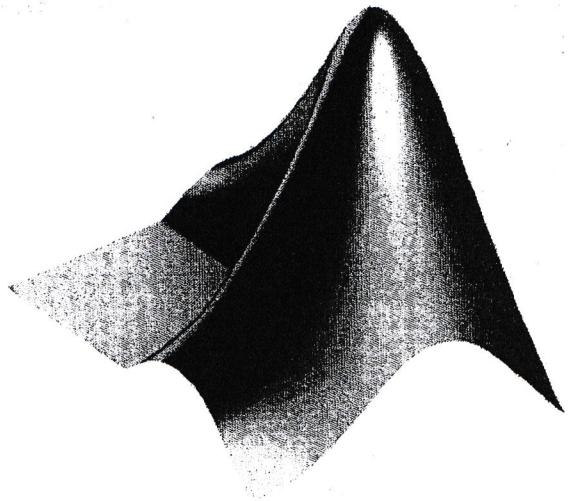
كلية التربية للعلوم الصرفة / قسم الفيزياء

المرحلة الثانية

مختبر الحاسبات

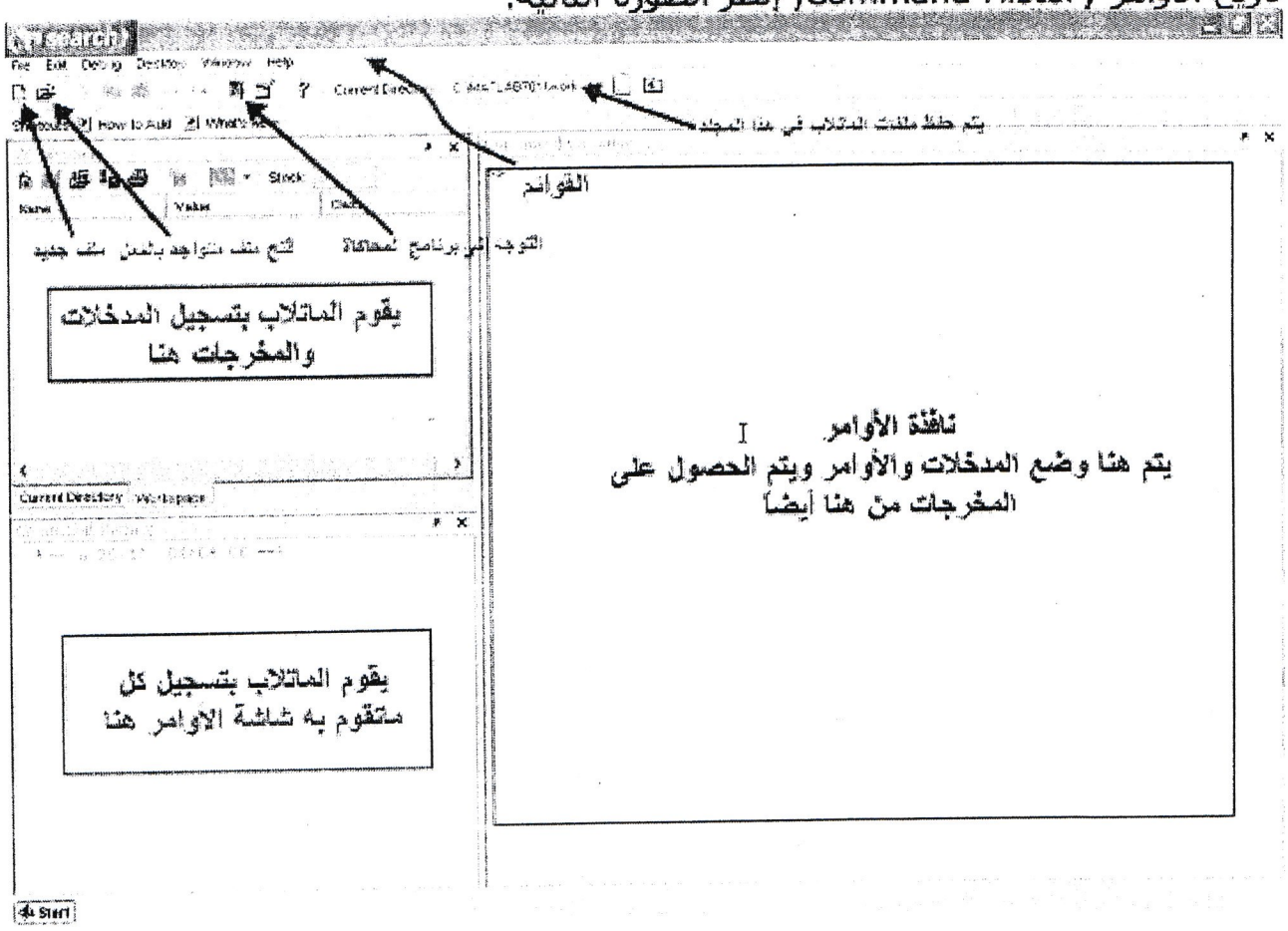


MATLAB®

العام الدراسي 2020-2021

واجهة البرنامج

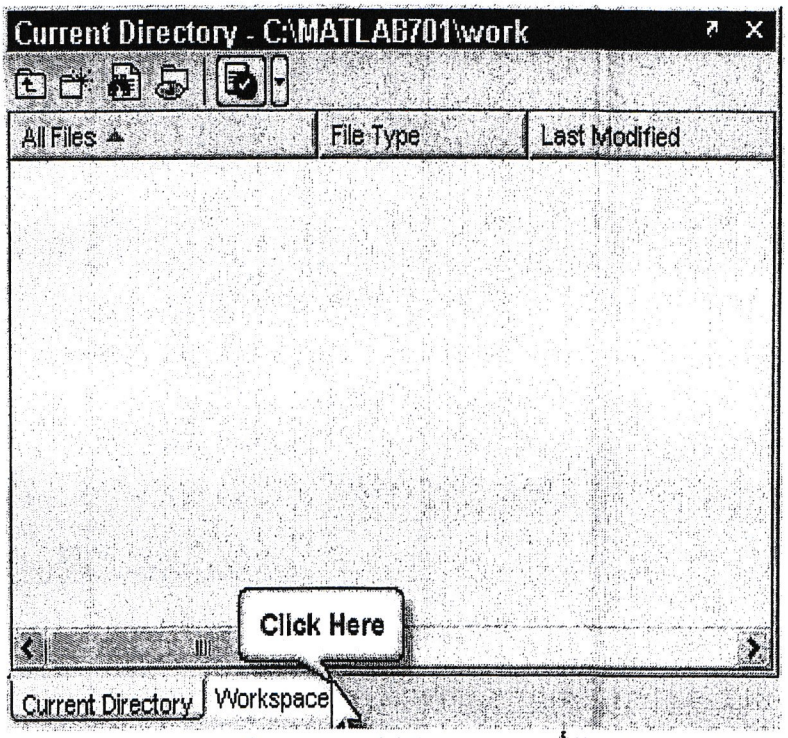
تتسم واجهة البرنامج بالسهولة في التعامل معها, حيث يتم تقسيم مناطق العمل بها إلى ثلاث مناطق رئيسية, وهي كالتالي نافذة الأوامر Command Window و منطقة العمل Workspace و تاريخ الأوامر Command History, إنظر الصورة التالية.



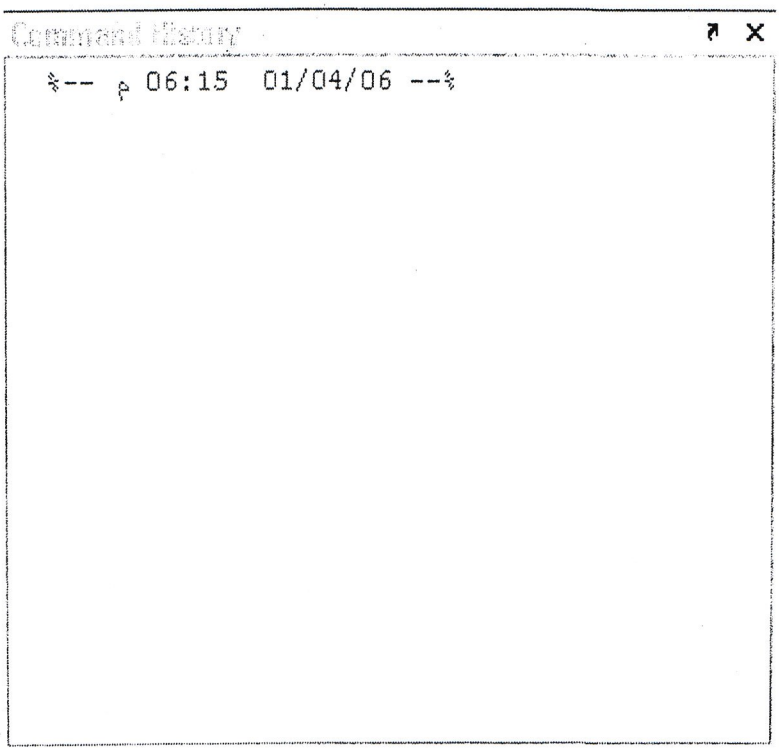
نافذة الأوامر Command Window: حيث يتم إدخال المدخلات Inputs والأوامر Commands, ويعمل الماتلاب على تحليل تلك البيانات ومدى مطابقة المدخلات للوظيفة المطلوبة منه, حتى تحصل على النتائج في نفس الشاشة.

