پیش‌گفتار

در این فصل، اصول اولیه‌ی برنامه‌نویسی در نرم‌افزار MATLAB مورد بررسی قرار می‌گیرد. این بررسی شامل معرفی انواع داده‌ها، عملگرها و تابع‌های پایه موجود در این نرم‌افزار می‌باشد. علاوه بر این، شیوه‌های مختلف برنامه‌نویسی در این نرم‌افزار همراه با نحوه‌ی تولید تابع‌های دلخواه ارائه می‌گردد. در پایان فصل، از خواننده این انتظار می‌رود که بتواند الگوریتم‌های عددی ساده را در MATLAB پیاده‌سازی کرده و خروجی‌های مرتبط را تولید نماید. یادگیری تک تک این اصول هر چند ساده است، اما فهم ارتباط بین آنهاست که به خواننده کمک می‌کند تا بتواند یک سیستم مخابراتی پیچیده را شبیه‌سازی و تحلیل نماید. تفاوت آنچه در این کتاب ارائه می‌شود با راهنمای اصلی نرم‌افزار MATLAB، در بیان همین ارتباط‌ها می‌باشد. به عبارت دیگر، یادگیری همه‌ی دستورهای این نرم‌افزار لزوماً به معنای ایجاد توانایی در جهت شبیه‌سازی و تحلیل نمی‌باشد. از این رو، در این فصل و سایر فصل‌ها تلاش می‌شود تا ارتباط بین قابلیت‌های این نرم‌افزار و نیازهای یک مهندس مخابرات جهت شبیه‌سازی و تحلیل سیستم‌های مخابراتی به خوبی تبیین گردد. بدین منظور، تلاش می‌شود با مطرح کردن مثال‌های متعدد و تا حد ممکن کاربردی به یادگیری بهتر و بیشتر کمک شود.

۱-۱- انواع داده‌ها در MATLAB

آرایه‌ها[1] پایه انواع داده‌های مورد استفاده در MATLAB می‌باشند. آرایه در مرسوم‌ترین شکل ماتریسی، متشکل از اعداد حقیقی (یا مختلط) است و در حالت خاصی که این ماتریس دارای یک عنصر باشد، نماینده‌ی یک عدد اسکالر خواهد بود. در حالت خاص دیگری که تنها یکی از ابعاد برابر یک باشد، یک بردار سطری یا ستونی خواهیم داشت. به جز استفاده از تابع‌های آماده (یا تابع‌های نوشته شده توسط کاربر) برای تولید و معرفی ماتریس‌ها، یکی از راه‌های مرسوم برای معرفی یک ماتریس عددی در MATLAB وارد کردن عناصر ماتریس به صورت صریح می‌باشد. برای این منظور باید سه قاعده‌ی زیر را استفاده نمود:

- عناصر یک سطر با استفاده از یک فاصله یا کاما از یکدیگر مجزا می‌گردند.
- از سمیکالن (;) برای مشخص کردن پایان هر سطر استفاده می‌شود.
- دو طرف لیست اعضا با استفاده از کروشه بسته می‌شود.

1. Arrays

مثال ۱-۱: نحوه‌ی معرفی یک ماتریس عددی دو بُعدی

برای معرفی ماتریس $A=\begin{bmatrix}1 & 5 & 4\\ 2 & 0 & 9\end{bmatrix}$، از دستور زیر استفاده می‌نماییم:

```
>> A = [1 5 4 ; 2 0 9]
```

با اجرای این دستور در پنجره‌ی دستور[1] نتیجه به صورت زیر نمایش داده خواهد شد:

```
A =
  1  5  4
  2  0  9
```

همان‌طور که ملاحظه شد، برای معرفی ماتریس A در پنجره‌ی دستور از هیچ علامتی در پایان دستور استفاده نشده است. در این مواقع MATLAB نتیجه‌ی کامل اجرای دستور را همان‌طور که دیده شد، نمایش می‌دهد. اگر بخواهیم نتیجه‌ی اجرای دستور نمایش داده نشود، باید از علامت سمیکالن (;) در انتهای دستور استفاده نماییم. در این حالت دستور تماماً اجرا می‌گردد ولی نتیجه‌ی آن نمایش داده نمی‌شود. در هر صورت، بعد از اجرای دستور متغیر A به صورت خودکار در محیط کار MATLAB قابل فراخوانی است.

