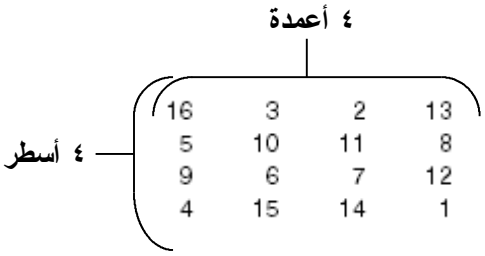
الفصل الأول
المصفوفات

ـــ مقدمة :

ذكرنا سابقا أن MATLAB يتعامل مع جميع البيانات على أنها مصفوفات ، لذلك لابد لنا أن نقف عند المصفوفات لنرى كيف نتعامل معها في MATLAB ، لأنها اللبنة الأساسية لهذا البرنامج .

المصفوفات هي عبارة عن مجموعة من الأعداد المرصوصة على شكل مربع أو مستطيل ، مثلا :
١ ـــ مصفوفة مربعة : يكون فيها عدد الأسطر مساوي لعدد الأعمدة فمثلا المصفوفة التالية مصفوفة مربعة ذات ٤ أسطر و ٤ أعمدة :



٢ـــ مصفوفة مستطيلة : يكون فيها عدد الأسطر لا يساوي عدد الأعمدة فمثلا المصفوفة التالية مصفوفة مستطيلة ذات ٤ أسطر و ٥ أعمدة :

```
16   3   2  13   0
 5  10  11   8   0
 9   6   7  12   0
 4  15  14   1  17
```

· ملاحظات :
١. العدد المفرد يعتبر مصفوفة مربعة ذات سطر واحد و عمود واحد ، مثلا : (٥) هي مصفوفة مربعة ذات سطر واحد و عمود واحد .
٢. عادة نقول أن مصفوفة أبعادها (m , n) ، أي أنها ذات m سطر و n عمود .
٣. تسمى المصفوفات ذات الأبعاد (1, n) أو (m , 1) بالأشعة .

ـــ إدخال المصفوفات في MATLAB :

هناك أربع طرق لإدخال المصفوفات في MATLAB :
١. إدخال المصفوفة على شكل قائمة من العناصر (الأعداد) .
٢. إدخال المصفوفة و ذلك عن طريق تحميلها من ملف .
٣. توليد المصفوفة عن طريق توابع و دوال مبنية في برنامج MATLAB .
٤. إنشاء المصفوفة عن طريق توابع و دوال نقوم بإنشائها نحن باستخدام ملفات (M_Files) .

لإدخال مصفوفة عل شكل قائمة من العناصر نقوم بما يلي :
١) عناصر السطر الواحد نترك بينها مسافات فارغة أو فواصل (,) .
٢) نهاية كل سطر نضع فاصلة منقوطة (;) .
٣) نحيط قائمة العناصر هذه بقوسين مربعين [ ] .
مثلا : عند كتابة السطر التالي في الـــ (Command Window) :

```
A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]
```

PDF created with pdfFactory trial version www.pdffactory.com

ستظهر لنا النتيجة التالية في الـــ (Command Window) :

```
A =
    16     3     2    13
     5    10    11     8
     9     6     7    12
     4    15    14     1
```

يقوم MATLAB بتخزين عناصر المصفوفة السابقة في متغير اسمه (A) ، أصبحت المصفوفة (A) موجودة و مخزنة في فضاء العمل (Work Space) ، و عندما نكتب في الـــ (Command Window) الحرف (A) ستظهر لنـــا عناصر هذه المصفوفة ، و عندما نفتح الـــ (Work Space) ستظهر لنا المتغيرات التي استخدمناها في برنامجنا إضافة إلى أبعادها و كم هو حجمها في الذاكرة إضافة إلى نوع عناصرها ، و في مثالنا هذا سيظهر لنا أن اسم المصفوفـــة هـــو (A) و حجمها (2×2) و نوع عناصرها هو (Int) .