منطقة العمل Workspace: حيث يقوم الماتلاب بتسجيل المدخلات Inputs والمخرجات Outputs في هذه الشاشة.

ملاحظة: عند بدء العمل على الماتلاب لأول مرة, لاتظهر نافذة Workspace , وحتى تظهر إضغط بزر الفأرة على كلمة Workspace كما في الصورة التالية



نافذة تسجيل الأوامر Command History: يتم تسجيل كل ما يقوم به المستخدم على برنامج الماتلاب في هذه النافذة. إنظر الصورة التالية

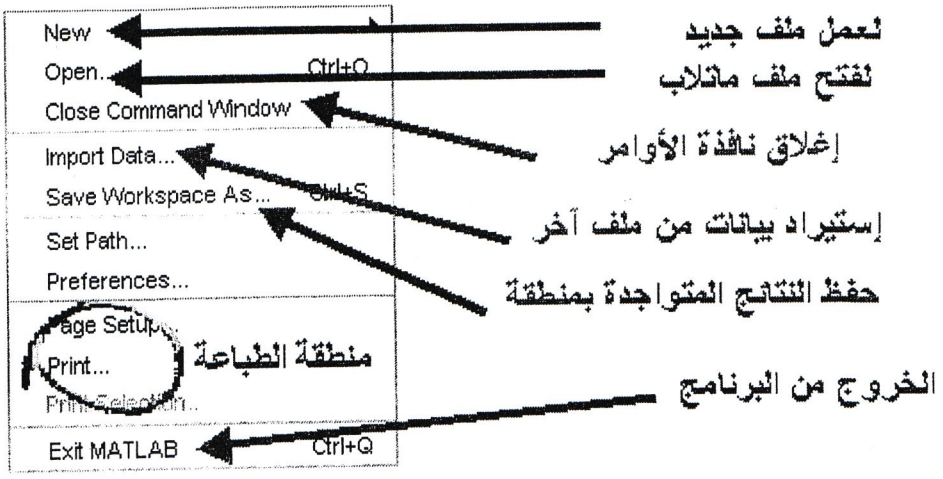


قائمة إبدأ Start: تستخدم هذه القائمة للوصول إلى التطبيق المراد تنفيذه, تستخدم هذه القائمة في المراحل المتقدمة في برنامج الماتلاب
صورة 4

بعض الأساسيات الهامة لمستخدمي برنامج الماتلاب
سنتعرف بإذن الله على القوائم, وما يقوم به كل إختيار.

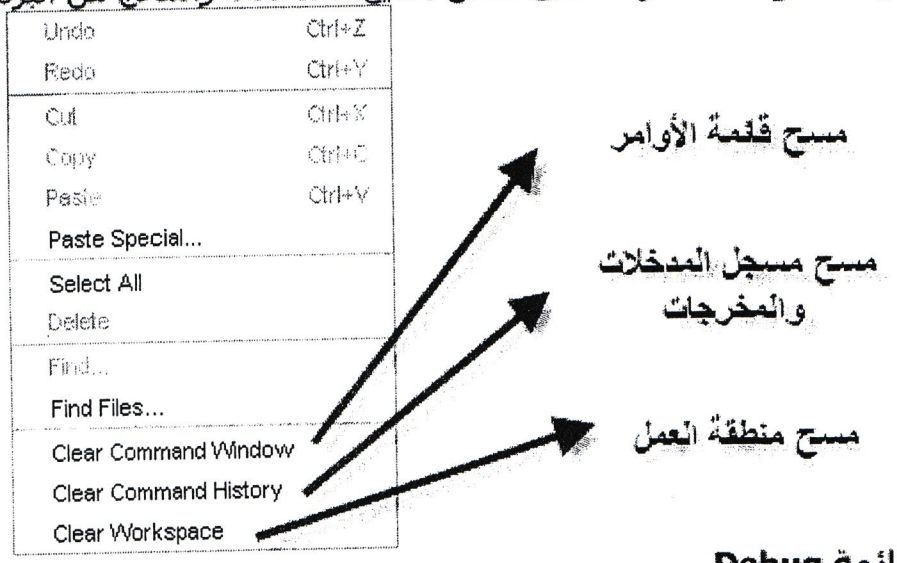
قائمة ملف File

تتكون هذه القائمة من العديد من الخيارات, والتي تنفذ كل منها وظيفة محددة باقي البرامج



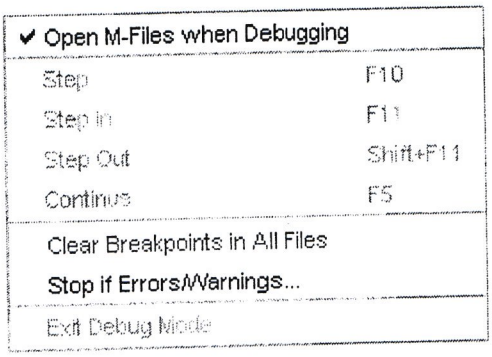
قائمة التعديل Edit

فكما تعودنا في تلك القائمة أن نجد أوامر (نسخ Copy, قص Cut, لصق Paste, بحث Find),
ولكن هنالك ثلاث أدوات هامة بها وهم
Clear Command Window
Clear Command History
Clear Workspace
حيث تعمل تلك الأدوات على مسح جميع المدخلات والنتائج من البرنامج



قائمة Debug

هذه القائمة خاصة بمعالجة البيانات, والطريقة المتبعة من قبل برنامج الماتلاب في مواجهة الأخطاء.
أنظر الصورة التالية

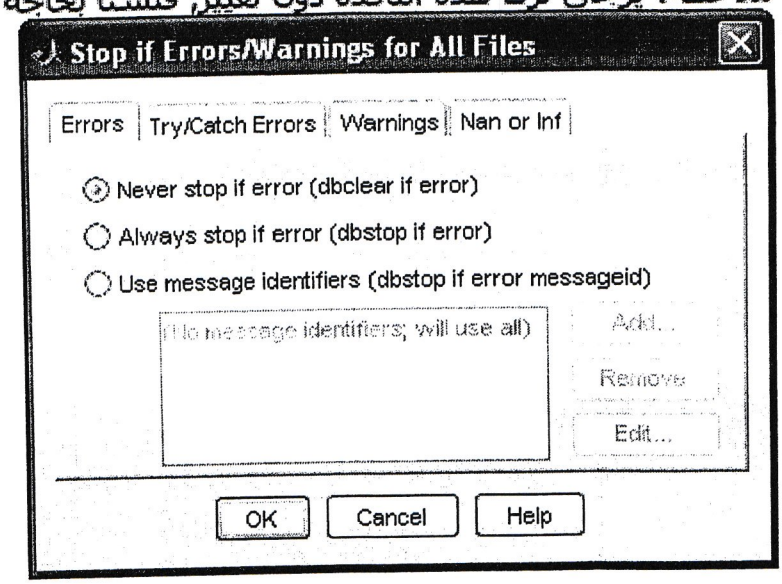


تختص هذه المنطقة بعملية معالجة البيانات، وإحتمالات حدوث الخطأ في برنامج الماتلاب

فمثلاً قم بإختيار ...Stop If Errors/Warnings

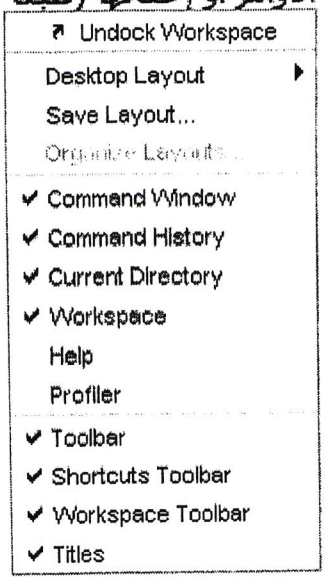
ستلاحظ ظهور نافذة, تعطيك حرية الإختيار في تصرف برنامج الماتلاب عند حدوث أخطاء أو تحذيرات

ملاحظة: يرجى ترك هذه النافذة دون تغيير, فلسنا بحاجة لها الآن.