برای دسترسی به عناصر مختلف یک ماتریس روش‌های متنوعی وجود دارد. به عنوان مثال، A(i,j) نمایشگر عنصری از ماتریس A است که در سطر i ام و ستون j ام آن قرار دارد. به عنوان مثالی دیگر، A(:,j) اشاره به ستون j ام ماتریس A دارد. به طریق مشابه، A(i,:) برابر سطر i ام ماتریس A می‌باشد. علامت دونقطه (:) در حقیقت یکی از مهم‌ترین عملگرهای مورد استفاده در MATLAB می‌باشد. مثلاً برای معرفی یک بردار که از مقدار مشخص a شروع شده و با افزایش معادل b تایی در هر مرحله تا مقداری که حداکثر مساوی c است ادامه پیدا می‌کند، از عبارت (a:b:c) استفاده می‌شود. همچنین (a:c) برای معرفی برداری استفاده می‌شود که از مقدار مشخص a شروع شده و با افزایش یک واحدی در هر مرحله تا مقداری که حداکثر مساوی c است ادامه پیدا می‌کند. استفاده از نماد A(:) باعث تولید یک بردار ستونی از عناصر ماتریس A می‌شود که ترتیب خواندن عناصر A در آن به صورت ستون به ستون است. با اشاره تک بُعدی به ماتریس دو بعدی A، در حقیقت عناصر A(:) فراخوانی می‌گردند. مثلاً با در نظر گرفتن ماتریس A که در مثال ۱-۱ تعریف شده است، A(3) همان A(1,2) است. در هنگام اشاره به عناصر مختلف ماتریس، نماد end نشانگر آخرین عنصر در یک بُعد مشخص است. مثلاً A(i,end) نشانگر آخرین عنصر سطر i ام است. لازم به ذکر است که برای دسترسی به عناصر یک بردار، مشخص کردن یک اندیس کافی است. هر چند مثلاً در مورد یک بردار سطری به طول n مانند B، می‌توان آن را یک ماتریس با

1.Command Window

ابعاد 1×n در نظر گرفته و به جای B(j) از عبارت B(1,j) استفاده نمود.

مثال ۱-۲: بررسی روش‌های مختلف برای دسترسی به عناصر یک ماتریس

با در نظر گرفتن ماتریس A که در مثال ۱-۱ تعریف شده، می‌توان به عناصر مختلف آن به شیوه‌های ذکر شده، دسترسی پیدا کرد.

```
>> A(2,3)

ans =

    9

>> A(1,:)

ans =

    1    5    4

>> A(:,2:3)

ans =

    5    4
    0    9

>> A(:,2)

ans =

    5
    0

>> A(1:end,[1 3])

ans =

    1    4
```

```
    2     9

>> A(:,3:-2:1)

ans =

    4     1
    9     2

>> B = A(:)

B =

    1
    2
    5
    0
    4
    9

>> A(5)

ans =

    4

>> B(5,1)

ans =

    4

>> B(5)

ans =

    4
```

لازم به توضیح است دستوری مانند A(1:end,[1 3]) که در این مجموعه دستورها به آن اشاره شد به معنای انتخاب عناصری از ماتریس A است که در تمامی سطرها و تنها در ستون‌های اول و آخر قرار دارند. دستوری که حاصل آن با توجه به ابعاد ماتریس A، ماتریسی 2×2 است.

توجه: در مثال‌ها، نماد << در کنار یک عبارت، به این معناست که این دستور در پنجره‌ی دستور اجرا شده است و عبارت زیر آن نتیجه‌ی حاصل از اجرای آن می‌باشد.

