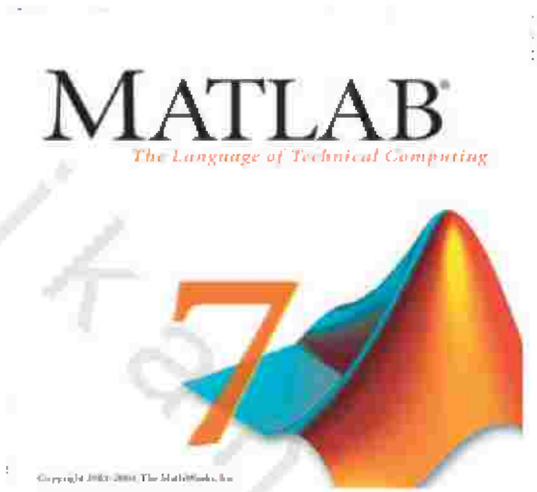
الفصل الأول

مبادئ برنامج MATLAB

(١,١) مقدمة في MATLAB

MATLAB هو برنامج عالي الأداء، صُمّم لإجراء الحسابات الرياضية المتقدمة، ويتميز بكونه برنامجاً متخصصاً ييسّر عمل الباحثين والدارسين في مختلف مجالات الدراسات العليا والدراسة الجامعية، ويضمّ المئات من الدوال الجاهزة التي توفر للمبرمج وقتاً وجهداً عند كتابة البرامج. أصول MATLAB ترجع إلى محاولة الرياضيين لتسهيل حساب المصفوفات، ومن هذا المنطلق فإن معنى كلمة MATLAB هو معمل المصفوفات "Matrix Laboratory" و قد بدأ تطويره من قبل شركة MathWorks في عام ١٩٨٤م ، مع أن تصميمه الأول كان على يد العالم Cleve Moler في ١٩٧٠م، الذي كان يشغل منصب رئيس قسم علوم الحاسب في جامعة نيو ميكسيكو بالولايات المتحدة الأمريكية. يوجد لبرنامج MATLAB عدة نسخ صدرت على مدى أعوام، مثل النسخة MATLAB 5 و 6 و 7. نستخدم هنا في عرضنا النسخة السابعة MATLAB 7 الشكل رقم (١,١). وهو يعمل على كل أنظمة Windows 95/98، Windows 2000/NT، Windows XP ، كما يوجد إصدار مماثل يخصّ Macintosh و Unix. ويمكن لـ MATLAB التعامل مع مستخدم واحد single user ،

أو أكثر من مستخدم للشبكات Workstation ، وهو عبارة عن حزمة من البرمجيات الجاهزة (Toolboxes) يحصل عليها المستخدم بحسب الحاجة.



الشكل رقم (١.١). الإصدار السابع من MATLAB.

إن برنامج MATLAB عمليٌ جداً، وخاصة في تمثيل البيانات بالرسم سواء منحنيات أو مسطحات، ويمكّن المستخدم من التحكم بالرسم بسهولة. و رغم أن MATLAB لغة متطورة في البرمجة إلا أنه من الممكن للمبرمج المبتدئ التعامل معه ببساطة. ويحتوي MATLAB على توابع و دوال جاهزة لتغطية أكثر المسائل شيوعاً. فمثلاً تتطلب كل من لغة Fortran ، C و Pascal عشرات الأسطر لكتابة برنامج لإيجاد تحويل فورير Fourier Transform بينما لدى MATLAB أمر ضمني واحد (fft) لإنجاز هذه العملية. وكذلك فإن عدم تعريف المتغيرات وحجمها مسبقاً من

المميزات لـ MATLAB على باقي لغات البرمجة ، ولكنه أبطأ لأنه يقوم بترجمة البرنامج المصدري جملة جملة interpative language بينما تقوم اللغات Pascal ، Fortran و C بترجمة البرنامج المصدري كاملاً وتحويله إلى لغة الآلة compiled language ، لذلك يتم تنفيذه بطريقة أسرع. فبرنامج MATLAB يمكن الاستفادة منه بطريقتين ، إما للحسابات المكثفة بالدوال الجاهزة ، أو بالبرمجة مع إدراج الدوال الجاهزة لتبسيط و تسريع أداء البرامج.

برنامج MATLAB يتعلق بالحسابات الرياضية ، والهندسية والمحاكاة. ويستخدم في الصناعات المختلفة ، كما يستخدم للأغراض الأكاديمية ، وخصوصاً أغراض البحث العلمي في الغالبية العظمى من جامعات العالم. والكثير من شركات الطيران المدني والعسكري ، تستخدم MATLAB في الحسابات الهندسية ، والنمذجة والمحاكاة ، مثل شركة إيرباص. كما يُعتمد عليه في تصميم الطائرات التي تطير بدون طيّار ، و في أبحاث الفضاء لشركة ناسا في مجال معالجة البيانات من قبل الباحثين والمختصين ليتمكن الباحث من إجراء مئات التجارب ، الأمر الذي يستحيل فعله بطريقة يدوية. وهو الآن شائع الاستخدام في التدريس خاصة في مواد الجبر الخطي ، والتحليل العددي وتطبيقاته المختلفة.

يهدف الكتاب إلى عرض بعض التطبيقات الرياضية على MATLAB لمستوى المقررات الجامعية الأولى ، و ذلك لمساعدة طالب العلوم الطبيعية أو الهندسة على ترسيخ المفاهيم الرياضية ، خاصةً التي تعتمد على التحليل العددي والحسابات العددية المكثفة. يحتاج القارئ إلى الإلمام بمبادئ البرمجة البسيطة والطرق الرياضية المختلفة ، وذلك لأن الكتاب يركز في عرضه على كيفية الاستخدام الجيد لبرنامج MATLAB من أجل كتابة برامج توضح هذه المفاهيم المختلفة. كما أن الكتاب يوضح طرقاً مختلفة

لتسخير MATLAB كآلة مساعدة لتوضيح المفهوم الرياضي، بنفس الطريقة التي تُستخدم فيها الآلة الحاسبة أو الأدوات الهندسية في إقناع الطالب بصحة القواعد والنتائج. وبعد الاطلاع على الكتاب وأداء التطبيقات، أدعو القارئ للاكتشاف بنفسه قوة MATLAB عن طريق التطبيقات المتخصصة في مجاله. من الصعب ذكر كل دوال MATLAB واستخداماتها الرياضية في كتاب واحد، وإنما نحتاج إلى عدة كتب لتغطية ذلك، و لو بحث القارئ في أيّ محرك بحث على الإنترنت عن مفهوم رياضي وتطبيقاته باستخدام MATLAB لحصل على عشرات المواقع التي تتحدث عن أمثلة جديدة وتطبيقات مبتكرة. يحتوي الكتاب الحالي على التطبيقات الرياضية الأساسية والتي تعطي الخطوط العريضة guidelines لقدرات MATLAB وتساعد القارئ على استخدامها وتسخيرها لتطبيقاته المختلفة.

(١,٢) الأوامر الرئيسة في MATLAB

يُعدّ MATLAB واحداً من أهم البرامج التي تقدم حلولاً متكاملة في مجال الرياضيات و الاختصاصات المعتمدة عليها والتي لا حصر لها، وهو يوفر للمستخدم:

١- التعامل بسهولة مع المصفوفات وحساباتها.

٢- عدداً كبيراً من الدوال الجاهزة والبرامج المخزنة، وهي في تطوير دائم مع كل نسخة جديدة.

٣- الرسم المتقن في بعدين و ثلاثة أبعاد.

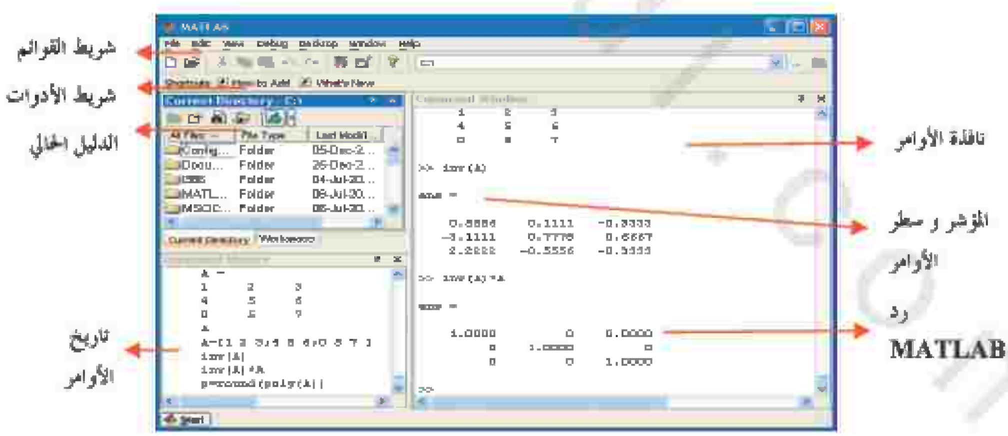
٤- القدرة على البرمجة للاستفادة من MATLAB في كافة المجالات العلمية.

عند فتح برنامج MATLAB تظهر عدة نوافذ مختلفة تظهر في الشكل رقم (١.٢) وهي :

١- **نافذة الأوامر الحالية Command window :** وهي النافذة الأساسية التي تتخاطب من خلالها مع برنامج MATLAB وهي تشمل مجال العمل Workspace أو الذاكرة المؤقتة ، حيث يتم فيها حفظ جميع المتغيرات التي استُعملت إلى حين إغلاق MATLAB ما لم يتم تنفيذ الأمر clear.

٢- **نافذة تاريخ الأوامر Command History :** و تعرض جميع الأوامر التي كتبت في مجال العمل Workspace الحالي.

٣- **الدليل الحالي Current Directory :** و يعرض موقع الملف الحالي لـ Workspace.



الشكل رقم (١.٢). النافذة الأساسية في MATLAB.

أهم الأوامر الخاصة بالنافذة الأساسية

• يُعدّ الأمر *help* المستخدم للمساعدة الطريقة الأساسية لمعرفة صيغ الدوال وتطبيقاتها، ويقوم بعرض المعلومات الضرورية مباشرة في نافذة الأوامر، ولاستخدامه اكتب الأمر *help* عند المؤشر >> ، وبعدها بمسافة اكتب اسم الدالة المطلوبة للاستعلام، ثم اضغط زر الإدخال Enter . وسوف يقوم MATLAB بالرد على هذا الاستعلام بشرح بسيط للدالة كما هو موضح في التالي.

أمر الاستعلام

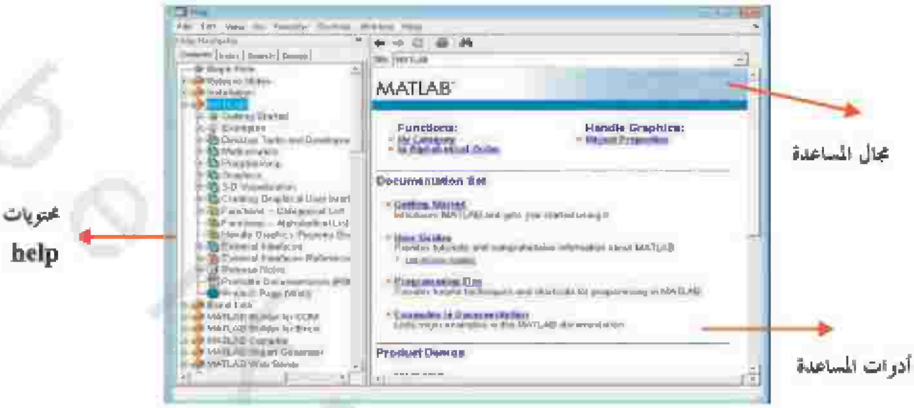
```
>> help cos
```

```
COS    Cosine.
COS(X) is the cosine of the elements of X.
See also acos, cosd.
```

رد MATLAB

كذلك توجد نافذة help المبينة في الشكل رقم (١.٣) والتي يتم فتحها من شريط المهام في MATLAB .