يمكننا تحديد عنصر ما في مصفوفة بطريقتين :

· تحديد العنصر عن طريق السطر و العمود : في مثالنا السابق ، لنفرض أننا نريد العنصر ذو السطر الثالث و العمود الرابـــع ، لإنجاز ذلك نقوم بكتابة ما يلي في الـــ (Command Window) :

```
A( 3 , 4 )
```

و ستظهر عندها النتيجة التالية :

```
ans=
    12
```

· تحديد العنصر عن طريق ترتيبه : لكل عنصر في المصفوفة ترتيب ، يبدأ الترقيم من العنصر الأول فـــي الزاويـــة اليســـرى العلوية ثم بعد ذلك العنصر الذي تحته يأخذ الرقم ٢ و الذي تحته يأخذ رقم ٣ و هكذا حتى ينتـــهي العمـــود الأول ، نتـــابع الترقيم ابتداء من العمود الثاني و هكذا حتى ينتهي العمود الثاني ، ثم العمود الثالث ... و هكذا .

ففي مثالنا السابق ، لنفرض أننا نريد العنصر ذو الترتيب ٧ ، عندئذ نكتب في الـــــ (Command Window) الســـطر التالي :

```
A( 7 )
```

و ستظهر لنا النتيجة التالية :

```
ans=
     6
```

Ü ملاحظات :

١. لاحظ أنه عندما لا نضع متحول خرج فإن MATLAB يعطي متحول خرج افتراضي اسمه (ans) ، أي أنه فـــي مثالنـــا السابق عندما نكتب :

```
X = A( 3 , 4 )
```

فإنه ستظهر لنا النتيجة التالية :

```
X =
    12
```

و هذا يعني أن MATLAB قد خزن العدد 12 في مصفوفة اسمها X ، و عندما نفتح الـــ (Work space) فإننـــا ســـنجد إضافة للمصفوفة A السابقة ، سنجد المصفوفة X ، و هذه المصفوفة أبعادها (1×1) .

أما عندما لا نضع متحول خرج فإن متحول الخرج الافتراضي في MATLAB هو (ans) .

٢. يمكن إجراء العمليات الحسابية على عناصر محددة في المصفوفة ، ففي مثالنا السابق ، عندما نكتب السطر التالي :

```
A(1 , 2) + A(3 , 4) - A(2 , 2 )
```

ستظهر لنا النتيجة التالية :

```
ans=
    18
```

٣. عندما نحدد عنصرا خارج المجال فإنه ستظهر لنا رسالة الخطأ التالية :

```
Index exceeds matrix dimensions.
```

PDF created with pdfFactory trial version www.pdffactory.com

٤. إذا قمنا بتخزين قيمة جديدة في مصفوفة ، و كانت هذه القيمة خارج مجال المصفوفة ، فـــإن MATLAB ســـيكمل بقيـــة العناصر الفارغة في المصفوفة بأصفار ، ففي المثال السابق ، عندما نكتب الأسطر التالية :

```
X = A;
X(4,5) = 17
```

ستظهر لنا النتيجة التالية :

```
X =
    16     3     2    13     0
     5    10    11     8     0
     9     6     7    12     0
     4    15    14     1    17
```

ـــ المعامل (:) :

يعتبر المعامل (:) من أهم المعاملات في MATLAB ، و هو يقوم بإنشاء مصفوفات سطرية (أشعة سطرية) كما أنه يمكننا من تحديد مجموعة من العناصر المتتالية ، و سنذكر بعض الأمثلة على ذلك حتى تتضح الصورة بشكل أفضل .

١. عند كتابة السطر التالي في الـــ (Command Window) :

```
1:10
```

فإن MATLAB سيقوم بإنشاء مصفوفة سطرية { المصفوفة السطرية أبعادها (1 , n) } تبدأ عناصرها بالعدد (1) و تتزايد بمقدار (1) حتى تصل إلى العدد (10) ، أي ستظهر لنا النتيجة التالية :

```
1     2     3     4     5     6     7     8     9    10
```

٢. عند كتابة السطر التالي في الـــ (Command Window) :

```
100:-7:50
```

فإن MATLAB سيقوم بإنشاء مصفوفة سطرية تبدأ عناصرها بالعدد (100) و تتناقص بمقدار (7) حتى تصـــل إلـــى العدد (50) ، أي ستظهر لنا النتيجة التالية :

```
100    93    86    79    72    65    58    51
```

٣. عند كتابة السطر التالي في الـــ (Command Window) :

```
0:pi/4:pi
```

فإن MATLAB سيقوم بإنشاء مصفوفة سطرية تبدأ عناصرها بالعدد (0) و تتزايد بخطوة مقدارها (pi/4) حتى تصـــل إلى العدد (pi = 3.1415926) ، أي ستظهر لنا النتيجة التالية :

```
0    0.7854    1.5708    2.3562    3.1416
```

