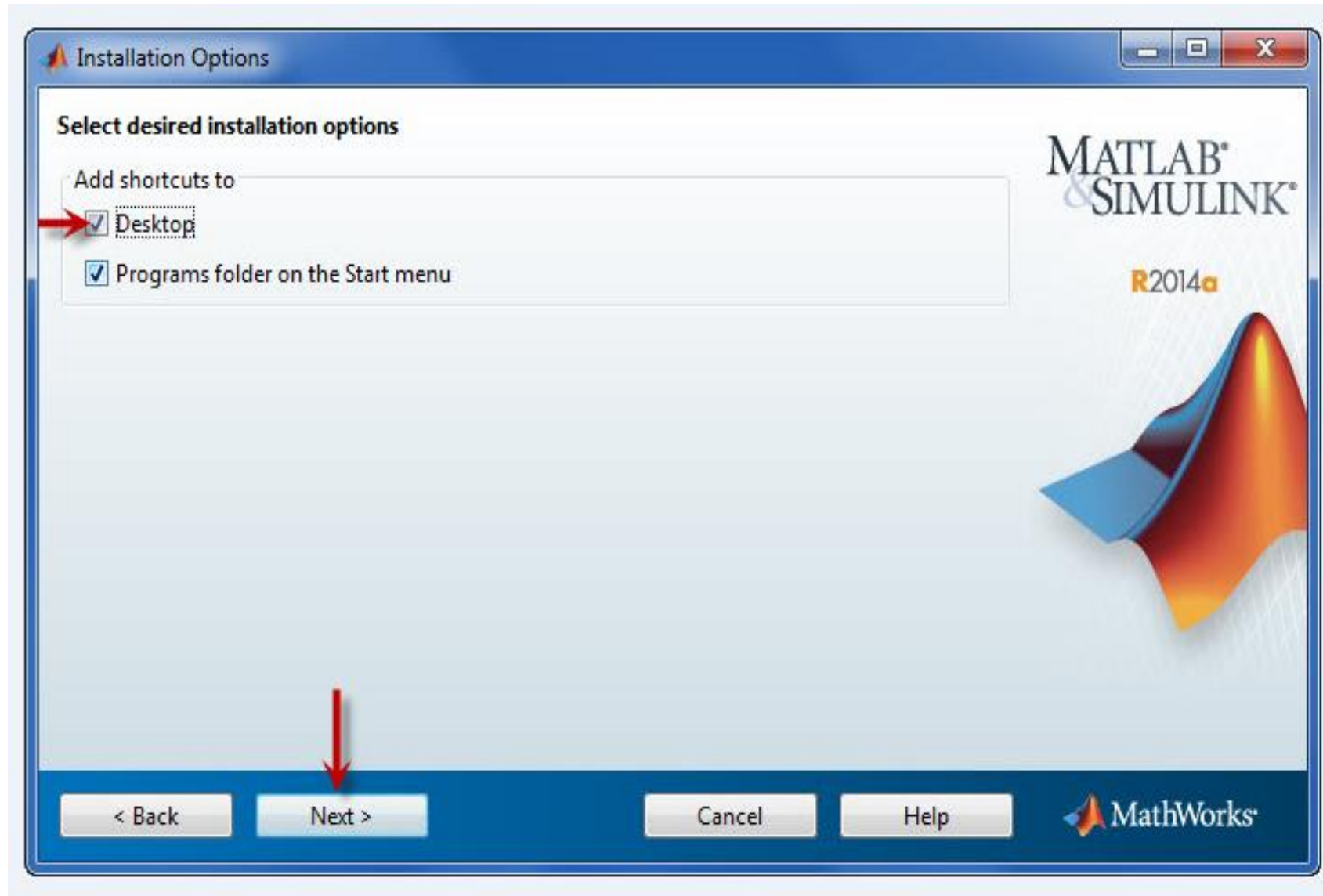
✓ در این پنجره در صورت نیاز می‌توانید با استفاده از کلید Browse مسیر نصب برنامه را تغییر دهید و در غیر اینصورت کافیست روی Next کلیک کنید.



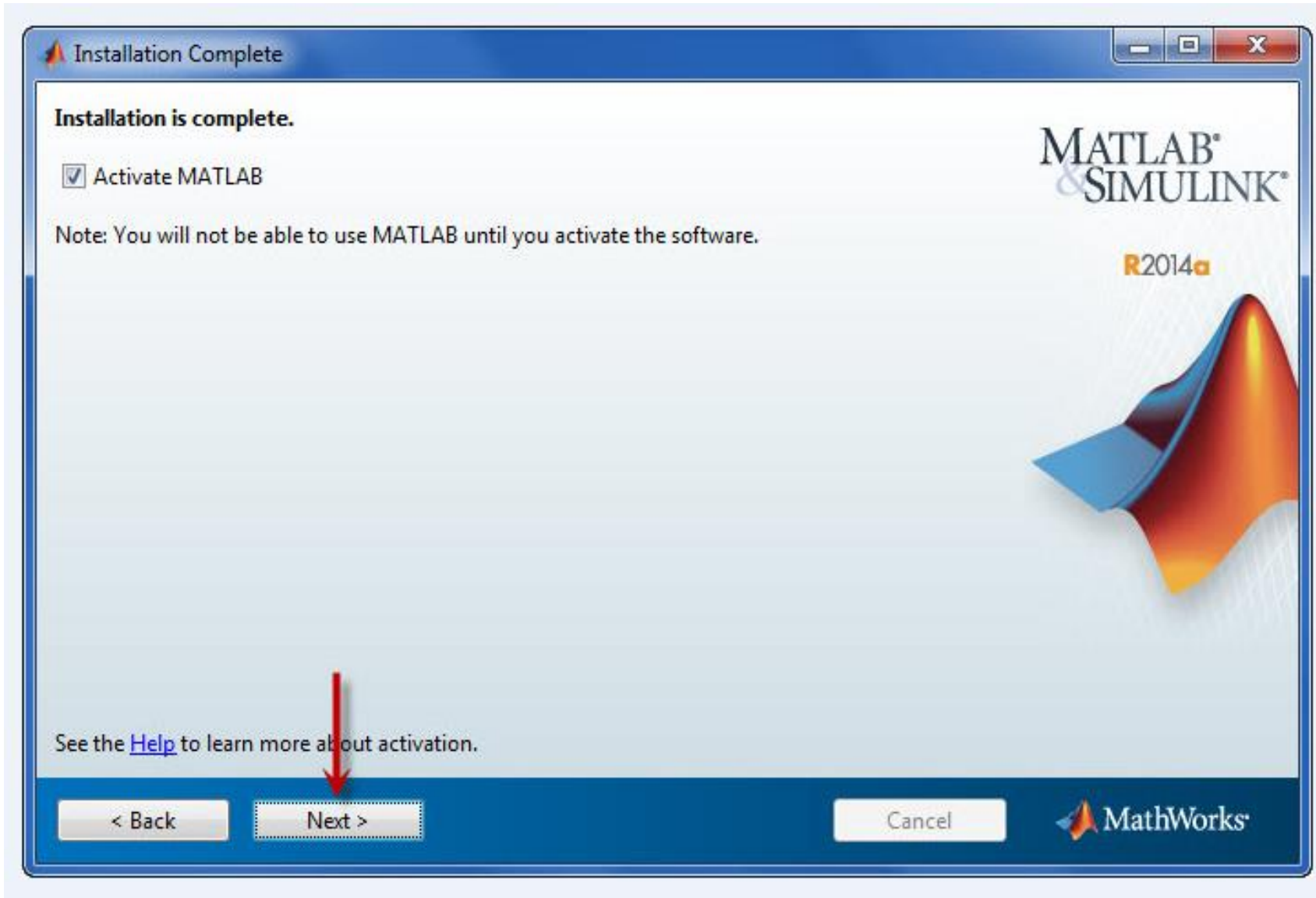
کامپوننت های مورد نظرتان را انتخاب و بر روی Install کلیک کنید.