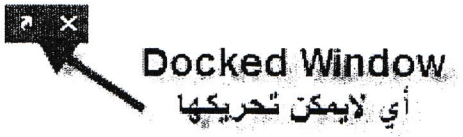


قائمة Desktop:

في هذه القائمة يتم التحكم بمحتوى الواجهة الخاصة ببرنامج الماتلاب, فمثلاً يمكننا إظهار نافذة الأوامر أو إخفائها (طبعاً لو أخفيناها مش حنعرف نشتغل), أنظر الصورة



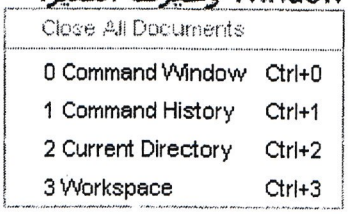
معلومة هامة:
تكون النوافذ في أحد الوضعين
1- Docked: حيث تكون النافذة غير قابلة للتحريك من مكانها.
2- Undocked: حيث تكون النافذة قابلة للتحريك وتعديل مقاسها أيضاً



يتبقى لدينا قائمتان هما

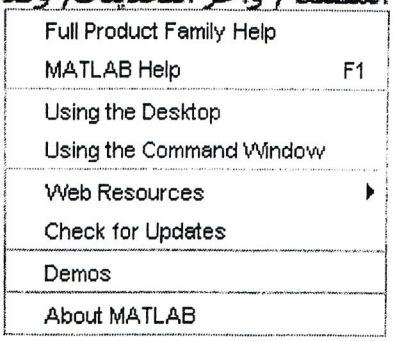
قائمة Window:

حيث يمكنك التنقل بين ملفات الماتلاب المختلفة, وكذلك النوافذ مثل نافذة الأوامر Command Window وغيرها الكثير.



قائمة Help:

حيث تقوم تلك القائمة, بتوفير المساعدات الضرورية في البرنامج, ووسائل الإتصال بالشركة المصنعة, وآخر التحديثات, وكذلك تعلم الماتلاب باللغة الإنجليزية



بعض الدوال الاساسية المستخدمة في برنامج ماتلاب

ELEMENTARY MATH FUNCTIONS

Elementary math functions include logarithms, exponentials, absolute value, rounding functions, and functions used in discrete mathematics.

Common Computations:

abs(x) : Finds the absolute value of x.

```
abs(3)
ans

 3
```

sqrt(x) : Finds the square root of x.

```
 sqrt(85)
ans

 9.2195
```

nthroot(x,n) : Finds the real *n*th root of x. This function will not return complex results.

```
nthroot(2, 3)
ans
1.2599
```

sign(x) : Returns a value of -1 if x is less than zero, a value of 0 if x equals zero, and a value of 1 if x is greater than zero.

```
sign(8)
ans

1
```

rem(x,y) Computes the remainder of x/y.

```
rem(25,4)
ans  1
```

The following are true by convention:

rem(X,0) is NaN.

rem(X,X) for X~=0 is 0.

rem(X,Y) for X~=Y and Y~=0 has the same sign as X.

exp(x) : Computes the value of e^x, where e is the base for natural logarithms, or approximately 2.7183.
exp(10)

ans

2.2026e+04 =$2.2026 \times 10^{+04}$ (note)

log(x) : Computes ln(x), the natural logarithm of x (to the base e).
log(10)
ans

2.3026

log10(x) : Computes log10(x), the common logarithm of x (to the base 10).
log10(10)
ans

1

round(x) : Rounds x to the nearest integer.

round(8.6)
ans

9

fix(x) : Rounds (or truncates) x to the nearest integer toward zero. Notice that 8.6 truncates to 8, not 9, with this function.
fix(8.6)
ans

8

floor(x) : Rounds x to the nearest integer toward negative infinity.

floor(8.6)
ans

8

ceil(x) rounds the elements of x to the nearest integers greater than or equal to x

ceil(3.9)

ans

4

ceil(3.2)

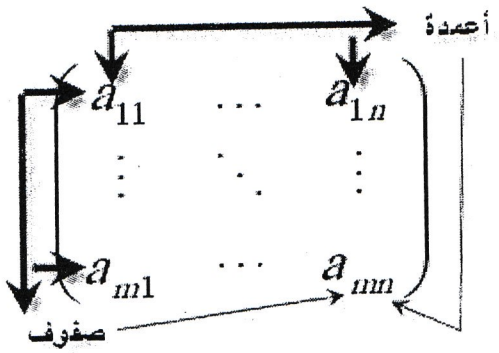
ans

4

المصفوفات
Matrices

ما هي المصفوفة:

هي مجموعة من البيانات والتي يتم وضعها في صورة صفوف وأعمدة, وتأخذ الشكل التالي :



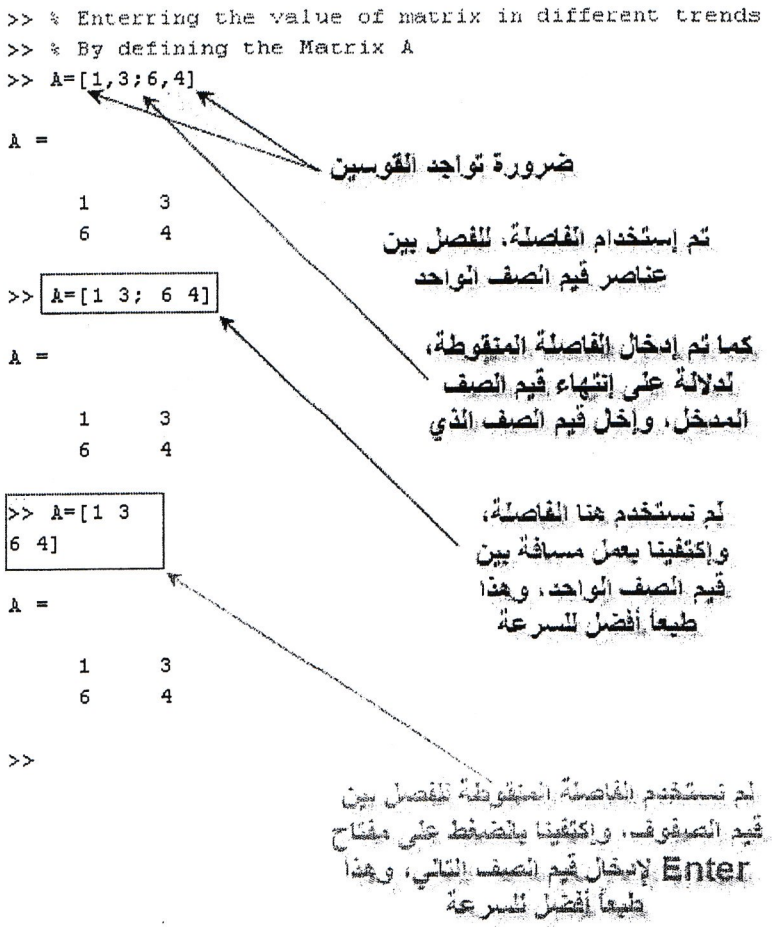
وتستخدم المصفوفات في حل كثيرات الحدود ,Polynomials وفي حل مجموعة من المعادلات, كما سيتم شرحه لاحقاً

كيفية كتابة المصفوفات في برنامج الماتلاب:

يتم إدخال المصفوفة بكتابة عناصر الصف الأول, ثم الثاني وهكذا . فمثلاً كتابة مصفوفة مثل التالية :

$$\begin{pmatrix} 1 & 3 \\ 6 & 4 \end{pmatrix}$$

ولكن قبل إدخال القيم التالية, على الجميع أن يعلم بأنه يتم كتابة عناصر الصف الأول ,ويتم الفصل بين أرقام الصف الأول إما بفاصلة (,) Comma أو بعمل مسافة Space بين الأرقام, بعد إدخال قيم الصف الأول يتم فصل عناصر الصف الأول عن عناصر الصف الثاني (الذي سيتم إدخال قيمه) إما بالضغط على مفتاح Enter أو باستخدام الفاصلة المنقوطة Semicolon (;) , أنظر الصورة التالية :



فكما نرى أساليب متعددة لإدخال قيم المصفوفات والشكل واحد في جميع الطرق.

العمليات الأساسية للمصفوفات :

الجمع:

قبل البدء في الشروع ببدء استخدام الماتلاب يجب أولاً أن نذكر شرط جمع مصفوفتين.