- الأمر *who* يعرض جميع متغيرات workspace في command window.
- الأمر *clear all* يقوم بمسح جميع محتويات workspace ويمكن مسح المتغير x فقط بالأمر *clear x* أو مسح المتغيرات التي تبدأ بحرف a فقط *clear a**.



الشكل رقم (١,٣). نافذة المساعدة help.

- للقيام بحفظ كل محتويات workspace في ملف اسمه mywork ندخل الأمر *save mywork* أما إذا أردنا حفظ متغير مثل x نكتب *save x*.
- ولاسترجاع الملف mywork ندخل الأمر *load mywork* ، ولاسترجاع متغير x نكتب *load x* .
- العلامة ; بعد أي أمر تمنع طباعة النتائج المحسوبة.
- العلامة % تسبق كل جملة تعليق comment line أي أن MATLAB يتجاهل كل ما كتب من أوامر في هذه الجملة.
- العلامة ... في نهاية السطر تعني تابع الجملة في السطر الجديد.

مثال رقم (١,١)

ندخل بعض من هذه الأوامر :

```
>> % some examples
>> x=4
x =
    4
>> who
Your variables are:
x
>> save x
>> load x
>> clear x
```

تم تخزين المتغير x

تم التحميل

مسح المتغير x

(١,٣) الحسابات البسيطة في MATLAB

عند تشغيل برنامج MATLAB تظهر العلامة << وتسمى Command Line ويمكن إدخال أمر بعدها ، وبعد ضغط ENTER يتم تنفيذ الأمر ، و يظهر رد MATLAB بعد عبارة ans . فمثلاً لحساب العبارة الرياضية البسيطة $(2+3.5^2-4(9))/12$.

```
>> (2+3.5^2-4(9))/12
ans =
    -1.8125
```

أدخل الأمر

رد MATLAB

ويمكن تعريف متغيرات لحساب العبارة:

```
>> x=2+3.5^2
x =
   14.2500
>> y=4*9
y =
   36
>> (x-y)/12
ans =
   -1.8125
```

يتعامل MATLAB مع جميع المتغيرات كمصفوفات، ويجب أن يكون اسم المتغير متكوناً من كلمة واحدة لا يفصل بينها مسافة، فمثلاً : myvariable وليس my variable، وألا يتعدى ٣١ رمزاً و يبدأ بحرف. يميز MATLAB بين الحروف الصغيرة small case والكبيرة capitals أي أن المتغير A يختلف عن المتغير a. ويجب مراعاة عدم استخدام المتغيرات الخاصة ببرنامج MATLAB مثل :

ans : متغير يرمز للنتائج

pi : القيمة ط

NaN : رقم غير مقبول

inf : مالا نهاية

تتم العمليات الحسابية باستخدام الإشارات التالية :

عملية الجمع	+
عملية الطرح	-
عملية الضرب	*
عملية الأس	^
عملية القسمة	/

يقوم MATLAB بالحسابات مستخدماً ١٥ خانة رقمية ، أي الدقة المضاعفة Double Precision وعادة يتم عرض خمس خانات فقط ، ويمكن عرض كل الخانات إذا تم إدخال الأمر :

```
>> format long
>> ans
ans =
-1.81250000000000
```

للعودة للعرض القصير ندخل *format short* وللعرض بصيغة scientific notation :

```
>> format short e
>> ans
ans =
-1.8125e+000
```

إذا أدخلت عبارة خاطئة فإن MATLAB يرد بعبارة بلون أحمر توضح الخطأ، فمثلاً :

```
>> (x-y/12
??? (x-y/12
Error: Incomplete or misformed expression or statement.
```

من أهم مزايا MATLAB هو أن الأعداد المركبة يتم إدخالها بسهولة كما لو كانت قيماً حقيقية، ويمكن استخدام الحرف i او j للدلالة على العدد التخيلي. وتتم العمليات الحسابية بالأعداد المركبة بسهولة.

```
>> c=1+2*i
 c =
   1.0000 + 2.0000i
 >> c=1+2*j
 c =
   1.0000 + 2.0000i
>> (1+2i)*(1-2i)
ans =
     5
```

نستطيع استخدام دالة *real* ، ودالة *imag* للتعرف على أجزاء العدد المركب الحقيقي و التخيلي :

```
>> imag(c)
ans =
     2
>> real(c)
ans =
1
```

(١,٤) المتجهات والمصفوفات

المتجهات والمصفوفات هي أساس العمل في بيئة MATLAB . يتم تعريف المتجهات عمودية أو صفية و هي مجموعة أرقام تفصلها فاصلة أو فراغ بين الأقواس المربعة [ ] وقد تمثل السرعة أو القوة أو أي قيمة فيزيائية أخرى. ويتم إدخال المتجه الصفي كالآتي :

```
>> u=[2 3.6 0.5 sqrt(3)]
u =
   2.0000    3.6000    0.5000    1.7321
```

أما المتجه العمودي فنفصل بين عناصره بالفاصلة المنقوطة :

```
>> v=[1;3;7.8;pi]
v =
1.0000
3.0000
7.8000
3.1416
```

المصفوفة matrix هي عبارة عن مجموعة من الأعداد الحقيقية (أو المركبة) عناصرها مرتبة في جدول مستطيل، يسمى كل سطر أفقي من عناصر المصفوفة صفاً (row) ويسمى كل سطر رأسي عموداً (column) فمثلاً المصفوفة A ذات الأبعاد 3×4 تكتب :

```
>> A=[1 2 3 4;6.2 4 5 -3;1/2 1/3 4 -4]
A =
   1.0000    2.0000    3.0000    4.0000
```

```
  6.2000    4.0000    5.0000   -3.0000
  0.5000    0.3333    4.0000   -4.0000
```

وتقوم الفاصلة المنقوطة بمنع ظهور ناتج الأمر في command window بعد تنفيذه، وهذا يفيد في حال المصفوفات الكبيرة.

(١,٤,١) رتبة المصفوفة

لإيجاد رتبة المصفوفة matrix rank نستخدم الأمر *rank* :

```
>> rank(A)
ans =
     3
```

(١,٤,٢) بُعد المصفوفة

أما لمعرفة أبعاد المصفوفة matrix size فندخل *size* :

```
>> size(A)
ans =
     3     4
```

(١,٤,٣) طرق التعامل مع المصفوفة

لرؤية عنصر معين في المصفوفة فعلينا تحديد موقعه بالمصفوفة، فمثلاً إذا أردنا العنصر الواقع في الصف الأول و العمود الثاني :

```
>> A(1,2)
ans =
     2
```

ولاستخراج مصفوفة جزئية من A :

```
>> A(1:2,2:3)
ans =
     2     3
     4     5
```

ولإيجاد الصف الثاني من المصفوفة نكتب :

```
>> A(2,:)
ans =
     6.2000    4.0000    5.0000   -3.0000
```

ولإيجاد العمود الثاني من المصفوفة ندخل :

```
>> A(:,2)
ans =
    2.0000
   4.0000
   0.3333
```

كما يمكن تبديل الصفوف في المصفوفة ، فمثلاً لو أردنا تبديل الصف الأول والثالث نكتب :

```
>> A([3,2,1],:)
ans =
    0.5000    0.3333    4.0000   -4.0000
   6.2000    4.0000    5.0000   -3.0000
  1.0000    2.0000    3.0000    4.0000
```

وبنفس الطريقة يتم تبديل العمودين الثاني والثالث، ولكن يتغير وضع الفاصلة :

```
>>A(:,[1,3,2,4])
ans =
    1.0000    3.0000    2.0000    4.0000
    6.2000    5.0000    4.0000   -3.0000
   0.5000    4.0000    0.3333   -4.0000
```

يمكن تبديل الصف أو العمود بأي قيمة جديدة ، فمثلا لتغيير قيمة الصف الثاني إلى [2 3 5 6] نقوم بإدخال :

```
>> A(2,:)=[2 3 5 6]
 A =
   1.0000    2.0000    3.0000    4.0000
   2.0000    3.0000    5.0000    6.0000
  0.5000    0.3333    4.0000   -4.0000
```

أما إذا احتجنا إلى تغيير شكل المصفوفة ندخل أمر *reshape* الذي يضم اسم المصفوفة و الشكل الجديد، مثل :

```
>> reshape(A,1,12)
ans =
   Columns 1 through 9
  1.00   2.00   0.500   2.00   3.00   0.333   3.00   5.00  4.00
  Columns 10 through 12
  4.00   6.00  -4.00
```

(١,٤,٤) إنشاء متجهات في MATLAB

وإذا أردنا إدخال متجه أو مصفوفة كبيرة، فإن كتابتها عنصراً بعنصرٍ ليس أمراً عملياً لذلك إن وجدت قاعدة معينة أو نمط متكرر للعناصر نستطيع استخدامها مع إشارة العمود (:) فمثلاً التعبير (1:10) = x يمثل متجهاً صفياً يحتوي على الأرقام الصحيحة من ١ إلى ١٠. ومن أجل الحصول على متتالية حسابية أساسها غير الواحد، يمكننا أن نستخدم إشارة العمود بإدخال القيمة الابتدائية للمتجه، ومقدار الإضافة في كل خطوة، ثم القيمة النهائية. على سبيل المثال نريد كتابة متجه x يحتوي على العناصر (1,1.1,1.2,1.3,...,5.9,6) ندخل :

```
>> x=1:0.1:6
```

وسينتج متجه سطري يحتوي على 51 عنصراً. نستطيع أيضاً استخدام الأمر *linspace* لنفس الغرض، فنحدد البداية، والنهاية وعدد العناصر :

```
>> x=linspace(1,6,51)
```

وهذا يساعد في تقسيم الفترات إلى فترات متساوية، فعلى سبيل المثال الفترة $[0,2\pi]$ تقسم إلى ٢٠ فترة باستخدام الأمر *linspace(0,2*pi,20)* . ولإنشاء متجه صفي مكون من ١٠ عناصر كلها لها القيمة واحد، نستخدم الأمر *ones* مع تحديد حجم المتجه :

```
>> x=ones(1,10)
 x =
 1   1   1   1   1   1   1   1   1   1
```

أو لو احتجنا متجهاً صفرياً فنستخدم الأمر zeros :

```
>> y=zeros(1,10)
y =
   0   0   0   0   0   0   0   0   0   0
```

عند إنشاء المصفوفة الصفرية z بالحجم $n \times m$ نحدد ذلك في الأمر z =zeros(n,m). ولإنشاء مصفوفة w بالحجم $n \times m$ ، ولكن عناصرها اختيرت عشوائياً ، فنستخدم الأمر w=rand(n,m) .

كما يمكن إنشاء مصفوفات أكثر تعقيداً باستخدام مصفوفات أخرى ، فمثلاً العبارتان التاليتان :

```
 >> C=[6.5 4.7 ;.5 4.1];
>> D=[C zeros(size(C)); zeros(size(C)) ones(size(C))]
D =
   6.5000   4.7000        0        0
   0.5000   4.1000        0        0
        0        0   1.0000   1.0000
        0        0   1.0000   1.0000
```

تعطينا الناتج : المصفوفة الجديدة D بضعف حجم المصفوفة C .