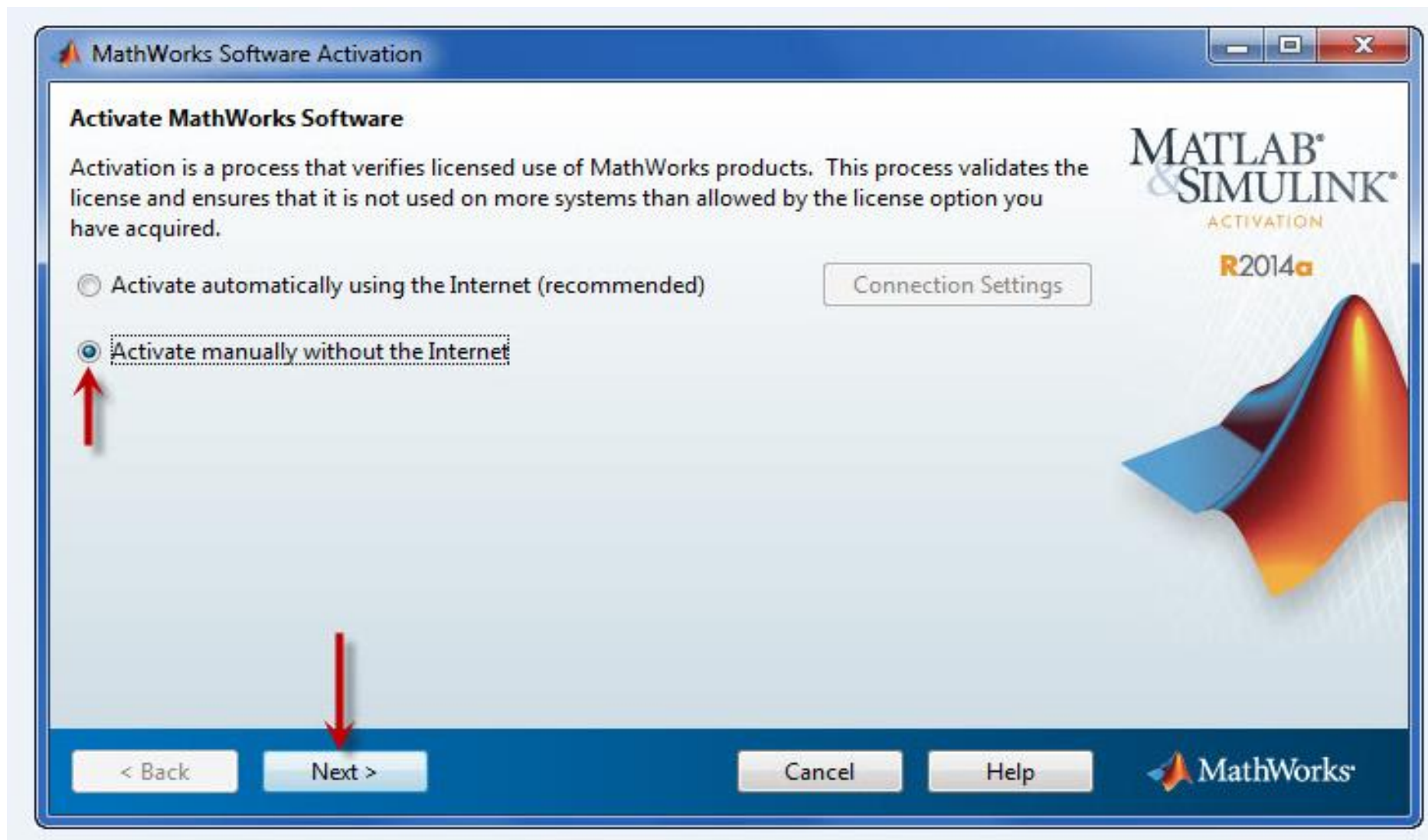
✓ تیک مربوط به ایجاد Shortcut بر روی Desktop را فعال کرده سپس روی کلید Next کلیک کنید، install را زده و سپس Next



✓ روی کلید Next کلیک کنید.

