فصل دوم

مقدمات متلب

عمليات محاسباتي

از نرم افزار متلب می توان به عنوان یک ماشین حساب استفاده کرد. مثال زیر را در نظر بگیرید. $>> 3^2 - (5 + 4)/2 + 6*3$ اگر در سمت چپ عبارت بالا متغیری تعریف نکنیم , متلب آنرا در متغیری به نام ans قرار می دهد. >> ans² + sqrt(ans) همانطور که میبینید متلب مقدار جدیدی را به متغیر ans اختصاص داد. می توانیم مقدار محاسبه شده را در متغیر خاصی قرار دهیم. >> u = cos(10)توابع مثلثاتي با راديان كار مي كنند. و مثالی دیگر >> v = sin(10)از دو متغیر حاصله میتوان در محاسبات بعدی استفاده کرد. $>> u^2 + v^2$ متلب به صورت پیش فرض با format short کار می کند. برای تغییر فرمت و افزایش دقت خروجي عبارت زير را تايپ کنيد. >> format long حالا با دستور زیر دقت برنامه را ببینید. >> 1/3 برای بازگشت به فرمت قبلی دستور زیر را وارد کنید. >> format short

مسائل جبری

برای اینکه بتوانیم با متغیر ها به صورت سـمبلیک رفتار کنیم , باید ابتدا متغیر را سـمبلیک کنیم.

```
>> syms x y
>> (x - y)*(x - y)^2
pt = (x - y)*(x - y)^2
>> expand(ans)
>> factor(ans)
>> factor(ans)
>> factor(ans)
>> simplify((x^3 - y^3)/(x - y))
```

متلب تابع دیگری دارد که یک عبارت ریاضی را به همه شـکل های ممکن سـاده می کند. < simple((x^3 - y^3)/(x - y))

عبارات پارامتری و محاسبات دقیق

در متلب عدد پی با pi نشـان داده می شـود. جواب عبارت زیر را ببینید.

>> cos(pi/2)

```
جواب باید صفر شود. ولی به دلیل دقیق نبودن عدد پی کمی با صفر فاصله دارد. تابع sym
این عدد را به عددی سـمبلیک و دقیقا برابر pi/2 تبدیل می کند.
```

```
>> cos(sym('pi/2'))
```

به این نکته دقت کنید که عبارت syms x معادل عبارت ('x=sym('x') اسـت. با این تفاوت که sym تاثیر موقت دارد ولی syms تاثیر دائم دارد.

به نحوه جمع کردن دو عدد 1/2 و 1/3 به صورت سمبلیک توجه کنید.

```
>> sym('1/2') + sym('1/3')
```

```
حالا ببینید چگونه می توان عدد رادیکال 2 را تا 50 رقم اعشار بدست آورد.
```

>> vpa('sqrt(2)', 50)

تابع vpa برای گرفتن دقت بیشتر بکار می رود. برای مثال نتیجه عبارتهای (45^3)vpa و ('vpa(5')vpa را با هم مقایسـه کنید. در کاربرد های تجربی بین این دو مقدار فرقی وجود ندارد ولی بعضی مواقع به عدد دقیق نیاز اسـت که باید عبارت دوم را بکار برد.

مدیریت متغیر ها

در متلب سـه طبقه متفاوت داده وجود دارد. عدد و رشـته و اعداد سـمبلیک. با اسـتفاده از دسـتور whos می توان خلاصه ای از اسـامی و متغیر هایی که تا به حال تعریف شـده مشـاهده کنیم.

>> whos

تا حالا 5 متغیر تعریف کردیم. X و y به دلیل به کار بردن syms سمبلیک هستند. متغیر ans هم به دلیل این که جمع دو متغیر سمبلیک است سمبلیک شده است. متغیر های u و v هم اسکالر هستند. در قسمت سایز هم از سمت چپ به ترتیب تعداد سطر ها و ستون های ماتریس را نشان می دهد.