وإذا أردنا إنشاء مصفوفة قطرية diagonal matrix بقيم معينة نستخدم diag :

```
>> D=diag([1 2 3])
D =
   1   0   0
   0   2   0
   0   0   3
```

الأمر sum يقوم بحساب مجموع عناصر المتجه ، فإذا رجعنا للمصفوفة D فإن D(1:k,j) هي العناصر الـ k الأولى في العمود j ، وإذا أردنا حساب مجموع عناصر

العمود الرابع في D نكتب sum(D(1:4,4)) . كما توجد طريقة أخرى للحصول على نفس النتيجة وذلك باستخدام sum(D(:,end)) حيث إن end تدل على العمود الأخير في المصفوفة.

(١,٥) جبر المصفوفات Matrix Algebra

سنتناول بعض الأمور الجبرية المتعلقة بالمصفوفات، ونبين الأوامر التي تنفذها على MATLAB ، وفق التعاريف المعتمدة في علم الجبر الخطي.

(١,٥,١) جمع وطرح المصفوفات

إذا كانت المصفوفتان A و B من النوع n x m فإن مجموعهما الذي يمثل بـ A+B هو مصفوفة من النوع n x m ، عناصرها هي $a_{ij}+b_{ij}$ لكل i=1,2,...n و j=1,2,...m

مثلاً:

```
>> B=[1 3 4;1 2 5;0 1 4]
B =
   1    3    4
   1    2    5
   0    1    4
>> C=[0 3 4;1 2 3;4 5 6]
C =
   0    3    4
   1    2    3
   4    5    6
>> B+C
ans =
   1    6    8
   2    4    8
   4    6   10
```

والفرق بين المصفوفتين:

```
>> B-C
ans =
     1    0    0
```

```
  0   0   2
 -4  -4  -2
```

(١,٥,٢) ضرب المصفوفات

نفرض A مصفوفة من النوع n x m و B مصفوفة من النوع m x p ، تمثل مصفوفة جداء A في B بالرمز A*B ، وهي مصفوفة من النوع n x p وتأخذ عناصرها الشكل

$$c_{ij} = \sum_{k=1}^{m} a_{ik} b_{kj} \quad \text{لكل} \quad j=1,2,...p \quad \text{و} \quad i=1,2,...n$$

أي يمكن اعتبارها مجموع جداءات عناصر الصف i من A بالعناصر المقابلة من العمود j من B، لذلك يجب تساوي عدد صفوف B مع عدد أعمدة A . وفي حال عدم توفر ذلك فإن البرنامج يقوم بتنبيه المستخدم .

```
>> A
A =
   1   2   3
   4   5   6
   0   5   7

>> B
B =
   1   6   1   0
   2   8   9   1
   3   7   9   2

>> A*B
 ans =
14    43    46    8
32   106   103   17
31    89   108   19

>> B*A
??? Error using ==> mtimes
Inner matrix dimensions must agree.
```

ويمكن حساب أي عبارة رياضية تحتوي على مصفوفات وأعداد حقيقية، مثلاً :

```
>> 2*A-1
ans =
     1     3     5
     7     9    11
    -1     9    13
```

```
>>5*(A*A)^3
ans =
     1101285    3119175    4131720
     2591400    7339635    9722220
     2431300    6886200    9121595
```

ولكن لإجراء عمليات جبرية على مستوى العناصر نضع نقطة قبل العملية مثل

*. أو ./ أو ^.

```
>>5>>5*(A.*A).^3
ans=
       5        320        3645
   20480      78125      233280
       0      78125      588245
```

وتختلف النتيجة ما بين العملية * ونفس العملية مرفوقة بالنقطة *. لأن العملية الأولى هي عملية ضرب مصفوفتين بينما العملية الثانية هي عملية ضرب العنصر بالعنصر، كما تبرهن الخطوات التالية :

```
>> a=[1 2;3 4]
a =
   1    2
   3    4
>> b=[5 6;7 9]
 b =
      5    6
      7    9
 >> a*b
 ans =
      19    24
```

```
     43    54
>> a.*b
ans =
      5    12
     21    36


>> a.^b
ans =
      1         64
      2187      262144
```

في العملية الأخيرة كل عنصر من *a* رفع للقوة المقابلة في b.

(١,٥,٣) مصفوفة الوحدة و معكوس المصفوفة

المصفوفة المربعة ببعد $n \times n$ التي تحتوي على أصفار ما عدا على القطر تأخذ القيم ١ ، تُدعى مصفوفة الوحدة identity matrix وتنتج بالأمر *eye(n)* ، فمصفوفة الوحدة من الرتبة الثالثة هي :

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

وإذا كان لدينا مصفوفة مربعة A فيمكن حساب محدد المصفوفة matrix determinant بإدخال الأمر *det(A)* :

```
>> A
A =
   1    2    3
   4    5    6
   0    5    7
>> det(A)
 ans =
       9
```

عندما يكون المحدد للمصفوفة المربعة غير صفري ، فيمكن حساب معكوس المصفوفة matrix inverse بإدخال A^{-1} أو inv(A) :

```
>> inv(A)
ans =
0.5556   0.1111   -0.3333
-3.1111   0.7778   0.6667
2.2222   -0.5556   -0.3333
```

بحيث يكون $I = A\,A^{-1} = A^{-1}A$ و I مصفوفة الوحدة بحجم A ، وبذلك تكون المصفوفة غير شاذة nonsingular . ويمكن التأكد بحساب $A\,A^{-1}$:

```
>> inv(A)*A
ans =
   1.0000        0          0.0000
      0       1.0000        0
     0        0            1.0000
```

إذا كانت المصفوفة B غير مربعة m × n بحيث m > n فيمكن لـ MATLAB حساب شبه المعكوس pseudo- inverse بدالة *pinv* ، ويمكن الاطلاع على التعريف الرياضي لشبه المعكوس بالأمر *help pinv*.

```
>> B=[1 2 3;4 5 9;7 11 18;-2 3 1;7 1 8];
>> pinv(B)
ans =
  -0.0080   0.0031   -0.0210   -0.0663    0.0966
   0.0117   0.0092    0.0442    0.0639   -0.0805
   0.0036   0.0124    0.0232   -0.0023    0.0162
>>help pinv
MATLAB Function Reference   pinv
Moore-Penrose pseudo inverse of a matrix
B = pinv(A)
B = pinv(A,tol)

Definition
The Moore-Penrose pseudo inverse is a matrix B of the same dimensions as
```

A' satisfying four conditions:
*A*B*A = A*
*B*A*B = B*
*A*B is Hermitian*
*B*A is Hermitian*
If A is square and not singular, then pinv(A) is an expensive way to compute inv(A). If A is not square, or is square and singular, then inv(A) does not exist. In these cases, pinv(A) has some of, but not all, the properties of inv(A).

(١,٥,٤) القيم الذاتية للمصفوفة

الأمر poly يعطي متجهاً يحتوي على معاملات المعادلة المميزة characteristic equation للمصفوفة A، حيث إن المعادلة المميزة للمصفوفة هي det(λ I-A)=0. ويمكن حساب القيم الذاتية Eigenvalues للمصفوفة A عن طريق تطبيق الأمر roots على المعادلة المميزة للمصفوفة. كما يوجد في MATLAB دالة جاهزة لحساب القيم الذاتية للمصفوفة تُدعى eig .

نطبق على مثالنا السابق ونلاحظ تطابق الطريقتين، حيث إن جذور المعادلة المميزة لـ A ($\lambda^3-13\lambda^2+9\lambda-9=0$) تعطي القيم الذاتية للمصفوفة .

```
>> p=round(poly(A))
p =
   1  -13    9   -9


>> roots(p)
ans =
         12.3292
          0.3354 + 0.7858i
          0.3354 - 0.7858i
>> eig(A)
ans =
          0.3354 + 0.7858i
          0.3354 - 0.7858i
         12.3292
```

(١,٥,٥) منقول المصفوفة

منقول المصفوفة matrix transpose $A = (a_{ij})$ من النوع n×m هو المصفوفة $A^t = (a_{ji})$ من النوع m×n بحيث تصبح الصفوف في A أعمدة في A^t والعكس.
ولحساب المنقول نستخدم علامة الاقتباس المفردة (') وتقوم هذه العملية بقلب عناصر المصفوفة وفق قطرها الرئيس على النحو التالي :

```
>> A'
ans =
     1    4    0
     2    5    5
     3    6    7
```

```
>> B=[ 1 2 ;3 4;5 6]'
ans =
     1    3    5
     2    4    6
```

حتى في حال مصفوفة غير مربعة

تسمى المصفوفة المربعة متناظرة symmetric matrix إذا كان $G = G^t$ مثل :

```
>> G
G =
   6    4   -3
   4   -2    0
  -3    0    1
>> G'
ans =
     6    4   -3
     4   -2    0
    -3    0    1
```

إذا كانت المصفوفة تحوي أعداداً مركبة فإن (') تعطي المنقول المرافق المركب complex conjugate transpose :

```
>> a=[1+2*i 3+5*i;4+2*i 3+4*i]
 a =
1.0000 + 2.0000i   3.0000 + 5.0000i
4.0000 + 2.0000i   3.0000 + 4.0000i
```

```
>> a'
1.0000 - 2.0000i   4.0000 - 2.0000i
3.0000 - 5.0000i   3.0000 - 4.0000i
```

لحساب المنقول المركب من غير المرافق complex transpose نسبق العلامة بنقطة أي ('.) :

```
>> a.'
   1.0000 + 2.0000i   4.0000 + 2.0000i
   3.0000 + 5.0000i   3.0000 + 4.0000i
```

(١,٥,٦) نظيم المتجه و نظيم المصفوفة

يحسب MATLAB النظيم *norm* للمتجهات أو للمصفوفات على حسب نوع النظيم بالأوامر المذكورة في *help* :

```
For matrices...
    NORM(X,2) is the same as NORM(X).
    NORM(X,1) is the 1-norm of X, the largest column sum,
              = max(sum(abs(X))).
    NORM(X,inf) is the infinity norm of X, the largest row sum,
              = max(sum(abs(X'))).
    NORM(X,'fro') is the Frobenius norm, sqrt(sum(diag(X'*X))).
For vectors...
    NORM(V,P) = sum(abs(V).^P)^(1/P).
    NORM(V) = norm(V,2).
    NORM(V,inf) = max(abs(V)).
```

فمثلاً النظيم ℓ_∞ norm المُعرف $\|x\|_\infty = \max_i |x_i|$ للمتجه $x =$[-1 1 -3] يحسب بالأمر التالي :

```
norm(x,inf)
ans =
    3
```

(١,٥,٧) العدد الشرطي للمصفوفة

يمكن إيجاد العدد الشرطي لمصفوفة باستخدام دالة *cond* وذلك حسب التعريف $cond(A)=\|A\|\|A^{-1}\|$ ، فمثلاً إذا تم اختيار النظيم الإقليدي ℓ_2 norm (p=2) على المتجه $x=(x_1,x_2,\dots x_n)^t$ فإن نظيم المصفوفة :

$$\|A\|_2=\max\|Ax\|_2 \quad s.t. \quad \|x\|_2=\left\{\sum_{i=1}^{n}x_i^2\right\}^{1/2}$$

يمكن تحديده بالأمر *cond* (A, 2) . وتُعد المصفوفة حسنة الشروط well-conditioned كلما اقترب العدد الشرطي من الواحد، وتدعى سيئة الشروط ill-conditioned عندما يكون العدد الشرطي أكبر من الواحد بصورة ملحوظة.

```
>> A=[3 2 -1;-1 3 2;1 -1 -1 ] ;
>> cond(A)
  63.0547
```

(١,٦) الدوال المخزّنة على MATLAB

توجد مجموعةٌ كبيرةٌ من الدوال مخزنة ذاتياً في MATLAB، ويمكن استحضار الدوال بطباعة اسم الدالة والعوامل المرتبطة بالدالة. وقد تعطي هذه الدوال ناتجاً أو أكثر على حسب نوعها. الجدول رقم (١.١) يعرض بعض هذه الدوال ، حيث يبين اسم الدالة ووظيفتها، و مثالاً على طريقة استعمالها. ونلاحظ استخدام الحروف الصغيرة في جميع أسماء الدوال. بعض الدوال تحتاج إلى أكثر من عامل إدخال لتحصل على النتائج المرجوة، مثل *bessel(n,x)* وهي دالة البسل بدرجة n للمتغير x. وفي المقابل هناك الدالة *[y,I]=sort(x)* وهي مثال على دالة الترتيب التي تعطي قيمتي إخراج، y هي المصفوفة المرتبة وI هي المصفوفة التي تحتوي على معاملات العناصر لهذا الترتيب.