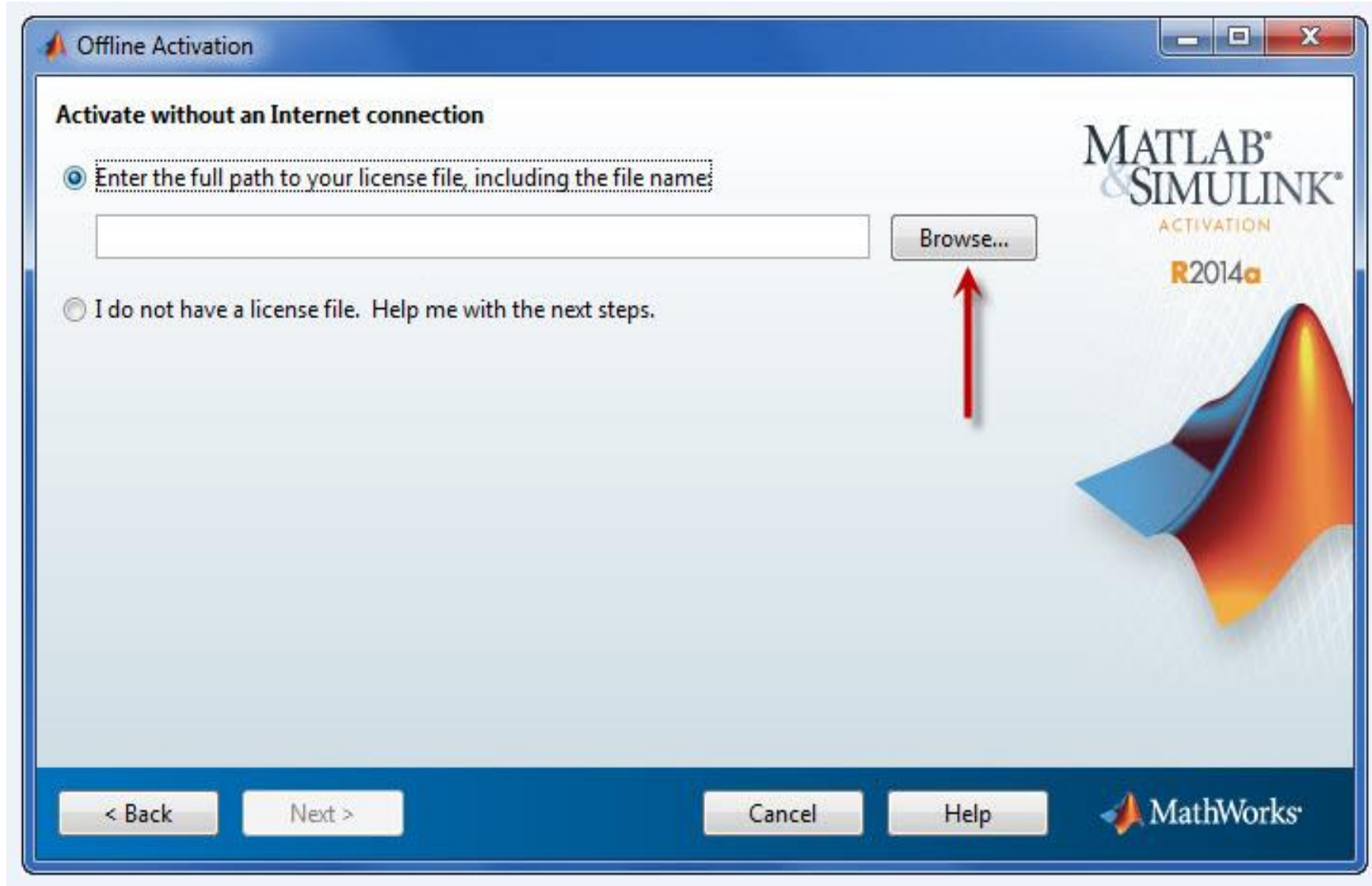


✓ در این پنجره گزینه Activate Manually without the Internet را انتخاب کرده و بر روی Next کلیک کنید.