بنویسـید u=pi و v='pi' و (whos و سـپس دسـتور whos را وارد کنید تا تفاوت این داده ها را ببینید.

برای مشاهده مقدار هر متغیر نام آن را در پنجره command window بنویسید و Enter بزنید.

شـما باید تشـخیص دهید که هر تابع به چه نوع ورودی ای نیاز دارد. وگرنه دادن ورودی نا مناسـب به آن منجر به پیام eror می شـود.

برای پاک کردن همه متغیر ها از حافظه دستور clear all یا clear all را وارد کنید. شـما همچنین می توانید بنویسـید clear x y برای این که بتوانید فقط متغیر های x وy را پاک کنید. شـما معمولا باید قبل از شـروع محاسـبات جدید متغیر های قبلی را پاک کنید.

توجه کنید که اطلاعات مشاهده شده از طریق دستور whos در پنجره workspace هم قابل مشاهده است.

می توانیم متغیر هایی که تا به حال در حافظه موقت ذخیره شده است را ذخیره کنیم. برای این کار در قسمت workspace روی گزینه save کلیک کنید و برای ذخیره آن آدرس دهید تا با فرمت mat. ذخیره شود. بعد از بسته شدن نرم افزار متلب , هر وقت خواستید دوباره آموزش را ادامه دهید , نرم افزار را باز کنید و در قسمت workspace به بخش import data بروید و فایل ذخیره شده را انتخاب و گزینه finish را بزنید.

خطا در ورودی

یکی از رایج ترین خطا ها در ورودی که منجر به دریافت پیام eror می شـود , خطای ضرب اسـت. به مثال زیر توجه کنید.

>> 3u^2

در نتیجه متلب eror می دهد. خطا به دلیل فقدان اپراتور ضرب * است.

یکی دیگر از خطا های معمول به پرانتز مربوط می شـود که معمولا پرانتز باز می شـود و بسـته نمی شـود.

عدم تشخیص نوع ورودی مورد نیاز هم یکی خطاهای رایج در متلب است. در ادامه توضیحات بیشتری ارائه می شود.

متغیر ها و نام گذاری آنها

به مثال زیر توجه کنید.

```
>> x = 7
>> syms y
>> z = x<sup>2</sup> - 2*x*y + y
```

همان طور که می بینید به x عدد دادیم و y را سـمبلیک کردیم. برای محاسـبه z , متلب به جای متغیر x مقدار 7 قرار می دهد.

```
>> clear x; syms x y
>> z = x<sup>2</sup> - 2*x*y + y
```

در اینجا متغیر قبلی x را حذف کردیم و دوباره x را سـمبلیک کردیم.

>> 5*y*z

همان طور که دیدید به جای z مقدار حاصل از دستورات قبلی را قرار داد.

حل معادلات

می توانیم عبارات شـامل متغیر ها را با اسـتفاده از دسـتورات solve و fzero حل کنیم. به مثال زیر توجه کنید.

>> solve('x² - 2*x - 4 = 0')

این دستور ریشه های معادله درجه دوم مورد نظر را بدست می دهد. جواب به صورت سمبلیک است. برای مشاهده اسکالر باید دستور (double(ans یا برای رقم های بیشتر دستور (vpa(ans را وارد کنید. دستور solve همچنین می تواند چند جمله ای های درجه بالاتر را هم حل کند. همچنین می تواند معادلاتی با بیش از یک متغیر را حل کند. اگر تعداد معادلات کمتر از تعداد متغیر ها باشـد باید مشخص کنیم کدام متغیر ها باید حل شوند. دستور زیر γ را بر حسب متغیر x بدست می دهد.

>> solve('2*x - log(y) = 1', 'y')

می توانیم با استفاده از دستور solve بیش از یک معادله را حل کنیم.

>> $[x, y] = solve('x^2 - y = 2', 'y - 2*x = 5')$

این سیستم معادلات , دو جواب دارد. جواب اول x با جواب اول y همزمان رخ می دهند. جواب دوم هم به همین صورت.