الجدول رقم (١,١) . دوال جاهزة في MATLAB.

الدالة	وظيفتها	مثال
exp(x)	e^x	3*exp(4)
floor(x)	تقريب باتجاه ∞ -	floor(3.6)
ceil(x)	تقريب باتجاه ∞ +	ceil(4.9)
gcd(x)	القاسم المشترك الأكبر	gcd(6,8)
lcm(x)	المضاعف المشترك الأصغر	lcm(12,10)
sqrt(x)	الجذر التربيعي	sqrt(5,8+2)
max(v)	القيمة العظمى	max([2,4,7])
min(v)	القيمة الصغرى	min([2,4,7])
imag(z)	الجزء التخيلي من العدد المركب	imag(4+6*i)
real(z)	الجزء الحقيقي من العدد المركب	real(4+6*i)
fix(x)	تقريب باتجاه الصفر	1-fix(5.9)
round(x)	تقريب باتجاه أقرب رقم صحيح	3*round(6.8)
sum(v)	حاصل جمع العناصر	sum([2,3,6)]
abs(x)	القيمة المطلقة	3*abs(-4.5)

نعرض بعض الدوال المثلثية ومعكوس كلٍ منها في الجدول رقم (١,٢).

الجدول رقم (١,٢) . الدوال المثلثية في MATLAB.

sin(x)	دالة الجيب المثلثية
cos(x)	دالة الجيب التمام المثلثية
tan(x)	دالة الظل المثلثية
sec(x)	دالة القاطع المثلثي
csc(x)	دالة القاطع التمام المثلثي

تابع الجدول رقم (١,٢) .

cot(x)	دالة الظل التمام المثلثي
acos(x)	معكوس cos
asin(x)	معكوس sin
acot(x)	معكوس cot
atan(x)	معكوس tan

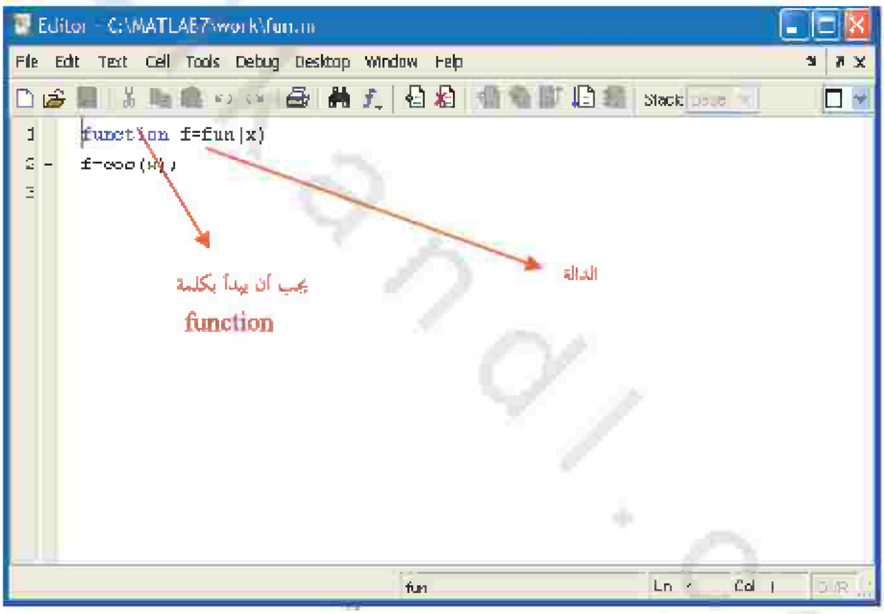
(١,٧) تعريف دوال في MATLAB

(١,٧,١) تعريف الدالة في ملف m-file

يستخدم MATLAB نوعين من الملفات التي تدعى m-files. النوع الأول هو ملف لكتابة البرامج ويخزن باسم معين للبرنامج ويمكن استحضار البرنامج وتشغيله بطباعة اسم الملف في نافذة الأوامر أو في أي برنامج آخر. النوع الثاني هو ملف لتعريف الدوال function m-file ويخزّن باسم الدالة المراد تعريفها ، ويتم استخدام الدالة عند طباعة اسم الدالة في نافذة الأوامر مثل أي دالة جاهزة في MATLAB. ويمكن للمستخدم تعريف دوال خاصة وفق صيغة معينة ، وبذلك يتمكّن من القيام بتطبيقات عديدة. الصيغة العامة هي :

```
function output=function_name(input)
function body
```

وسوف يتم حفظ الملف بنوع fun. m ويجب تعريف الدالة في أول سطر بكلمة *function* ثم يلي ذلك أسطر تحتوي على حسابات الدالة. مع مراعاة احتواء الدالة على عامل الإدخال input عند طباعة اسمها، وستظهر النتيجة في عامل الإخراج output الشكل رقم (١,٤).



الشكل رقم (١,٤). نافذة تعريف الدوال Function m-file.

مثال رقم (١,٢)

عَرِّف الدالة البسيطة $f(x)=\left(\frac{x}{2}\right)^3-x+\sin\left(\frac{\pi x}{2}\right)$ في ملف يحمل اسم *fun*

واحسب قيمتها عند (2.5).

الحل :

نعرف الملف كالتالي :

```
function p=fun(x)
% A simple function definition
x=x/2;
p=x^3-2*x+sin(pi*x);
```

ويمكن حساب قيمة الدالة عند (2.5) بطباعة *y = fun(2.5)* . أو باستخدام الأمر . feval('fun',2.5)

مثال رقم (١،٣)

عرف الدالة في متغيرين $f(x,y) = x + ye^{x^2+y^2}$ واحسب قيمتها عند (1.2).

الحل :

نستطيع تعريفها في ملف fun2 كالآتي :

```
function z=fun2(x,y)
z=x+y.*exp(x.^2+y.^2);
```

ومن ثم تُستخدم الدالة لإيجاد قيمتها عند النقطة المطلوبة ، بالأمر fun2(1.2) لتعطي القيمة 297.8263 .

كما يمكن تطبيق الدوال الجاهزة في MATLAB على الدوال المعرفة من قبل المستخدم ، مثل دالة *fzero* التي تستخدم لإيجاد جذور دالة بطرق عددية. فلإيجاد جذر *fun* في الفترة ما بين 2 و 4 نكتب *fzero('fun',[2,4])* (ملحوظة : كتابة اسم الدالة تتم

داخل علامات اقتباس) وسيُظهر MATLAB الحل في شاشة الأوامر 3.1264 .

(١,٧,٢) أوامر تعريف الدوال

طريقة أخرى لتعريف دالة في سطر واحد هي استخدام الأمر *inline*. ويُحدد اسم الدالة بين علامات الاقتباس والمتغيرات مثل :

```
>> g=inline('exp(-x.^2)','x')
g =
   Inline function:
   g(x) = exp(-x.^2)
```

ثم يمكن حساب قيم الدالة بسهولة عند أي ثابت بالأمر *feval* :

```
>> feval(g,0)
ans =
    1.0000
```

أو مباشرة عند أي متجه :

```
>> x=0:3
x =
    0    1    2    3
>> g(x)
ans =
   1.0000    0.3679    0.0183    0.0001
```

كما يمكن استخدام إشارة @ لتعريف الدالة مثل :

```
>> fh = @(x,y)y*sin(x)+x*cos(y)
fh =
   @(x,y)y*sin(x)+x*cos(y)
>> fh(pi,2*pi)
ans =
   3.1416
```

حساب الدالة عند (π,2 π)

(١,٨) الإدخال والإخراج في MATLAB

(١,٨,١) أوامر الإخراج

أبسط طريقة في الإخراج أو عرض البيانات على MATLAB هي عدم كتابة الفاصلة المنقوطة في نهاية عبارة التعريف، ولكن هذه الطريقة لا تظهر النتائج بطريقة مرتبة، فالأفضل استخدام أوامر خاصة بالإخراج، مثل: الأمر disp الذي يعرض النتائج و العبارات النصية.

لعرض محتويات مصفوفة نستخدم disp(A) و لعرض عبارة نص يجب كتابتها ضمن إشارة الاقتباس ' ' :

```
>> disp('this is text')
this is text
```

ويمكن عرض عبارة تحتوي على نص و قيم رقمية وذلك بفصلها عن بعضها بالفاصلة، وتكون داخل أقواس مربعة ويجب استخدام دالة *num2str* لتحويل الأرقام إلى نص:

```
>> x
x =
  10.8000
>> t
t =
    3
>> disp(['this is the value   ',num2str(x),'   at the time    ',num2str(t)])
this is the value   10.8    at the time    3
```

أما للعرض بشكل منسق، فهناك الأمر الأكثر مرونة وهو *fprintf* ، ويمكن العرض على الشاشة أو على ملف. ويأخذ الصيغة التالية بعد تحديد المتغيرات وأنواعها وعدد خانات العرض مسبوقة بعلامة ٪ :

```
>> fprintf('filename','format string',list);
```

```
>> fprintf('x=%8.2f t=%8.6f',x,t)
x=   10.80 t=3.000000
```

(١,٨,٢) أوامر الإدخال

أما بالنسبة لإدخال نص أو بيانات من لوحة المفاتيح ، فنستعمل دالة *input*. وهي تأخذ الصيغة الأولى في حال إدخال قيمة رقمية ، و تأخذ الشكل الثاني إذا أردنا إدخال نص. ستظهر على الشاشة كلمة *text* وينتظر البرنامج من المستخدم إدخال القيمة المناسبة باستخدام لوحة المفاتيح :

```
>> x=input('text');
>> x=input('text','s');
```

لإدخال بيانات كثيرة يفضّل استخدام الأمر *load* الذي يمكّن المستخدم من إدخال بيانات تم حفظها في ملف ، ويكتب اسمه بعد الأمر ، أي *load filename* . عند وجود البيانات في ملف يمكن قراءة البيانات بعد فتح الملف بدالة *fopen* عن طريق أحد الأوامر *fscanf* ، *fget1* أو *fread* وكل من هذه الدوال لها صيغة معينة ، وتعتمد على نوع الملف ونوع البيانات (*binary* ، *text* ، ...). ويجب إقفال الملف عند الانتهاء من قراءة البيانات بالأمر *fclose* . لمزيد من التفاصيل عن دوال الإدخال نقترح اللجوء إلى *help* .

(١,٩) الرسم على MATLAB

من أهم مميزات MATLAB القدرة على رسم المنحنيات ثنائية و ثلاثية الأبعاد 3D graphics والتي يصعب رسمها ببقية لغات البرمجة. ويقدم MATLAB

وسائل مساعدة للتحكم بالرسوم وتعديلها، سواء من ناحية تحديد المحاور أو تغيير نمط ولون خط الرسم أو تحريك الشكل. يوفر MATLAB دوال عديدة للرسم ويمكن استخدامها مباشرة على الرسم أو كتابتها على شكل أوامر في نافذة الأوامر.

