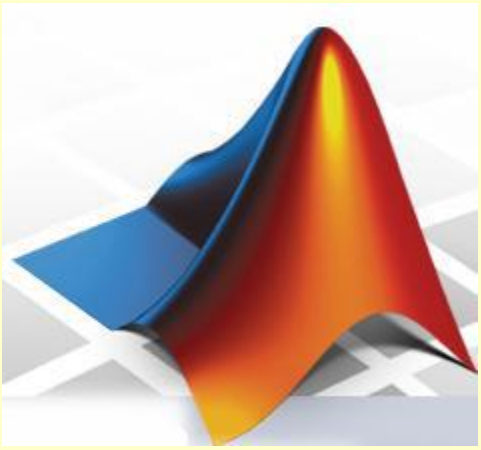
ویرایش اول

راهنمای نرم افزار

MATLAB



MATLAB Software Helpful

مهندس علی عالم

تابستان ۱۳۹۳

دیباچه

با عنایت به رشد سریع علم در دهه های اخیر و گستردگی پژوهش های علمی و افزایش حجم فعالیت های تحقیقاتی درحوزه فناوری های نوین خصوصاً زمینه های فضایی ، اتمی ، نانو ، الکترونیک و سیستم های کنترلی و صنایع رباتیک، کلیه دانشمندان ومتخصصان و مهندسان را به سمت تدارک و استفاده ازابزار های محاسباتی پیشرفته وروش های شبیه سازی قوی هدایت نموده است که ازآن جمله می توان به نرم افزارهای توانمند اشاره کرد ، دراین کتاب تلاش شده است که نرم افزار متلب(MATLAB) برای خواننده گرامی معرفی شودو کاربرد های این نرم افزار قدرتمند بیان گردد.

مطالب این کتاب از راهنمای نرم افزار متلب ترجمه و تخلیص شده و موارد کاربردی ومثال ها توسط خود نرم افزار اجرا شده است.

این کتاب را به جامعه علمی و پژوهشی کشورمان تقدیم می کنم و امیدوارم به عنوان ذره ای کوچک از انبوه تولیدات علمی محسوب گردد و مورد استفاده دانشجویان و دانش پژوهان عزیز قرار گیرد.

درپایان نیز ازتمام زحمات فرهیختگان میهن واساتید بزرگوارم و پدر ومادر ارجمندم و همسرمهربانم کمال تشکر وامتنان رادارم .

علی عالم

مرداد ۹۳

AliAlam@chmail.ir

فهرست

۵- برنامه نویسی گرافیکی (GUI)

۵-۱ آشنایی با واسط گرافیکی کاربر

۵-۲ آشنایی با کنترلر ها

۵-۳ ویژگی کنترلر (property inspector)

۵-٤ کامپایل کردن برنامه(compile)

٦- کاربرد های نرم افزار متلب

٦-۱ کاربرد متلب در داده های آماری

٦-۲ کاربرد متلب در بهینه سازی خطی

٦-۳کاربرد متلب در برنامه ریزی صفر و یک

٦-٤ کاربرد متلب در محاسبات عددی(درون یابی)

۱- مقدمه نرم افزار متلب

مسلماً یکی از قویترین ابزارهای محاسباتی در رشته های مهندسی کامپیوتر ها می باشندکه در انجام محاسبات مهمترین نقش را دارا می باشند.

حال که بر حسب نیازمان کامپیوتر را بعنوان ماشین حسابگر قدرتمند تعریف کردیم نیاز به برنامه ای داریم تا با توجه به نیازهایمان آن را به کاربگیریم.

در این جا به معرفی نرم افزارقدرتمند MATLAB خواهیم پرداخت و شما را با کاربردها و تواناییهای آن آشنا خواهیم ساخت به این امید که مورد توجه و عنایت شما عزیزان قرار بگیرد.

MATLAB یک نرم افزار قوی جهت دانشجویان و محققین رشته های ریاضی و مهندسی است که اولین نگارشهای آن در دانشگاه نیومکزیکو و استانفورد در سال ۱۹۷۰ در جهت حل مسائل تئوری ماتریسها، جبرخطی و آنالیز عددی بوجود آمد و امروزه صدها هزار کاربر دانشگاهی، پژوهشی، صنعتی دارد که در زمینه های بسیار متنوع نظیر ریاضیات پیشرفته، جبر خطی، مخابرات، مهندسی سیستم وشبیه سازی، مهندسی فضایی وسیستم های ترابری ازاین نرم افزار بهره می برند نرم افزار MATLAB بعنوان یکی از اولین محیط های محاسباتی و تکنیکی است که قادر به حل مسائل پیچیده می باشد.

ریاضیات، زبان مشترک بسیاری از علوم مهندسی است .ماتریسها، معادلات ،رشته های عددی، کاربرد و اهمیت زیادی درحل مسائل مهندسی دارند .تحلیل داده ها ،ترسیمات و گرافها از لوازم اصلی بکار گرفته در ریاضیات می باشند ونرم افزار MATLAB یک زبان برنامه نویسی قدرتمند و یک محیط نرم افزاری کامل برای حل بسیاری از محاسبات علمی و مهندسی می باشد.

در نرم افزار متلب شما می توانید بسادگی توابع و برنامه های خاص خودتان را با استفاده از کدها و توابع نرم افزار بنویسید و در صورتیکه تعداد آنها زیاد باشد با اختصاص یک زیر شاخه برای آنها از مجموعه آنها یک جعبه ابزار درست کنید . در حقیقت این نرم افزار یک زبان آسان با مشخصات بسیار پیشرفته و ساده تر از زبانهای برنامه نویسی نظیر C++ ، C است .این نرم افزار یک محیط پرقدرت برای تصویر کردن اطلاعات از طریق کامپیوتری و یاقابلیتهای گرافیکی را فراهم می کند.

دراین کتاب فقط به موارد کاربردی تر می پردازیم و با توجه به گستردگی نرم افزار متلب توصیه می کنیم که در صورتی که نیاز به اطلاعات بیشتر دارید می توانید به help های موجود درخود نرم افزار مراجعه نمایید.

۱-۱-آشنایی با پنجره های نرم افزار متلب

هنگامی که برای اولین بار نرم افزار متلب را اجرا می کنید پنجره هایی مشاهده می شود، البته می توانید بر اساس نیاز خود پنجره های اضافه را بسته و یا پنجره هایی که نمایش داده نشده اند راباز کرد.

پنجره command window

همان طورکه ازنامش مشخص است همه دستورات برای اجرادرنرم افزاردراین پنجره وارد می شوند هردستوربه طور سطری دراین پنجره تایپ شده ونرم افزار پس ازاجرا نتیجه دستور رانیز درهمین پنجره نمایش می دهد.

پنجره command history

کلیه دستورات انجام شده در پنجره دستور دراین پنجره بایگانی می گردد که می تواند جهت مراجعات بعدی مورد استفاده قرار گیرد.

پنجره workspace

یکی ازمهمترین پنجره های MATLAB پنجره workspace می باشد .در این پنجره می توانیم تمام متغیرهایی را که در Matlab تعریف کرده ایم را لیست وار مشاهده کنیم و حتی در آنها تغییراتی ایجاد کرده و یا آنهارا بصورت نمودار مشاهده کنیم.

پنجره current directory

این پنجره محل فایل هایی راکه برنامه درحال اجرای آن می باشد را نشان می دهد ومی توان کلیه فایل های اجرا شده توسط نرم افزار را دراین پنجره مشاهده کرد .مسیر قرار گیری و ذخیره این فایل ها درموقع نصب نرم افزار توسط کاربر تعیین می گردد.

محتویات تمام این پنجره ها را می توان از طریق منوی Edit ویرایش کرد.

تفاوت دستورات پنجره فرمان و برنامه

برنامه ها در فایل هایی به نام M-File ذخیره میشوند. برنامه هایی که در آنها تعدادی فرمان یک جا اجرا میشوند Script M-File نام دارند و برنامه هایی که یک تابع را تعریف میکنند Function M-File نام دارند .

۱-۲-مقدمات کار با MATLAB

MATLAB اعمال ساده ریاضی را به راحتی یک ماشین حساب انجام می دهد.

```
>>2+4-1
ans =
5
>>3+8/2
ans =
7
```

روش دیگر انجام محاسبات این است که مقادیر را در چند متغیر ذخیره کرده و روی متغیرها عملیات محاسباتی را انجام دهیم:

```
>> a=2
a =
2
>> b=4
b =
4
>> c=1
c =
1
>> a+b-c
ans =
5
```

در نام گذاری متغیرها باید موارد زیر را رعایت کرد:

۱) MATLAB نسبت به حروف کوچک و بزرگ حساس است

۲) اسامی متغیرها حداکثر می تواند ۳۱ کاراکتر باشد.

۳)اسامی متغیرها حتما باید با حرف شروع شود .(کاراکتر اول نباید عدد باشد.)

۴) جز کلمات تعریف شده برای MATLAB نباشد.(این کلمات به رنگ آبی نوشته می شود مثل for)

برای شناسایی کلمات کلیدیMATLAB میتوان از دستورiskeyword استفاده کرد.

این تابع در صورتی که عبارت ،یک کلمه کلیدی باشد مقدار یک و در غیر این صورت مقدار صفر را برمی گرداند.

```
>> iskeyword('for')
ans =
1
>> iskeyword('keyword')
ans =
0
>> iskeyword('ans')
ans =
1
>> iskeyword('if')
ans =
1
```

۵) تمام دستورات و عبارات کلیدی MATLAB با حروف کوچک نوشته می شود .بنابراین می توان اسامی آن ها را با حروف بزرگ برای نام گذاری متغیرها به کار برد.

همان طورکه در مثال بالا دیدیدMATLAB حاصل متغیرa+b-c را درans ذخیره کرده است .این متغیر به طور پیش فرض برای ذخیره اطلاعات به کار می رود مگر این که کاربر نتایج را در یک متغیر دیگر ذخیره کند.

```
>> D=a+b-c
D =
5
```

اگر بخواهیم چند دستور را در یک خط بنویسیم باید از کاما (،) و سمیکالن (؛) استفاده کنیم .سمیکالن باعث می شود محاسبات انجام شود ولی نتایج نمایش داده نشود.

```
>> a=3 , b=4 ; c=1;
a =
3
```

برای انجام اعمال ساده ریاضی می توان از عملگرهای معمول جمع (+)، تفریق (-) تقسیم(/) ضرب (*)توان (^) استفاده کرد.

برای صرفه جویی در وقت با استفاده از کلیدهای upوDown (کلیدهای جهتی) می توانید دستوراتی که قبلا اجرا شده را مرور کنید.

همچنین سایر کلیدهای ویرایشی (... ,Page Up ,End ,Home) وظایف استاندارد خود را دارا می باشند.

درMATLAB متغیرهای ویژه ای وجود دارد که هر یک مقادیر خاصی را در خود ذخیره میکنند.

ans:برای ذخیره مقادیری که کاربر متغیری را برای ذخیره آنها در نظر نگرفته است .

Pi :مقدار عدد پی ۳،۱۴۱۶

eps:کوچکترین عدد مثبت بزرگتر از صفر

i: عدد موهومی$\sqrt{-1}$

inf:به عنوان علامت بینهایت

NaN یا**nan:** مقدار غیر عددی (نتیجه تمام عملگرها روی NaN ، NaN است)

بهترین روش برای خواناتر شدن یک برنامه استفاده از جملات توضیحی می باشد درMATLAB این جملات بعد از علامت % می آید.

MATLAB عبارت بعد از % را ویرایش نمی کند.

گاهی اوقات یک فرمان ممکن است آن قدر طولانی باشد که نتوان آن را در یک خط نمایش داد . برای حل این مشکل می توان در آخر خط سه نقطه (. . .) گذاشته و ادامه دستور را در خط بعد تایپ کرد.

برای توقف پردازش برنامه از Ctrl + c استفاده کنید.

در MATLAB اعداد با فرمت های مختلفی به نمایش در می آیند. از مهمترین آن ها می توان به Format short اشاره کرد که فرمت پیش فرض است و اعداد را با دقت ۴ رقم اعشار نمایش می دهد

همچنین Format bank که بر اساس سیستم بانکی(دلار و سنت)ایجاد شده اعداد را با دقت ۲ رقم اعشار نمایش می دهد.

برای گرد کردن اعداد روش های مختلفی وجود دارد تمام این روش ها را می توان در MATLAB یافت:

fix:گرد کردن به طرف صفر

floor:گرد کردن به طرف منفی بینهایت

ceil:گرد کردن به طرف مثبت بینهایت

round:گرد کردن به طرف نزدیکترین عدد صحیح

۱-۳-مدیریت فایل ها

Dir:همه فایل های موجود در دایرکتوری رانمایش می دهد .

Cd:این دستور مسیر current directory را مشخص می سازد .

Clear: این دستور تمام یا تعدادی از متغیرها را پاک کرد.

```
>> clear a
>> a
??? Undefined function or variable 'a'.
>> b
b =
4
>> clear
>> b
??? Undefined function or variable 'b'.
```

delete:می توان فایل های مورد نظر و موجود در دایرکتوری جاری را پاک کرد.

به عنوان مثال دستور زیر تمام p-file های موجود در دایرآتوری جاری را پاک می کند.

```
>> delete *.p
```

mkdir:این دستوریک فایل دایرکتوری با نام وارد شده را می سازد .

rmdir: برای حذف دایرکتوری مورد نظر استفاده می شود .

load: متغیر های x , y,z را ازفایل مورد نظر بارگذاری می کند.

load filename X Y Z ...

save: متغیر معرفی شده رادر شاخه current directory ذخیره می نماید.

save('filename', 'var1', 'var2', ...)

open: نام فایل یا متغیر مورد نظر درداخل پرانتز به صورت کاراکتر وارد می نماییم ونرم افزار آن رابا توجه با نوع وپسوند فایل با ابزار مناسب باز می نماید .

open(name)

Dlmwrite: این دستور ماتریس مشخص شده را با برحسب تنظیمات موردنظر درداخل یک فایل متنی می ریزد.

dlmwrite(filename, M, 'attrib1', value1, 'attrib2', value2, ...)

Dlmread: این دستور یک فایل متنی رابرحسب داده های محدود اسکی با تنظیم مورد نظر می خواند که R شماره سطر وc شماره ستون مورد نظر می باشد . می توان از متغییر range=[R1, C1, R2, C2] استفاده کرد که(R1 C1) گوشه سمت چپ بالایی می باشد.

M = dlmread(filename, delimiter, R, C)

Textread: این دستور معادل textscan می باشد و می توان از منوی file گزینه import data را معادل این دستور استفاده کرد. این دستور فایل مورد نظر را با فرمت مورد نظر وبه تعداد N بار ازورودی می خواند.