شرط جمع مصفوفتين:

لنفترض أن لدينا مصفوفتين A & B, فشرط جمعهما أن يكون كلاهما له نفس عدد الصفوف m وكذلك نفس عدد الأعمدة n . فمثلاً المصفوفتان التاليتان يمكن جمعهما لأنها يحملان نفس عدد الصفوف والأعمدة .

$$A=\begin{pmatrix}1 & 2\\ 3 & 4\\ 5 & 6\end{pmatrix}_{3\times 2}$$

$$B=\begin{pmatrix}7 & 8\\ 9 & 10\\ 11 & 12\end{pmatrix}_{3\times 2}$$

كما نرى فإن عدد الصفوف في المصفوفة الأولى مساوياً لعدد الصفوف في المصفوفة الثانية، وكذلك عدد الأعمدة لكلتا المصفوفتين

كيف تتم عملية جمع صفوفتين:

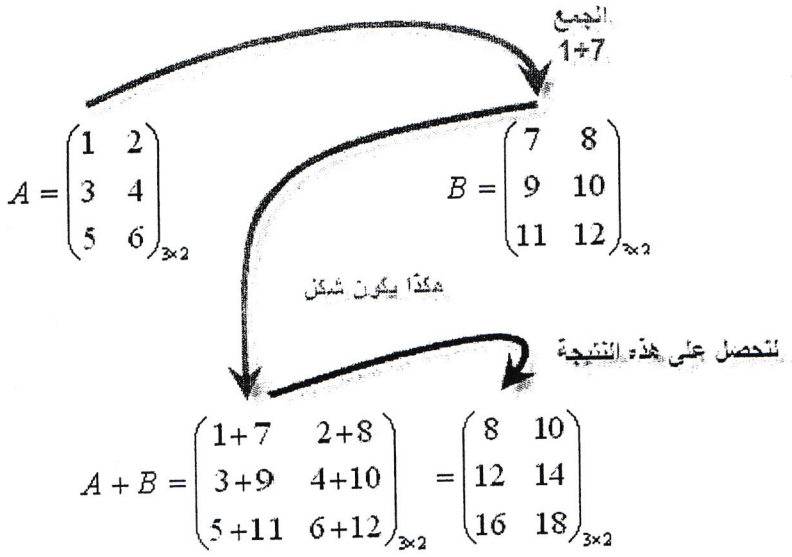
تتم عملية الجمع بجمع العنصر الأول للصف الأول مثلاً في المصفوفة الأولى وما يناظره في المصفوفة الثانية, وبالتالي نكون قد جمعنا العنصر الأول للصف الأول.وبالتالي نكون قد جمعنا

1+7=8

جمع الصف الأول العنصر الثاني: نجمع العنصر الثاني للصف الأول في المصفوفة الأولى وما يناظره في المصفوفة الثانية, وبالتالي نكون قد جمعنا

2+8=10

ونستمر هكذا حتى إتمام كامل المصفوفة, ويمكن تلخيص العملية في الصورة التالية :



$$A+B=\begin{pmatrix}1+7 & 2+8\\ 3+9 & 4+10\\ 5+11 & 6+12\end{pmatrix}_{3\times 2}=\begin{pmatrix}8 & 10\\ 12 & 14\\ 16 & 18\end{pmatrix}_{3\times 2}$$

الجمع في الماتلاب :

يجب أولاً كتابة المصفوفتين ,A&B كما تعلمنا سابقاً ثم استخدام رمز الجمع (+) للتتم عملية الجمع, أنظر الصورة التالية

```
>> % Today We're going to discuss the basic operation on Matrices
>> % By Defining the Matrix A
>> A=[1 2;3 4;5 6]

A =

     1     2
     3     4
     5     6

>> % By Defining the matrix B
>> B=[7 8;9 10;11 12]

B =

     7     8
     9    10
    11    12

>> % By making addition to both A&B

>> % Assume that the Result of summation would be denoted as C
>> C=A+B

C =

     8    10
    12    14
    16    18
```

طرح المصفوفات :

شرط طرح المصفوفات هو نفس شرط الجمع, حيث يشترط أن تكون المصفوفات التي يتم جمعها أو طرحها لها نفس القوة $m \times n$ حيث m هي عدد الصفوف وحيث n خي عدد الأعمدة أنظر الصورة التالية

$$A = \begin{pmatrix} 1 & 2 \\ 4 & 6 \\ 9 & 8 \end{pmatrix}_{3\times2}$$

$$B = \begin{pmatrix} 0 & 4 \\ 3 & 9 \\ 3 & 7 \end{pmatrix}_{3\times2}$$

كما ترى فلابد أن يتكون
المصفوفات التي يتم طرحها لها
نفس القوة
وفي المثال قوة المصفوفة هي
٣ صفوف
٢ عمود

$$A - B = \begin{pmatrix} 1 & 2 \\ 4 & 6 \\ 9 & 8 \end{pmatrix}_{3\times2} - \begin{pmatrix} 0 & 4 \\ 3 & 9 \\ 3 & 7 \end{pmatrix}_{3\times2} = \begin{pmatrix} 1 & -2 \\ 1 & -3 \\ 6 & 1 \end{pmatrix}_{3\times2}$$

لنقم الآن بعمل نفس المثال على برنامج الماتلاب أنظر الصورة التالية :

```
>> % By Defining the Matrix A
>> A=[1 2;4 6;9 8];
>> % By Defining the Matrix B
>> B=[0 4;3 9;3 7];
>> % C=A-B
>> C=A-B

C =

    1   -2
    1   -3
    6    1
```

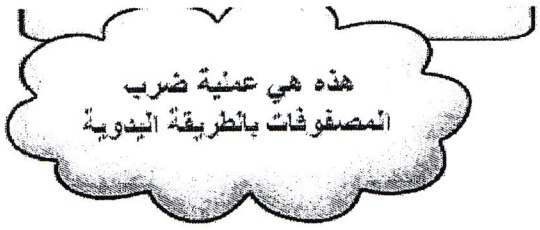
كما ترى فلقد حصلنا
على نفس الناتج السابق

ضرب المصفوفات :

شرط ضرب أي مصفوفتين هو أن يكون عدد أعمدة المصفوفة الأولى n_1 مساوياً لعدد الصفوف في المصفوفة الثانيةm_2 أنظر الصورة التالية :

$$A=\begin{pmatrix}1 & 2\\ 4 & 6\\ 9 & 8\end{pmatrix}_{3\times2}$$

$$B=\begin{pmatrix}0 & 3 & 3\\ 4 & 9 & 7\end{pmatrix}_{2\times3}$$



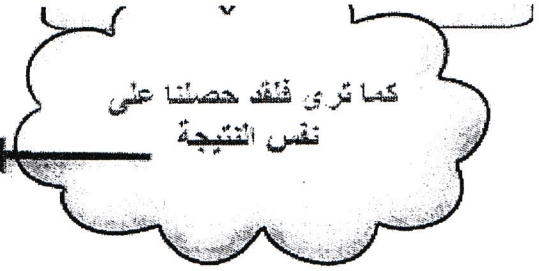
$$C=A\times B=\begin{pmatrix}1 & 2\\ 4 & 6\\ 9 & 8\end{pmatrix}_{3\times2}\times\begin{pmatrix}0 & 3 & 3\\ 4 & 9 & 7\end{pmatrix}_{2\times3}$$

$$C=\begin{pmatrix}(1\times0)+(2\times4) & (1\times3)+(2\times9) & (1\times3)+(2\times7)\\ (4\times0)+(6\times4) & (4\times3)+(6\times9) & (4\times3)+(6\times7)\\ (9\times0)+(8\times4) & (9\times3)+(8\times9) & (9\times3)+(8\times7)\end{pmatrix}_{3\times3}$$

$$C=\begin{pmatrix}8 & 21 & 17\\ 24 & 66 & 54\\ 32 & 99 & 83\end{pmatrix}_{3\times3}$$

لنقوم الآن بإدخال نفس المثال على الماتلاب أنظر الصورة التالية :

```
>> % By defining the Matrix A
>> A=[1 2;4 6;9 8];
>> % By Defining the Matrix B
>> B=[0 3 3;4 9 7];
>> % C=A*B
>> C=A*B

C =

   8   21   17
  24   66   54
  32   99   83

>>
```



قسمة المصفوفات :