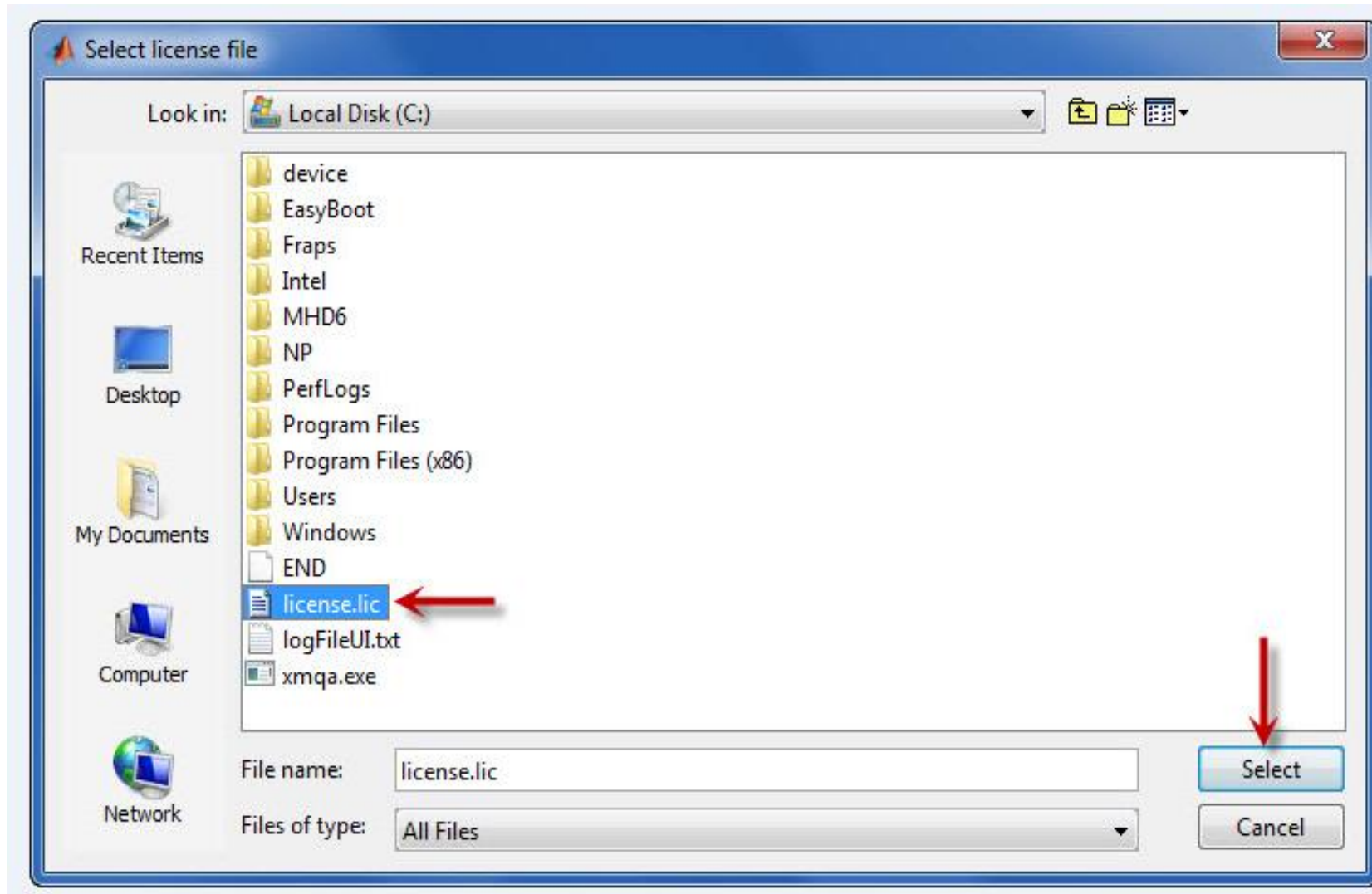




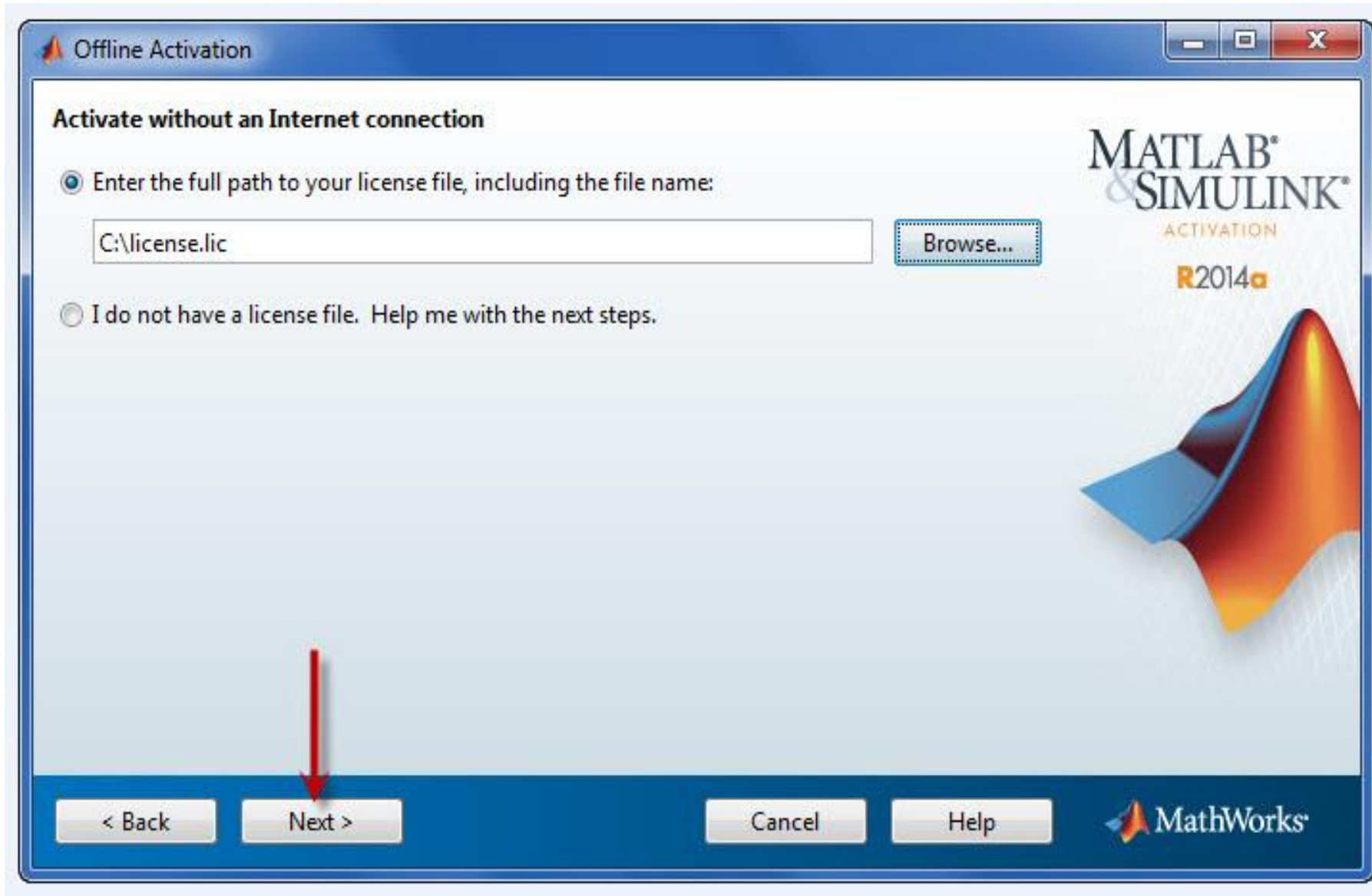
✓ در پنجره بعدی گزینه Enter the full path... را انتخاب کرده و سپس روی کلید Browse کلیک کنید.



✓ از اتوران روی کلید لایسنس برنامه کلیک کنید و آن را در محل دلخواهی از حافظه Extract نمایید سپس فایل license.lic را انتخاب و بر روی Select کلیک کنید.

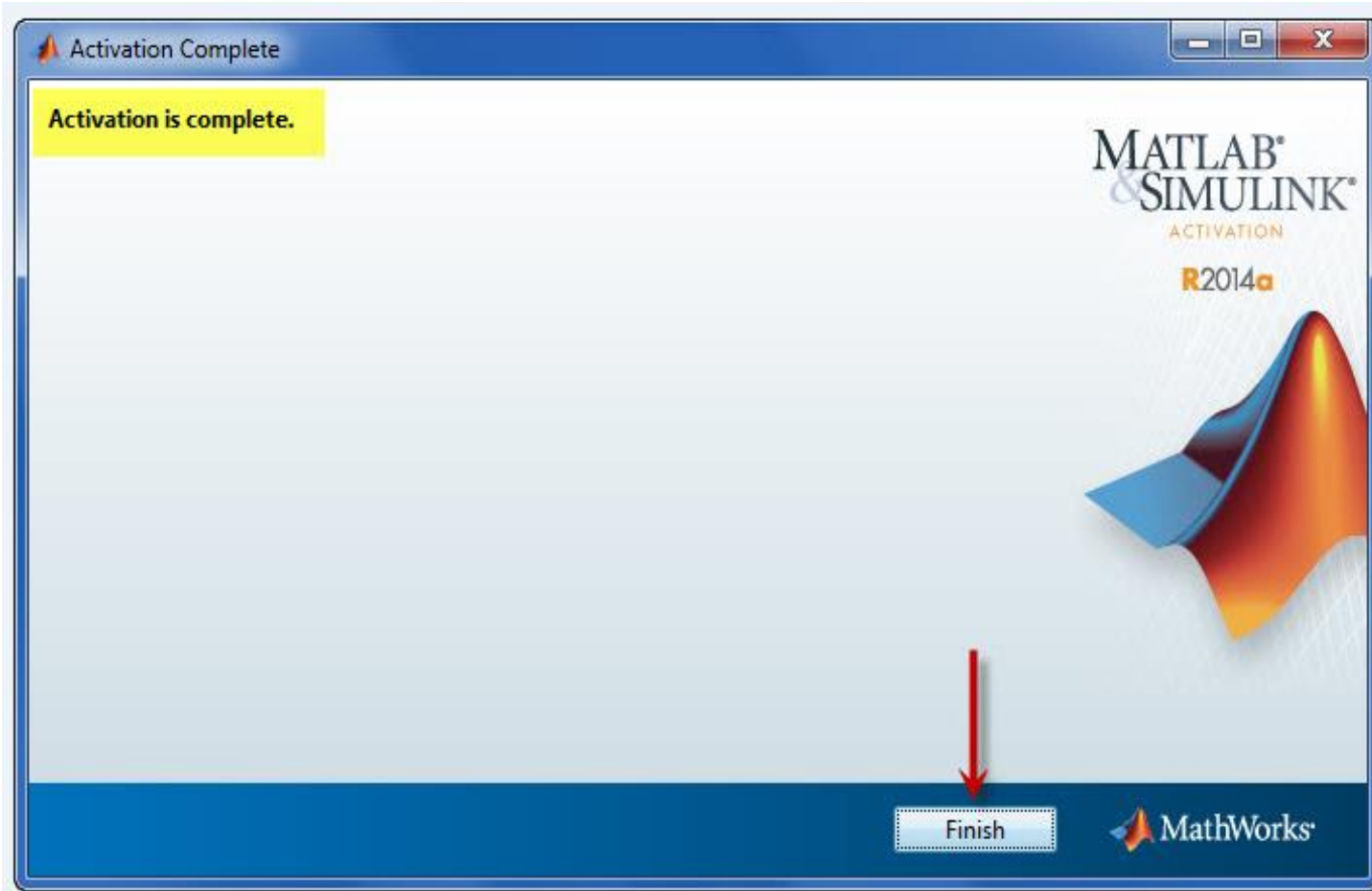


✓ در این پنجره بر روی Next کلیک کنید.





✓ در این پنجره بر روی Finish کلیک کنید.



- ✓ از اتوران روی کلید **فعال ساز Patch** کلیک کرده و آن را در محل نصب برنامه نصب نمایید.
- ✓ برنامه را اجرا کنید؛ همانطوریکه مشاهده می کنید برنامه به صورت کامل فعال شده است.