[A,B,C,...] = textread(filename,format,N)
C = textscan(fid, 'format', N, param, value, ...)

fopen:این دستورفایل مورد نظر(filename) رابادسترسی مطابق (permission)باز می کند و با دستور مخصوص خواندن بیت یا بایت های فایل (machineformat) فایل را درصورت وجود باز ودرصورت عدم وجود در دایرکتوری نرم افزار متلب ایجاد می نماید .

fileID = fopen(filename, permission, machineformat)

قسمت **permission** رشته ای است که نوع دسترسی به فایل را شرح می دهد .که می تواند خواندنی ،نوشتنی ،الحاقی یا به روزرسانی است که به صورت دودویی (باینری) وحالت متنی است.

بازکردن فایل هادرحال دودویی ویژه مطابق جدول ذیل است:

شرح حالت علائم	علائم
فایل رادرحالت خواندن باز می کند	'r'
فایل رادرحالت خواندن بازیا ایجاد می کند	'w'
فایل رادرحالت الحاق داده ها بازیا ایجاد می کند.	'a'
فایل جدید برای حالت خواندن ونوشتن بازیا ایجاد می کند.	'w+'

'a+'	فایل جدید برای حالت خواندن ونوشتن بازیا ایجاد می کند وداده ها به آخر فایل الحاق می شوند.
'A'	الحاق بدون خالی کردن اتوماتیک حافظه
'w'	نوشتن بدون خالی کردن اتوماتیک حافظه

fprintf:این دستور امکان ترکیب رشته و عدد را دربرنامه فراهم می آوردوامکان قالب بندی آن را به شکل دلخواه در خروجی فراهم می آورد علائم %f,%g,%G,%e,%E,%x,%X,%u,%o,%i,%d علائم تعیین فرمت خروجی Format Specifier هستند.علائم \t,\n,\b,... مشابه دستورات c++ دستوراتی برای تعیین حالت چاپ را درخروجی ایجاد می نمایند.

```
fprintf(fileID, format, A, ...
```

fscan:این دستورداده ها را ازیک فایل متنی می خواند وداده هارا به یک آرایه مثل A تبدیل می نماید.

```
A = fscanf(fileID, format)
```

شاید بخواهیم عملیاتی را که در یک دوره انجام داده ایم ذخیره کرده و از آن پرینت گرفته و یا بعدها از آن استفاده کنیم. برای این کار از دستورDiary استفاده می کنیم.
با اجرای دستور diary on ، MATLAB مانند یک دفترچه یادداشت عمل کرده و تمام مطالب موجود درprompt MATLAB ، در یک فایل ذخیره می شود تا هنگامی که diary off اجرا شود.
اگر دستور format compact را وارد کرده Enter بزنیدMATLAB خطوطی را که بصورت خالی بین خروجی قرار می دهد را حذف می کند.عکس این دستور format loose است که خطوط خالی حذف شده را بر می گرداند.
بوسیله دستور format compact می توانیم داده های بیشتری را در پنجره Command window جا دهیم .
همان طور که قبلا توضیح دادشده است با استفاده از دستورات Doc وHelp می توان به متن راهنمای یک دستوریا تابع دست پیدا کرد.به این طریق می توان با عملکرد آن دستور یا تابع آشنا شده و روش های به کاربردن آن را آموخت . همچنین درادامه با سایر دستورات مرتبط آشنا می شوید.

در هنگام اجرای دستورات تکراری یا هنگام آزمایش کردن مقادیر مختلف در یک متغیر ممکن است تایپ دستورات خسته کننده باشد برای حل این مشکل MATLAB یک راه حل دارد و آن استفاده از M-file می باشد.

با استفاده از M-file ها می توانید دستورات را در یک فایل ذخیره کرده و با باز کردن آن همانند آن که آن ها را در خط فرمان تایپ کرده باشید اجرا کنید.

برای ایجاد یک M-file می توانید از گزینه New M-file استفاده کنید.

دستور format

به وسیله این دستور می توان دقت پاسخ ارائه شده نرم افزار را تنظیم کرد.

انواع تنظیمات دستور format مطابق جدول ذیل است:

شرح دستور	حالت دستور
دو رقم اعشار	Format bank
چهار رقم اعشار	Format short
چهار رقم اعشار ممیز شناور	Format short e
بهترین حالت اعشار	Format short g
پانزده رقم اعشار	Format long
پانزده رقم اعشار ممیز شناور	Format long e
بهترین حالت اعشار	Format long g
براساس هگزا دسیمال	Format hex
به صورت کسر منطقی	Format rat
به صورت تابع علامت sign	Format +

۲- آرایه ها در متلب

در MATLAB آ را یه ها به سادگی ایجاد می شوند ساده ترین و ابتدایی ترین روش، تایپ تمام مقادیر بین دو [] می باشد:

```
>> a=[1,2,3,4]
a =
1 2 3 4
```

برای تولید آرایه های بزرگتر استفاده از روش بالا بسیار وقت گیر است MATLAB چند دستور برای تولید آرایه ها دارد:

linspace(a,b,c)

این دستور c نقطه با فاصله های مساوی در بازه [a,b] را برمی گرداند.

logspace(a,b,c)

این دستور نیزc نقطه با فواصل لگاریتمی دربازه a^{10} , b^{10} بر می گرداند.

[a:b:c]

مفهوم کلی این دستور را می توان به این صورت بیان کرد؛ ازa شروع کن b تا b تا جلو برو تا به c برسی دراین روش که گام حرکت نام دارد b می تواند منفی باشد .

با استفاده از دستورات فوق و توابع ریاضی می توان ماتریس های متنوعی تولیدکرد.

```
<<x=0:.5:2*pi;
y=sin(x)

y=

Columns 1 through 10

-۰.۹۷۷۵ -۰.۷۵۶۸ -۰.۳۵۰۸ ۰.۱۴۱۱ ۰.۵۹۸۵ ۰.۹۰۹۳ ۰.۹۹۷۵ ۰.۸۴۱۵ ۰.۴۷۹۴ ۰

Columns 11 through 13

-۰.۲۷۹۴ -۰.۷۰۵۵ - ۰.۹۵۸۹
<<z=tanh(x)

z=

Columns 1 through 10

۰.۹۹۹۸ ۰.۹۹۹۳ ۰.۹۹۸۲ ۰.۹۹۵۱ ۰.۹۸۶۶ ۰.۹۶۴۰ ۰.۹۰۵۱ ۰.۷۶۱۶ ۰.۴۶۲۱ ۰

Columns 11 through 13

۱.۰۰۰۰ ۱.۰۰۰۰ ۰.۹۹۹۹
```

در متلب آرایه ها تنها به آرایه های سطری محدود نمی شوند بلکه می توان آرایه های ستونی نیز معرفی کرد برای اینکه نرم افزارتشخیص دهد که قصد تولید سطر جدید را داریم از سمیکالن(؛) استفاده می کنیم.

```
>> b=[1;3;5;7]
b =
1
3
5
7
>> b=linspace(1,7,2) '
b =
1
3
5
7
```

همان طور که در مثال بالا مشاهده کردید بااستفاده از عملگر ترانهاده (') یک آرایه سطری را به آرایه ستونی تبدیل کردیم.

برای دسترسی به درایه ها از اندیس آن استفاده می شود. b(3) سومین درایه آرایه b را نشان می دهد.

به عنوان مثال

```
<<b(3)
ans=
5
```

b(3)=5 را برمی گرداند.

اگر بخواهیم درایه های خاصی از یک آرایه را انتخاب کنیم می توان با استفاده از دو نقطه به آنها دسترسی داشته باشیم.اگر طول آرایه مشخص نباشد می توان برای دسترسی به آخرین عنصر از end استفاده کرد.

```
>> z(end:-3:4)
ans =
1.0000 0.9998 0.9951 0.9051
```

همچنین می توانیم از یک آرایه با مقادیر یک آرایه دیگر با ترتیب مورد نظر خودمان استفاده کنیم.

```
>> c=[1,5,3,5,1];
>> z(c)
ans =
0 0.9640 0.7616 0.9640 0
```

همان طور که مشاهده کردید با استفاده از آرایه c عناصر اول،پنجم،سوم،پنجم و اول را فراخوانی کردیم .

نکته دیگر که باید به آن توجه کرد این است که اندیس یک عدد صحیح و مثبت است .اگر کاربر یک عدد غیر صحیح و یا منفی را به عنوان اندیس وارد کند نرم افزار متلب یک پیغام خطا بر می گرداند .

```
>> b(2.7)
??? Subscript indices must either be real positive integers or logicals.
>> b(-3)
??? Subscript indices must either be real positive integers or logicals.
```

دو آرایه مفروض a,b را می توان با استفاده از دستورات الحاق سطری [a,b] و الحاق ستونی [a;b] به یکدیگر الحاق کرد.

و به این ترتیب می توان آرایه هایی با سطرها و ستون های متعدد داشت،البته در الحاق سطری تعداد سطرها و در الحاق ستونی تعداد ستون های دو آرایه باید برابر باشد.همچنین می توان تعداد آرایه های الحاقی را به طور دلخواه افزایش داد.

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

```
4 5 6
7 8 9
```

به این ترتیب متوجه شدیدکه آرایه ها در برنامه متلب می توانند دارای سطرها و ستون های متعدد باشند.

حال ممکن است این سوال پیش بیاید که چگونه می توان این آرایه ها را تولید کرد.همان طور که قبلا ملاحظه نمودید برای معرفی ستون ها از فاصله یا کاما (،) و برای سطر ها از سمیکالن (؛) استفاده می شود

راه دیگر ایجاد سطر این است که بعد از معرفی سطر اول با زدن کلیدEnter به خط بعد رفته و به معرفی سطر بعد بپردازید.

```
>> A=[1 2 3;6 5 4]
A =
1 2 3
6 5 4
>> B= [10 11 12
13 14 15]
B =
10 11 12
13 14 15
```

در این مورد به چند نکته باید توجه کرد:

تعداد ستون ها در هر سطر باید برابر باشد در غیر این صورت پیغام خطایی نمایش داده می شود.

تعداد فواصلی که برای جدا کردن اعداد به کار می رود برای متلب مهم نیست.یعنی متلب فضاهای خالی اضافی را تشخیص داده و حذف می کند.

۲-۱-محاسبات عددی آرایه ها

نرم افزار متلب عملگرهای فراوانی برای اعمال بر روی آرایه ها دارد.تعدادی از آنها را در مثال های زیر مشاهده می کنید:

```
>> A
A =
1 2 3
6 5 4
>> A-2
ans =
-1 0 1
4 3 2
>> A*2 + B
ans =
12 15 18
25 24 23
>> A + B
ans =
11 13 15
19 19 19
```

همان طور که در بالا دیدید اعمال ساده ریاضی را می توان روی آرایه ها انجام دهید. A-2 باعث می شود از درایه های آرایه A دو واحد کم شود. همچنین دستور A+B تمام درایه های نظیر به نظیر دو آرایه را جمع می کند.

ضرب و توان ماتریس ها نیز به سادگی قابل تعریف اند:

```
>> A
A =
1 2 3
6 5 4
>> C=[1 2;3 4];
>> C*A
ans =
13 12 11
27 26 25
>> C^2
ans =
7 10
15 22
```

حال اگر بخواهیم تک تک درایه ها را به توان n برسانیم یا درایه های دو آرایه یا ماتریس را نظیر به نظیر در هم ضرب یا تقسیم کنیم باید قبل از این عملگرها یک نقطه(point) قرار دهیم.به مثال های زیر توجه کنید:

```
>> A.*B
ans =
10 22 36
78 70 60
>> B.^A
ans =
10 121 1728
4826809 537824 50625
>> C.^2
ans =
1 4
9 16
```

۲-۲-آرایه های استاندارد

ones(n,m): ماتریسی با ابعاد n در m وبا درایه های یک ایجاد می کند.

```
>> ones(2,3)
ans =
1 1 1
1 1 1
```

zeros(n,m): ماتریس nسطر و m ستون وبا درایه های صفر ایجاد می کند.

```
>> zeros(2,4)
ans =
0 0 0 0
0 0 0 0
```

eye(n): این دستور هم ماتریسی همانی n*m بعدی ایجاد می کند که درایه های روی قطر اصلی آن یک است.

```
 >> eye(3,4)
ans =
1 0 0 0
0 1 0 0
0 0 1 0
```

rand(n,m): این دستور هم یک ماتریس با درایه های اتفاقی بین صفر و یک ایجاد می کند.

```
>> rand(2,3)
ans =
0.9501 0.6068 0.8913
0.2311 0.4860 0.7621
```

randperm(n): این دستور اعداد 1 تا n را به صورت اتفاقی در یک بردار سطری قرار می دهد.

```
>> randperm(8)
ans =
2 4 1 5 8 6 3 7
>> randperm(10)
ans =
10 7 1 8 2 5 9 6 4 3
```

magic(n): این دستور ماتریسی که به ماتریس جادویی معروف است را تولید می کند.ویژگی این ماتریس این است مجموع درایه های هر سطر، ستون و قطر با هم برابر است.

```
>> magic(3)
ans =
8 1 6
3 5 7
4 9 2
```

توجه به این نکته لازم است که دو دستور آخر تنها یک ورودی دارند،اما دستورات اول می توانند دارای دو آرگومان ورودی باشند.در صورتی که دستورات اول با یک آرگومان به کار بروند یک ماتریس مربعی n x n ایجاد می شود.

Primes(n): این تابع اعداد اول کوچکتر یا مساوی n رادرخروجی ارائه می دهد.

Factor(n): این تابع عوامل اول عدد n را به صورت غیرتوانی در خروجی نمایش می دهد .(عدد را به عوامل اول تجزیه می کند).

Factorial(n): حاصل ضرب اعداد صحیح 1 تا n می باشدکه مساوی مقدار فاکتوریل n می باشد. این دستورمعادل دستور prod(1:n) می باشد.اگر n حالت آرایه ای داشته باشد این دستور فاکتوریل همه مقادیر را درخروجی ارائه می دهد.

gcd(a,b): بزرگترین مقسوم علیه مشترک دو عدد را محاسبه می کند.

lcm(a,b): بزرگترین مضرب مشترک دو عدد را محاسبه می کند.

۲-۳-توابع آرایه ای

در نرم افزار متلب دستوراتی وجود دارد که به وسیله آنها می توان اطلاعاتی در مورد یک ماتریس از قبیل تعداد سطر،ستون و تعداد کل عناصر آن را به دست آورد.

```
>> a= [1 2 3
1 2 3]
a =
1 2 3
1 2 3
>> size(a)
ans =
2 3
```

Size(n):این دستور همان طور که از اسم آن مشخص است اندازه(تعداد سطر و ستون)ماتریس را برمی گرداند.دردستوراتی مانند دستور فوق که 2 خروجی یا بیشتر دارند می توان هر خروجی را در یک متغیر ذخیره کرد.