قد يستغرب البعض□ن وجود كلمة القسمة للمصفوفات, ولكن الحقيقة أنها□وجودة و□ستخد□ة بكثيرة ولكننا لا ننتبه لوجودها, فبهذه القسمة نقوم بحل المعادلات والتي سيتم شرحها لاحقاً بإذن الله وقبل أن أشرح لكم كيفية عمل القسمة, لابد□ن شرح كيفية حل المعادلات كثيرة الحدود لنفترض أن لدينا□معادلتان كالآتي

$$3X + 3Y = 3$$
$$2X + 3Y = 5$$

وكلتا المعادلتان يمكن حلهما ليكون الناتج

X=-2

Y=3

فكيف يتم ذلك؟

يمكن وضع المعادلتان في صورة مصفوفة كما في الشكل التالي :

$$\begin{pmatrix} 3 & 3 \\ 2 & 3 \end{pmatrix}\begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} 3 \\ 5 \end{pmatrix}$$

المعادلتان في صورة المصفوفة

وهنا نذكر أن هنالك طريقتان لحل المعادلتان

1 - طريقة الحذف

2- قسمة المصفوفات

وسأذكر سريعاً طريقة الحذف, أنظر الصورة التالية :

By Multiplyin g by $(\frac{3}{2}\times R_2 - R_1)$

$$\begin{pmatrix} 3 & 3 \\ (\frac{3}{2}\times 2-3) & (\frac{3}{2}\times 3-3) \end{pmatrix}\begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} 3 \\ (\frac{3}{2}\times 5-3) \end{pmatrix}$$

$$\begin{pmatrix} 3 & 3 \\ 0 & 1.5 \end{pmatrix}\begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} 3 \\ 4.5 \end{pmatrix}$$

$\therefore 1.5Y = 4.5$

$\therefore Y = 3$

$\because 3X + 3Y = 3$

$\therefore 3X + (3\times 3) = 3$

$\therefore X = -2$

طريقة الحذف في حل المصفوفات

أما الطريقة الثانية هي قسمة المصفوفات لنعود إلى الصورة التالية مرة أخرى

$$\begin{pmatrix} 3 & 3 \\ 2 & 3 \end{pmatrix}\begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} 3 \\ 5 \end{pmatrix}$$

المعادلتان في صورة المصفوفة

نجد أنه يمكننا أن نضعها في الصيغة التالية

$$AX = B$$

وبالتالي من أجل الحصول على X يجب قسمة B على A كما في الصورة التالية

$$X = \frac{B}{A}$$

ولكن ماذا تعني $\frac{1}{A}$ من ناحية المصفوفات وليست الأعداد؟

$$\frac{1}{A} = inv(A)$$

Where inv( ) is the inverse function

وهذا ما يسمى قسمة المصفوفات , ولكن يشترط عند إيجاد inv أن تكون المصفوفة مربعة (أي عدد الصفوف يساوي عدد الأعمدة) وبالتالي يمكن إيجاد قيمة X & Y عن طريق وضع المعادلة في الصورة التالية, مع الأخذ في الاعتبار أن تتوفر شرط عملية الضرب بين المصفوفتين

$$\begin{pmatrix} X \\ Y \end{pmatrix} = inv\left(\begin{pmatrix} 3 & 3 \\ 2 & 3 \end{pmatrix}_{2\times2} \times \begin{pmatrix} 3 \\ 5 \end{pmatrix}_{2\times1}\right)$$

يجب الإنتباه لشرط عملية ضرب المصفوفة

استخدام المصفوفات في حل نظام المعادلات الخطية باستخدام ماتلاب:

حل نظام المعادلات الاتي:

$3x_1-40x_2-30x_3=15$.....(1)

$18x_1+11x_2+93x_3=10$..... (2)

$90x_1+5x_2-30x_3=18$.....(3)

الحل : نكتب العلاقة بالشكل التالي:

$$ax=b$$

حيث a هي المصفوفة الناتجة من مضروبات قيم x في المعادلات و b هي الشعاع العمودي الناتج من قيم المعدلات بعد اشارة المساواة:

$$\begin{pmatrix} 3 & -4 & -3 \\ 18 & 11 & 93 \\ 90 & 5 & -30 \end{pmatrix} \mathbf{X} = \begin{pmatrix} 15 \\ 10 \\ 18 \end{pmatrix}$$

نكتب البرنامج التالي في ماتلاب من اجل ايجاد قيم x :

```
a=[3 -4 -3;18 11 93;90 5 -30];
>> b=[15;10;18];
>> x=a\b
x =
   0.5485   <-------- x1
  -3.6647   <-------- x2
   0.4348   <-------- x3
```

حل نظام المعادلات التالي:

$6x_1-40x_2-30x_3+70x_4=15$.....(1)

$10x_1+11x_2+93x_3=11$..... (2)

$90x_1+5x_2-30x_3=18$.....(3)

$5x_1+14x_2+7x_3-x_4=80$.....(4)

الحل:

```
a=[6 -40 -30 70;10 11 93 0;90 5 -30 0;5 14 7 -1];
>> b=[15;11;18;80];
>> x=a\b
x =
  -0.3559
   6.4022
  -0.6007
   3.6457
```

العمليات على المصفوفات والمتجهات :

الفرق بين المتجهات والمصفوفات :

المتجهات هي مصفوفة ولكن إما بعمود واحد Column Vector أو صف واحد Row Vector فمثلاً الصورة التالية لمتجه صفي

```
>> A=[1 2 3 4 5 6 7 8 9 10]

A =

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

Row Vector

وهذه صورة لمتجه عمودي :

```
>> B=[1;2;3;4;5;6;7;8;9;10]

B =

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

Column Vector

أما المصفوفة فهي التي يزيد عدد صفوفها وأعمدتها عن صف واحد أو عمود واحد

العمليات على المتجهات :

لنقوم بتعريف متجه صفي لدى الماتلاب كما في الصورة التالية :

```
Command Window
>> A=[1 2 3 4 5 6 7 8 9 10]

A =

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

تعريف متجه صفي

```
Command Window
>> A=[1 2 3 4 5 6 7 8 9 10];
>> % It's required to get the length of A
>> length(A)

ans =

   10
```

فالمقصود بـ length هو عدد العناصر الموجودة في المتجه
وكما هو واضح أن عدد العناصر هو ١٠

والآن نقوم بالعملية الأولى وهي طول المصفوفة

يمكن عمل نفس العملية على متجه عمودي

إضافة عنصر:

```
Command Window
>> A=[1; 2; 3; 4; 5; 6; 7; 8; 9; 10]

A =

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

تعريف متجه عمودي

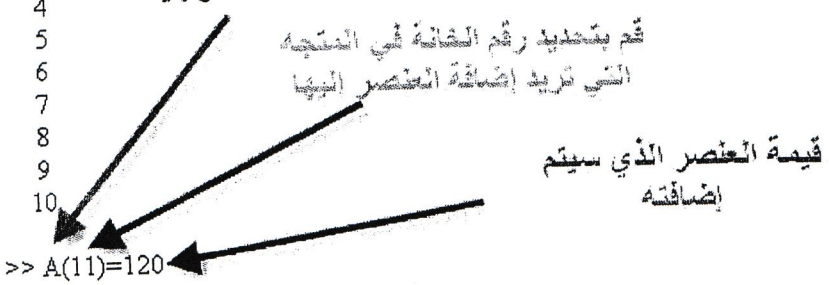
لنقوم بوضع متجه عمودي في الماتلاب, كما في الصورة التالية :

كما هو واضح, أن عدد العناصر الموجودة في هذا المتجه هو 10, وللتأكد قم بعمل الأمر length في نافذة الأوامر للماتلاب, أنظر الصورة التالية :