٤. عند كتابة السطر التالي في الـــ (Command Window) :

```
A(1:k,j)
```

فإن MATLAB سيقوم بتحديد أول (K) عنصر من العمود (j) في المصفوفة (A) ، فمثلا عند كتابة :

A(1:3,3)

فإنه ستظهر لنا النتيجة التالية :

2 11 7

PDF created with pdfFactory trial version www.pdffactory.com

٥. عند كتابة السطر التالي في الـــ (Command Window) :

A(: , :)

فإن هذا يعني تحديد جميع عناصر المصفوفة (A) ، و ستظهر لنا المصفوفـــة (A) كمـــا لـــو أننـــا كتبنـــا (A) فـــي الـــ (Command Window) .

٦. عند كتابة السطر التالي في الـــ (Command Window) :

A(:,end)

فإن MATLAB سيقوم بتحديد عناصر آخر سطر في المصفوفة A ، و ستظهر لذلك النتيجة التالية :

```
     4    15    14     1
```

٧. عند كتابة السطر التالي في الـــ (Command Window) :

A(end , :)

فإن MATLAB سيقوم بتحديد عناصر آخر عمود في المصفوفة A ، و ستظهر لذلك النتيجة التالية :

```
    13
     8
    12
     1
```

ـــ التعابير في MATLAB :

كأي لغة برمجية يحتوي MATLAB على التعابير ، و التعابير تقسم إلى نوعين :

- تعابير رياضية .
- تعابير منطقية .

و التعابير (بنوعيها الاثنين) تتكون من أربعة أقسام :

١) متحولات .
٢) أعداد .
٣) معاملات .
٤) توابع و دوال .

و سنقوم بتفصيل كل واحد من هذه الأقسام .

أولا : المتحولات :

لا نحتاج في MATLAB إلى تعريف المتحولات (المصفوفات) التي نستخدمها في برامجنا ، كما أننا لا نحتاج إلـــى تحديد أبعاد هذه المتحولات (المصفوفات) و لا نوعها .

عندما نولد متحولا جديدا في MTALAB ، فإن MATLAB يقوم أوتوماتيكيا بتوليد مصفوفة هذا المتغير و يحجز الذاكرة المناسبة له , و إذا كان المتغير موجود سابقا في الـــ (Work Space) فإن MATLAB يقوم بتغيير خصائص هذا المتغير بحيث يأخذ الخصائص الجديدة (أبعاد ، حجم الذاكرة ، نوع العناصر) ، على سبيل المثال ، عند كتابة السطر التالي :

student = 25

ينشأ لنا متغـــير اســـمه (student) ذو (1×1) و يخـــزن فيـــها العـــدد (25) و نـــوع العنـــاصر فـــي هـــذا المتغـــير (المصفوفة) هو (Int) .

الآن عند كتابة السطر التالي :

student = ‘ 25 ‘

PDF created with pdfFactory trial version www.pdffactory.com

يقوم MATLAB بتغيير خصائص المتغير (student) لتصبح أبعادها (2 × 1) و يخزن فيها سلسلة المحارف (25) و نوع العناصر في هذا المتغير هو (Char) .

و يجب أن تبدأ أسماء المتغيرات بحرف ، و قد يحتوي اســـم المتغـير أرقـام أو underscore (_) ، و يتعـرف MATLAB على أول ٣١ رمز (حرف ، رقم ، underscore) من اسم المتغير ، كمـا أن MATLAB يتحســس لحالـة الحرف (هل هو كبير أم صغير) ، فالمتغير (A) يختلف عن المتغير (a) ، و لرؤية قيم عناصر متغير ما فيكفي أن نكتـب اسم هذا المتغير في الـ (Command Window) .

ثانيا : الأعداد :

يتعامل MATLAB مع الأعداد العشرية الصحيحة مثل (10) و مع الأعداد ذات الفاصلة العشرية مثـل (10.15) كما يتعامل مع الأعداد ذات الصيغة العلمية { لها رياضيا الشكل (7×10^5) سواء كـان الأس موجـب أو سـالب } و ذلـك باستخدام العدد (e) { الذي يعبر عن الرفع إلى قوة أساسها العدد (10) } متبوعا بالأس (سالب أو موجب) ، أما الأعـداد العقدية فيتم التعبير عنها باستخدام الحرف (i) أو (j) ، و المثال التالي يوضح هذه الأنواع من الأعداد :

```
3             -99           0.0001
9.6397238     1.60210e-20   6.02252e23
1i            -3.14159j     3e5i
```

ثالثا : المعاملات :

تستخدم التعابير مجموعة من المعاملات الحسابية ، و يبين الجدول التالي هذه المعاملات :

+	Addition
-	Subtraction
*	Multiplication
/	Division
\	Left division
^	Power
'	Complex conjugate transpose
( )	Specify evaluation order

و تخضع هذه المعاملات إلى قوانين الأفضلية في التعامل و هي :

١ــ يتعامل MATLAB مع الأقواس الداخلية قبل الخارجية .