(١,٩,١) الرسم في بعدين

الأمر الأساسي في الرسم في بعدين هو *plot(x,y)* ويقوم هذا الامر بفتح شاشة خاصة بالرسم تسمى figure file و يتم رسم المتجه x في محور السينات والمتجه y في محورالصادات. كما توجد دوال أخرى مثل:

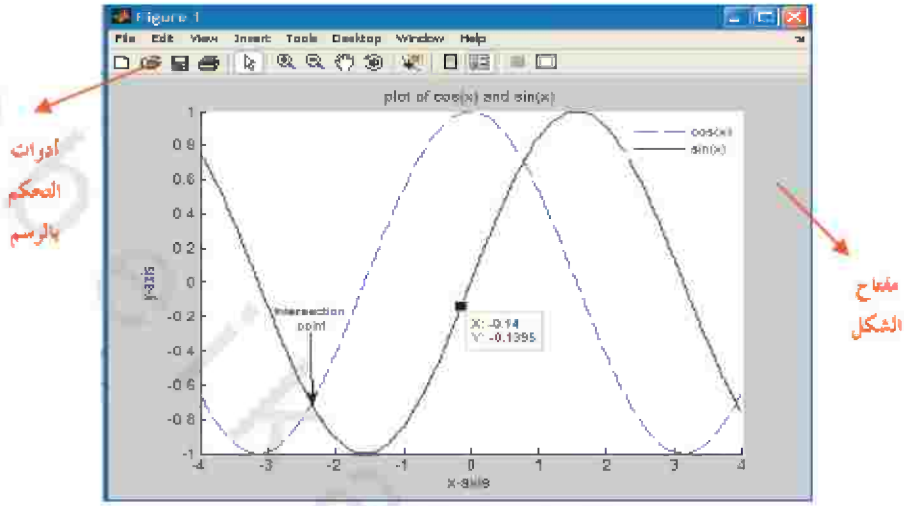
- *hold*: تقدم هذه الدالة إمكانية رسم أكثر من منحنى على نفس الرسم، حيث يتم تفعيلها بـ *hold on* و هذا بعد إنشاء الرسم الأول، ثم يتم رسم المنحنى التالي، ويمكن إيقافها بـ *hold off* .

- *axis*: تقوم بتحديد المحاور مع العلم بأن MATLAB يقوم بذلك بشكل ذاتي ولكن هذه الدالة تستخدم لإظهار أفضل شكل من تكبير أو تصغير أو حتى إظهار جزء من الرسم.
- *title* : تقوم بإدراج عنوان في أعلى الرسم.
- *xlabel* : تستخدم لتحديد تسمية المحور الأفقي للرسم.
- *ylabel* : تستخدم لتحديد تسمية المحور العمودي للرسم.
- *figure* : تستخدم لفتح نافذة للرسم جديدة للعرض.
- *fplot* : تستخدم لرسم دالة معرّفة مسبقا بقيم مختلفة.
- *Plot(x1,y1,x2,y2,x3,y3,...)* : تستخدم لرسم عدة منحنيات على نفس المحاور، باعتبار الأزواج المرتبة ...(x1,y1), (x2,y2), (x3,y3)

وهناك وسائل تحكم عديدة على شاشة الرسم، مثل تغيير طريقة الرسم من خط مستقيم إلى خط مقطّع، أوخط يحتوي على رمز توضيح، ويمكن تغيير سُمك الخط ولونه من الجدول رقم (١,٣) مع مراعاة كتابتها بين علامات اقتباس. يستطيع المستخدم إضافة نصّ في أي مكان على الرسم، وكذلك إضافة مفتاح للتمييز بين المنحنيات.

```
>> figure
>> x=-4:.02:4;
>> y=cos(x);

>> plot(x,y, '--');              خط مقطع

>> xlabel('x-axis'); ylabel('y-axis');

>> hold                          يسمح hold برسم منحنى z مع y

Current plot held

>> z=sin(x);

>> plot(x,z, 'k');               اللون الأسود

>> title('plot of cos(x) and sin(x)');      عنوان الرسم
```

يظهر الرسم الموضح في الشكل رقم (١,٥) على نافذة مرقمة خاصة بالرسم تسمى *Figure*. ويتم التحكم بالخطوط والألوان عن طريق رموز، أدرجنا بعضها في الجدول رقم (١,٣)، وتكتب داخل أقواس دالة *plot*، ويمكن كتابة أكثر من رمز للحصول على الرسم المناسب مثل *plot(x,y,':*r')* التي ترسم المنحنى بخط أحمر مقطع تتخلله نجوم. كما يمكن التحكم بالشكل من خلال شريط الأوامر بالنافذة.



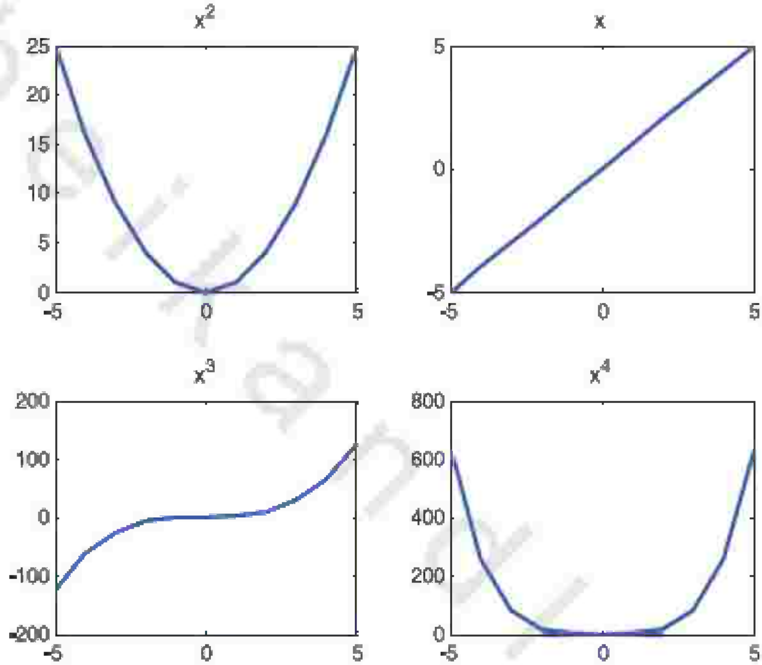
الشكل رقم (١,٥). نافذة الرسم **Figure**.

الجدول رقم (١,٣). أدوات التحكم بالرسم.

اللون	أحمر	أخضر	أزرق	سماوي	أرجواني	أصفر	أسود	أبيض
الرمز	r	g	b	c	m	y	k	w
نوع الخط	: مقطع بنقاط	- مقطع بمواصلة	-. مقطع بنقطة و واصلة	* خط تتخلله نجوم	+ خط يتخلله +	x خط يتخلله x	. خط يتخلله .	بدون رمز خط مستقيم

ولرسم الدوال x^4 ، x^3 ، x^2 ، x مستخدماً الدالة *subplot* التي تمكننا من تجزيء العرض إلى عدد من الخلايا ورسم كل دالة في خلية كما هو موضح في الشكل رقم (١,٦).

```
>> x=-5:5;
>>subplot(2,2,1); plot(x,x.^2);title('x^2')
>> subplot(2,2,3); plot(x,x.^3);title('x^3')
>> subplot(2,2,2); plot(x,x);title('x')
>> subplot(2,2,4); plot(x,x.^4);title('x^4')
```



الشكل رقم (١,٦). مثال على الأمرsubplot.

مثال رقم (١,٤)

لرسم دالة *f(x)* غير متصلة ومعرفة كالآتي :

$$f(x)=\begin{cases}\tan x & -\pi/4 \le x \le \pi/4\\ \sin(x) & \pi/4 \le x \le \pi/2\\ e^x & \pi/2 \le x \le 3\end{cases}$$

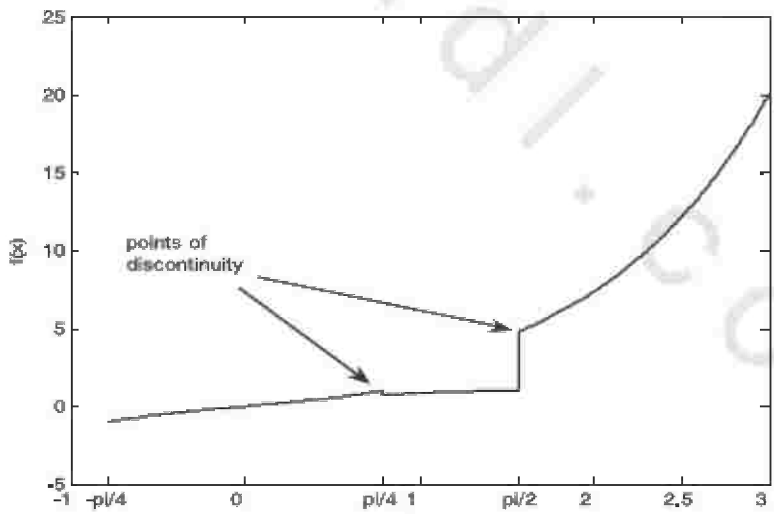
نُدخل كل جزء من مجال الدالة بمتجهات x3 ,x2, x1 وكل جزء من الدالة بمتجهات y1, y2, y3 :

```
>> x1=-pi/4:pi/200:pi/4;
>> y1=tan(x1);
>> x2=pi/4:pi/200:pi/2;
>> y2=sin(x2);
>> x3=linspace(pi/2,3);
>> y3=exp(x3);
```

ثم نضم متجهات المجال في متجه واحد x و بالمثل ننشئ متجه y :

```
>> x=[x1 x2 x3];
>> y=[y1 y2 y3];
>> plot(x,y)
```

ونحصل على رسم الدالة *f(x)* الذي يتضح عليه نقاط عدم الاتصال كما في الشكل رقم (١,٧).



الشكل رقم (١,٧). رسم الدالة *f(x)* .