برای این کار به صورت زیر عمل می کنیم:

```
>> [s t]= size (a)
s =
2
t =
3
```

Length(n): این دستور بزرگترین مقدار بین سطر و ستون را برمی گرداند.

```
 >> length(a)
ans =
3
```

numel(n): این دستور هم تعداد عناصر ماتریس (number of elements)را برمی گرداند.

```
>> numel(a)
ans =
6
```

۲-٤-ایجاد تغییر در ماتریس ها

نرم افزار متلب با اختصاص یک اندیس به هر عضو آرایه روش های زیادی را برای ایجاد تغییر در درایه های ماتریس ها به وجود می آورد.

برای دسترسی به عناصر آرایه ها 2 روش وجود دارد.

روش اول با 2 آرگومان ورودی که سطر و ستون را مشخص می کنند.

```
c=
1  0  0
0  2  0
0  0  8
```

```
<<c(3,3)
ans=
 8
```

روش دوم استفاده از یک آرگومان که در این صورت اندیس هر درایه از آرایه تعیین می شود.

```
<<c(9)
ans=
8
```

c(:,n)=m: مفهوم این دستور این است که تمام سطرها در ستون n را مساوی m قرار بده.

```
>>c(:,1)=3
c=
۳  ۰  ۰
۳  ۲  ۰
۳  ۰  ۸
```

c([1 end],[1 end])=m: این دستور یعنی در سطر اول و آخر، مقدارستون های اول و آخر را مساوی m قرار می دهد.

```
 >> c([1 end],[1 end])=4
c=
۴  ۰  ۴
۳  ۲  ۰
۴  ۰  ۴
```

c(n,:)= []: این دستورمقادیرسطر n ام را مساوی صفر قرار می دهد و موجب حذف سطر n ام می شود.

```
<<c(3,:)=[]

c=


۴  ۰  ۴
۰  ۲  ۳
```

تغییراتی که در ماتریس ها می توان ایجاد کرد تنها به تغییر عناصر آن محدود نمی شود بلکه می توان ابعاد،ترتیب و جای درایه ها را تغییر داد.

flipud(a): این دستور ماتریس را از بالا به پایین می چرخاند یعنی جای سطرها را نسبت به سطر وسط عوض می کند.

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

Fliplr(a) : این دستور هم مانند دستور فوق است با این تفاوت که روی ستون ها عمل می کند.

```
>> b=fliplr(a)
b =
3 2 1
6 5 4
```

rot90(b): این دستور ماتریس را 90 درجه در خلاف حرکت عقربه های ساعت می چرخاند.البته اگر این دستور به صورت rot90(a,n) به کار رود تعداد دفعات چرخش را نیز مشخص می کند.

```
>> rot90(b)
ans =
1 4 7
2 5 8
3 6 9
```

triu(a): این دستور ماتریس بالا مثلثی ایجاد می نماید .

```
<<triu(b)
ans=
۱  ۲  ۳
۰  ۵  ۴
۰  ۰  ۷
```

tril(b): این دستور ماتریس پایین مثلثی ایجاد می نماید .

```
<<tril(b)
ans=
۳  ۰  ۰
۶  ۵  ۰
۷  ۸  ۹
```

horzcat(a,b):این دستور به صورت افقی دو آرایه a و b را به هم الحاق می نماید .

```
<<c=horzcat(ans,b)

c=

 ۳  ۰  ۰  ۳  ۲  ۱
۶  ۵  ۰  ۶  ۵  ۴
۹  ۸  ۷  ۹  ۸  ۷
```

vertcat(a,b): این دستور به صورت عمودی دو آرایه a و b را به هم الحاق می نماید .

```
<<d=vertcat(a,b)

d=
```

```
۳  ۲  ۱
۶  ۵  ۴
۹  ۸  ۷
۱  ۲  ۳
۴  ۵  ۶
۷  ۸  ۹
```

Reshape(a,n,m):این دستور آرایه a را به حالت n سطر و m ستون که موردنظربرنامه نویس می باشد تبدیل می کند .

```
<<reshape(d,2,9)
ans=
۴  ۹  ۳  ۵  ۸  ۲  ۶  ۷  ۱
۷  ۱  ۶  ۸  ۲  ۵  ۹  ۳  ۴
```

Repmat(a,n,m): اگر بخواهیم ابعاد یک ماتریس را تغییر بدهیم می توانیم از این دستور استفاده کنیم.البته باید تعداد عنصر ها قبل و بعد از تغییر با هم برابر باشد. این دستور با تکرار یک ماتریس ماتریسی با ابعاد جدید تولید می کند. ماتریس a را به عنوان یک عنصر در نظر گرفته n * m بار تکرار می کند.

```
<< repmat(d,1,2)
ans=
۳  ۲  ۱  ۳  ۲  ۱
۶  ۵  ۴  ۶  ۵  ۴
۹  ۸  ۷  ۹  ۸  ۷
۱  ۲  ۳  ۱  ۲  ۳
۴  ۵  ۶  ۴  ۵  ۶
۷  ۸  ۹  ۷  ۸  ۹
<<repmat(ans(:,1),2,4)
ans=
۱  ۱  ۱  ۱
۴  ۴  ۴  ۴
۱  ۱  ۱  ۱
۴  ۴  ۴  ۴
```

cat(n,a,b): این دستور از دستورات کاربردیMATLAB است . در این دستور aوb دو ماتریس است وn مشخص کننده جهت الحاق می باشد مثال های زیر بهتر مفهوم مورد نظر را می رساند.

مثال سوم نمونه ای از ماتریس های 3 بعدی می باشد،به نحوه نمایش این ماتریس ها دقت کنید.

```
<<d=cat(1,a,b)
d=
۳  ۲  ۱
```

```
۶  ۵  ۴
۹  ۸  ۷
۱  ۲  ۳
۴  ۵  ۶

<<d2=cat(2,a,b)
d2=
۱  ۲  ۳  ۳  ۲  ۱
۴  ۵  ۶  ۶  ۵  ۴
۷  ۸  ۹  ۹  ۸  ۷

<<d3=cat(3,a,b)
=d3(:,:,1)
۳  ۲  ۱
۶  ۵  ۴
۹  ۸  ۷
= d3(:,:,2)
۱  ۲  ۳
۴  ۵  ۶
۷  ۸  ۹
```

kron(a,b): این دستور به ماتریس کرونکراشاره دارد و اگر a ماتریس m*n و b ماتریس p*q باشد ماتریس کرونکر یک ماتریس با ابعاد m*pسطر و n*q ستون خواهد بود و مقادیر درایه ها به صورت ذیل محاسبه می شود :

درایه های ماتریس a در عناصر ماتریس b ضرب می گردد و به درایه های حاصل جایگزین در آرایه a می گردد . برای مثال ذیل داریم:

```
[ X(1,1)*Y X(1,2)*Y
  X(2,1)*Y X(2,2)*Y]
>> a=[1 2;3 4];
>> b=[2 3];
>> kron(a,b)
ans =
2  3  4   6
6  9  8 12
>> kron(b,a)
ans =
2   4   3   6
6   8   9 12
```

مثال اخیر را می توان به این صورت نیز نمایش داد:

```
>> [ b(1)*a , b(2)*a ]
ans =
2   4  3   6
6   8  9 12
```

diag(n): اگربخواهیم عناصر قطر اصلی یک ماتریس را به دست آوریم.

```
>> diag(a)
```

```
ans =
1
2
```

diag(a,n):این دستور قطر n ام ماتریس a را نشان می دهد که n می تواند مثبت یا منفی باشد.

درصورتی که a یک بردار باشد دستور(diag(a ماتریسی ایجاد می کند که قطر اصلی آن بردارa و سایر عناصر آن صفر است.

```
>> c= [1 2 3];
>> diag (c)
ans =
1 0 0
0 2 0
0 0 3
```

Eig(x):این دستور مقادیر ویژه ماتریس x را به خروجی می برد.

Disp(x): مقادیر یک متغیر را بدون نمایش نام آن متغیر چاپ می کند.

```
>> disp('       Corn        Oats        Hay')
disp(rand(5,3))
   Corn       Oats       Hay
  0.8147    0.0975    0.1576
0.9058    0.2785    0.9706
0.1270    0.5469    0.9572
0.9134    0.9575    0.4854
0.8003   0.9649   0.6324
```

Trace(n):این دستورحاصل جمع عناصر قطر اصلی را درخروجی ظاهرمی کند.

```
a=
۱  ۴  ۳
۲  ۵  ۸
۱  ۷  ۴
<<trace(a)
ans=
10
```

rank(a):مرتبه ماتریس a را که بزرگترین درجه ای که ماتریس می تواند مستقل باشد رانشان می دهد به عبارتی تعداد سطریاستون که مستقل خطی هستند.

minfo(a):این دستور اطلاعاتی را درباره خود ماتریس درخروجی ظاهر می نماید.

```
<<minfo(a)
3 rows  3 cols: regular MATLAB matrix
```

dot(A,B):حاصل ضرب داخلی دو بردار a, b را به صورت عددی درخروجی ارائه می دهد .

cross(A,B):حاصل ضرب خارجی دوبردار a,b را به صورت یک بردار درخروجی ارائه می دهد.

Complex(a,b):این دستور عدد مختلط a+bi را با قسمت های حقیقی و موهومی ایجاد می کند .

angle(z):زاویه هر عنصر مختلط آرایه ها را بر حسب رادیان برمی گرداند.

conj(z):عدد مزدوج متناظر با عدد مختلط z را در خروجی ارائه می دهد .

real(z):قسمت حقیقی عناصر آرایه مختلط z را برمی گرداند.

۲-۵-مرتب کردن آرایه ها

یکی از امکانات مفید نرم افزار متلب وجود دستوری برای مرتب کردن آرایه ها می باشد.در زیر ابتدا یک ماتریس درایه های بین 0 و 20 تولید می کنیم و سپس با استفاده از دستور(sort(a,n که n مشخص کننده سطر یا ستون می باشد آن را مرتب می کنیم.

```
<<x=fix(20*rand(3,4))
x=
۱۹  ۵  ۱۸  ۱۶
۳  ۱۰  ۱۲  ۱۸
۱۹  ۱۹  ۱  ۲


<<sort(x,1)
ans=
۳  ۵  ۱  ۲
۱۹  ۱۰  ۱۲  ۱۶
۱۹  ۱۹  ۱۸  ۱۸

<<[sor,pos]=sort(x,2)
sor=
۱۹  ۱۸  ۱۶  ۵
۱۸  ۱۲  ۱۰  ۳
۱۹  ۱۹  ۲  ۱
pos=
۴  ۲  ۱  ۳
۱  ۲  ۳  ۴
۴  ۳  ۱  ۲
```

همان طور که مشاهده می کنید در صورتی که به نرم افزار متلب دستور دو خروجی بدهیم آنگاه دو ماتریس را برمی گرداند؛ ماتریس اول همان ماتریس مرتب شده و ماتریس دوم اندیس مربوط به درایه های مرتب شده را نمایش می دهد.به عبارت دیگر ماتریس pos نشان دهنده مکان درایه قبل از مرتب شدن می باشد.

در این گونه دستورات در صورتی که n توسط کاربر مشخص نشود 2 حالت پیش می آید .

1- در صورتی که ماتریس 2 بعدی یا یک بردار ستونی باشد ستون ها مورد بررسی قرار می گیرند.

2- در صورتی که ماتریس یک بردار سطری باشد سطرها بررسی می شوند.

۲-٦-جستجو در آرایه ها

درMATLAB دستورات زیادی برای جستجو در آرایه ها، پیدا کردن عناصر خاص و ... وجود دارد .در زیر به برخی از آنها اشاره می شود.

nnz(a):تعداد عناصر غیرصفر ماتریس a را نمایش می دهد .

```
<<a=[2 0 3;4 6 2; 0 2 7];
<<nnz(a)
ans=
۷
```

find(a): اندیس درایه های غیرصفرماتریس a رانمایش می دهد.

```
<<find(a)
ans=
۱
۲
۵
۶
۷
۸
۹
```

nonzeros(a): این دستور خود درایه های غیر صفر را درخروجی نمایش می دهد.

```
<<nonzeros(a)
ans=
۲
۴
۶
۲
۳
۲
۷
```

all(a)، **all(a,n)** : در صورتی که تمام درایه های سطر یا ستون(بستگی به n دارد) غیر صفر باشند مقدار یک و و در غیر این صورت صفر را به خروجی می برد.

```
<<all(a)
ans=
۱  ۰  ۰

<<all(a,2)
ans=
```

۰
۱
۰

any(a,2): این دستور در صورتی که یکی ازعنصرها غیر صفر باشند مقدار یک و در صورتی که همه صفر باشند مقدار صفر را برمی گرداند.

```
>> any(a,2)
ans =
1
1
1
```

unique(a):این دستور یک بردار شامل کلیه عناصرغیرتکراری a را که به ترتیب صعودی مرتب شده اند را در خروجی نمایش می دهد.

```
<<unique(a)
ans=
۰
۲
۳
۴
۶
۷
```

۲-۷-دستورات منطقی

islogical():این دستور تعیین می کند که کدام ورودی آرایه منطقی است و برای مقادیر درست (true) عدد 1وبرای مقادیرنادرست (false) عدد 0 را برمی گرداند .

isa():این دستورتعیین می کند که آیا ورودی شی ای از کلاس گرفته شده می باشد .درصورتی که ورودی متعلق به کلاس باشد جواب منطقی درست(true) 1ودرصورت عدم قرارگیری درکلاس جوای منطقی نادرست (false) 0 را درخروجی ارائه می دهد.

```
<<isa(rand(3,4),'double')

ans=

 1
```

isempty(a): این دستور شرط منطقی رابررسی کرده و در صورتی که ماتریس یک ماتریس تهی باشد مقدار 1 را بر می گرداند.

```
<<isempty(a)
```

```
ans=
٠
```

ismember(b,a):درصورتی که اعضای ماتریس b عضو a نیز باشد مقدار 1 و درغیر این صورت مقدار 0 را بر می گرداند.

```
<<b=[ 2 3 5 7 11]
b=
١١  ٧  ٥  ٣  ٢
<<ismember(b,a)
ans=

٠  ١  ٠  ١  ١
```

isequal(a,b):این دستور برای مقایسه دو ماتریس a و b به کار می رود در صورتی که دو ماتریس برابر باشند مقدار یک را برمی گرداندو درغیراینصورت مقدار صفر را می دهد.

issorted(a):این دستور اگر مقادیر ماتریس a در حالت مرتب شده باشد مقدار یک و درغیراین صورت مقدار صفر را درخروجی ارائه می کند.

isprime(a):این دستور برای آرایه هایی که عد اول هستند عدد یک و برای آرایه های غیر اول عدد صفر را درخروجی نمایش می دهد .

isreal(a):این دستور برای اعدادی که شامل قسمت موهومی نمی باشند عدد 1و درغیر این صورت عدد 0 را برمی گرداند.

```
x=3+4i;
y=5-4i;
isreal(x+y)

ans =
    1
```

۲-۸-ماتریس به عنوان ضرایب چند جمله ای

یک دیگر از کاربردهای ماتریس ها استفاده از آنها به عنوان ضرایب یک چندجمله ای است.