می توانیم جواب اول x و y را با استفاده از دستورات زیر بگیریم.

>> x(1) >> y(1)

در دستور بالا ابتدا ans به x اختصاص می یابد و سـپس به y. جواب دوم می تواند با دستورات (2)x و (2)y بدست آید.

توجه کنید که اگر در سـمت چپ دسـتور solve بردار تعریف نکنیم ِ متلب نتیجه را نمایش تخواهد داد.

برای مثال

>> sol = solve('x² - y = 2', 'y - 2*x = 5')

در اینجا برای بدست آوردن بردار x تایپ کنید x=sol.x و همچنین برای y تایپ کنید y=sol.y . برای گرفتن مقادیر اول تایپ کنید sol.x(1) .

در بعضی معادلات جواب نمی تواند سـمبلیک باشـد. به همین دلیل متلب جواب اسـکالر می دهد.برای مثال

>> solve('sin(x)=2-x')

گاهی اوقات جواب بیشتر از یکی است و جواب دلخواهمان را نمی گیریم.برای مثال

>> solve('exp(-x) = sin(x)')

جواب , یک عدد مختلط است. حرف i در انتهای عدد دوم نمایانگر رادیکال (1-) است. با وجودی که این یک جواب معتبر برای این معادله است , این معادله جواب های حقیقی هم دارد.

نمودار برخورد (sin(x و exp(-x) در شـکل زیر نشـان داده شـده اسـت. نحوه رسـم این نمودار در ادامه توضیح داده می شـود.



هر کدام از نقاط برخورد یکی از جواب های معادله است.

با تابع fzero می توان جواب های معادله را بصورت عددی بدست آورد. در واقع جواب معادله بالا , صفر های معادله (exp(-x)-sin(x است. دستور زیر این عبارت را بصورت تابعی بر حسب x تبدیل می کند.

```
>> inline('exp(-x) - sin(x)')
```

در مورد دستور inline به طور کامل توضیح داده خواهد شد. دستور fzero برای اجرا در اولین آرگومان به تابع نیاز دارد.

در دستور fzero در اولین آرگومان یک تابع وجود دارد و در دومین آرگومان عددی که نزدیک یکی از ریشـه های معادله اسـت. در نتیجه برای بدسـت آوردن ریشـه های دقیق ابتدا باید مکان تقریبی ریشـه ها را بدسـت آوردن جواب معادله در نزدیکی 0.5 دسـتور زیر را بنویسـید.

```
>> fzero(inline('exp(-x) - sin(x)'), 0.5)
```

برای یافتن ریشه بعدی , باید جواب را در نزدیکی عدد 3 جست و جو کنید.

بردار ها

یک بردار دنباله ای از اعداد است. در متلب می توانیم برداری به طول دلخواه با نوشتن مجموعه ای از اعداد با فاصله در داخل براکت ایجاد کنیم. برای مثال

>> Z = [2, 4, 6, 8]

>> Y = [4 -3 5 -2 8 1]

فرض کنید که می خواهید برداری از اعداد صحیح 1 تا 9 ایجاد کنیم. در زیر روش این کار بدون تایپ اعداد را نشـان می دهد.

>> X = 1:9

در دستور بالا مقدار استپ 1 بود. حالا اگر بخواهیم به فرض استپ را 2 قرار دهیم باید یک آرگومان جدید به آن اضافه کنیم.

>> X = 0:2:10

نمو همچنین میتواند کسری و یا منفی باشد. برای مثال 0:0.1:2 و 1:0-:01 .

آرایه های بردار X میتواند به صورت X(1) و X(2) و ... از بردار استخراج شود. برای مثال << (3) <<

برای تغییر بردار X از یک بردار سطری به یک بردار سـتونی , بعد از X علامت پریم (') قرار دهید.