هناك أنواع مختلفة من الرسومات، ومن أهمها :

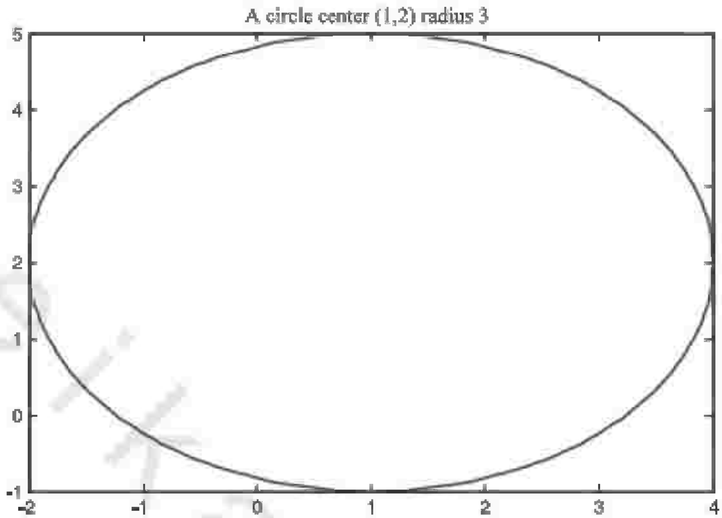
• إذا كانت الدوال معرّفة بمنحنيات وسيطية parametric curves فترسم بنفس الطريقة ، فمثلاً لرسم دائرة بمركز (1,2) ونصف قطر ٣ معرّفة بـ $x=1+2cost$ و $y=1+2sint$ على الفترة $0 \leq t \leq 2\pi$ نُدخل الأوامر التالية لنحصل على الشكل رقم (١,٨) .

```
>> t=0:2*pi/200:2*pi;
>> plot(1+3*cos(t),2+3*sin(t))
```

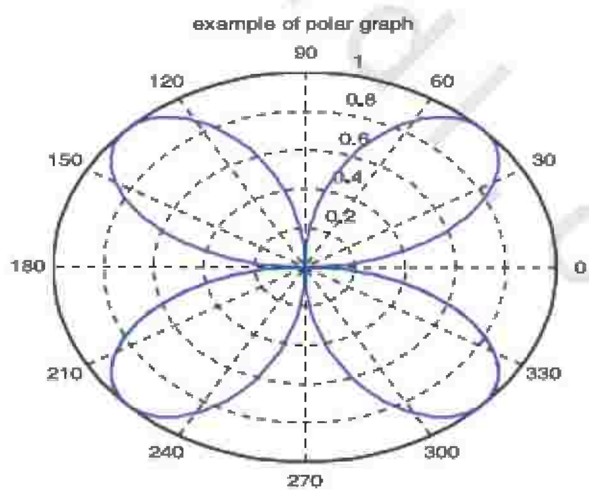
• الدالة *polar(theta, r)* تنتج رسماً بمحاور قطبية بالزوايا مدخلة في المتجه theta و القيم مدخلة في المتجه r ، فمثلاً الأمر *polar(x,sin(2*x))* يعطي الشكل رقم (١,٩) .

• نرغب أحيانا في تمثيل البيانات على شكل Bar chart أو على شكل Pie chart أو نقاط متفرقة Scatter points ويقدم MATLAB دوال جاهزة لذلك. على سبيل المثال pie(a,b) حيث a متجه يمثّل نسبة درجات الطلاب في مادة معينة ، و b طريقة التجزيء (هنا اخترنا أكبر قطعة تمثل التقدير C) تعطي الشكل رقم (١,١٠) .

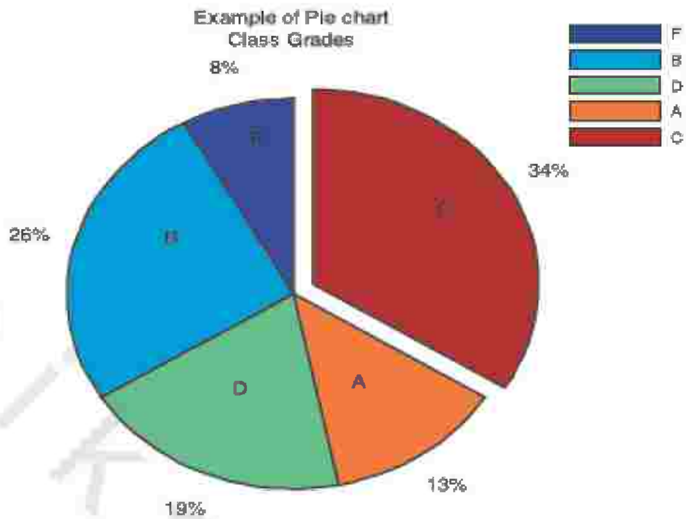
• الأمر bar(x) ينتج الرسم في الشكل رقم (١,١١) .



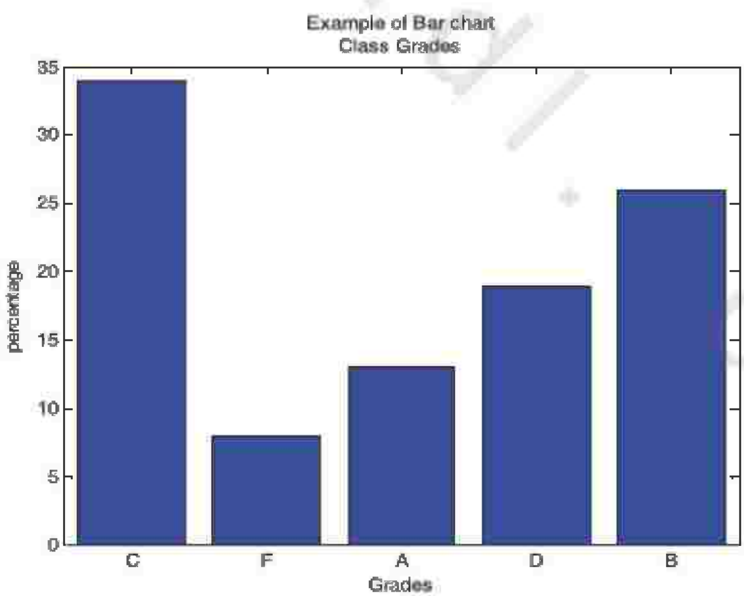
الشكل رقم (١,٨). رسم دائرة بنصف قطر ٣ ومركز (1,2).



الشكل رقم (١,٩). مثال لرسم قطبي.



الشكل رقم (١,١٠). مثال لرسم Bar chart.



الشكل رقم (١,١١). مثال لرسم Bar chart .

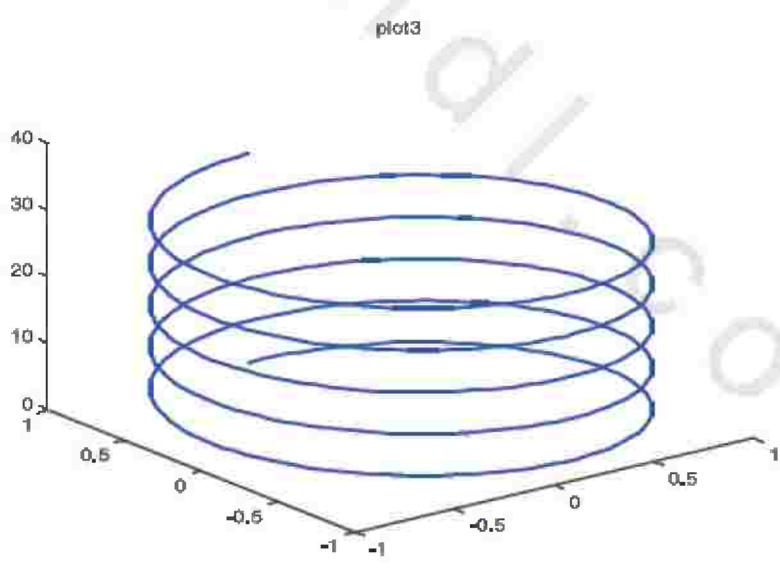
(١,٩,٢) الرسم في الأبعاد الثلاثية

يمكّن MATLAB المستخدم من العرض البياني في ثلاثي الأبعاد على شكل خطوط ذات ثلاثة أبعاد أو مسطحات متنوعة ، وسنعرض في هذا الجزء نبذة عن بعض الدوال التي تتحكم في هذا النوع من الرسوم ، وللتعرف أكثر على ما يمكن استخدامه من أوامر أخرى ننصح بالرجوع لأمر المساعدة *help graphics*.

الدالة *plot3(x,y,z)* ترسم رسماً ثلاثي الأبعاد ، وقد استخدمنا الأوامر :

```
t = 0:pi/50:10*pi
plot3(sin(t),cos(t),t )
```

والناتج هو الرسم الحلزوني في الشكل رقم (١,١٢).



الشكل رقم (١,١٢). مثال لرسم بالأمر plot 3 .

كما توجد دوال أخرى للرسم مثل *meshgrid, mesh, surf, contour, contour3*، ولكن في البداية نحدد مجال الدالة بتعريف مصفوفتين للإحداثيات x و y ونستخدم دالة *meshgrid* :

```
>> [x y]=meshgrid(-8:0.5:8);
```

ثم نعرّف الدالة z ، ونستخدم الرقم eps $=2.2x10^{-16}$ لتفادي القسمة على صفر.

```
>> r=sqrt(x.^2+y.^2)+eps;
>> z=sin(r)./r;
```

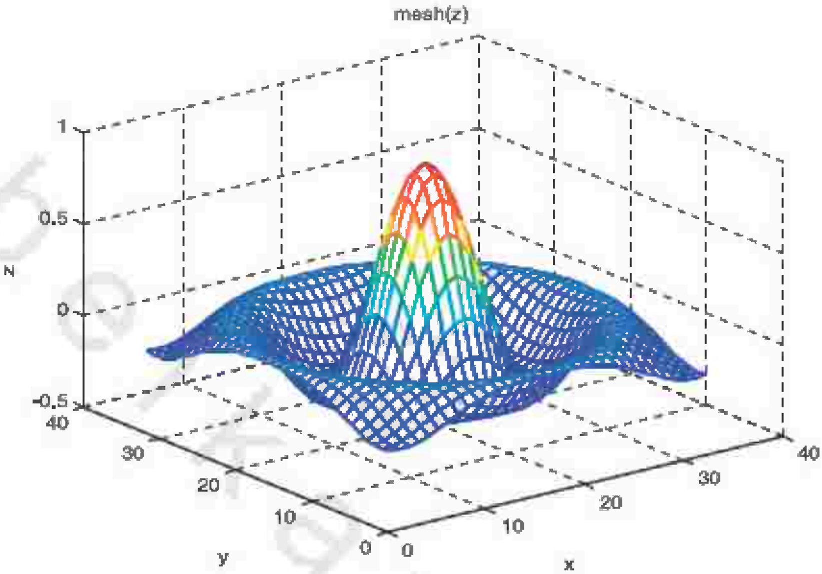
وهنا بعض الأمثلة لرسم القبعة المكسيكية باستخدام دوال مختلفة :

١- الأمر *mesh* : ينتج الرسم على شكل شبكة كما في الشكل رقم (١,١٣)، بحيث تأتي نقاط تقاطع الأعمدة والصفوف في الشبكة من المصفوفات المحددة في *meshgrid* .

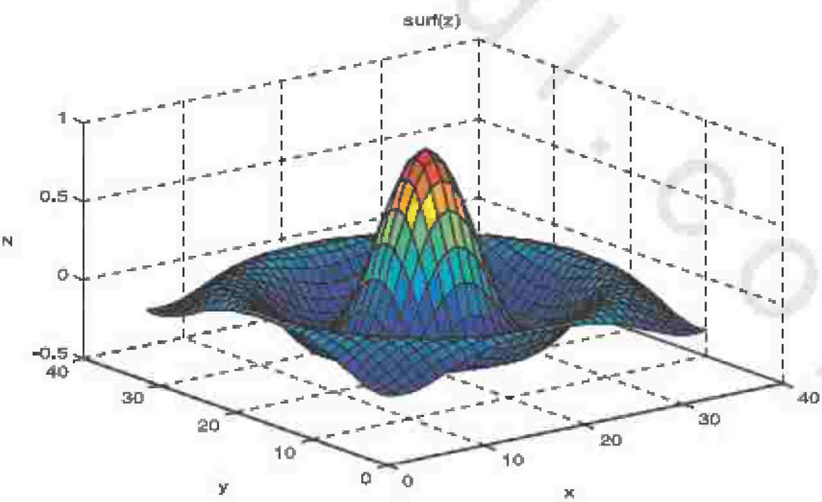
٢- لرسم مسطح نستخدم الدالة *surf* ، وينتج الشكل رقم (١,١٤).

٣- يمكن رسم خطوط محددة للرسم بدالة *contour3* المبينة بالشكل رقم (١,١٥).