فرض کنید تعدادی داده آماری دارید و می خواهید برای ارتباط دادن آنها با یکدیگر تابعی را پیدا کنید MATLAB این کار را به راحتی و به وسیله **polyfit(x,y,n)** انجام می دهد در اینجا x, y داده ها وn مشخص کننده درجه چندجمله ای مورد نظر است.نتیجه این تابع به صورت یک ماتریس است.

```
<<x=[2 4 5 8 3]
<<y=[2 5 3 8 6]
<<p=polyfit(x,y,3)
p=
  ۰.۳۹۲۹    ۵.۶۵۳۶    - ۲۴.۵۲۶۸    -۲۷.۵۲۲۰
```

polyval(p,x):از این تابع برای بدست آوردن حاصل یک چند جمله ای مثل p به ازای مقادیر مشخص x استفاده می شود .

```
<<polyval(p,2.5)
ans=
۴.۵۹۸۶
```

polyvalm(p,x):این تابع با جایگزین کردن ماتریس x در چند جمله ای p نتیجه را با حساسیت ماتریسی درخروجی نمایش می دهد.

Polyint(p,k):این دستور تابع انتگرال چند جمله ای p را به ازای مقدار ثابت k نشان میدهد درحالتی که k ذکر نشود به طور پیش فرض k=0 فرض می شود .

```
<<polyint(p3)

ans=


  ۰.۲۵۰۰   -۰.۳۳۳۳  ۰  ۱.۰۰۰۰  ۰
```

polyfit:نقاط وارد شده را با بهترین حالت روی منحنی با درجه معلوم n برازش می دهد.

```
p = polyfit(x,y,n)
```

x,y مختصات نقاط وارد شده و n درجه چند جمله ای برازش شده می باشد.برای انجام صحیح این دستورلازم است از تابع کمکی ginput(n) استفاده نماییم . این دستور در فضای ترسیم منتظرگرفتن نقاط به تعداد n می ماند.که با انتخاب هرنقطه مقادیرx,y درنظرگرفته می شود.

```
[x,y]=ginput(10);
f=polyfit(x,y,3);
z=polyval(f,x);
plot(x,y,x,z);
```

roots(p):ازاین دستور برای محاسبه ریشه های یک چند جمله ای استفاده می شود .

```
<<p=[2 4 -5 3]
ans =
  -3.0000
   0.5000 + 0.5000i
   0.5000 - 0.5000i
```

poly(r):این دستور عکس دستورroots(p) می باشد.یعنی با داشتن ریشه های یک چندجمله ای می توانید ضرایب آن را به دست آورید.

conv(p1,p2,...):این دستور برای ضرب کردن چند جمله ها به کاربرده می شود .

```
>> p1=[1 0 2 -1];
>> p2=[1 3];
>> conv(p1,p2)
ans =
1 3 2 5 -3
```

deconv(p1,p2):این دستور برای تقسیم دو چند جمله ای به هم استفاده می شود .

```
>> deconv(p1,p2)
ans =
1.0000 -2.0000 5.5000
```

جمع و تفریق چندجمله ای ها نیز به سادگی جمع و تفریق ماتریس ها می باشد.البته دو ماتریس باید هم مرتبه باشند.

```
>> p3=[1 -1 0 1];
>> p1 + p3
ans=
۲ -۱ ۲ ۰
```

Ployder(a): این تابع ماتریس ضرایب مشتق a را نمایش میدهد.

Ployder(a,b): این تابع ضرایب حاصل ضرب مشتق a و مشتق b را نمایش می دهد.

```
<<a = [3 6 9];
b = [1 2 0];
<<polyder(a,b)
ans=
 ۱۲ ۳۶ ۴۲ ۱۸
```

Residue(a,b):این تابع خارج قسمت تقسیم چند جمله ای a را به b به پایه باقی مانده تبدیل می کند ونمایش می دهد.

```
<<residue(p1,p3)
ans=
۰.۴۲۴۲ - ۰.۹۵۶۶ i
۰.۴۲۴۲ + ۰.۹۵۶۶ i
  - ۰.۹۱۳۲
```

cumtrapz(Y):مجموع تقریبی بردار یا ماتریس y را با استفاده ازروش فواصل مساوی ذوزنقه ای محاسبه می نماید.

diff(X,n): مشتق بردار یا ماتریس برحسب x را تا مرتبه n ام در خروجی ارائه می دهد.

gradient(F): بردار گرادیان تابع f را برحسب متغیرهای آن درخروجی نمایش می دهد.

۲-۹-ماتریس به عنوان مجموعه

یکی دیگر از کاربردهای گسترده ماتریس ها در نظر گرفتن آنها به عنوان یک مجموعه می باشد.

```
<<a=randperm(6)
a=
۲  ۳  ۱  ۵  ۴  ۶
<<b=2:3:9
=b
۸  ۵  ۲
```

union(a,b):این دستوراجتماع دو مجموعه a , b را به خروجی می برد.

```
<<union(a,b)
ans=
۸  ۶  ۵  ۴  ۳  ۲  ۱
```

intersect(b,a): این دستور اشتراک دو مجموعه a , b را به خروجی می برد.

```
<<intersect(b,a)
ans=
۲  ۵
```

setdiff(a,b):این دستور تفاضل مجموعه a و b را درخروجی ارائه می کند . عناصری از مجموعه a که در مجموعه b وجود ندارند. به عبارتی مجموعه (a-b)

```
<<setdiff(a,b)
ans=
۱  ۳  ۴  ۶
```

setxor(a,b):این دستور اعضایی که یا فقط درa هستند یا فقط در b .به عبارت دیگر اجتماع دو مجموعه منهای اشتراک دو مجموعه (تفاضل متقارن a, b)

```
<<setxor(a,b)
ans=
۸  ۶  ۴  ۳  ۱
```

۲-۱۰-محاسبات ماتریسی

inv(a): معکوس ماتریس a را درخروجی نمایش می دهد.

```
<<a=[2 4 8;4 3 9;1 0 5]
a=
۲  ۴  ۸
۴  ۳  ۹
۱  ۰  ۵
<<inv(a)
ans=

-۰.۳۱۵۸        ۰.۵۲۶۳        -۰.۳۹۴۷
```

۰.۲۸۹۵ - ۰.۰۵۲۶ - ۰.۳۶۸۴

۰.۰۷۸۹ - ۰.۱۰۵۳ ۰.۲۶۳۲

det(a):دترمینان ماتریس a را درخروجی نمایش می دهد.

<<det(a)

ans=

-38

a':ترانهاده ماتریس a را در خروجی نمایش می دهد .

<<a'

ans=

۲ ٤ ۱
٤ ۳ ۰
۸ ۹ ۵

nargin(fun):تعداد آرگومان های معین ورودی هر تابع را درخروجی ارائه می دهد.

nargout(fun): تعداد آرگومان های معین خروجی هر تابع را درخروجی ارائه می دهد.

۳-رسم نمودار ها در نرم افزار متلب

دسته دیگر توابع MATLAB توابع مربوط به رسم نمودار می باشند.

نمودارها قادر به انتقال اطلاعاتی هستند که شاید خیلی از جداول و لیست ها قادر به انتقال آن نباشند.به همین دلیل این بخش را به معرفی توابع مربوط به رسم نمودار اختصاص دادیم .

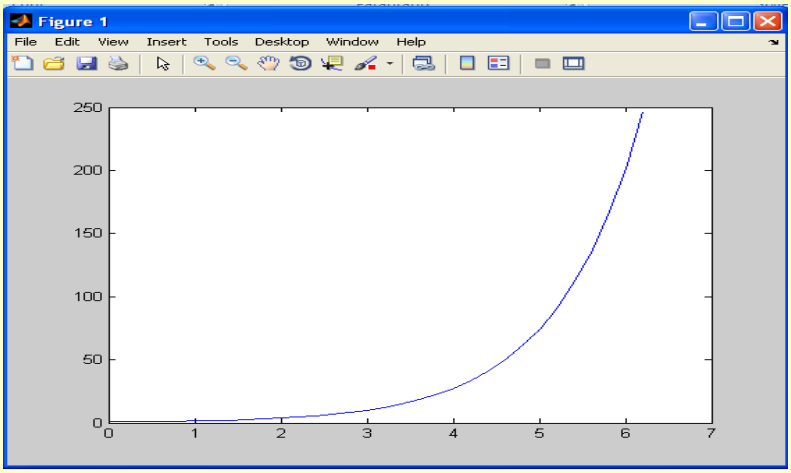
plot :متداول ترین تابع رسم نمودارهای دو بعدی این تابع می باشد.این تابع مجموعه ای از آرایه های داده ها را بر روی محورهای مختصات رسم کرده و نقاط تعیین شده را با خطوط مستقیم به هم متصل می کند.

```
<<x = 0:0.2:2*pi;
<< y=cosh(x)
<<plot(x,y)
```

در مثال بالا x محور افقی و y محور عمودی را می سازند(در تابع آرگومان اول محور افقی و آرگومان دوم محور عمودی را مشخص می کند.)

تابع plot پنجره گرافیکی figure را باز میکند ،سپس اندازه محورهای مختصات را مطابق داده ها تنظیم می کند بعد از رسم نقاط آنها را با خطوط راست به یکدیگر متصل می کند.

در زیر نتیجه دستورات بالا و پنجره figure را مشاهده می کنید.

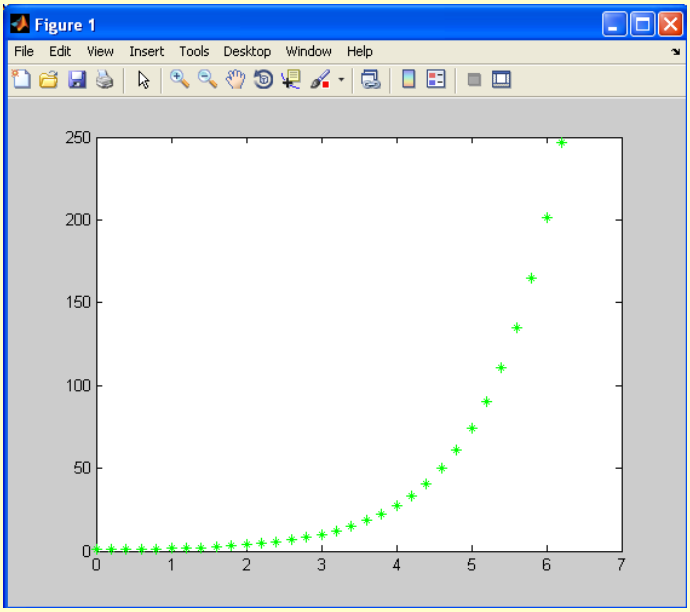


تابع plot را می توان به همراه آرگومان سومی نیز به کار برد این آرگومان که پس از x , y می آید یک رشته کاراکتری است(' ') که مشخص کننده نوع خطوط و رنگ آنها می باشد، این رشته شامل یک یا چند کاراکتر از جدول زیر است.

شکل خط نمودار	علائم	شکل خط نمودار	علائم
triangle (down)	v	point	.
triangle (up)	^	circle	o
triangle (left)	<	x-mark	x
triangle (right)	>	plus	+
hexagram	h	star	*
pentagram	p	square	s
dashdot	-.	diamond	d
dotted	:	dashed	--
		solid	-

رنگ نمودار	انتخاب رنگ	علائم
آبی	blue	b
سبز	green	g
قرمز	red	r
آبی فیروزه ای	cyan	c
بنفش	magenta	m
زرد	yellow	y
مشکی	black	k

<<plot(x,y,'g*')

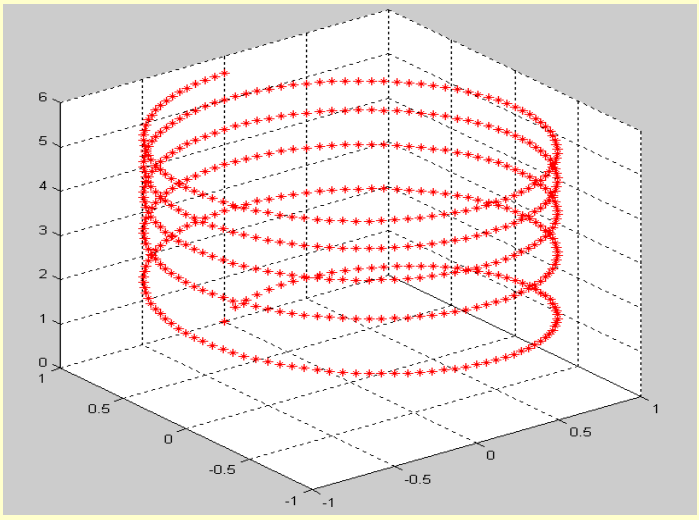


Plot3:برای ترسیم نمودارهای سه بعدی برای مجموعه ای از نقاط از این تابع استفاده می شود.

```
plot3(X1,Y1,Z1,LineSpec,...)
```

دراین دستور X1,Y1,Z1 مجموعه نقاط و LineSpec ،سبک ،علامت ورنگ خط نمودار رامشخص می نماید.

```
t = 0:pi/50:10*pi;
plot3(sin(t),cos(t),t.^0.5,'*r')
grid on
axis square
```



۳-۱-رسم سطوح ولایه ها

لایه ها و سطوح یک فضای پیوسته می باشند که با داشتن ضابطه مشخصی بین متغیرها می توانیم درفضای سه بعدی اقدام به رسم آنها نماییم .

Peaks:این تابع درخود نرم افزار متلب قرار گرفته تا بتوانیم به راحتی اقدام به رسم سطوح ولایه های دلخواه نماییم.

```
[X,Y,Z] = peaks(n);
```

دردستور بالا n به عنوان دقت ترسیم می باشد .جهت رسم سطوح لازم است که کلیه نقاط موجود به فاصله یکسان ازهم قرار گیرند و مقادیر x,y,z به صورت تابع بیان شوند.

mesh(X,Y,Z):چنانچه یک تابع فضایی از Y و X مانند Z داشته باشیم این دستور نقاط فضائی با مختصات X,Y,Z، را طوری به هم وصل میکند که یک شکل فضایی پدید آید .