>> X'

می توانیم روی مولفه های بردار عملیات ریاضی انجام دهیم. برای مثال همه ی آرایه های بردار X را به توان 2 می رسانیم.

>> X.^2

نقطه بکار برده شـده در دسـتور بالا خیلی اهمیت دارد. این دسـتور کاری می کند که اعداد بردار X درایه به درایه و جدا گانه مربع شـوند. اگر تایپ می کردیم X^2 به متلب دسـتور می داد که از ضرب ماتریسـی اسـتفاده کند و بردار X را در خودش ضرب می کرد و در این مورد پیام eror می داد. (در ادامه در مورد ماتریس ها بطور کامل بحث می شـود.)

به طریق مشابه برای ضرب و تقسیم درایه به درایه باید به ترتیب تایپ کنید *. و /. . فقط به یاد داشته باشید که طول بردار ها باید مساوی باشد. برای مثال برای ضرب کردن مولفه های بردار X با مولفه های نظیر از بردار Y تایپ کنید.

>> X.*Y

و

بیشتر دستورات در متلب به صورت مولفه به مولفه به ورودی اعمال می شوند. برای مثال در عملیات جمع و تفریق نقطه نمی گذاریم. مثلا می توانیم تایپ کنیم (exp(X تا نمایی همه اعداد بردار X را بدست آوریم.

ماتریس ها

ماتریس یک آرایه مستطیلی از اعداد است . بردار های سطری و ستونی که در بالا روی آنها بحث شد نمونه ای از ماتریس بودند.ماتریس 4×3 زیر را ملاحظه کنید.

	(1	2	3	4	
A =	5	6	7	8	
	9	10	11	12)	

این ماتریس می تواند در متلب با دستور زیر وارد شود.

>> A = [1, 2, 3, 4; 5, 6, 7, 8; 9, 10, 11, 12]

توجه کنید که مولفه های هر سطر با کاما و سطر ها توسط سمی کالن از هم جدا شده اند.

اگر بین مولفه های هر سطر کاما نگذاریم هم فرقی نمی کند. ولی فاصله نباید فراموش شود.

اگر دو ماتریس A و B دارای سایز برابر باشـند , جمع مولفه با مولفه آنها با نوشـتن A+B بدسـت می آید. همچنین می توان یک عدد اسـکالر را به تک تک مولفه های یک ماتریس اضافه کرد. به طریق مشـابه A-B تفاضل دو ماتریس A و B را به دسـت می دهد. همچنین A-C یک عدد اسـکالر را از همه مولفه های ماتریس A کم می کند.

اگر A قابلیت ضرب ماتریسـی در B را داشـته باشـد (یعنی اگر A ماتریسـی m×n و B ماتریسـی ا×m باشـد .) سپس حاصل ضرب ماتریسـی A*B یک ماتریس ا×n خواهد بود. یاد آوری می کنم که مولفه ماتریس A*B در i مین سطر و j مین سـتون برابر اسـت با مجموع حاصل ضرب مولفه های سـطر i ام از A در سـتون j ام از B . به عبارتی دیگر

$$(\mathbf{A} * \mathbf{B})_{ij} = \sum_{k=1}^{m} \mathbf{A}_{ik} \mathbf{B}_{kj}, \ 1 \le i \le n, \ 1 \le j \le \ell.$$

ضرب عدد اسکالر c در بردار A به صورت C*A نوشـته می شـود و دسـتور 'A ماتریس ترانهاده مزدوج A را به دسـت می ده*د*.

ضرب ماتریس 4×3 A در بردار 'Z که 1×4 است می تواند مثال خوبی باشد.

>> A*Z'

که حاصل دستور بالا , یک ماتریس ستونی 1×3 است.

متلب دستورات دیگری نیز در این زمینه دارد که در فصل های بعدی به آن اشاره خواهد شد.