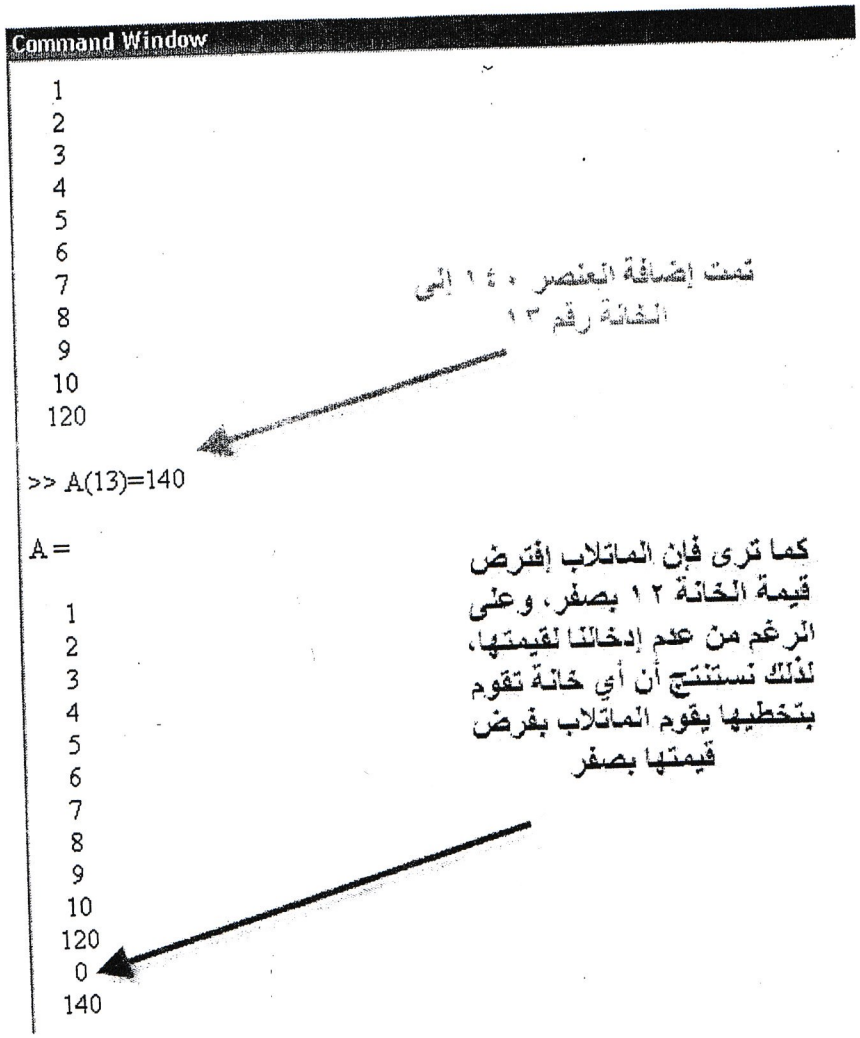
```
Command Window
>> A=[1; 2; 3; 4; 5; 6; 7; 8; 9; 10]

A =

    1
    2
    3
    4
    5
    6
    7
    8
    9
   10

>> length(A)

ans =

   10
```

لنقل أننا نريد إضافة الرقم 120 في الخانة الحادية عشرة, أي الخانة التالية للخانة العاشرة, أنظر الصورة التالية :

```
A =

     1
     2
     3
     4
     5
     6
     7
     8
     9
    10

>> A(11)=120

A =

     1
     2
     3
     4
     5
     6
     7
     8
     9
    10
   120
```



ملاحظة: في المثال السابق تمت إضافة الرقم 120 إلى الخانة 11, فماذا إذا قمنا بإضافة رقم جديد ولكن في الخانة رقم 13, فماذا ستكون قيمة الخانة 12 التي لم يتم إضافة أي عنصر لها, أنظر الصورة التالية



إضافة أكثر من عنصر متتالي :

لنفترض أننا نريد إضافة مجموعة من العناصر المتتالية في الخانات 11و 12 و 13 ويمكن بدلاً من إدخال كل رقم على حدى,كما في الصورة التالية :

```
>> A=[1; 2; 3; 4; 5; 6; 7; 8; 9; 10];
>> A(11)=11;
>> A(12)=12;
>> A(13)=13;
>> A

A =

    1
    2
    3
    4
    5
    6
    7
    8
    9
   10
   11
   12
   13
```

تم إضافة ثلاثة عناصر فقط
في المتجه

ولكن قد يبدوا ذلك مستنفذاً للوقت, إذا تم إدخال 100 رقم متتالي أو 1000 رقم, فما العمل؟ هنالك طريقة في الماتلاب تستخدم إذا أردت أن تضيف مجموعة من الأرقام المتتالية فمثلاً عندما نريد أن نذكر مجموعة من الأرقام المتتالية من 1 إلى 10 نكتب التالي1:10 وعندما نريد كتابة مجموعة من الأرقام المتتالية من 10 إلى 1200 نكتب 10:1200 وبالتالي إذا أردنا كتابة مجموعة من الأرقام المتتالية من 11 إلى 13 كما في مثالنا نكتب 11:13 وبالتالي تكون الكتابة في الماتلاب كما في الصورة التالية

```
Command Window
>> A=[1; 2; 3; 4; 5; 6; 7; 8; 9; 10];
>> A(11:13)=[11 12 13]

A =

    1
    2
    3
    4
    5
    6
    7
    8
    9
   10
   11
   12
   13
```

تم تحديد الخانات المتتالية
من ١١ إلى ١٣

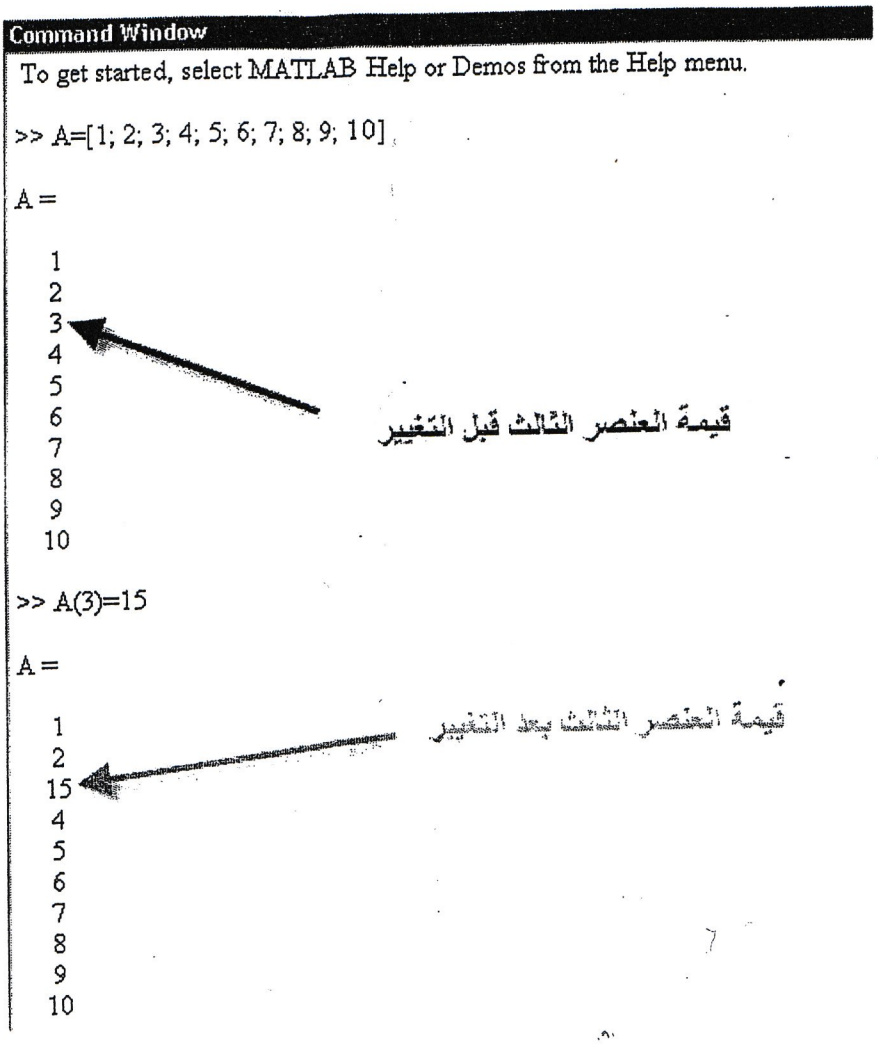
استبدال عنصر :

عملية إستبدال عنصر تتطلب عدة شروط :

1 - أن يكون العنصر موجوداً بالفعل

2 - أن تحدد مكان هذا العنصر

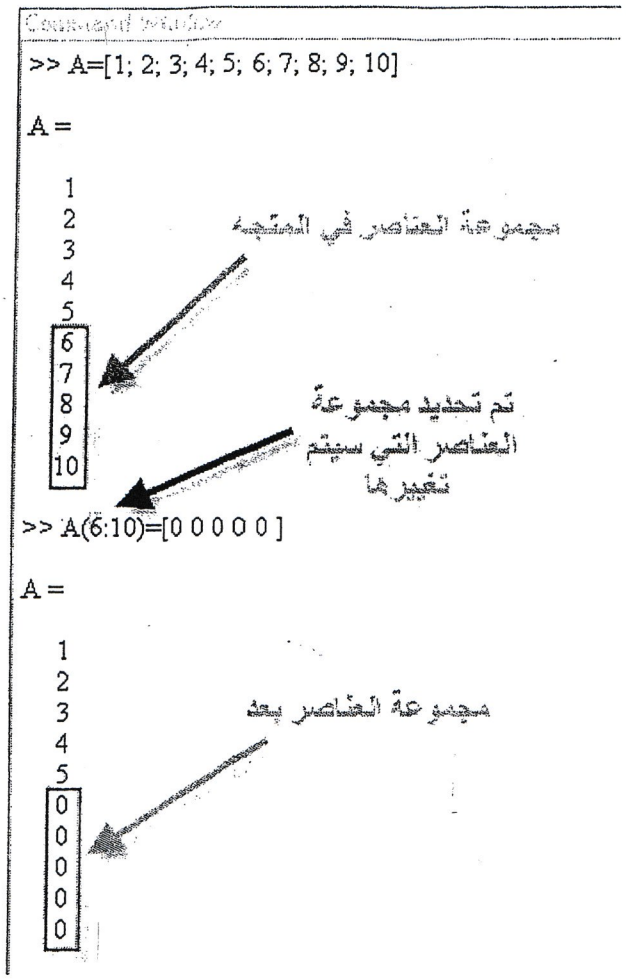
ففي المثال التالي أردنا أن تستبدل العنصر الثالث بدلاً من الرقم 3 إلى الرقم 15 كل ما علينا فعله هو كتابة التالي A(3)=15 حيث A هي المتجه الذي يحتوى العنصر الذي تريد تغييره