Meshgrid

این دستوریک شبکه(خطوط عمود به هم) برای رسم سطوح می سازد

این دستور برای ساختن یک شبکه همگن وکامل ساخته شود که برای ساختن سطوح ولایه ها لازم می باشد

```
[X,Y,Z] = meshgrid(x,y,z)
```

اگر b وa به ترتیب بردارهای n,m عنصری باشند عبارت [X,Y] = meshgrid(a,b) باشند ماتریس X را با ردیف هایی مساوی a در m ردیف و ماتریس y را در با ستون های مساوی b در n ستون می سازد.درنتیجه ماتریس های x, y دو ماتریس همسان خواهند بود.

Surf

این دستور سطوح را با رنگ و نور مشخص می کند به گونه ای که می توان با نورپردازی به سطوح عمق بخشید.

Contour

منحنی همسان (کانتور) نقاط هم ارتفاع یا هم وزن یا هم پتانسیل را به هم وصل می کند.

منحنی های همسان (کانتور) همدیگر را قطع نمی کنند. دایره های کوچک تر دارای مقادیر کمینه یا بیشینه هستند.

Meshc

این دستور منحنی mesh را به همراه منحنی همسان (کانتور) آن یکجا رسم می کند.

Surfc

این دستور سطوح surf را به همراه منحنی همسان (کانتور) آن دریکجا رسم می کند.

Contour3: این دستور منحنی همسان(کانتور) سه بعدی را نمایش می دهد . یعنی هر منحنی همسان(کانتور) درارتفاع مربوط به خود نمایش داده می شود.

توابعی در نرم افزار متلب وجود دارند که به توابع آسان ترسیم (easy plotting)معروف هستند و فقط با معرفی تابع نمودار آنها رسم می شود ونیازی به تعریف دامنه ندارند.

ezplot: این دستور به طور پیش فرض و بدون نیاز به بازه نمودار تابع را به صورت دوبعدی رسم می کند.

ezplot3: این دستور با تعریف هر چند متغیر به عنوان ورودی نمودار تابع به صورت سه بعدی رارسم می کند.

ezmesh:این دستور یک شبکه برای یک تابع سه بعدی تعریف شده می سازد .

ezsurf:این دستور نیز ترسیم سطوح را به صورت رنگ و سایه انجام می دهد.

polar(t,r,s):این دستور برای رسم مختصات قطبی با پارامترهای t که مشخص کننده زاویه و rکه مشخص کننده مقدار شعاع و sکاراکتوری است که مشخص کننده رنگ و نوع خط رسم می باشد.

[THETA,RHO,Z] = cart2pol(X,Y,Z) : این دستور برای تبدیل مختصات دکارتی به مختصات قطبی مورد استفاده قرار می گیرد.

[X,Y,Z] = pol2cart(THETA,RHO,Z):این دستور برای تبدیل مختصات قطبی به مختصات دکارتی مورد استفاده قرار می گیرد.

sphere = [X,Y,Z]این دستور مختصات فضائی یک کره را داخل سه ماتریس قرار میدهد.

cylinder = [X,Y,Z]این دستور مختصات فضائی یک استوانه را داخل سه ماتریس قرار میدهد.

view = [a b] زاویه دید فعلی نموداررا میدهد عدد a زاویه چرخش افقی و عدد b زاویه چرخش عمودی شکل را نسبت به دید مستقیم از روبرو میدهند.

دستور Getframe:با این دستور میتوان از هر یک از زوایای گراف یک عکس گرفت.

۳-۲-بر چسب ها، تنظیمات

xlabel و ylabel: این دستورات برچسب محورها را مشخص می کنند.

```
xlabel('string')
```

title: این دستور عنوان را بالای نمودار قرار می دهد.

title('string')

legend : این دستور راهنمای نمودار را نشان میدهد که در صورت رسم چند نمودار روی یک پنجره می تواند مفید باشد.

grid on: این دستور خطوط شبکه ای را روی نمودار فعال می کند

grid off:این دستور خطوط شبکه ای رااز روی نمودار آنها، حذف می کند.

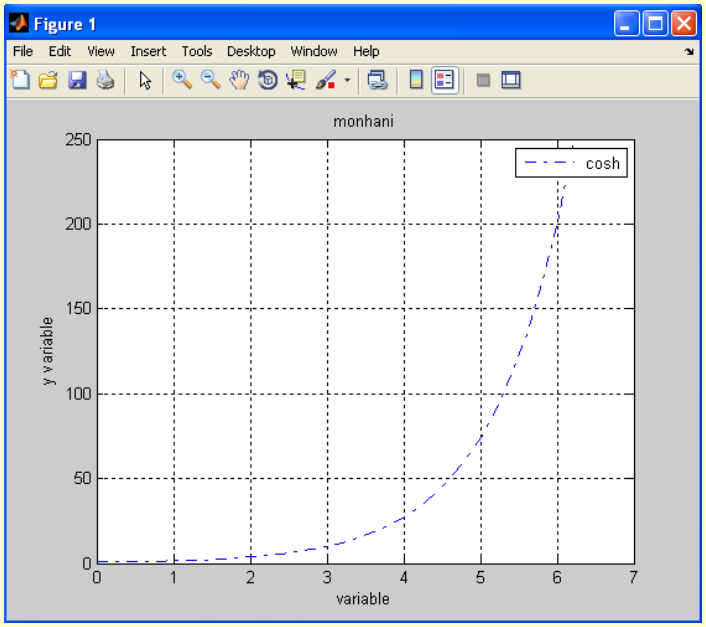
Text: اگر بخواهیم متنی را روی نمودار قرار دهیم از این دستور استفاده می کنیم.

clf: این دستور پنجره figure را پاک می کند .

clc: این دستور نیز پنجره command window را پاک می کند .

حال در مثال زیر روش استفاده از این دستورات را مشاهده می کنید.

```
>> clf
>>  plot (x,y,'-.')
>>  xlabel (' variable ')
>>  ylabel (' y variable')
>>  title (' monhani ')
>> legend (' cosh')
>>  grid
```



grid: این دستور در صورتی که به تنهایی به کار رود ،در صورتی که شبکه ها روشن باشد آن را خاموش و در صورت خاموش بودن آنها را روشن می کند.

:text(x,y,z,'string')

در دستورtext آرگومان اول و دوم وسوم مختصات ابتدای متن و آرگومان بعدی به صورت رشته متن مورد نظر ما است.

درصورتی که مختصات متن را ندانیم می توانیم از دستور gtext استفاده نماییم با اجرای این دستور خطوط متقاطعی روی صفحه نمایش داده می شود و مکان مورد نظر با کلیک ماوس تعیین می شود.

حالت کلی این دستور به این شکل است کهTEXT متن مورد نظر است.

```
gtext ( 'TEXT' )
```

همان طور که مشاهده نمودید تقریبا در تمام دستورات فوق از رشته های کراکتری استفاده می شود .

نرم افزار متلب علاوه بر رشته های معمولی امکاناتی دارد تا بتوان متن هایی شامل کراکترهای ویژه مثل (∂ ∞ ≠)و در چند خط، همچنین عبارات توان دار و اندیس دار را به نمودارها اضافه کرد.

اضافه کردن کاراکترهای ویژه به راحتی انجام می گیرد.با قرار دادن کد مربوط به نمادها میتوان آن را به متن اضافه کرد.

برای ایجاد متن های چند خطی می توانید از آرایه های رشته ای به صورت زیر استفاده کنید.

```
text ( {'LINE1' , 'LINE2' } )
```

برای قرار دادن توان بر روی یک عبارت از علامت توان بعد از عبارت استفاده می شود.در صورتی که عبارتی که در توان قرار می گیرد بیش از یکدکاراکتر باشد آن را بین دو { } قرار می دهیم .و برای ایجاد اندیس ازکاراکتر '_' استفاده می کنیم.

حتی با استفاده از دستورfontsize نیز می توان اندازه متن را نیز مشخص کرد.

همچنین ممکن است تنها نمایش قسمتی از نمودار برای ما مهم باشد .دستور axis با مشخص کردن حدود محورها این کار را انجام میدهد.همان طور که در زیر می بینید آرگومان ورودی دستور شامل یک برداراست که مشخص کننده حدود محورها می باشد.

```
axis ( [XMIN XMAX YMIN YMAX] )
```

بسیاری از دستوراتی که در بالا توضیح داده شد بدون تایپ در پنجرهcommand و از طریق منوی Insert واقع در پنجره figure قابل دسترسی هستند.اگر احتیاجی به یاد گرفتن دستورات بالا نمی بینید می توانید به این طریق عمل کنید.

تا به حال نمودارهایی را رسم کردیم که محورهای مختصات آنها به صورت خطی تقسیم بندی شده بود؛ولی در برخی از مواقع لازم است که یک یا هر دو محور را با تقسیمات لگاریتمی نمایش دهیم. برای این کارنیز دستوراتی وجود دارد.

از تابع semilogx برای نموداری که محورx آن برحسب مقدار لگاریتمی تقسیم بندی شده و از تابع semilogyبرای نموداری با محورy لگاریتمی استفاده کنید.

همچنین تابع loglog نموداری رسم می کند که هر دو محور آن لگاریتمی است. آرگومان های ورودی این توابع مانند تابع plot میباشد .

۳-۳-نمودارهای چندتایی

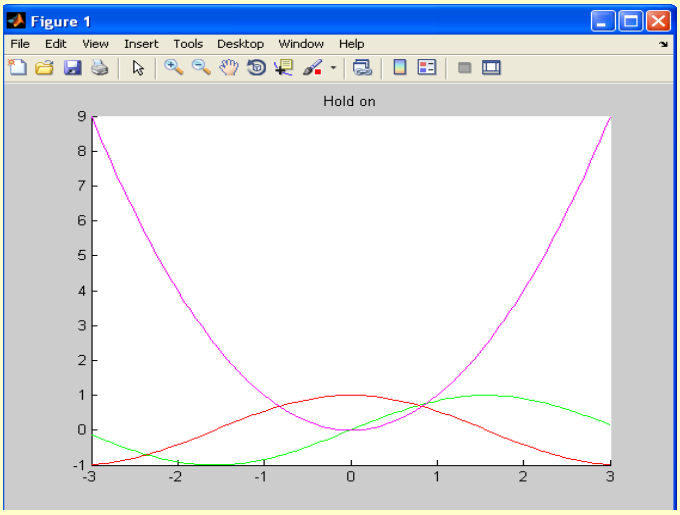
تا اینجا دستورات متنوعی برای رسم نمودار هاآموختیم.ولی امکان دارد که بخواهیم چند نمودار را به طور همزمان روی یک شکل داشته باشیم اما می دانیم که این خواسته با توجه به این که پنجره figure با رسم نمودار جدید پاک می شود ونمودارجدید جایگزین قبلی می شود به روش معمولی امکان پذیر نیست.

در زیر چند روش را برای این کار بیان می کنیم.

روش اول :

در این روش از دستور hold استفاده می شود این دستور محتویات پنجره figure را نگه داشته و نمودار جدید را روی نمودار قبلی رسم می کند در این روش با توجه به این که نمودارها به یک رنگ رسم می شوند بهتر است رنگ و نوع خط نمودار توسط کاربر مشخص شود.

```
>> x=linspace(-3,3);
>> y=sin(x); s=cos(x);t=x.^2;
>> hold on
>> plot(x,y,'g')
>> plot(x,s,'r')
>> plot(x,t,'m')
>> title('Hold on')
>> hold off
```

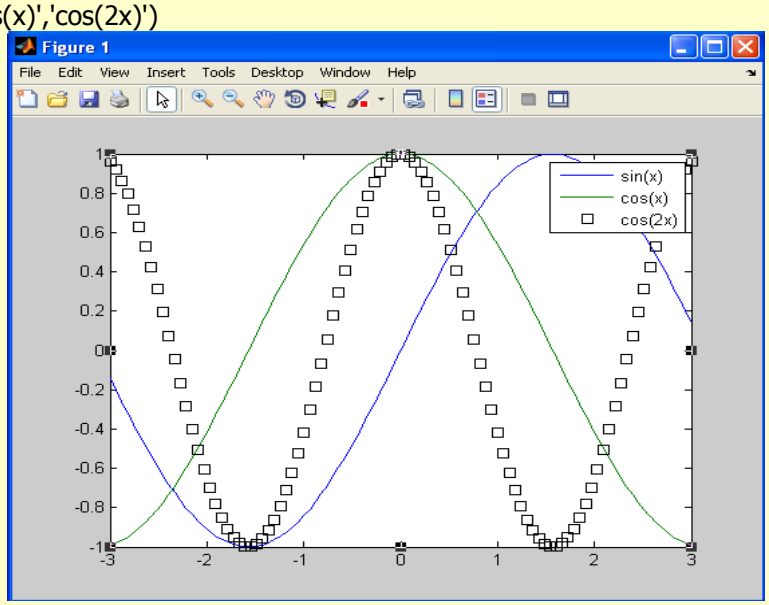


روش دوم :

در این روش از تابعplot استفاده می شود به طوری که در این تابع می توان بعد از جفت آرگومان اول ،جفت آرگومان مربوط به نمودار بعدی را به عنوان آرگومان های بعدی وارد کرد به این ترتیب این تابع می تواند بیشمار آرگومان ورودی داشته باشد.

نرم افزار متلب این نمودارها را با رنگ های مختلف رسم می کند ..در صورتی که بخواهیم نوع خط و ...را مشخص کنیم باید بعد از هر جفت آرگومان این کار را انجام دهید.

```
<<plot (x,y,x,s,x,cos(2*x),'sk')
legend('sin(x)','cos(x)','cos(2x)')
```



روش سوم :

در این روش دو نمودار با محورx مشترک و محورy مختص به خود که تقسیم بندی متفاوتی دارند رسم می شوند این کار توسط تابع plotyy انجام می گیرد.این تابع حداکثر دو نمودار را رسم می کند، به این ترتیب شوداین دستور دارای دو جفت آرگومان ورودی است.حالت کلی آن را در زیر می بینید:

```
<<plotyy(x1,y1,x2,y2,' fun1' ,' fun2')
```

دو آرگومان آخر مشخص کننده نوع محورهای مختصات برای نمودار اول و دوم می باشد؛ و می تواند یکی از موارد زیر باشد.

semilogx, semilogy, plot, loglog, stem

در این روش نمی توان به سادگی تابع plot نوع خطوط و رنگ آنها و ... را مشخص کرد.برای این کار باید از اشاره گرها یا روش های دیگر استفاده کرد .