متوقف كردن خروجى

تایپ علامت سمی کالن (;) در انتهای خط ورودی , مانع چاپ خروجی خواهد شد. زمانی که بخواهیم بردار یا ماتریس بزرگی تعریف کنیم از سمی کالن استفاده می کنیم. برای مثال بردار (x = -1:0.1:2) چاپ نشود بهتر است. همچنین در هر جایی که نیازی به چاپ شدن خروجی نباشد می توانیم از سمی کالن استفاده کنیم.

توابع

در متلب هم می توانید از توابع پیش فرض و هم از توابع ساخت خودتان استفاده کنید.

توابع پیش فرض

متلب تعداد زیادی تابع غیر قابل تغییر دارد. از جمله آنها توابع sqrt و sin و cos و tan و log و exp و atan (برای arctan) همچنین تعدادی تابع ریاضی خاص مثل gamma و erf و besselj. متلب همچنین تعداد زیادی ثابت پیش فرض دارد. از جمله pi (عدد پی) و i (رادیکال منفی یک) و inf (∞). در زیر چند مثال می زنیم

```
>> log(exp(3))
```

تابع log لگاریتم طبیعی می گیرد و در ریاضیات با نماد " ln " نشـان داده می شـود.

```
>> sin(2*pi/3)
```

در بالا برای گرفتن جواب دقیق باید از آرگومان سمبلیک استفاده کنید

```
>> sin(sym('2*pi/3'))
```

توابع تعريف شده توسط كاربر

در این بخش نشـان می دهیم که چگونه می توان از تابع inline برای تعریف یک تابع اسـتفاده کرد.

به نحوه تعریف تابع چند جمله ای f(x) توجه کنید

>> f = inline('x² + x + 1', 'x')

آرگومان دوم بکار برده شده در inline , متغیر مستقل را مشخص می کند. می توانیم این قسمت را حذف کنیم. متلب از طریق قوانین متغیر مستقل حدس می زند که چه باید باشد. در ادامه در این مورد بیشتر بحث خواهد شد.

وقتی که تابع را تعریف کردیم , میتوانیم به ازای ورودی های مختلف از آن خروجی بگیریم. <> (4) (4)

تابع تعریف شـده در بالا فقط می تواند به عدد اسـکالر اعمال شـود. برای این که بتواند برای بردار هم به کار رود از تابع vectorize متلب اسـتفاده می کنیم.

تابع زیر تابع برداری شده معادله f است.

>> f1 = inline(vectorize('x² + x + 1'), 'x')

توجه کنید که ^ باید با ^. جایگزین می شد. حالا می توانیم به f1 ورودی بردار دهیم.

>> f1(1:5)

می توانیم f1 را از طریق بخش گرافیک متلب از راه های مختلف رسـم کنیم. این بخش را با اشـاره به این نکته که می توان تابعی با چند متغیر تعریف کرد به پایان می رسـانیم.

>> g = inline('u² + v²', 'u', 'v')

گرافیک

در این بخش دو دسـتور رسـم مقدماتی متلب را تعریف می کنیم و نحوه اسـتفاده از آنها را توضیح خواهیم داد.

رسم نمودار با استفاده از ezplot

ساده ترین راه رسـم یک تابع تک متغیره اسـتفاده از ezplot اسـت. برای رسـم معادله "x²+x+1" در بازه 2- تا 2 داریم

>> ezplot('x² + x + 1', [-2 2])

نمودار در پنجره جدیدی به نام "Figure 1 " نشان داده می شود.

قابل ذکر است که ezplot هم آرگومان رشته ای (مانند بالا) و هم عبارت سمبلیک را قبول می کند.

نمودار گرفته شده در مثال قبل را از طریق دستور زیر نیز می توان مشاهده کرد.

>> syms x >> ezplot(x² + x + 1, [-2 2])



نمودار ها می توانند گمراه کننده باشند اگر به محور ها توجه نکنیم.

برای مثال نمودار "x²+x+3" نیز دقیقا مانند نمودار بالا اسـت با این تفاوت که اعداد روی محور فرق می کند.