## نحوه ورود به برنامه :



برای وارد شدن به این نرم افزار بر روی آیکون برنامه در دسک تاب کلیک می کنیم :

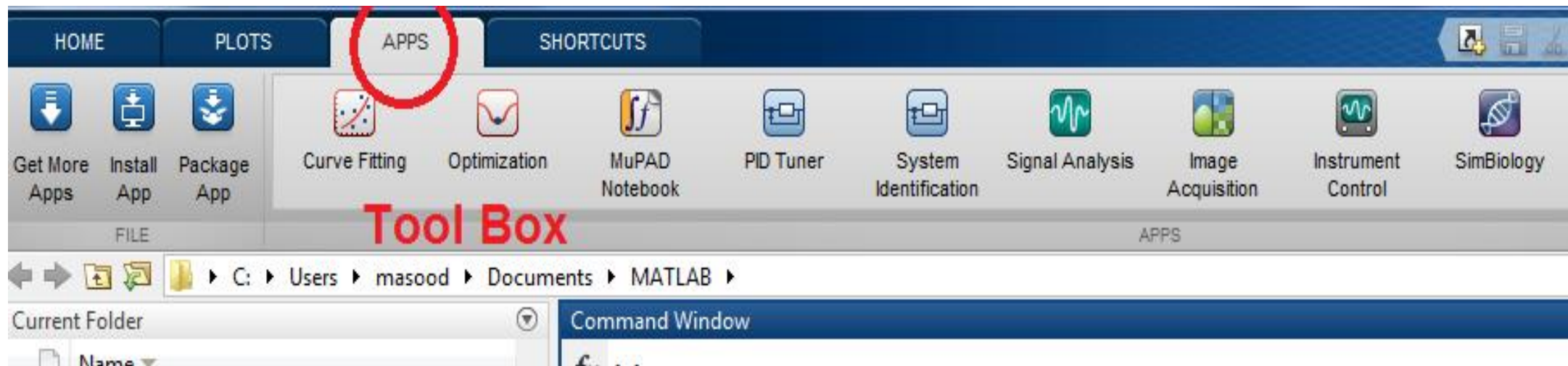
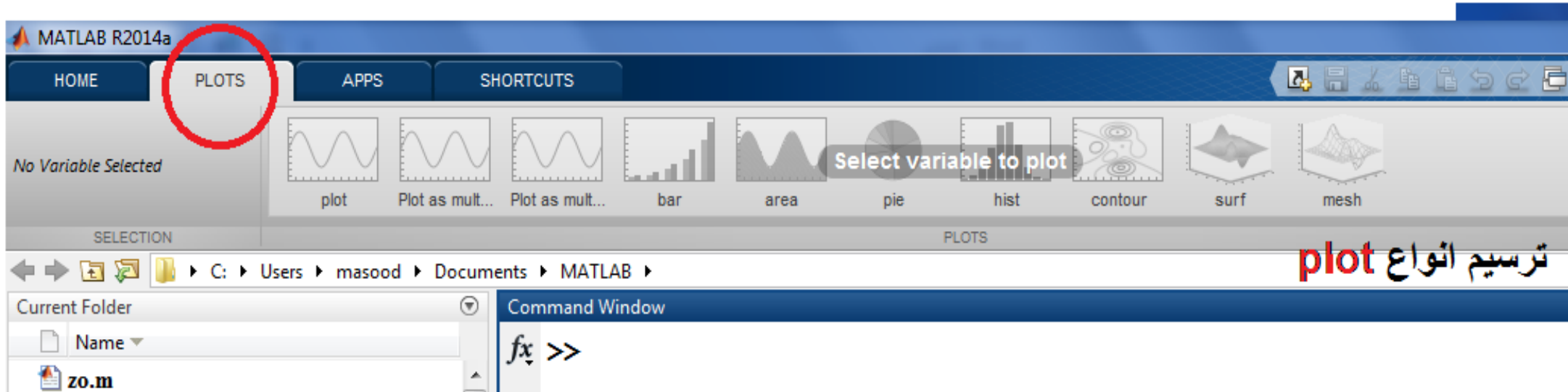
اولین پنجره باز شده بصورت زیر است :

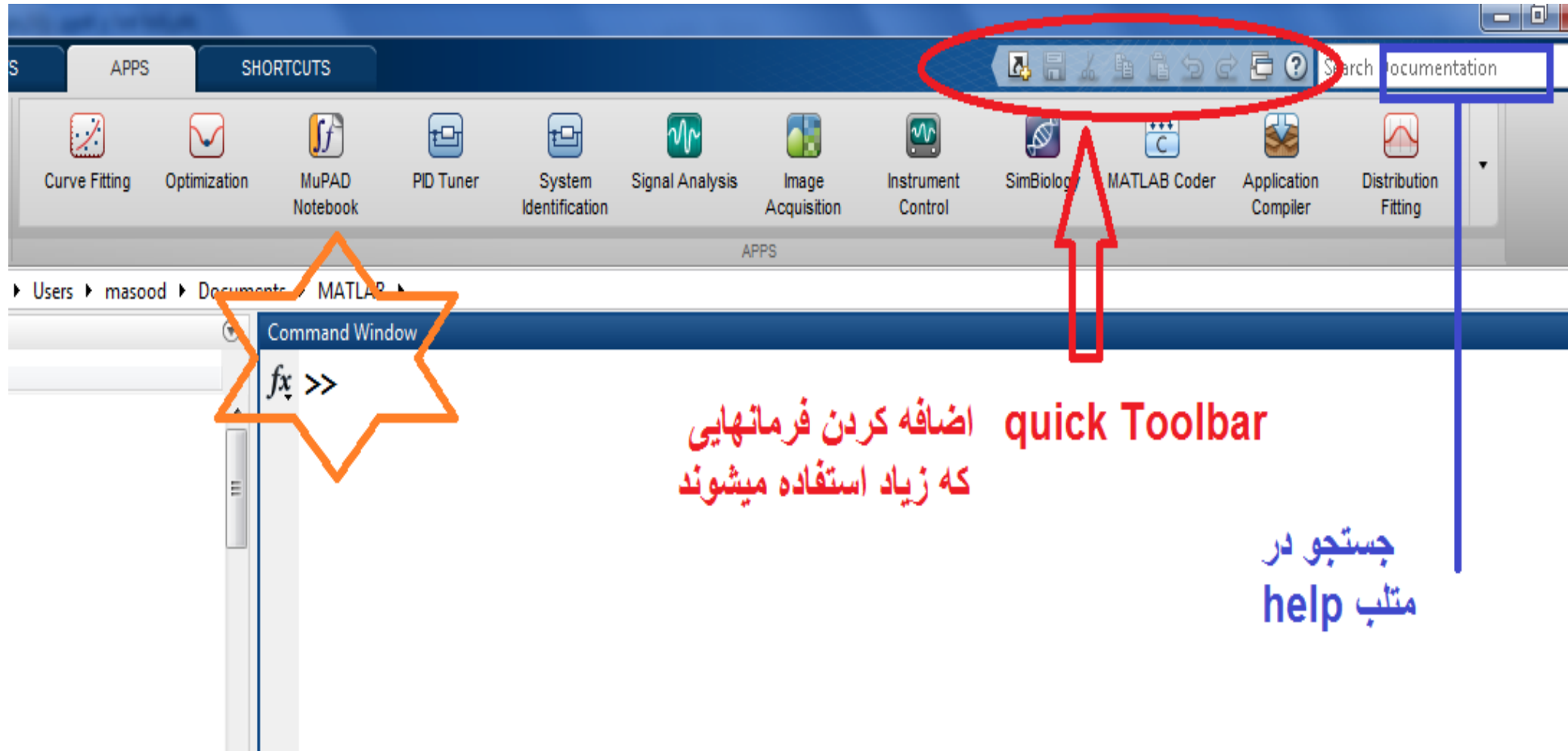
The screenshot shows the MATLAB R2014a interface with several components highlighted and labeled:

- Tab:** A red circle highlights the top tabs: HOME, PLOTS, APPS, and SHORTCUTS.
- ریون (Ribbon):** A red circle highlights the ribbon area, which is divided into sections: FILE, VARIABLE, CODE, SIMULINK, ENVIRONMENT, and RESOURCES.
- کار با فایل ها (File operations):** A green arrow points to the FILE section, which includes icons for New Script, New, Open, and Compare.
- کار با متغیرها (Variable operations):** A blue arrow points to the VARIABLE section, which includes icons for New Variable, Open Variable, and Clear Workspace.
- کار با کدها و آنالیز کدها (Code operations):** A purple arrow points to the CODE section, which includes icons for Analyze Code, Run and Time, and Clear Commands.
- محیط برنامه (Simulink):** A yellow circle highlights the Simulink Library icon in the SIMULINK section.
- منابع و help (Resources):** A brown arrow points to the RESOURCES section, which includes icons for Help, Community, Request Support, and Add-Ons.

Additional labels and arrows in the screenshot include:

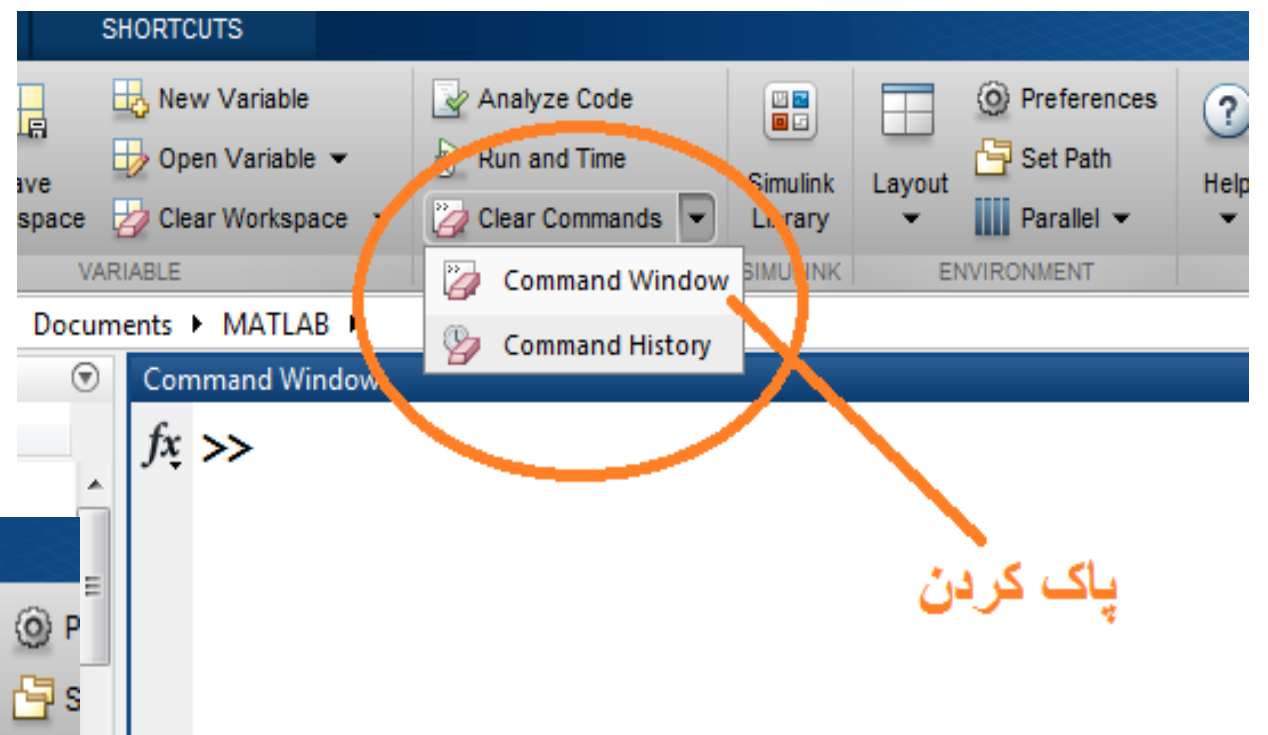
- A green arrow pointing to the Current Folder browser on the left, which lists files like zo.m, x.m, Untitled2.m, test7.m, test6.m, and test5.m.
- A blue arrow pointing to the Command Window on the right, which shows the prompt `fx >>`.
- An orange box highlights the ENVIRONMENT section (Layout, Preferences, Set Path, Parallel).
- A brown box highlights the RESOURCES section (Help, Community, Request Support, Add-Ons).