روش چهارم :

در این روش از پنجره های متعدد استفاده می شود.به این طریق که برای هرتابع رسم نمودار از دستور figure(n) استفاده می کنیم که n مشخص کننده شماره پنجره ی است که برای فراخوانی پنجره از آن استفاده می شود. این دستور پنجره figure جدیدی را باز کرده و نمودار را در این پنجره رسم میکند.

دستورات زیر را تایپ کنید و نتیجه آن را مشاهده کنید.

```
>> figure(1)
>> x=linspace(0,5);
>> y=sin(x);s=cos(x);
>> plot(x,y,'g')
>> figure(2)
>> plot(x,s)
```

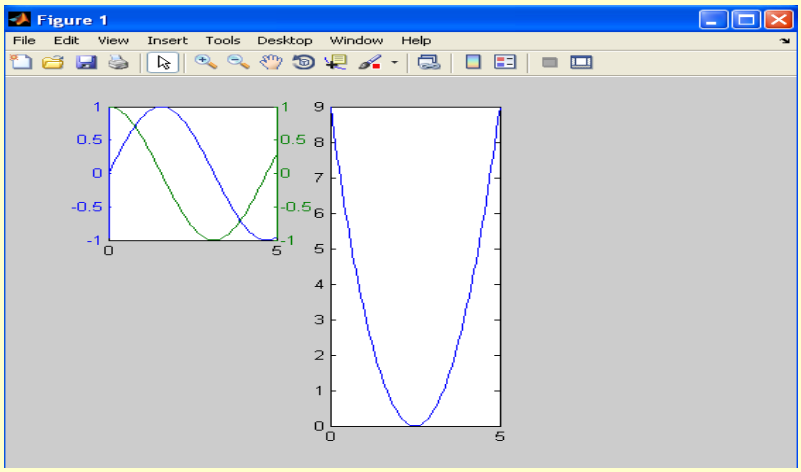
روش پنجم :

در این روش پنجره figure را به چند قسمت تقسیم کرده و هر نمودار را در یکی از این قسمت ها رسم می کنیم .این تقسیم توسط دستور subplot انجام می شود. حالت کلی این دستور به صورت زیر است :

subplot (m,n,p) or subplot(mnp)

این دستور پنجره figure را به یک ماتریس mxn تقسیم می کند و p امین خانه آن را انتخاب می کند.شماره هر خانه به صورت ردیفی تعیین می شود.

```
>> subplot (2,3,1)
>> plotyy (x,y,x,s)
>> subplot (1,3,2)
>> plot (x,t)
```



روش ششم :

در این روش نمودار جدید با محورهای جدید ولی با مقیاس متفاوت روی نمودار قبلی قرار می گیرد . تابع مورد استفاده در این روش axes می باشد حالت کلی آن به صورت زیر است :

axes (' position' , [left, bottom, width, height])

این دستور دارای دو آرگومان ورودی است ؛آرگومان اول یک رشته کاراکتری به صورت بالا و آرگومان بعدی یک بردار است .دو عنصر اول بردار مشخص کننده مکان نمودار جدید و دو عنصر بعدی مشخص کننده اندازه آن است.

این دستور مختصات (0،0)را برای گوشه پایین سمت چپ و(1،1)را برای گوشه بالا سمت راست در نظرمی گیرد.

٤-توابع و متغیرها

MATLAB چند دستور برای آگاهی کاربر از متغیرها و فایل های موجود دارد که در زیر به آنها اشاره می شود.

What:این دستور کلیه m-file های موجود دردایرکتوری های ذخیره شده درپوشه MATLAB رانمایش می دهد.

برای تغییر دایرکتوری می توانید همانند سیستم عامل dos از دستورcd استفاده کنید .

٤-١-ساخت function file

تا به حال تنها از توابعی استفاده می کردیم که قبلا برای نرم افزار تعریف شده بود ولی ممکن است این توابع نتوانند نیازهای ما را پاسخ دهند، یا بخواهیم توابعی باکاربری خاص تعریف نماییم .

یک تابع(function file) مانند یک M-file است با این تفاوت که خط اول آن به صورت زیر است:

function [out1, out2, ...] = funname(in1, in2, ...)

به طور ساده تر

function [outputs]= name(inputs)

این خط مشخص می کند که اینM-file یک تابع است . همچنین تعداد ورودی ها و خروجی ها را مشخص کرده و هر یک را در یک متغیر قرار می دهد.در صورتی که تنها یک ورودی داشته باشیم نیازی به کروشه[] نیست . name نیز نام تابع را مشخص می کند.

بهتر است برای خواناتر شدن برنامه از عبارات توضیحی استفاده کنیم.این عبارات باید ورودی ها و خروجی ها را مشخص کند.همچنین می توان نام برنامه نویس و تاریخ نوشتن آن را نیز مشخص کرد .

این خطوط با اجرا دستور >>help name به نمایش درمی آیند .

به عنوان مثال M-file زیر تابع محاسبه میانگین وانحراف استاندارد را ارائه می کند.

```
function [mean,stdev] = stat(x)
n = length(x);
mean = sum(x)/n;
stdev = sqrt(sum((x-mean).^2/n));
```

٤-٢-انواع عملگرها

٤-٢-١عملگرهای رابطه ای

این عملگرها شامل موارد زیر می باشد:

عملگرهای رابطه ای	شرح عملگرها
>	کوچکتر از
=>	کوچکتر یا مساوی
<	بزرگتر
=<	بزرگتر یا مساوی
==	مساوی با
=~	مخالف با(نامساوی)

٤-٢-٢-عملگرهای منطقی

این عملگرها را در جدول زیر مشاهده می کنید

عملگرمنطقی	شرح عملگر
&	AND
|	OR
~	NOT
xor(x,y)	OR انحصاری(در صورتی که تنها یکی (x,y) مقدار درستی داشته باشند True برمی گرداند

٤-٣-حلقه های تکرار

این دستورات در اغلب زبان های برنامه نویسی به خصوص زبان C برای انجام یک دستور به صورت تکراری وجود دارند.

حلقه For.... End

این حلقه این امکان را به وجود می آورد که برخی از دستورات به تعداد دفعات از قبل تعیین شده تکرارشوند.شکل کلی آن به صورت زیر است:

```
for variable = a
statement 1
statement 2
...
```

```
statement n
end
```

که در حالت بالا a یک ماتریس است.

به این ترتیب در هر بار تکرار حلقه یک ستون ماتریس a درvariable قرار می گیرد. که بدین صورت حلقه به تعداد ستون های a تکرارمیشود .

این حلقه را می توان به صورت تو در تو استفاده کرد .مثال زیر با استفاده از حلقه های تو در تو جدول ضرب ایجاد می کند.

```
for i=1:9
for j=1:9
 p(i,j)=i*j;
end
end
```

حلقه While....End

این حلقه چند دستور را به تعداد دفعات نامحدود تکرار می کند .از این دستور هنگامی استفاده می شودکه تعداد دفعات تکرار مشخص نباشد .شکل کلی این دستور به صورت زیر است:

```
while expression
statements
end
```

expression: یک عبارت شرطی است و تا هنگامی که درست باشد، حلقه تکرار می شود.

statements: کلیه دستوراتی هستند که تا موقعی که عبارت شرطی برقرار باشد اجرا می گردند.

٤-٤-ساختارهای تصمیم

شرط If ...End

درحالتی که دربرنامه نیازبه وجود شرطی می باشد ازاین دستوراستفاده می گردد که در صورت برقرار بودن شرط دستورات اجرا می گردد .شکل کلی دستور به صورت ذیل می باشد

```
if expression
statements
end
```

درصورتی که شرط برقرار نباشد برنامه به خط پس از end منتقل می گردد.

شرط If - Else – End

حتماًبا عملکرد این دستور در زبان های برنامه نویسی دیگر آشنا شده اید.شکل کلی این دستور را در زیرمی بینید.

```
if expression 1
statements 1
elseif expression 2
statements 2
...
elseif expression n
statements n
```

```
else
statements
end
```

همان طور که مشاهده می کنید در حالت کلی می توان از یک If بیشمار Elseif و یک Else و یک End استفاده کرد.

استفاده از Elseif و Else اختیاری است .

اگر شرط مقابل If درست باشد دستورات شماره ۱ اجرا می شوند، در غیر این صورت (Elseif)شرط مقابل ۲ بررسی می شود در صورتی که درست باشد دستورات ۲ ودر غیر این صورت شرط ۳ بررسی می شود .

در صورتی که n شرط بررسی شد و درست نبود دستورات قسمت Else اجرا می شود.

شرط Switch...Case

از این ساختار برای تصمیم گیری چندگانه بر اساس مقادیر مختلف یک عبارت استفاده می شود.به طورکلی در تمام تصمیم گیری هایی که بیش از ۳ انتخاب وجود داشته باشد از این دستور استفاده می شود.

حالت کلی این دستور را مشاهده می کنید:

```
switch switch_expr
 case case_expr
    statement, ..., statement
  case {case_expr1, case_expr2, case_expr3, ...}
    statement, ..., statement
  otherwise
    statement, ..., statement
end
```

به چند نکته در این مورد باید دقت کرد:

۱- پس از اجرای هر یک از دستورات روند اجرا برنامه به بعد از End منتقل می شود و سایر Case ها کنترل نمی شوند.

۲- در بالا در مورد Case دوم در صورتی که عبارت مورد نظر با هر یک از ۳ عبارت موجود در داخل کروشه{} برابر باشد دستورات اجرا می شوند.

۳- استفاده از Otherwise نیز اختیاری است .

شرط logical if...else ...if

این ساختاربرای تصمیم گیری شرطی براساس نتایج گزاره های منطقی به کار می رود .

```
if
        then
        elseif
        elseif
          .
          .
          .
        else
```

بلوک Try-Catch

شکل کلی این دستور به این صورت می باشد:

```
try
commands
catch exception
commands
end
```

عملکرد این دستور به این صورت است که دستورات بعد از Try اجرا می شوند در صورتی که خطایی رخ دهد کنترل برنامه به Catch منتقل شده و دستورات موجود در این قسمت اجرا می شود. این خاصیت باعث می شود از آن برای خطایابی برنامه ها استفاده شود.

٤-٥-توقف روند اجرای برنامه

Break

هنگامی که این دستور اجرا می شود نرم افزار متلب به اولین دستوری که بعد از حلقه For قرار دارد می رود.در صورتی که این دستور در حلقه های تو درتو (ForیاWhile) به کار رودنرم افزار متلب فقط از حلقه جاری خارج می شود.

Error

این دستور باعث توقف اجرا برنامه شده و می تواند یک رشته کار اکتری را برگرداند.

```
error (' STATEMENT ')
```

Return

هر گاه روند اجرا برنامه به این دستور برسد مقدار مورد نظر را برمی گرداند در(Command window)نمایش می دهد و ادامه اجرای برنامه متوقف می شود.

از این دستور برای نمایش زود هنگام مقادیر یعنی قبل از به پایان رسیدن کامل برنامه استفاده می شود .به این ترتیب هرگاه جواب مورد نظر به دست آمد روند اجرای برنامه نیز متوقف می شود و مقدار مورد نظر رابرمی گرداند.

Continue

این دستور در داخل حلقه for یا while قرار می گیرد وهرموقع که اجرا شود کنترل را به سطراول حلقه برمی گرداند.

٤-٦-توابع زمانی

جدول زیر بعضی از توابع زمانی را نشان میدهد:

نام تابع	عملکرد تابع
tic	شروع تایمر
toc	نشان دهنده زمان تایمر
clock	روز و ساعت
etime(t1,t2)	فاصله زمان بین t2,t1
Pause(t)	وقفه (مکث) t ثانیه
cputime	زمان cpu بعد از شروع متلب
date	تاریخ روز
calendar	تقویم ماه جاری
calendar(yyyy,mm)	تقویم سال وماه معین

٤-٧-توابع خاص

نمونه هایی از توابع پیشرفته ریاضی به همراه دستور مربوطه در زیر آمده اند:

تابع لژاندر	تابع بسل	تابع گاما	تابع بتا
legendre(n,x)	**bessel(n,x)**	**gamma(x)**	**beta(x)**

٤-٨-توابع مبین منظم متلب

Regexp:

```
[v1, v2, ...] = regexp('str', 'expr', q1, q2, ...)
```

Regexprep:

```
s = regexprep('str', 'expr', 'repstr', options)
```

regexptranslate:

```
regexptranslate(type, s1)
```

٤-٩-عبارات وتوابع نمادین (Symbolic)

درریاضیات ما اغلب نیاز دارم عبارات چند جمله ای یا توابع پارامتری ایجاد نماییم .مشتق ،انتگرال گیری ،حد گیری، سری،تبدیلات لاپلاس و فوریه و z ،حل دستگاه های چند جمله ای و معادلات دیفرانسیل از این جمله هستند.

تعریف توابع نمادین (symbolic):

مامی توانیم یک حرف یا کلمه را به عنوان یک متغیر غیر عددی یا پارامتری معرفی نماییم .

تابع sym و syms جهت تعریف یک یا چند متغیر به کار می رود .

```
>>X=Sym( 'x')
```

هچنین می توان یک متغیر را تابعی از چند متغیر دیگر قرار داد وبا استفاده از دستور findsym نشان داد که هرمتغیر به چه متغیرهای پارامتری دیگروابسته است .

```
>>syms x y z
>>f=sin(x)4cos(y);
>>g=f*z;
>>findsym(g)
ans=
x, y, z
```

دستور subs:با این دستور می توان در یک عبارت پارامتری یک متغیر پارامتری را با یک عدد یا پارامتر جدید وحتی یک ماتریس جایگزین کرد

```
>> subs(f,'x',1)
ans=
sin(1)+ cos(y)
>> subs(f,y,'theta')
ans=
sin(x)+cos(theta)
>> subs(f,{x,y},{'theta','alfa'})
```

```
>> subs(f,{x,y},{pascal(3),magic(3)})
```
اگر تمامی متغیرهای عبارت سمبلیک با عدد یا ماتریس های عددی جایگزین شوند عبارت سمبلیک به یک عبارت عددی تبدیل می شود.

تابع vpa:این تابع بخش های قابل تبدیل عبارت سمبلیک را به عدد تبدیل می کند .هر قسمت از عبارت سمبلیک که ارزش عددی ومحاسباتی دارد با مقدارعددی جایگزین می شود.

آرگومان دوم این تابع نشان میدهد که عبارت با چه دقتی محاسبه ونمایش داده شود.

```
>>a=subs(f,'x',2)
a=
sin(2)+cos(y)
>>vpa(a)
ans=
>>vpa(a,5)
ans=
```

تابع eval:این دستور مقدارگذاری تابع را به ازای مقادیر مشخص نشان می دهد.

تابع diff: برای مشتق گرفتن از یک عبارت عبارت جبری وپارامتری از این تابع استفاده می شود.