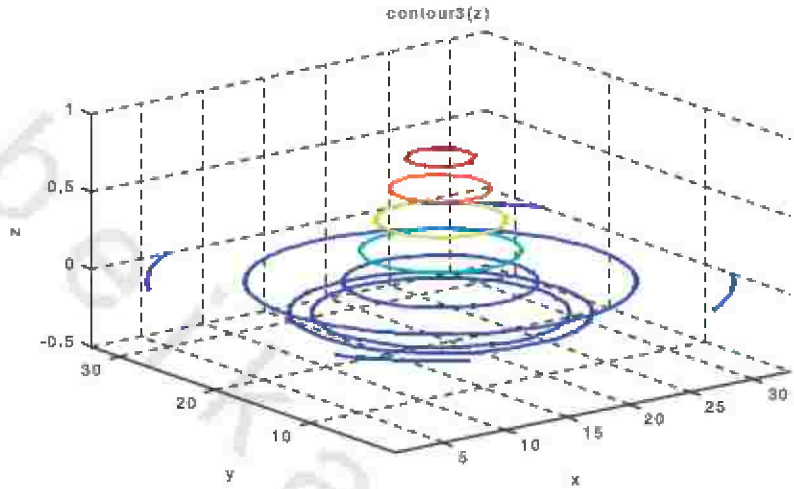
٤- يمكن رسم الخطوط تحت المسطح أو الرسم الشبكي بدالتي *surfc* و *meshc* مثل الشكل رقم (١,١٦).



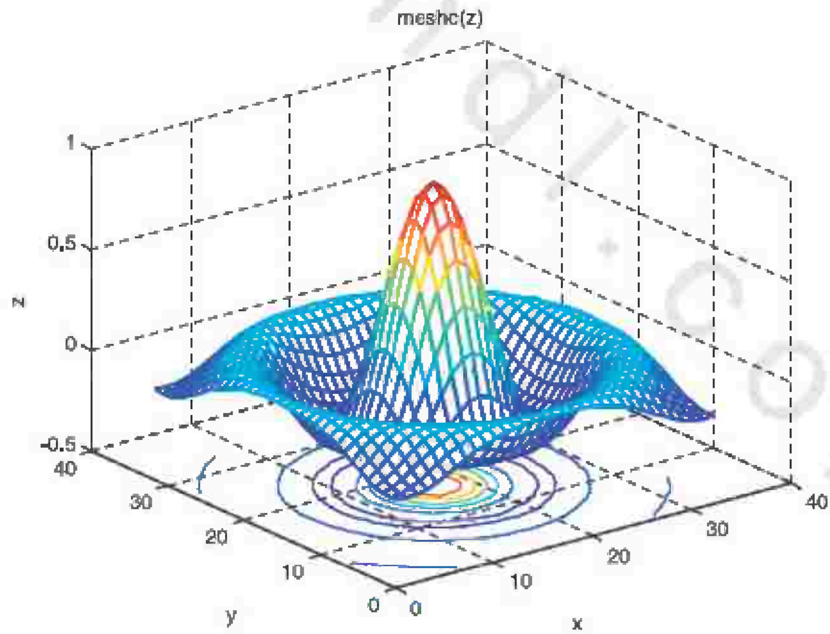
الشكل رقم (١,١٣). مثال لرسم بالأمر **mesh**.



الشكل رقم (١,١٤). مثال لرسم بالأمر **surf** .



الشكل رقم (١,١٥). مثال لرسم بالأمر contour3 .



الشكل رقم (١,١٦). مثال لرسم بالأمر meshc .

(١,١٠) العلاقات وعمليات المنطق الرياضي في MATLAB

بالإضافة للعمليات الرياضية التقليدية التي يمكن القيام بحسابها في MATLAB فإنه يقوم بإجراء عمليات وعلاقـات في المنطـق الرياضي Logical Operators. ويقدم MATLAB مبدأ جديداً وهو المتجه المنطقي logical vector وهذه خاصية قوية بـ MATLAB ولا توجد في أي لغة أخرى. الهدف من عمليات المنطق الرياضي هو الرد على الأسئلة الصائبة و الخاطئة ، و يقوم بهذه العملية MATLAB ويخزّن النتائج في المتجه المنطقي الذي يحتوي على القيمة 1 للصائب و القيمة 0 للخاطئ. أما أدوات الربط في المنطق الرياضي على MATLAB فهي موضحة في الجدول رقم (١,٤). نستخدم أدوات الربط للمقارنة بين متجهين بنفس الحجم ، أو لمقارنة عدد ثابت بمتجه. ونقارن الأعداد ببعضها وتكون الإجابة بصائب (1) أو خاطئ (0).

أكثر استخدام لهذه المقارنات هو في مجال البرمجة ، لأن استخدامها يعطي البرامج سرعة ويجعل المهمة التي تحتاج أسطراً عديدة لتنفيذها تُنفذ بأمر واحد من MATLAB، وسوف نعرض أمثلة على البرمجة في الجزء القادم.

الجدول رقم (١,٤). أدوات المنطق الرياضي.

أقل	<
أقل من أو يساوي	<=
اكبر من أو يساوي	>=
يساوي	==
أكبر	>
عدم المساواة	~=
النفي	~
وَ	&
أو	|

مثال رقم (٥,١)

نفرض أن لدينا متجهين A و B :

```
» A=1:11, B=16-A,

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

B =
   15   14   13   12   11   10   9    8    7    6
```

إذا قمنا بمقارنة المتجه A وعدد ثابت 5 :

```
» r=A>5
r =
   0   0   0   0   0   1   1   1   1   1   1
```

فإن العناصر الصفرية في المتجه المنطقي r تدل على $A \leq 5$ والعناصر التي تحتوي على الرقم واحد تدل على $A>5$:

```
» r=(A==B)
r =
   0   0   0   0   0   0   0   1   0   0   0
```

في هذه المقارنة نبحث عن العناصر من A التي تساوي عناصر من B . كما يستخدم ~ لنفي الجملة :

```
» R=~(A>5)
R =
   1   1   1   1   1   0   0   0   0   0   0
» R=(A>5)&(A<8)
R =
     0   0   0   0   0   1   1   0   0   0
```

في هذه المقارنة ينتج العدد واحد عندما تكون A أكبر من 5 وأصغر من 8.

إضافةً لما سبق من عرض لأدوات المنطق الرياضي يوفر MATLAB للمستخدم العديد من الدوال المنطقية، مثل الدوال في الجدول رقم (١،٥):

الجدول رقم (١،٥). أدوات المنطق الرياضي.

دالة	xor(x,y)
تنتج صائب(1) إذا كان أي عنصر في x غير صفري	any(x)
تنتج صائب(1) إذا كان كل عنصر في x غير صفري	all(x)
صائب إذا كانت العبارة الحجية فارغة	isempty(x)
صائب إذا كان A و B متساويين	isequal(A,B)
صائب إذا كانت x مالا نهاية	isinf(x)
يبحث عن شرط معين	find

نمثل لطرق استخدام بعض هذه الدوال في التالي:

```
>> b=[0 inf 0 0 0  9];
>> find(isinf(b))
ans =
    2
```

الإجابة 2 تعطي رقم العنصر الذي يحقق الشرط:

```
>> a
a =
   0.5000    1.0000    1.6000    1.2000    0.8000    2.1000
>> find(isempty(a))
ans =
    [ ]
>> find(isequal(a,b))
ans =
    [ ]
```

نلاحظ إجابة MATLAB بـ [ ] لدالة البحث *find* ، مما يدل على أنه لا يوجد الشرط الذي نبحث عنه.

```
>> all(b)
ans =
   0
>> any(b)
ans =
   1
```

الإجابة 0 تدل على أنه ليس كل عناصر b صفرية ، أما الإجابة 1 تدل على وجود عناصر غير صفرية في b .

(١,١١) البرمجة في MATLAB

هناك تشابه كبير بين البرمجة في MATLAB ولغات البرمجة المعروفة ذات المستوى الرفيع. وقواعد MATLAB للبرمجة قريبة جداً من لغة FORTRAN و PASCAL مع بعض الإضافات من لغة C . يختلف MATLAB عن اللغات الأخرى بكونه بيئة interactive وجميع البرامج يتم ترجمتها في MATLAB جملة جملة interpreting بدلاً من ترجمة البرنامج compiling كما يجري في اللغات الأخرى. لأن MATLAB يتعامل مع المتغيرات على أنها مصفوفات، فهو يوفر برامج عديدة متطورة و جاهزة للمستخدم في مجال المصفوفات مثل حلّ نظام معادلات خطية أو إيجاد القيم الذاتية لمصفوفة.

الأوامر for ، if و while تعدّ أساسيات البرمجة على MATLAB. نقوم بالبرمجة على MATLAB باستخدام الحلقات loops وهي مجموعة من الأوامر التي تنفّذ بشكل تكراري حتى يتحقق شرط معين مثل for loop ، while loop .

١ – **for :** يتم تنفيذ تكرار العبارة داخل الحلقة بعددٍ محدد من المرات يحدّد

بوساطة عدّاد الحلقة counter variable الذي يبين متغير العدّاد، والقيمة البدائية والقيمة النهائية للعداد.

الشكل العام :

```
for counter variable=initial value:final value
action
end
```

يمكننا الأمر for من إنشاء المصفوفة c ذات الشكل الخاص بالبرنامج التالي :

```
» for i=1:5
for j=1:5
if i==j
c(i,j)=i*(5-i+1);
elseif j>i
c(i,j)=c(i,j-1)-i;
else j<i
c(i,j)=c(j,i);end
end
end
end
>> c
   5   4   3   2   1
   4   8   6   4   2
   3   6   9   6   3
   2   4   6   8   4
   1   2   3   4   5
```

لحساب مضروب الأعداد من ١ الى ١٠ نكتب برنامجاً باستخدام *for* :

```
>> n=10;
>> fact=1;
>> for k=1:n
fact=k*fact;
disp([k fact])
end
   1     1
   2     2
   3     6
   4     24
   5     120
   6     720
```

```
7     5040
8     40320
9     362880
10    3628800
```

ولتوفير الجهد على المستخدم توجد دالة جاهزة في MATLAB تُدعى *factorial* لحساب مضروب أي عدد ، فمثلاً :

```
>> factorial(10)
ans =
   3628800
```

مثال رقم (٦ ,٩)

استخدم حلقة for لحساب نهاية المتسلسلة $x_n = ax_{n-1}/n$ حيث a=10 (سنرى في الجزء القادم دالة جاهزة في MATLAB تقوم بحساب النهايات بأمر واحد). بما أن MATLAB لا يتعامل مع المتجهات غير المنتهية فإننا نستخدم قيمة كبيرة لعدد الحدود n لتظهر الصورة العامة للنهاية ، وهنا افترضنا أن n=10 حيث نرى نهاية العبارة تتقارب إلى 2.7557×10^3 :

```
>> a=10;
>> x=1;
>> k=10;

>> for n=1:k
x=a*x/n;
disp([n x])
end
    1        10
    2        50
    3        166.6667
    4        416.6667
    5        833.3333
    6        1.3889e+003
    7        1.9841e+003
    8        2.4802e+003
    9        2.7557 e+003
    10       2.7557e+003
```

٢- **While** : تستخدم لتنفيذ عبارات برمجية طالما أن الشرطَ محققٌ.

الشكل العام :

```
while condition
action
increment action
end
```

في المثال نلاحظ استخدام أدوات الربط للمنطق الرياضي التي تسهل كتابة البرنامج :

```
while sum(x)~=max(y)
x=x.^2;
y=y+x;
end
```

٣- **if** : تقوم بفحص شرط منطقي وبعد ذلك يتم الانتقال إلى تنفيذ تعليمات معينة إذا تحقق الشرط. ونستطيع استخدام أكثر من عبارة if داخل بعضها على شكل nested loops .