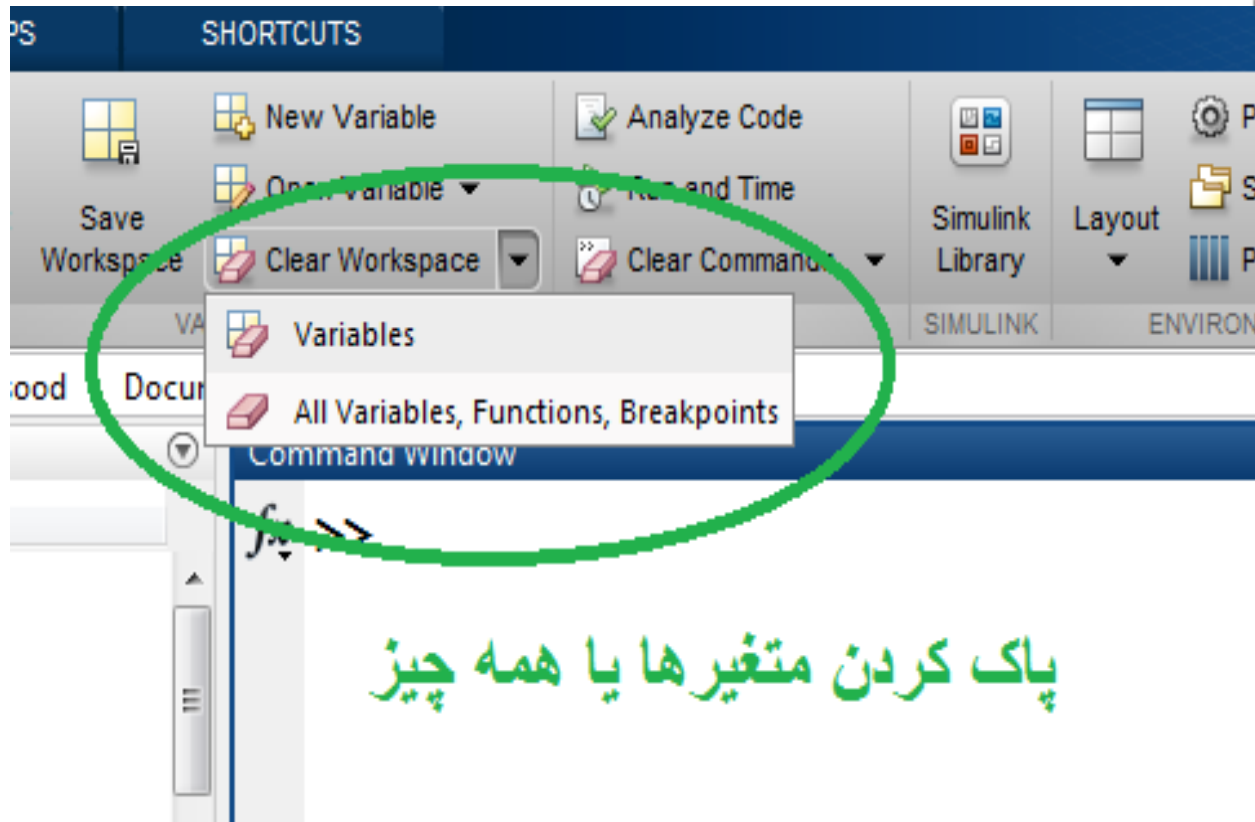


quick Toolbar  
اضافه کردن فرمانهایی  
که زیاد استفاده میشوند

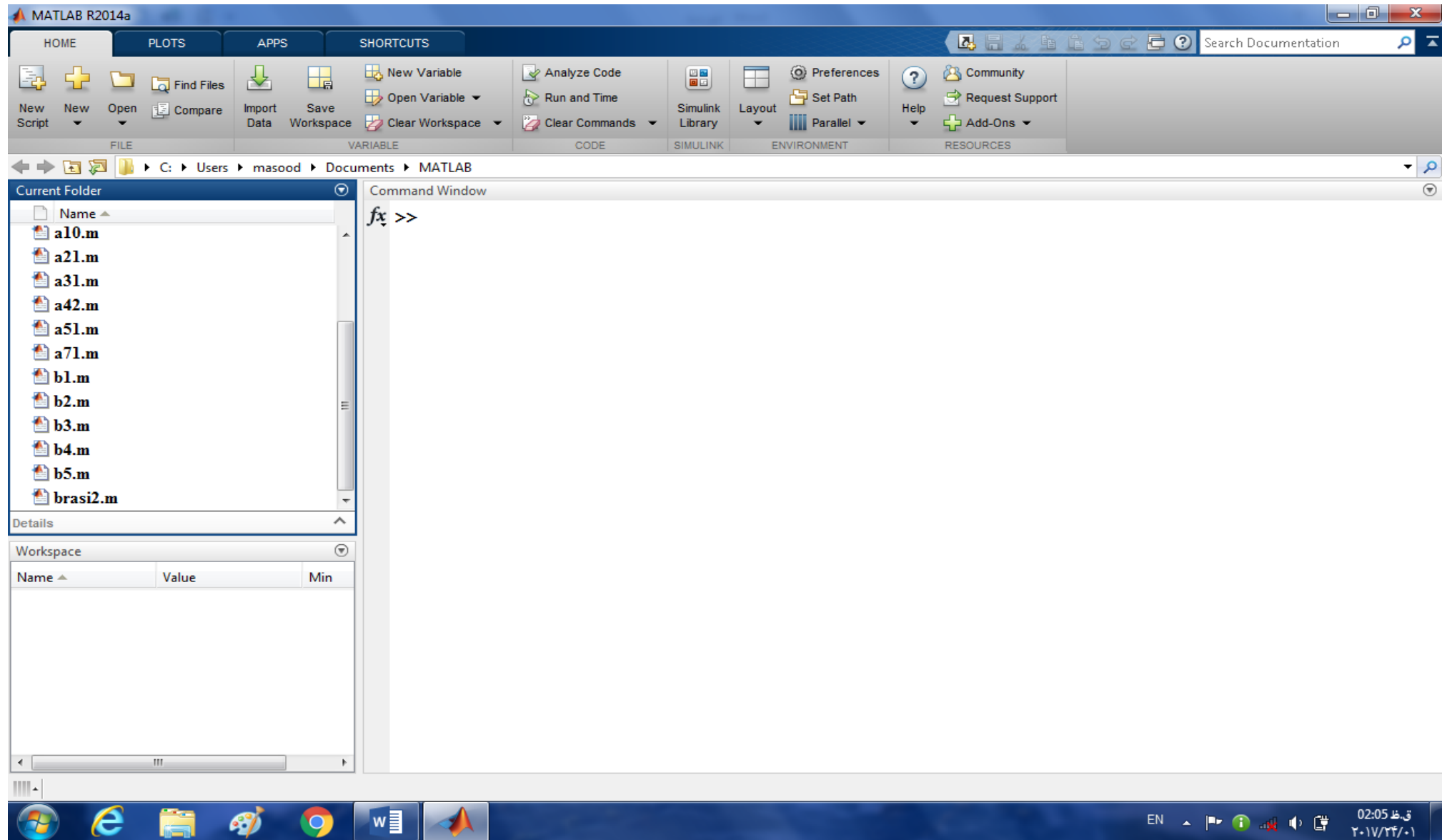
جستجو در  
متلب help



پاک کردن



پاک کردن متغیرها یا همه چیز



## که شامل چند پنجره است :

**Command Window**: در این پنجره می توان مستقیماً دستورات را نوشته و با زدن کلید Enter جواب را در همان پنجره دید.

**Current Folder**: در این پنجره فایل‌های ذخیره شده قبلی نمایش داده می شود.

**Workspace**: متغیرهایی که از قبل تعریف شده همراه با مقادیرشان در این مکان نمایش داده می شوند. (قابل ویرایش)

**Command History**: دستوراتی که قبلاً استفاده شده در این قسمت ذخیره می شوند.

- در سربرگ Home قسمت Layout ( طرح بندی ) می توان نمایش این پنجره ها را تغییر داد که معمولاً در حالت Default ( قراردادی ) قرار می دهیم.

**نکته** : در این نرم افزار بین حروف کوچک و بزرگ فرق می باشد و دستورات با حروف کوچک وارد می شوند .

## در Matlab چهار نوع آرایه می توان تعریف کرد :

- ۱ - اعداد اسکالر که تک عضوی هستند. ( یک ماتریس  $1 \times 1$  )
- ۲ - بردارها که شامل یک سطر و  $n$  ستون می باشند. ( یک بعدی )
- ۳ - ماتریسها ( دو بعدی )
- ۴ - آریه های با ابعاد بیش از دو بعد

اعضای یک آرایه می توانند عدد و یا حروف باشند.

در این نرم افزار تمامی محاسبات و ورود اطلاعات بر اساس ماتریس صورت می گیرد.



# ماتریس چیست؟

ماتریس به آرایشی مستطیلی شکل از اعداد یا عبارات ریاضی که به صورت سطر و ستون شکل یافته گفته می‌شود. به طوری که می‌توان گفت که هر ستون یا هر سطر یک ماتریس، یک بردار را تشکیل می‌دهد. هر یک از عناصر ماتریس درایه خوانده می‌شود.

ماتریس‌های هم اندازه (با تعداد سطر و ستون برابر) را می‌توان با هم جمع یا از هم تفریق کرد. ضرب دو ماتریس تنها در صورتی ممکن است که تعداد ستون‌های ماتریس نخست با تعداد سطرهای ماتریس دوم برابر باشد.