دراین تابع اولین آرگومان نام تابع ودومین آرگومان نام متغیری که برحسب آن مشتق گیری انجام می گردد و سومین آرگومان مرتبه مشتق گیری مورد نظر را نشان می دهد .درصورت عدم ذکرنام متغیر، نرم افزار متغیر xرا به طور پیش فرض انتخاب می کند.

```
>>syms x y z
>>g=sin(x*y)*z;
>>diff(g,x,2)
ans=

>>diff(g,y,3)
ans=
```

تابع int:برای انجام عملیات انتگرال گیری معین ونا معین به کار میرود .آرگومان اول نام تابع ،آرگومان دوم نام متغیری است که انتگرال گیری برحسب آن انجام می شود.درانتگرال های معین آرگومان سوم حد پایین انتگرال وآرگومان چهارم حد بالای انتگرال را نشان می دهد .برای گرفتن انتگرال های دو ،سه وچندگانه از انتگرال های تو در تو استفاده می نماییم.

```
>> syms x y z t
>> w=sin(x)+cos(y)+exp(z)+t^2;
>> int(w)
ans=
>> int(w,t,1,10)
Ans=
```

تابع **symsum**:برای بدست آوردن مجموع مقادیر یک تابع برای یکی از متغیر ها در یک بازه مشخص می باشد که نوعی انتگرال برای مقادیر گسسته وصحیح می باشد که دارای دو نوع معین ونامعین می باشد.

```
>> syms a b c x y z
>> f1=a*b;
```

```
>> f2=a*x^2+b*y+exp(-c*z);
>> symsum(f1)
>> symsum(f1,a,0,3)
```

تابع **limit**: برای بدست آوردن حد تابع مورد نظر دریک نقطه وبه صورت چپ وراست مورد استفاده قرار می گیرد.درتابع حد آرگومان اول نام تابع ،آرگومان دوم نام متغیر،آرگومان سوم عددی که متغیر به سمت آن میل داده می شود ودرآرگومان چهارم عبارت left (چپ) یا right (راست) ذکر می شود.

```
>> syms x y z
>> f=x/sin(x);
>> g=y+ceil(z);
>> limit(g,z,0,'right')
```

درصورت عدم برابری حد چپ و راست عبارت NAN درخروجی نمایش داده می شود.

توابع تبدیل (فوریه ،لاپلاس وZ):تبدیلاتی که در حوزه ی عبارات SYMBOLIC مطرح می گردد عبارت است از تبدیل فوریه ومعکوس آن ،تبدیل لاپلاس و معکوس آن وتبدیل Z و معکوس آن.

تعریف تبدیل فوریه :

$$f = f(x) \Rightarrow \quad F = F(w) \quad F(w) = \int_{-\infty}^{+\infty} \mathrm{f(x)e^{-iwx}}\, dx$$

درنرم افزار متلب تابع F به صورت fourier(f) تعریف می شود درصورتی که بخواهیم نام متغیر غیر از w باشد نام متغیر را در آرگومان دوم می آوریم ،آرگومان سوم نام متغیری که انتگرال گیری براساس ان انجام می شود ذکر می گردد .

```
>> syms x y u
>> f=sin(x);
>> g=sin(x)+sin(y);
fourier(f)>>
fourier(f,u)>>
>> fourier(g,y,u)
```

تابع **dirac** همان تابع ضربه است که به شکل ذیل تعریف می شود :

```
>>int(heaviside(x),x,-inf,inf)
ans=
dirac(x)
```

تابع $\mathrm{heaviside(x)}$: همان تابع پله است که به ازای مقادیر X>0 تابع 1و به ازای مقادیر X<0 تابع صفر می باشد همچنین تابع درنقطه X=0 تعریف نشده است.

تابع **ifourier**:این تابع به عنوان معکوس تابع فوریه وبا تعریف ذیل به کار می رود .

$$F = F(w) \quad => \quad f = f(x) \qquad \mathrm{f(x)} = {}^{1}/_{2\pi} \int_{-\infty}^{+\infty} F(w)\mathrm{e}^{\mathrm{iwx}}\, dw$$

آرگومان اول نام متغیر تابع f و آرگومان دوم نام متغیری است که معکوس تابع فوریه براساس آن انتگرال گیری می شود.

```
>> syms x u w
>> f=۱/w;
>> g=u+w;
>> ifourier(f)
>> ifourier(g,u,x)
```

تابع تبدیل لاپلاس :این تابع براساس تعریف ذیل برای تابع f تعریف شده است :

$$P(s) = L(f) = \int_0^{\infty} e^{-st} f(t)dt$$

تابع laplace(f):این تابع مقدار لاپلاس تابع f را محاسبه وبه خروجی می برد . تابع f بر حسب t می باشد ومقدار لاپلاس بر حسب s محاسبه می گردد .

```
>> syms x y t s r
>> f=heaviside(x);
>> g=x+sin(t);
>> h=x+sin(y);
>> laplace(h)
>> laplace(g,x,r)
```

تابع معکوس لاپلاس :این تابع برای محاسبه معکوس تابع لاپلاس با تعریف ذیل به کار می رود.

$$f(t) = L^{-1}(s) = \int_{c-i\infty}^{c+i\infty} e^{st} L(s)ds$$

تابع ilaplace:نرم افزار متلب با استفاده از این دستور معکوس لاپلاس تابع f را محاسبه وبه خروجی می برد.

تابع تبدیل z:این تبدیل همسان زمان گسسته تابع تبدیل لاپلاس می باشد وبه صورت ذیل تعریف می شود:

$$f = f(n) => \quad F = F(z) \;\; F(\mathrm{z}) = \sum_{0}^{\infty} \frac{\mathrm{f(n)}}{\mathrm{z^n}}$$

لازم است که عبارت f تابعی از n باشد وتابع تبدیل Z آن نیز تابعی از z می باشد .

دستور ztrans(f):این دستور تبدیل z تابع f را که برحسب n می باشد انجام داده و بر حسب z نمایش می دهد. درصورتی که f تابعی از z باشد تابع تبدیل z برحسب w در خروجی نمایش داده می شود.

```
>> syms n k w z u
>> f=n^2;
>> ztrans(f)
```

```
>> ztrans(f,k,z)
```

اگرf تابعی از n باشد وبخواهیم تابع تبدیل متغیر z راداشته باشد تابع تبدیل ztrans به صورت ذیل تعریف می شود :

Ztrans(f,k,z)

واگر *f* تابعی از *z* باشد وبخواهیم تابع تبدیل متغیر *w* را داشته باشد تابع تبدیل *ztrans* به صورت ذیل تعریف واستفاده می شود:

$$f = f(z) => \ F = F(w) \ F(\mathrm{w}) = \sum_{0}^{\infty} \frac{\mathrm{f}(z)}{\mathrm{w}^{\mathrm{z}}}$$

Ztrans(f,z,w)

به طور مثال :

```
>> syms n k w z u
>> f=n^2;
>> ztrans(f)
>> f=(k+n)^2;
>> ztrans(f)
```

تابع معکوس z :

این تابع به صورت ذیل تعریف می شود و برای محاسبه تابع f با متغیر n از تابع تبدیل F با متغیر w به کار می رود.

$$F = F(z) => f = f(n) \ , f(n) = {}^{1}/_{2\pi i} \oint F(z) z^{n-1} dz \quad n = 1,2,3, \ldots$$

تابع iztrans:این تابع برای محاسبه تابع معکوس تبدیل z توسط نرم افزار متلب به کار می رود.

```
>> syms z w
>> f = ٣*z/(z-١)^٢;
>> iztrans(f)
>> f=z/(z+٢*w);
>> iztrans(f)
```

توابع ترکیبی: این توابع به صورت ترکیبی از توابع تعریف شده ایجاد می گردند . درنرم افزار متلب این توابع با استفاده از دستور compose ساخته می شوند .

نحوه قرار گیری نام توابع و نام متغیر ها در ورودی تابع compose ،تعیین کننده نحوه عملکرد تابع compose می باشد.

```
<<syms x y z t
<<f= x /(x+x^2);
<<g=cos(y);
<<h=(x+1)^t;
<<compose(f,g)
ans=
cos(y)/(cos(y)^2 + cos(y))
<<compose(h,g,x,z)
```

= ans

(cos(z) + 1)^t

معکوس توابع :

اگر f تابعی از x باشد دراین صورت تابع f^{-1}را معکوس تابع f تعریف می کنیم به طوری که $f(f^{-1}(x)) = x$ باشد.

تابع finverse:این تابع معکوس تابعی را که در ورودی می دهیم را در خروجی برمی گرداند .

درصورتی که تابع مورد نظر ما بیش از یک متغیر داشته باشد لازم است نام متغیری که مایلیم معکوس تابع برحسب آن محاسبه شود رابه عنوان آرگومان دوم تابع finverse درتابع ذکر نماییم.

```
syms x y z
>> finverse(exp(x))
>> finverse(cos(y) - sin(x),y)
```

تابع Jacobian:این تابع ژاکوبین توابع f_1 , f_2 , f_3...را برحسب متغیرهای x_1, x_2, x_3...به صورت ماتریس درخروجی نمایش می دهد وبه صورت ذیل تعریف می شود:

```
Jacobian([f1;f2;f3,...],[x1,x2,x3,...])
```

که توابع عبارات هایی از متغیر ها می باشند.

```
>>Syms f1,f2,f3,x,y,z,w
>>f1=2*x+y+z
>>f2=x+2*y+z+w
>>f3=x+y+2*z+w
>>Jacobian([f1;f2;f3],[x,y,z,w])
```

توابع آسان ساز:برای نمایش واضح توابع نمادین ازاین توابع استفاده می شود :

تابع ()prrety:یک عبارت ریاضی را به فرم جبری خوانا تبدیل می کند :

```
>> syms x
>> f1 = (5+4*cos(x))^3*sin(x)^2*(1+sin(x));
>> pretty(f1)
```

تابع ()collect:برای دسته بندی عبارت جبری به صورت بسته به کار می رود.

```
>> syms x y
>> f = x^2*y + y*x - x^2 - 2*x;
>> collect(f)
```

تابع ()expand:برای دسته بندی عبارت جبری به صورت باز به کار می رود .

```
>> g = (2*x+5)^2
>> expand(g)
```

مشخص است که دو تابع فوق مکمل یکدیگر هستند .

تابع horner():عبارت جبری را آنقدرساده می کند تا هیچ عبارت توانی باقی نماند وتمامی عبارات به ضرایبی از متغیرها یا ترکیبات بدون توان آنها تبدیل شود.

```
>> syms x y
>> horner(x^2+y^2+x*y)
```

تابع factor():از عبارت جبری به صورت مشخصی فاکتور گیری می نماید.

```
>> syms x
>> g = x^3-1
>> gf = factor(g)
```

تابع simplify:این تابع ساده ترین شکل عبارت جبری را در خروجی نمایش می دهد.

```
>> syms x y
>> simplify(exp(y)*exp(x))
```

تابع simple():این تابع کلیه دستور های ساده سازی رابرروی یک عبارت جبری انجام می دهد.

```
>> syms x y
>>Simple(sin(x)*cos(x))
```

حل معادلات یک مجهولی ودستگاه معادلات چند مجهولی

معادلات چند جمله ای :

دستور roots(d):این دستور ریشه های چند جمله ای را به دست می هد که اعضای آن ضرایب بردار d=[n1,n2,n3,...] می باشد.

دستور roots():این دستور ریشه های موهومی وحقیقی را بدست می دهد.

دستور poly(b):این دستور چند جمله ای را بدست می دهد که ریشه های آن اعضای ماتریس b می باشند .

حل معادله با تابع fzero()

این تابع سعی می کند مقدار ریشه واقعی را حول و حوش یک نقطه شروع (حدس اولیه) که دستی وارد می شود را پیدا کند این تابع با پیدا کردن نقاط تغییر علامت توابع پیوسته به جواب ها (ریشه ها) می رسد.

تابع fzero(f,x_0):این تابع دارای دو آرگومان می باشد که اولی نام تابع و دومی مقدار نقطه شروع یا حدی اولیه که حتی الامکان نزدیک به ریشه می باشد.

۵-برنامه نویسی Graphical User Interface(GUI)

۵-۱-آشنایی با واسط گرافیکی کاربر

با دستور **guide** که مخفف **Graphical user interface development environment** می باشد ابزارتولید GUI در اختیارکاربر قرار میگیرد که شامل قسمت های زیر می باشد

Layout Editor(LE)
User Interface Controls (uicontrols)

در پنجره اول چهارچوب اصلی GUI را که پس از اجرا مشابه پنجرههای استاندارد محیط Windows خواهد بود، طراحی میکنیم .پنجره دوم یک میله ابزارtools bar است که دکمه کنترلهای ضروری جهت یک برنامه GUI را فراهم میکند .

این محیط مشابه محیط های برنامه نویسی شی گرا می باشد روش کار بدینگونه است که ابتدا کنترلر های لازم برای یک برنامه به قسمت le منتقل میشود، سپس به هر کنترلر وظیفه خاص خودش از طریق یک زیر برنامه مربوط به آن دکمه که Callback Function نام دارد محول میشود .البته اعمال فرعی دیگری نظیر تعیین ویژگی های کنترلر ها و پنجره هاوترازبندی کنترلرهاوویرایش آنها نیز انجام میشود .

۵-۲-آشنایی با کنترلر ها

محیط برنامه نویسی گرافیکی شامل پنجره ای می باشد که کنترلر های مورد نظر درآن قرار گرفته است که با انتخاب هرکدام ازآنها کنترل های مورد نظر برنامه نویس ایجاد می گردد درجدول ذیل نام و شرح ویژگی هریک ازاین کنترلرها آورده شده است:

نام کنترلر	شرح ویژگی های کنترلر
Select	درجابه جایی وانجام تنظیمات اشیا کاربرد دارد
Push button	با فشردن این کلید عملیات مربوطه به آن انجام می شود
slider	کلیدی است که با کشیدن آن مقداری به پارامترنسبت داده می شود
Radio button	با زدن این دکمه عملیات درحالت انتخاب قرار می گیرد
Check box	با زدن تیک عملیات درحالت انتخاب قرار می گیرد
Edit box	با زدن کنترلر امکان ویرایش متن یا رشته فراهم می شود
Static text	برای نمایش متن خروجی سیستم مورد استفاده قرار می گیرد
Pop-up menu	منو آبشاری که انتخاب گزینه های کنترلر در آن را انجام می گیرد
List box	لیستی از گزینه های موجود درکنترلر را نمایش می دهد
Toggle button	با فشردن آن عملیات مورد نظر انجام شده و تا فشردن مجدد دکمه در

	حالت انتخاب قرار می گیرد
Axes	کنترلر محلی را برای رسم و نمایش نمودار ایجاد می کند
panel	محلی برای تفکیک و دسته بندی کنترلر ها ایجاد می کند
Button group	محلی را برای قرار دادن کلید های گروهی مانند دکمه رادیو ایجاد می کند
Activex	این دکمه با توجه به پروتکل های موجود درسیستم کلیه ابزارهارا فراخوانی کرده و مورد استفاده قرار می دهد.