تفاوت عمده‌ای که نرم‌افزار MATLAB با سایر نرم‌افزارهای برنامه‌نویسی دارد، این است که در آن نیاز به معرفی نوع متغیر و یا اندازه‌ی آن نیست. در حقیقت وقتی MATLAB با یک متغیر جدید روبرو می‌گردد، به صورت خودکار نوع و اندازه‌ی آن را مشخص می‌کند و حافظه‌ی لازم را به آن اختصاص می‌دهد. در حالتی که متغیر قبلاً معرفی شده باشد، مقدار جدید به آن اختصاص داده می‌شود و همچنین در صورت لزوم اندازه‌ی آن تغییر می‌کند.

نام متغیرها در MATLAB با یک حرف الفبا (بزرگ یا کوچک) شروع می‌شود و توسط تعدادی حرف، عدد و یا علامت _ (Underline) ادامه پیدا می‌کند. لازم به ذکر است که در تعریف متغیرها، میان حروف بزرگ وکوچک تمایز وجود دارد. مثلاً در صورت معرفی متغیر A_1، نمی‌توان با نام a_1 آن را فراخوانی نمود. هر چند نام متغیر می‌تواند هر طولی داشته باشد، اما نرم‌افزار MATLAB فقط N کاراکتر اول نام را در نظر می‌گیرد. عدد N بسته به نسخه‌ی نرم‌افزار متفاوت می‌باشد. در نسخه‌های قبل از MATLAB 6.5 ، این عدد برابر ۳۱ و از این نسخه به بعد به ۶۳ افزایش داده شده است.

برای معرفی یک متغیر ماتریسی، همان‌طور که در مثال ۱-۱ دیده شد، می‌توان با استفاده از یک دستور این کار را انجام داد. هر چند اجباری در این کار نیست و با معرفی تک تک عناصر هم این کار قابل انجام است. اما با استفاده از روش دوم زمان بیشتری از پردازشگر گرفته می‌شود؛ زیرا با معرفی عناصر به صورت مجزا، MATLAB در هر مرحله به اندازه‌ی عنصر جدید، حافظه‌ی مورد نیاز برای متغیر را افزایش می‌دهد. در معرفی یک بازه (عنصر تدریجی) مقدار کل حافظه‌ی لازم برای ذخیره‌ی متغیر در یک مرحله به آن اختصاص داده می‌شود. این مسئله در ماتریس‌های کوچک چندان اهمیت ندارد، اما توجه کنید که در مسائلی که با ماتریس‌های بزرگ سروکار دارند، می‌تواند به لحاظ زمان پردازش مشکل‌ساز گردد. نکته‌ی دیگر این که در معرفی یک متغیر برابر متغیری دیگر، مقدار دو متغیر یکسان ولی حافظه‌ای مرتبط با آنها متفاوت است؛ پس با تغییر در هر یک از این دو، دیگری تغییر نمی‌کند. به عبارت دیگر، پس از عبارت تساوی، مقدارهای هر یک از

این دو متغیر را می‌توان جداگانه تغییر داد.

مثال ۱-۳: معرفی یکباره و تدریجی یک ماتریس

می‌خواهیم بردار [1 3 2] را در دو حالت یکباره و تدریجی به ترتیب با نام‌های A_1 و a_1 معرفی نماییم. به طوری که بردار a_2 مثل بردار A_1 باشد با این تفاوت که به عنصر دوم آن مقدار 1- داده شود.

```
>> A_1 = [1 3 2]

A_1 =

    1    3    2

>> a_1(2) = 3

a_1 =

    0    3

>> a_1(1) = 1

a_1 =

    1    3

>> a_1(3) = 2

a_1 =

    1    3    2

>> a_2 = A_1

a_2 =

    1    3    2
```

```
>> a_2(2) = -1

a_2 =

    1   -1    2
```