# ماتریس $m$ در $n$

$n$  ستون

از تغییر می‌کند

$a_{i,j}$

$m$   
سطر

از تغییر می‌کند

$a_{1,1}$

$a_{1,2}$

$a_{1,3}$

...

$a_{2,1}$

$a_{2,2}$

$a_{2,3}$

...

$a_{3,1}$

$a_{3,2}$

$a_{3,3}$

...

⋮

⋮

⋮

⋮

⋮

## درایه

به هر یک از عناصر که درون ماتریس می‌آیند درایه می‌گویند. برای مشخص کردن هر درایه باید عدد ردیف و ستون آن را به صورت پایین نویس با حرف کوچک نام ماتریس نوشت. برای نمونه اگر نام ماتریسی  $A$  باشد، درایه‌ای که در ردیف نخست و ستون دوم قرار دارد نوشته می‌شود  $a_{۱,۲}$  و خوانده می‌شود « درایه یک دو ».

## ابعاد

ابعاد یک ماتریس با تعداد سطر و ستون آن تعیین می‌شود. ابعاد ماتریسی با  $m$  سطر و  $n$  ستون به صورت  $m \times n$  نوشته و  $m$  در  $n$  خوانده می‌شود.

ماتریسی که تنها یک سطر دارد بردار سطری و ماتریسی که تنها یک ستون دارد بردار ستونی نامیده می‌شود. ماتریسی که تعداد سطر و ستون برابر دارد ماتریس مربعی نامیده می‌شود. ماتریس تهی ماتریسی است که سطر و ستونی ندارد.

# ماتریس :

برای نمایش ماتریس چهار روش وجود دارد :

مثلا برای وارد کردن یک ماتریس  $3 \times 3$

در روش اول پس از پایان هر سطر Enter را زده و سطر بعد را می نویسیم.

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

a =

     1     2     3
     4     5     6
     7     8     9

fx >>
```

Command Window

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

```
a =
```

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

```
fx >>
```

در **روش دوم** بعد از پایان هر سطر ;  
گذاشته سطر بعدی را می نویسیم.

## Command Window

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

```
a =
```

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

```
>> a=[1,2,3
```

```
4,5,6
```

```
7,8,9]
```

```
a =
```

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

**روش سوم و چهارم**

ترکیبی از دو روش قبلی

## عملگرهای محاسباتی :

+ عملگر جمع

- عملگر تفریق ( اگر دو ماتریس هم بعد باشند یعنی تعداد سطر و ستون آنها برابر باشد )

/ عملگر تقسیم (  $a : a/b$  تقسیم بر  $b$  و  $a \setminus b$  تقسیم بر  $a$  )

\* عملگر ضرب ( طبق قواعد ضرب ماتریسها یعنی  $m \times n \dots n \times l$  )

^ عملگر توان



## عملگرهای آرایه ای :

+ . جمع آرایه ای

- . تفریق آرایه ای

/ . تقسیم آرایه ای

\* . ضرب جمع آرایه ای ( ضرب درایه در درایه )

^ . توان آرایه ای

از این عملگرها آرایه ای زمانی استفاده می شود که دو ماتریس شرایط انجام عملیات محاسباتی را نداشته باشند یعنی تعداد ستون ماتریس اول مساوی تعداد سطر ماتریس دوم نباشد ، در نتیجه با اضافه کردن . قبل از عملگر محاسبات درایه ها در درایه انجام میشود که جوابی متفاوت دارد.

## عملگرهای رابطه ای :

علامت کوچکتر <

علامت بزرگتر >

علامت کوچکتر مساوی <=

علامت بزرگتر مساوی >=

برابری را بررسی می کند ==

نا برابری را بررسی می کند !=

مقدار را نسبت می دهد =

& و

| یا

در جواب عملگرهای رابطه ای یک به معنای جواب درست و صفر به معنای جواب نادرست می باشد.

## علامت ها :

**Square brackets** (کروشه یا براکت) : از [ ] برای ورود اطلاعات استفاده می شود.

**Parentheses** (پرانتز) : از ( ) برای چیزی که قرار است روی آن کار انجام شود.

**semicolon** (نقطه ویرگول ; )

✓ نشان دهنده انتهای هر سطر در ماتریس

✓ اگر در پایان دستوری نوشته شود اجرای آن دستور نمایش داده نمی شود.

**comma** (ویرگول , )

✓ اگر بجای فاصله از آن استفاده شود ماتریس نوشته شده از نظر نرم افزاری یک بردار شناخته می شود.

✓ اولویت بندی دستورها

**colon** (دونقطه : ) : بازه بندی کردن

## تعریف یک بردار :

برای تعریف بردارهای عددی حتماً باید از کروشه استفاده کرد ولی استفاده از کروشه برای متغیرهای حرفی الزامی نیست و باید آنها را داخل '' (سینگل کوتیشن) قرار داد.

Command Window

```
>> d=[1 2 3]
d =
     1     2     3
>> h=['ma']
h =
ma
>> k='ha'
k =
ha
```

**تولید بردار عددی :** تولید بردار عددی که اعضای آن به فاصله مساوی از هم قرار دارند.

`t = 1 : 5`

`t = 1 : 2 : 10`

برداری که از یک تا ده را با گام دو می‌شمارد

```
Command Window
>> t=[1 2 3 4 5]
t =
     1     2     3     4     5
>> t=1:5
t =
     1     2     3     4     5
>> t=1:2:10
t =
     1     3     5     7     9
```

# دستور linspace :

تقسیم به  $n$  قسمت مساوی بدون توجه به گام (ابتدا و انتها و تعداد بازها مهم است)

(تعداد بازهای بین ابتدا و انتها, انتهای بردار, ابتدای بردار) `linspace`

Command Window

```
>> linspace(3,20,8)
```

```
ans =
```

```
3.0000    5.4286    7.8571   10.2857   12.7143   15.1429   17.5714   20.0000
```

در متلب هر درایه را می توانیم به دو صورت آدرس دهی نماییم ( بصورت جداگانه مشاهده کرد ):

۱- اندیس ماتریس ( سطر و ستون )

( شماره ستون , شماره سطر ) a

۲- شماره درایه

اگر بخواهیم سطر و ستون خاصی از یک ماتریس را جدا کنیم :

a ( : , j )

نمایش عناصر ستون j ام از ماتریس a

a ( j , : )

نمایش عناصر سطر j ام از ماتریس a

a ( : )

نمایش کل ماتریس به صورت ستونی

# تمرین :

```
1:4:20
```

```
ans =
```

```
1 5 9 13 17
```

```
>> linspace(1,17,5)
```

```
ans =
```

```
1 5 9 13 17
```

```
linspace(1,9,5)
```

```
ans =
```

```
1 3 5 7 9
```

```
>> 1:2:9
```

```
ans =
```

```
1 3 5 7 9
```

```
linspace(1,30,6)
```

```
ans =
```

```
1.0000 6.8000 12.6000 18.4000 24.2000
```

```
30.0000
```

```
>> 1:5.8:30
```

```
ans =
```

```
1.0000 6.8000 12.6000 18.4000 24.2000
```

```
30.0000
```