تغيير دادن نمودار

*ب*ه چند طریق می توانیم نمودار ها را تغییر دهیم. برای تغییر عنوان بالای نمودار قبلی در زیر دستورات قبلی تایپ کنید

>> title 'A Parabola'

و اجرا کنید. در نتیجه عبارت بالای نمودار تغییر می کند.

می توانیم روی محور افقی نمودار توسط اضافه کردن دستور xlabel به دستورات قبلی برچسب ایجاد کنیم. برای این کار باید جلوی xlabel اسم دل خواه خود را قرار دهیم. برای محور عمودی نیز از دستور ylabel استفاده کنید. به این نکته توجه کنید که عبارت مقابل title حتما باید با علامت آپوستروف بسته شود ولی برای گذاشتن برچسب محور ها نیازی نیست.

همچنین می توانیم محدوده محور ها را به دلخواه تنظیم کنیم. مثلا در مثال فوق می توانیم محدوده نمایش روی محور عمودی را از 1 تا 4 محدود کنیم.

>> axis([-2 2 1 4])

دو عدد اول محدوده محور افقی را مشخص می کنند. در دستور بالا محدوده هر دو محور باید مشخص شود.

در فصل پنجم توضیحات بیشتری در این زمینه داده می شود.

رسم نمودار با استفاده از plot

دستور plot به داده های برداری عددی اعمال می شود. ساده ترین ترکیب این تابع (X,Y) plot است که در آن X و Y بردار هایی با طول یکسان هستند. برای مثال

```
>> X = [1 2 3];
>> Y = [4 6 5];
>> plot(X, Y)
```

این دستور نقطه (1,4) را با یک خط به نقطه (2,6) وصل می کند و به همین صورت برای نقطه بعدی.



برای ترسیم نمودار "X²+X+1" در بازه 2- تا 2 ابتدا باید برداری از مقادیر X تعریف کنیم . تعداد نقاط تعریف شـده باید به حدی باشـد که نمودار شـکسـته شـکسـته نباشـد. برای مثال زیر از اسـتپ 0.1 اسـتفاده می کنیم.

>> X = -2:0.1:2; >> plot(X, X.² + X + 1)



توجه کنید که برای جلو گیری از چاپ شدن 41 مولفه بردار X از سمی کالن استفاده کردیم. همچنین توجه کنید که دستور زیر همان نتیجه را خواهد داشت. زیرا قبلا تابع f1 را تعریف کرده ایم.

```
>> plot(X, f1(X))
```

در فصل پنجم دستورات گرافیکی بیشتری را توضیح خواهیم داد.

رسم منحنی های چندگانه

وقتی یک نمودار جدید را رسم می کنیم , متلب نمودار قدیمی را پاک می کند و به جای آن نمودار جدید را قرار می دهد. اگر بخواهیم دو یا تعداد بیشتری نمودار را روی هم رسم کنیم از دستور hold on استفاده می کنیم. این عبارت به متلب فرمان می دهد که نمودار قبلی را نگه دارد و نمودار های جدیدی که رسم می شود بر روی همان نمودار قدیمی رسم کند. این حالت تا وقتی که دستور hold off را وارد کنیم ادامه می یابد. در اینجا مثالی از به کار گیری تابع ezplot می آوریم.

```
>> ezplot('exp(-x)', [0 10])
>> hold on
>> ezplot('sin(x)', [0 10])
>> hold off
>> title 'exp(-x) and sin(x)'
```

نتیجه قبلا در همین بخش نمایش داده شـده بود. دسـتورات hold on و hold off با همه دسـتورات گرافیکی دیگر نیز به کار می رود.

با دستور plot می توان مستقیما چند نمودار را روی هم رسم کرد.

```
>> X = 0:0.1:10;
>> plot(X, exp(-X), X, sin(X))
```

فقط با این تفاوت که رنگ نمودار ها را متفاوت انتخاب می کند.