برای نوشتن اعداد در MATLAB از روش مرسوم نمایش دسیمال به همراه علامت مثبت یا منفی (مانند 1.6- یا 3.763) استفاده می‌شود. علاوه بر این، از نمایش علمی با نماد e (مانند 1.2e3- به جای 1200-) نیز می‌توان برای نوشتن اعداد استفاده نمود. برای مشخص کردن اعداد مختلط نیز می‌توان از نماد i و یا j (مانند 2+3.5j یا 2i-) استفاده نمود و این مسئله تضادی با معرفی متغیری به نام i و یا j به صورت هم‌زمان ندارد. اگر متغیری با این نام‌ها معرفی نشود، نماد i و j در محاسبات برابر عدد 1j (یا 1i) در نظر گرفته می‌شود.

برای ذخیره‌سازی اعداد در MATLAB از استاندارد ممیز شناور با دقت مضاعف IEEE استفاده می‌شود که دقت آن در حدود ۱۶ رقم اعشار (در مبنای ۱۰) است و صرف‌نظر از علامتش می‌تواند مقدارهایی در محدوده‌ی 10^{-308} تا 10^{+308} داشته باشد. این نوع ذخیره‌سازی عددها که با نام double در MATLAB شناخته می‌شود، دارای دقت بالایی است؛ البته باید به محدودیت‌های آن توجه کرد. مثلاً با محاسبه‌ی دقیق مقدار عبارت 1-3×(4/3-1) در محیط MATLAB، حاصل عبارت برابر عدد صفر نیست، بلکه برابر با 2^{-52}است. به عنوان مثال دیگر، با محاسبه‌ی مقدار عبارت 1-7×(8/7-1) در محیط MATLAB به عدد 2^{-51}دست پیدا خواهیم نمود. لازم به ذکر است که قسمت‌های مختلط و حقیقی اعداد به صورت جداگانه با استاندارد ممیز شناور IEEE در حافظه ذخیره می‌گردند.

نوع double که از هشت بایت برای ذخیره‌سازی هر عدد استفاده می‌کند، نوع عددی پیش‌فرض در MATLAB می‌باشد. علاوه بر این نوع داده، انواع داده‌ی عددی دیگری نیز در MATLAB قابل استفاده می‌باشند. نوع single، نوع دیگری از اعداد در MATLAB است که از چهار بایت برای ذخیره‌سازی هر عدد استفاده می‌کند و دقت آن در حدود هفت رقم اعشار (در مبنای ۱۰) است و صرف‌نظر از علامتش می‌تواند مقدارهایی در محدوده‌ی 10^{-38} تا 10^{+38} داشته باشد. لازم به ذکر است که برای ذخیره‌سازی نوع داده‌ی single در MATLAB ، از استاندارد ممیز شناور ۳۲ بیتی IEEE استفاده می‌شود. برای تولید این نوع داده در MATLAB ، باید از دستور Single استفاده نمود که یک ماتریس عددی را به عنوان آرگومان ورودی می‌پذیرد. مثلاً single([1 2 3])، برداری سطری به طول سه از نوع داده‌ی single تولید می‌نماید. نماد Inf (یا inf) نمادی

است که در MATLAB برای نشان دادن بزرگ‌ترین مقدار مثبت ممکن در انواع داده‌ی single و double استفاده می‌شود. نماد Inf- (یا Inf-) نیز نمادی است که برای نشان دادن کوچک‌ترین مقدار منفی ممکن در انواع داده‌ی single و double استفاده می‌شود. نمادهای Inf و Inf- در حقیقت به ترتیب معادل نمادهای ریاضی ∞ و $-\infty$ می‌باشد. مثلاً حاصل‌ضرب دو عدد 10^{200} و -10^{200} با نوع داده‌ی double در MATLAB ، inf- می‌باشد و یا از تقسیم عدد یک بر صفر در MATLAB، Inf حاصل می‌شود. نماد NaN (یا nan) نیز نمادی است که برای یک عدد نامشخص (تعریف نشده) در این دو نوع داده به کار می‌رود. به عنوان مثال نتیجه‌ی حاصل‌ضرب اعداد 0 و NaN، Inf خواهد بود.