٢ــ يتعامل MATLAB مع الرفع إلى قوة قبل الضرب و القسمة .

٣ــ يتعامل MATLAB مع الضرب و القسمة قبل الجمع و الطرح .

رابعا: التوابع و الدوال :

يحتوي MATLAB على العديد من التوابع و الدوال تتدرج من بسيطة تقوم بحساب الجذر التربيعي لعدد مـــا إلـــى معقدة كتلك التي تحسب تكاملات (Beta) و (Gamma) .

إذا أردنا أن نظهر قائمة بأسماء التوابع و الدوال البسيطة نكتب الأمر :

```
help elfun
```

أما إذا أردنا أن نظهر قائمة بأسماء التوابع و الدوال المعقدة نكتب الأمر :

```
help specfun
```

أما إذا أردنا أن نظهر قائمة بأسماء التوابع و الدوال الخاصة بالمصفوفات نكتب الأمر :

```
help elmat
```

PDF created with pdfFactory trial version www.pdffactory.com

بعض التوابع و الدوال في MATALB تكون موجودة ضمن نواة برنامج الـــــ (MATLAB) مثــل (sqrt) و (sin) و تسمى مثل هذه التوابع و الدوال بـــ (built_in functions) .

أما التوابع الأخرى فإنها تكون مزروعة ضمن برنامج الـــ (**MATLAB**) و مكتوبة باستخدام ملفات (**M_Files**) مثل (**gamma**) و (**sinh**) و تسمى مثل هذه التوابع و الدوال بـــ (**implemented functions**) .

يوجد عدة فروق بين هذين النوعين من التوابع ففي النوع الأول (built_in functions) يكون الـــ (code) مخفيا و لا نستطيع التغيير فيه أو حتى رؤيته ، كما تتميز هذه التوابع بالتوافق التام مع MATLAB .

أما في النوع الثاني (**implemented functions**) فإن هذه التوابع مكتوبـــة باســـتخدام ملفـــات (**M_Files**) و يمكننا رؤية الـــ (**code**) و التغيير فيه أيضا .

يحتوي **MATLAB** على ثوابت مفيدة في حساباتنا مثل العدد (**j**) و الذي يساوي إلى الجذر التربيعي للعـــدد (**-1**) ، كمـــا يحتوي على الثابت (لا نهاية) { مثلا القسمة على صفر تعطي (لا نهاية) } و تسمى في **MATLAB** (**inf**) ، كما يحتوي على الثابت (قيمة غير معرفة) (مثل قسمة صفر على صفر أو لانهاية على لا نهاية) و تسمى في **MATLAB** (**NaN**) ، كما يحتوي على الثابت (قيمة متناهية في الصغر) وتسمى في **MATLAB** (**eps**) و غيرها ... و يوضح الجدول التالي الثوابت المعرفة في **MATLAB** :

`pi`	3.14159265...
`i`	Imaginary unit, $\sqrt{-1}$
`j`	Same as `i`
`eps`	Floating-point relative precision, $\varepsilon = 2^{-52}$
`realmin`	Smallest floating-point number, 2^{-1022}
`realmax`	Largest floating-point number, $(2-\varepsilon)2^{1023}$
`Inf`	Infinity
`NaN`	Not-a-number

يمكننا أن نغير قيمة أي ثابت من هذه الثوابت كما نريد ، فمثلا السطر التالي يعطي قيمة جديدة للثابت (**eps**) :

```
eps = 1.e-6
```

و عندما نكتب (eps) في الـــ (Command Window) ستظهر لنا القيمة الجديدة لهذا الثابت ، و لإعادة قيمة الثابت الأصلية فإن ذلك يتم باستخدام التعليمة (clear) ، مثلا السطر التالي يعيد القيمة الأصلية للثابت (eps) :

```
clear eps
```

PDF created with pdfFactory trial version www.pdffactory.com