۵-۳- ویژگی کنترلر (property inspector)

روی le ویا روی کنترلر راست کلیک کرده و از منویی که باز میشود گزینه property inspector را کلیک میکنیم .پنجره مربوطه باز میشود این پنجره شامل موارد و گزینه هایی درمورد خواص و ویژگی های مربوط به کنترلر می باشد که مشابه سایرزبان های شی گرا می توان این ویژگی ها رابراساس برنامه مورد نظر تغییرداد

ویژگی Name or String

این ویژگی برای بعضی از اشیاء string و برای بعضی name می باشد با تایپ هرعبارت در این قسمت نام کنترلر مورد نظر مشخص می گردد.

تابع فراخوان (callback)

در داخل برنامه برای عملیاتی که با فشردن یک دکمه یا کلیک روی یک گزینه انجام میشود تابعی مینویسیم تابع فراخوان آن پنجره یا آن دکمه گفته می شود ودرمواقع خاص می تواند تابع را فراخوانی نماید.

تابع فراخوان دارای انواع مختلفی می باشد که به نحوه کاربرد آن در برنامه ارتباط دارد.

ساخت MENU

منو یک GUI ساده است که امکان انتخاب چند گزینه را فراهم میکند.

```
% mnu.m
k = 0;
while k < 4;
k = menu('Help Menu','Operators','Mod','Rem','Exit');
if k == 1
```

```
help \
elseif k == 2
help mod
elseif k == 3
help rem
else
a = input('Really Exit? (Y/N) ','s');
if (a == 'n') || (a =='N')
k = 3;
continue;
```

اجرا را به ابتدای حلقه while % منتقل و شرط را مجدد تست میکند

```
end
end
end
>> mnu
```

٥-٤-کامپایل کردن برنامه(compile)

هرنرم افزاری برنامه ای به نام کامپایلردارد که نقش ترجمه برنامه به زبان ماشین را برعهده دارد به طوریکه کامپایلر،برنامه نوشته شده را به زبان ماشین تبدیل می کند.

درمواقعی که برنامه ای با نرم افزار متلب نوشته شده و لازم است که کاربرانی ازآن استفاده نمایند می توان از حالت اجرایی یا exe برنامه استفاده کرد به طوری که کاربران می توانند بدون اینکه به منبع (source) برنامه دسترسی داشته باشند ازآن برنامه استفاده نمایند نتیجه این فرایند، ساخت یک فایل اجرایی با پسوند .exe می باشد که قابل استفاده درهرسیستم وبدون نیاز به نرم افزارمتلب می باشد.

برای نصب و راه اندازی اولیه کامپایلر از دستور mbuild-setup استفاده می شود.

درحین اجرای این دستور نرم افزار سوالاتی درخصوص تعیین نوع ونحوه انجام کامپایل از کاربر می نماید به طوری که نرم افزار به طورخودکارکامپایلرهای موجود درسیستم راشناسایی و پس ازانتخاب کاربر مورد استفاده قرار می دهد.

دستور matlab compiler command که به اختصار mcc خوانده می شود دستور اصلی کامپایل کردن نرم افزار متلب می باشد و می توان با حروفی پس ازاین دستور نحوه انجام کامپایل مورد نظر را تعیین کرد.

به طور مثال mcc-m به عنوان کامپایلر c می باشد.

mcc –S به عنوان کامپایلر Simulink S-Function و mcc –p به عنوان کامپایلر C++ عمل می کند.

همچنین می توانید برنامه توابع پیچیده را درنرم افزار متلب نوشته و به صورت فایل dll در سایر برنامه های visual مانند Basic و Delfiو studio دربرنامه خود فراخوانی واستفاده نمایید.

٦-کاربرد های نرم افزار متلب

٦-١-کاربرد متلب در داده های آماری

در نرم افزار متلب دستورات فراوانی برای ویرایش وفراکاوی داده ها و انجام دادن تحلیل های آماری وجود دارد .در زیر به مهمترین آنها اشاره می شود.

دستورات min(x) , max(x) به ترتیب مینیمم و ماکسیمم هر ستون را به دست می آورند .در صورتی که ماتریس یک بردار سطری باشد این کار را روی سطر انجام می دهد.

```
<<min(x)
ans=
 ٣  ٥  ١  ٢
<<[s,t]=max(x)
s=
١٩  ١٩  ١٨  ١٨
t=
   ١  ٣  ١  ٢
```

همان طور که ملاحظه می کنید خروجی دوم دستور sort مشخص کننده مکان درایه های مینیمم یا ماکزیمم خواهد بود.

در صورتی که این دستور به صورت max(a,b)یا (min)به کار رود که b یک ماتریس با ابعاد ماتریس a یا یک عدد باشد آنگاه خروجی یک ماتریس است با درایه های بزرگتر بین a , b که دو ماتریس a , b بایستی هم مرتبه باشند.

```
<<max(a,x)
ans=
٢  ١٠  ١٠  ٧  ١١
٨  ٩  ١١  ٦  ١١
١۴  ٢  ٩  ٩  ٢
```

mean(a,n):با این دستورمی توان میانگین هر سطر یا ستون را به دست آورد.

همچنین می توان میانگین و عضو میانی هر سطر یا ستون را به دست آورد.

```
<<mean(b,2)
ans=
٥.۶٠٠٠
```

Median(b,2): با این دستورمی توان عضو میانی هر سطر یا ستون را به دست آورد.

```
<<median(b,2)
ans=
٥
```

این دستورپس از مرتب کردن سطر یا ستون عنصر میانی را برمی گرداند .و به این نکته دقت کنید که تابع median درصورتی که تعداد سطرها یا ستون ها زوج باشد میانگین 2 عضو وسط را برمی گرداند.

Cov(x,y): این دستور مقدار کوواریانس دو بردار هم اندازه x, y را درخروجی ارائه می دهد .

Corrcoef(x,y):این دستور ضریب همبستگی (R) بین ماتریس x , y را درخروجی ارائه می دهد . سطرهای ماتریس ها مشاهدات و ستون ها متغیرها می باشند .

mode(X):این دستور مد(مقدار بیشترین فراوانی) بردار x را درآرایه ارائه می دهد .

std(X,flag,dim) : این دستور برای محاسبه انحراف استاندارد نمونه n تایی به کار می رود که اگرحالت برآورده نشده داشته باشیم یعنی تعداد نمونه n-1 در نظر بگیریم flag=0 و اگر حالت برآورده شده باشد یعنی تعداد نمونه n در نظر بگیریم flag=1 قرار داده می شود .

sum(b): این دستور مجموع هر سطر یا ستون را برمی گرداند.

```
<<sum(c)

ans=
۲۰  ۱۳  ۶
```

cumsum(b):این دستور حاصل جمع درایه با درایه های قبل را به صورت تجمعی درخروجی نمایش می دهد .

```
<<cumsum(c)
ans=
۰  ۷  ۳
۹  ۱۳  ۵
۲۰  ۱۳  ۶
```

prod(b):این دستور حاصل ضرب ستون ها را به عنوان سطر آخر درخروجی چاپ می کند

```
<<k=[3 4 6; 2 0 8;1 4 9];
<<prod(k)
ans=
۶  ۰  ۴۳۲
```

cumprod(b): این دستور حاصلضرب هر درایه در درایه های ماقبل را درخروجی نمایش می دهد. از این دستور می توان در ساخت تابع فاکتوریل استفاده کرد.

```
<<cumprod(k)
ans=
۶  ۴  ۳
۴۸  ۰  ۶
۴۳۲  ۰  ۶
Cumprod(1:n)=n!
<<cumprod(1:8)
ans=
```

۱ ۲ ۶ ۲۴ ۱۲۰ ۷۲۰ ۵۰۴۰ ۴۰۳۲۰

در دستورات می توان جهت انجام عملیات (سطر یا ستون)را مشخص کرد.در صورتی که جهت مشخص نشود مانند دیگر دستورها که در بالا گفته شد عمل می شود.

hist(Y,x):این دستور برای رسم هیستوگرام تابع y که متغیر x دارد به کار می رود.

Pie(a,b):این دستور برای رسم نمودار دایره ای به کار می رود.

Ribbon(x,y):این دستور نمودار خطی با خطوط دوبعدی به شکل نوار به کار می رود.

Stairs(x,y):این دستور برای رسم نمودار تابع y متغیری از x می باشد که به صورت پله ای رسم می شود.

٦-٢-کاربرد متلب در بهینه سازی خطی

فرمت کلی برنامه ریزی خطی به صورت زیر است :

```
Min f
S.t
Ax<b
Aeq=beq
lb<x<ub
```

که درآن f,x,A, ub,b,lb,beq بردار و A,Aeq ماتریس هستند برای حل این مساله برنامه ریزی خطی ازدستور linprog به صورت زیر استفاده می شود.

```
>> x=linprog(f,A,b,Aeq,beq,lb,ub)
```

دقت شود که درصورت نبودن هریک ازماتریس ها وبردار ها از علامت [] استفاده می شود

با تعریف مساله خطی زیر از برنامه ریزی خطی استفاده می کنیم:

Min f=$2x_1 + 3x_2$

$x_1 - 2x_2 \leq 6$

$x_1 + x_2 \leq 9$

x_1و$x_2 \geq 0$

```
>> f=[2;3]
>> A=[1 -2 ;1 1]
>> B=[6,9]
>> lb=zeros(1,2);
>> x=linprog(f,A,b,[],[],lb)
```

.Optimization terminated

X=

۰.۱۹۲۷

۰.۰۰۴۳

٦-۳-کاربرد متلب دربرنامه ریزی صفر و یک

Min f
S.t
Ax<b
Aeq=beq
X1, X2={0,1}

برای حل این مساله برنامه ریزی صفر ویک از دستور bintprog استفاده می شود

به طور مثال داریم

Minf=$3x_1 + 2x_2 + 4x_3$
s.t
$-x_1 + x_2 + x_3 \leq 3$
$2x_1 - x_2 + x_3 \leq 4$
$x_1 + x_3 \leq 2$
$x_j = 0,1 \quad j = 1,2,3$

```
>> f=[3;2;4]
>> A=[-1 1 1;2 -1 1 ;1 0 1]
>> b=[3;4;2]
 [x,fval]=bintprog(f,A,b)
Optimization terminated

X=

۰

۰

۰

Fval=

    0
```

٦-٤-کاربرد متلب در محاسبات عددی(درون یابی)

درون یابی به عنوان یکی از روش های تخمین مقادیر یک تابع با استفاده ازبدست آوردن یکسری از نقاط داده ای می باشد ،درون یابی ابزاری است که هنگام نبودن امکان محاسبه سریع مقدارتابع درنقاط میانی مورد نظربه کارمی رود.

ازروش های زیادی برای درون یابی توابع می توان استفاده کرد درنرم افزارمتلب دستورمشخصی برای انجام درون یابی توابع به کار می رود که این درون یابی می تواند دربیش از یک بعد انجام گیرد به طور مثال اگر تابع z=f(x,y) تابعی از دو متغیر x,y باشد می توان مقادیر x,y را برای محاسبه مقدار z درون یابی کرد .

در MATLAB درون یابی توابع در یک بعد از دستور interp1 استفاده می شود برای درون یابی توابع دوبعدی از دستور interp2 استفاده می شود همچنین این دستوربه طور پیش فرض به صورت خطی انجام می دهد.

```
>>hours=1:12;
temps=[20.2 20.9 21.1 21.6 22.2 22.7 23.1 24 25.3 25.9 26.1 27];
t=interp1(hours,temps,[2.5 3.9 7.8])
t=
```

۲۳.۸۲۰۰ ۲۱.۵۵۰۰ ۲۱.۰۰۰۰

به جای فرض کردن خط مستقیم برای اتصال نقاط داده ای می توان چندین چندین منحنی را برای ترسیم منحنی مناسب نقاط داده ای را درنظرگرفت . عمومی ترین فرض یک چند جمله ای درجه سوم مورداستفاده برای مدل سازی هرتکه ای بین نقاط داده ای متوالی می باشد ومشتق اول و دوم هر چند جمله ای درجه سوم روی نقاط داده ای قرار می گیرد این نوع درون یابی را مارپیچ (اسپلاین) درجه سوم یا دقیقاً مارپیچ ها نامیده می شود.

با استفاده ازآرایه interp1(x,y,z,'spline') می توان مقدارتابع z را بامتغیر های x,y با روش(method) اسپلاین درون یابی کرد :

```
s=interp1(hours,temps,[3.2 4.30 7.7],'spline')
=s
```

۲۳.۶۵۸۲ ۲۱.۷۸۲۱ ۲۱.۱۷۱۸

برای به دست آوردن یکسری ازداده ها درفاصله کم از درون یابی اسپلاین استفاده میشود

درون یابی دو بعدی حالت پیچیده ومشکل تر از یک بعدی است شکل کامل تابع interp2(x,y,z,xi,yi,method) می باشد که دراینجا x , y متغیر مستقل و z متغیر وابسته می باشد رابطه x,y با z عبارت است از :

Z(i,:)=f(x,y(i))

Z(:,j)=f(x(j),z(:,j))

با توجه به عبارات فوق می توان دریافت که با تغییر x i امین سطر از z با i امین عنصر از y یعنی y(i) مرتبط شده و با تغییر y ، j امین ستون از z با j امین عنصر از x یعنی x(j) مرتبط می شوند.

x_j آرایه ای از درون یابی مقادیر درطول محور x بوده و y_j آرایه ای ازدرون یابی مقادیر در طول محور y می باشد . پارامتر انتخابی method می تواند یکی از روشهای cubic , linear یا nearest باشد . دراین حالت cubic به معنی اسپلاین درجه دوم نیست بلکه الگوریتم دیگری با استفاده از چند جمله ای های درجه سوم می باشد.

روش linear درون یابی خطی مورد استفاده برای اتصال نقاط داده ای روی نمودارها می باشد.

روش nearest به سادگی نقطه داده ای ناهموار نزدیک به یک نقطه تخمینی را انتخاب می کند ودرتمام موارد ، متغیرهای مستقل x, y فضای خطی ویکنواخت فرض می شود .

```
>> time=1:12 ;
>> temp = [230 249 268 255 283 269 278 293 274 285 291 293 ]

>> interp1(time,temp,7.9,'spline')
= ans
۲۹۳.۳۲۶۲
>>interp1(time,temp,7.9,'cubic ')
= ans
۲۹۲.۶۸۱۲
```