مثال ۱-۴: مقایسه‌ی نوع داده‌های double و single

با تعریف ماتریس A=[1 4 2]، می‌خواهیم ماتریس B را مانند ماتریس A ولی از نوع single تولید کنیم که آن عنصر آخر آن برابر ∞ باشد. بعد از انجام این کار، می‌خواهیم عنصر سوم ماتریس A را برابر 10^{45} قرار دهیم. در نهایت مطلوب است بردار C مانند بردار B ولی از نوع double باشد.

```
>> A = [1 4 2];
>> B = single(A);
>> B(3) = 1e45

B =

    1     4   Inf

>> A(3) = 1e45

A =

  1.0e+045 *

   0.0000    0.0000    1.0000

>> C = double(B)

C =

    1     4   Inf
```

علاوه بر دو نوع double و single که برای ذخیره‌سازی اعداد اعشاری به کار می‌روند، انواع دیگری از داده‌ها صرفاً برای ذخیره‌سازی اعداد صحیح پیش‌بینی شده‌اند. لیست این نوع داده‌ها در جدول ۱-۱ آمده است. با توجه به این جدول و به عنوان مثال، برای معرفی عدد یک به عنوان یک عدد صحیح بدون علامت هشت بیتی می‌توان از دستور uint8(1) استفاده کرد. لازم به ذکر است که در تبدیل اعداد اعشاری به یک عدد صحیح، اعداد Inf و Inf- به ترتیب به بزرگ‌ترین و کوچک‌ترین اعداد ممکن در نوع داده‌ی صحیح مورد نظر تبدیل می‌شوند. علاوه بر این، NaN به عدد صفر تبدیل می‌شود.

هر چند در عمل به طور عمده با آرایه‌های دو بُعدی (یا ماتریس‌ها) سروکار داریم، اما در MATLAB امکان معرفی آرایه‌هایی با بیش از دو بُعد نیز وجود دارد. مثلاً با معرفی یک آرایه‌ی سه‌بُعدی A با ابعاد m×n×k، A(:,:,1) یک آرایه دو بُعدی m×n (یا یک ماتریس) است. در مثال ۱-۵، نحوه‌ی تعریف یک آرایه‌ی سه بُعدی برای نوع داده‌ی int16 نشان داده شده است.

جدول ۱-۱ - انواع داده‌های صحیح در MATLAB

نوع داده	محدوده‌ی مقدارهای	نماد مربوطه در MATLAB
صحیح و علامت‌دار ۸ بیتی	-2^{7} الی $2^{7}-1$	int8
صحیح و علامت‌دار ۱۶ بیتی	-2^{15} الی $2^{15}-1$	int16
صحیح و علامت‌دار ۳۲ بیتی	-2^{31} الی $2^{31}-1$	int32
صحیح و علامت‌دار ۶۴ بیتی	-2^{63} الی $2^{63}-1$	int64
صحیح و بدون علامت ۸ بیتی	0 تا $2^{8}-1$	uint8
صحیح و بدون علامت ۱۶ بیتی	0 تا $2^{16}-1$	uint16
صحیح و بدون علامت ۳۲ بیتی	0 تا $2^{32}-1$	uint32
صحیح و بدون علامت ۶۴ بیتی	0 تا $2^{64}-1$	uint64

مثال ۱-۵: آرایه‌های چند بُعدی

می‌خواهیم با استفاده از دو آرایه‌ی دو بُعدی از نوع int16 که یکی مختلط و دیگری حقیقی است، یک آرایه‌ی سه بُعدی مختلط تعریف کنیم.

```
>> A1 = int16([2 3i ; -3i+1 5]);
>> A2 = int16([3 0 ; 7 1]);
>> A(:,:,1) = A1;
```