الشكل العام :

```
if condition
statements
end
--------------
if condition
statements
else
statements
end
------------
if condition
<statements>
elseif condition2
statements
else
statements
end
```

مثال رقم (١,٧)

لو أردنا حساب جذور معادلة من الدرجة الثانية ax^2+bx+c باستخدام المميز $\frac{-b \pm \sqrt{b^2-4ac}}{2a}$ للمعادلة ($-10x+12+2x^2=0$) فإننا ندخل الخطوات التالية :

```
>> a=2;
>> b=-10;
>> c=12;
>> d=b^2-4*a*c;
>> if a~=0
if d<0
disp('çomplex roots')
else
x1=(-b+sqrt(d))/(2*a)
x2=(-b-sqrt(d))/(2*a)
end
end
```

نحصل على الجذور

```
x1 =
    3
x2 =
    2
```

(١,١٢) حسابات رمزية Symbolic Algebra

كما عرضنا في الأجزاء السابقة لدى MATLAB القدرة الفعّالة على التعامل مع الحسابات العددية ، فإننا في هذا الجزء نعرض قدرة MATLAB على التعامل مع الحسابات الرمزية symbolic manipulations أيضاً . وهي التي تحتوي على رموز ليس لها قيمٌ عددية معرفة مسبقاً ويعطي MATLAB نتائج بدلالة هذه الرموز. وهذه الميزة أضيفت في الإصدارات الجديدة من البرنامج ، ويتمكن MATLAB من إجراء ذلك عن طريق المحرك MAPLE .

للدخول في بيئة الحسابات بالرموز يجب طباعة الأمر *syms* وإدراج المتغيرات التي يراد استخدامها كرموز(يعتبر MATLAB الرموز ذات قيم صحيحة ، أما إذا كان

الرمز ذا قيمة حقيقية فيمكن إضافة كلمة real). فإذا أردنا حل معادلة من الدرجة الثالثة مثل $x^3-2x^2-x+2=0$ في x نستخدم الأمر *solve* :

```
>> syms x
>> solve(x^3-2*x^2-x+2)
```

والناتج هو الجذور الثلاثة :

```
ans =
 -1
 1
 2
```

أو إذا أردنا تحليل المعادلة نستخدم الأمر *factor* :

```
>> factor(x^3-2*x^2-x+2)
 ans =
 (x-1)*(x-2)*(x+1)
```

ويمكن إيجاد المشتقتين الأولى والثانية بالأمرين *diff(f(x))* و *diff(f(x),2)* :

```
>> diff(x^3-2*x^2-x+2)
ans =
3*x^2-4*x-1
>> diff(x^3-2*x^2-x+2,2)
 ans =
 6*x-4
```

أما حساب التكامل غير المحدود فيتم بالأمر *int(f(x))* :

```
>> int(x^3-2*x^2-x+2)
 ans =
 1/4*x^4-2/3*x^3-1/2*x^2+2*x
```

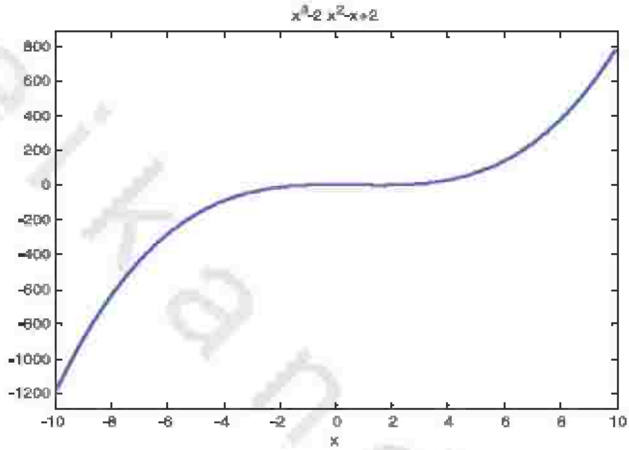
والتكامل المحدود من 0 إلى 3 لنفس الدالة هو :

```
>> int(x^3-2*x^2-x+2,0,3)
 ans =
 15/4
>>double(ans)
ans =
3.7500
```

لحساب القيمة العشرية للنتيجة السابقة

نستخدم الأمر **double (ans)**

كما يوجد أمر تنفيذي رمزي لرسم الدوال *ezplot(f,[a,b])*، حيث إن [a,b] هي الفترة المراد رسم الدالة عليها، فلنرسم دالتنا في الفترة [-10,10] لنصل للشكل رقم (١،١٧):



الشكل رقم (١،١٧). مثال لرسم بالأمر ezplot.

يقدم MATLAB دالة *dsolve* التي تعطي حلولاً لمعادلات تفاضلية. المعادلة ذات القيمة الابتدائية عند 1.5- :

$$y(-1.5)=-1 \text{ حيث إن } y' = y - x,$$

يمكن حساب الحل بتحديد المعادلة و القيمة الابتدائية و المتغير:

```
>> syms x y
>> h=dsolve('Dy=y-x,y(-1.5)=-1','x')
 h =
 x+1-1/2*exp(x)/exp(-3/2)
```

وإذا أردنا الحل العام لمعادلة تفاضلية من الدرجة الأولى $y' = x^2 + y$:

```
>> g=dsolve('Dy=x^2+y','x')
g =
-x^2-2*x-2+exp(x)*C1
```

أما لإيجاد نهاية العبارة الرياضية:

$$\lim_{x\to\infty}\frac{2x-1}{\sqrt{3x^2+x-1}}$$

فنكتب أمر *limit* مع تحديد الدالة ، والمتغير، والنهاية ، لنحصل على النتيجة:

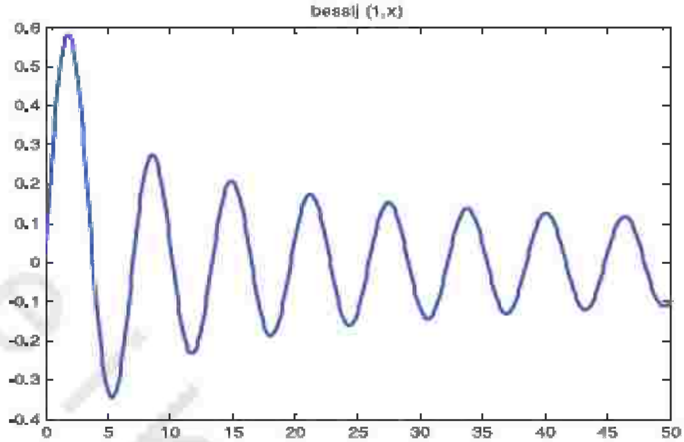
```
>> syms x, limit((2*x-1)/sqrt(3*x^2+x-1),x,inf)
ans =
2/3*3^(1/2)
```

أو النهاية من اليمين واليسار:

```
f =
(x-3)/abs(x-3)
>> a =limit (f,x,3,'left')
a =
-1
>> a =limit (f,x,3,'right')
a =
1
```

ويمكن تعريف دوال كثيرة جاهزة مثل laplace و الدالة العكسية ilaplace ، besselj أو gamma ورسمها بالشكل رقم (١,١٨) كذلك:

```
laplace(a)
ans =
1/s^2
>> laplace(t^2)
ans =
2/s^3
>> ilaplace(1/s^3)
ans =
1/2*t^2
besselj(1,[0:0.1:50])
plot([0:.1:50],besselj(1,[0:.1:50]))
```



الشكل رقم (١،١٨). مثال لرسم besselj .

وسيتم عرض كل هذه الأوامر الرمزية بتفصيل أكثر في الفصول القادمة حسب استخداماتها المختلفة. بصورة عامة في بيئة الحسابات الرمزية لاحظنا كيف يسّهل MATLAB إجراء الحسابات خاصة للطلاب في حلول التمارين المعقدة، والتي تستهلك الكثير من وقت الطالب لو أراد حلّها باليد، وبذلك يتمكن الطالب بوساطة MATLAB من الاطلاع على حلول لكمية أكبر من التمارين لترسيخ أي مفهوم رياضي.

(١،١٣) تمارين

١- اكتب العبارات التالية في أبسط صورة:

أ) $3.5-6/17(3^{2\pi})$

ب) $Sin(1.5)/5e^2$

جـ) 2>3&1

د) ~[1 2 0]*3

٢- المتجهان [3.2 1.25] و [1.9 5.5] يمثلان ضلعين في مثلث ، والضلع الثالث هو حاصل جمعهما ، احسب محيط المثلث.

٣- إذا كان A =[1 2 3 4 5] ، B =[4 5 6 7 8] ، C=[1 -1 2 -1 1] احسب (A.*B).*C و قارن مع A.*(B.*C)

٤- ارسم الدوال التالية :

أ) $y=2\sin 3x+3\cos 2x$ $\quad 0 \le x \le 2\pi$

ب) $t=2s/(1+s^2)$ $\quad -3 \le s \le 3$

جـ) $y=3\ln(1+x)$ $\quad -1/2 \le x \le 2$

٥- اكتب m-file لرسم الدالة التالية :

$$f(x)=\begin{cases} x-3-\cos\dfrac{\pi}{4}x & 2<x\le 4 \\ 1-\dfrac{x}{2}-\tan\dfrac{\pi}{8}x & 0\le x\le 2 \end{cases}$$

٦- استخدم دالة mesh لرسم $z=2xy/(x^2+y^2)$ على x=1:0.1:3 y=1:0.1:3 .
ثم قارن بـ surf ، و contour .

٧- اكتب برنامجاً لحساب القيمة العظمى والقيمة الصغرى للدالة $y = 2\sin(2x)-3\cos(x/2)$ على $[0, 2\pi]$ واستخدم الدوال min ، max والرسم للتأكد من النتائج.

٨- اكتب برنامجاً لاستخدام المميز لحساب جذري معادلة من الدرجة الثانية.
طبق البرنامج على $2x^2-12x+18=0$ واستخدم fprintf لطباعة النتائج.

٩- اكتب برنامجاً لإنشاء مصفوفة A 3×3 قطرية بالقطر [4 2 1] ثم أجرِ العمليات التالية :

أ) بدل العمودين الثاني والثالث.

ب) قم بإضافة عمود رابع صفري وسمِ المصفوفة B .

جـ) احسب المحدد، الرتبة ، المنقول وأوجد المعكوس لـ A .

د) احسب A*B وقارن بـ A.*B

هـ) هل المصفوفة A متناظرة ؟

و) تأكد من $(AB)^t = B^tA^t$

١٠ - حلّ المعادلة $20x^2-31x-12=0$ باستخدام *syms* .

١١- بسّط العبارة $(x^{12}-y^{12})/(x-y)$ للحصول على كثيرة حدود.

١٢- استخدم *ezplot* لرسم الدالة $y=\exp(-x^{50})$ باختيار المجال المناسب.

١٣- احسب القيم الذاتية للمصفوفة :

$$A=\begin{bmatrix}1 & 0 & 2\\ 0 & 1 & -1\\ -1 & 1 & 1\end{bmatrix}$$

١٤- أثبت باستخدام *inv* أن المصفوفة غير شاذة :

$$m=\begin{bmatrix}1 & 2 & 0\\ 2 & 1 & -1\\ 3 & 1 & 1\end{bmatrix}$$

١٥- ما هي قيم α ليصبح للمصفوفة التالية معكوس؟

$$A=\begin{bmatrix}1 & -1 & a\\ 2 & 2 & 1\\ 0 & a & -3/2\end{bmatrix}$$