وعلى هذا المنوال تستطيع أن تغير أي عنصر في المتجه

استبدال مجموعة عناصر متتالية :

كما شرحنا كيفية إضافة مجموعة عناصر متتالية, سنقوم باستبدال مجموعة عناصر متتالية كما في الصورة التالية :



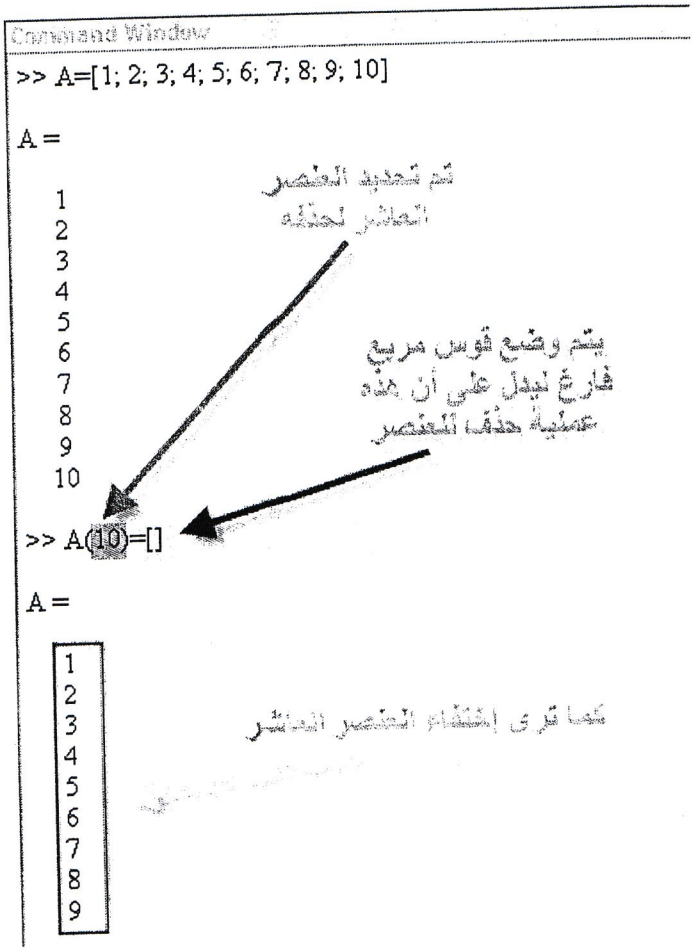
حذف عنصر من المتجه :

لتقوم بحذف عنصر من المتجه يجب أن يتوفر الشرطان التاليان

1 - تحديد العنصر الذي تريد حذفه

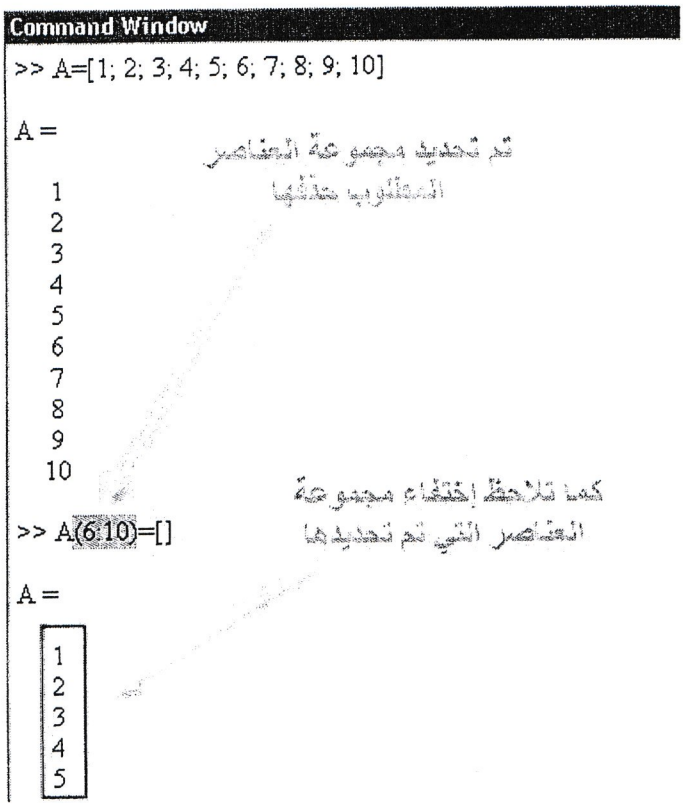
2 - وضع أقواس مربعة Square Brackets خالية من أي رقم

فالمثال التالي يوضح أننا نريد حذف العنصر في الخانة العاشرة, أنظر الصورة التالية :



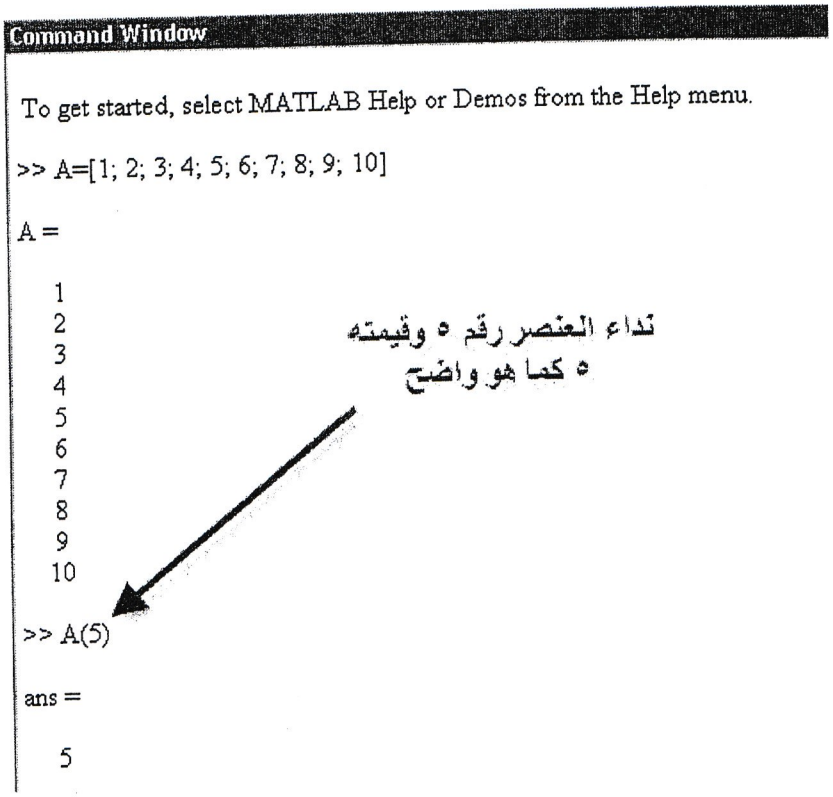
حذف مجموعة عناصر متتالية :

لحذف مجموعة عناصر متتالية, أنظر الصورة التالية :



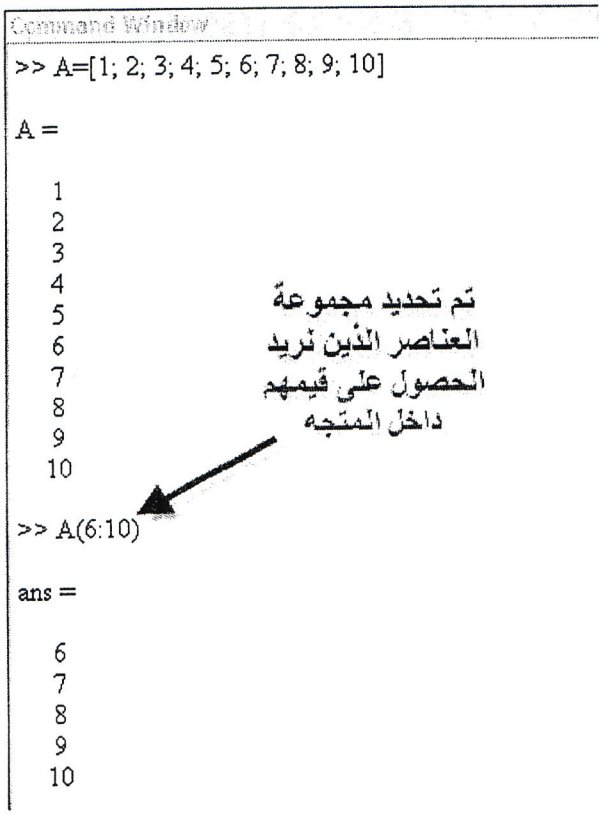
نداء عنصر :

نداء عنصر المقصود به هو الحصول على قيمة العنصر في أي مكان من المتجه ويمكن ذلك من خلال كتابة التالي :



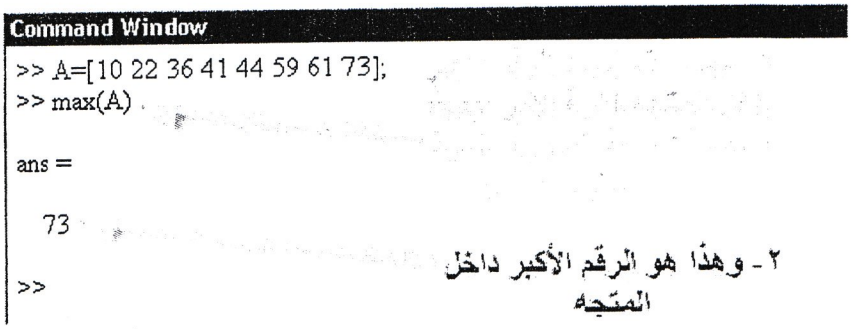
نداء أكثر من عنصر :

للحصول على قيم مجموعة عناصر محددة من متجه, قم بعمل الآتي على نافذة الأوامر Command Window



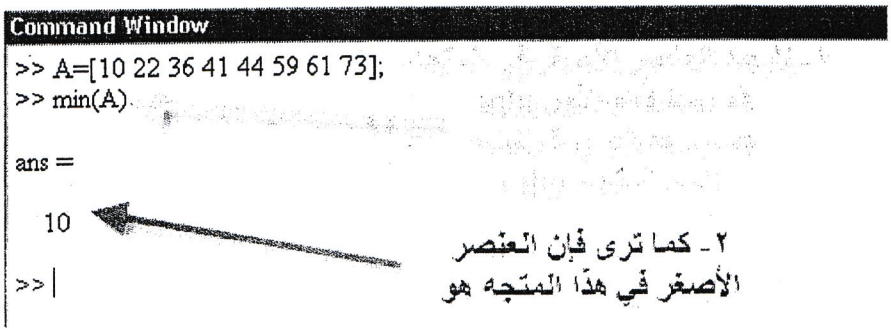
إيجاد العنصر الأكبر في المتجه :

لإيجاد العنصر الأكبر في متجه, يتم استخدام الأمر ,max حيث يمكن استخدامه في الماتلاب بالشكل التالي :



إيجاد العنصر الأصغر في المتجه :

لإيجاد العنصر الأصغر في المتجه, يجب استخدام الأمر min وهي اختصار لدى الماتلاب وهي اختصار لكلمة minimum أي الأقل ولإيجاد العدد الأصغر داخل المتجه في الماتلاب قم بعمل الآتي :

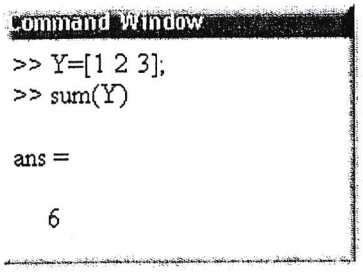


إيجاد مجموع عناصر المتجه :

يمكن جمع جميع عناصر المتجه, بإستخدام الأمر sum حيث أن هذا الأمر لابد أن يأخذ طريقة في تنفيذه فيجب أن ينفذ بالصورة التالية

Sum (إسم المتجه)

ولنقوم بعمل مثال في الماتلاب الآن



إيجاد حاصل ضرب العناصر في المتجه :

يوفر الماتلاب خاصية ضرب عناصر المتجه, وذلك باستخدام الأمر prod وهو اختصار product ويجب أن يأخذ هذا الأمر الصورة التالية في كتابته

prod (إسم المتجه)

والآن لنأخذ مثالاً تطبيقياً في الماتلاب

```
Command Window
>> Y=[1 2 3 4];
>> prod(Y)

ans =

   24
```

:

العمليات على المصفوفات

إيجاد حجم المصفوفة :

لإيجاد حجم المصفوفة أو دعونا نقول لإيجاد عدد الصفوف والأعمدة لمصفوفة, يجب استخدام الأمر size, حيث لا يصلح استخدام الأمر length,أمر length يستخدم في المتجهات وليس في المصفوفات, ولتوضيح الأمر دعونا نقوم بعمل مثال مبسط لشرح هذا الأمر, أولاً لنقوم بعمل مصفوفة غير منتظمة (أي أن عدد الصفوف لا يساوي عدد الأعمدة) كما في الشكل التالي

```
>> A=[3 4 9;2 4 5]

A =

   3   4   9
   2   4   5
```

أولا لنقوم بكتابة الأمر size لمعرفة حجم المصفوفة

```
>> A=[3 4 9;2 4 5]

A =

   3   4   9
   2   4   5

>> size(A)

ans =

   2   3
```

الأمر size

عدد الصفوف

عدد الأعمدة

أما إذا أردنا أن نعرف عدد الصفوف فقط نقوم بعمل الآتي

```
>> size(A,1)

ans =

   2
```

أما إذا أردنا أن نعرف عدد الأعمدة فقط نقوم بكتابة التالي :

```
>> size(A,2)

ans =

     3
```

إضافة عنصر إلى المصفوفة :

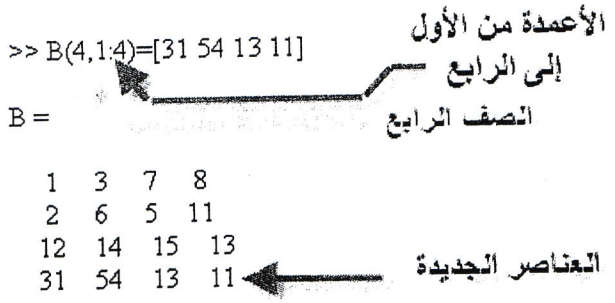
عملية إضافة عنصر أو عدة عناصر هي من العمليات الهامة جداً داخل الماتلاب, ودائماً نقوم باستخدامها في الكثير من البرامج المتقدمة كما سيتضح فيما بعد, ولتوضيح ذلك الأمر يجب أن نقوم بإعطاء مثال حتى تصل مرحلة الفهم التام لها لنقوم أولاً بتعريف مصفوفة في الماتلاب

```
>> B=[1 3 7 8; 2 6 5 11; 12 14 15 13]

B =

     1     3     7     8
     2     6     5    11
    12    14    15    13
```

لنفترض أننا نريد أن نقوم بوضع رقم 42 في الصف الثاني والعمود الخامس ,نقوم بكتابة التالي في الماتلاب

```
>> B(2,5)=42

B =

     1     3     7     8     0
     2     6     5    11    42
    12    14    15    13     0
```

كما تلاحظ فإن الصف الأول والصف الثالث للعمود الخامس, لم يتم وضع قيم بهما , لذلك قام الماتلاب بافتراضهما صفراً. فماذا إذا أردنا إضافة عدة عناصر في المصفوفة؟ يمكن إيضاح ذلك باستخدام المثال التالي لنقوم أننا نريد إضافة الأعداد31 و 54 و 13 و 11 في الصف الرابع و العمود الأول الثاني والثالث و الرابع على التوالي, يمكن ذلك من خلال الماتلاب بالشكل التالي



استبدال عنصر:

قد تكون هذه العملية نادراً ما يتم استخدامها, ولكنها هامة جداً, حيث توفر إمكانية استبدال عنصر أو عدة عناصر داخل المصفوفة, ولتوضيح هذه الخاصية, سنقوم بتعريف مصفوفة كما ذكرنا مسبقاً

```
>> B=[1 3 7 8; 2 6 5 11; 12 14 15 13]

B =

   1   3   7   8
   2   6   5   11
   12  14  15  13
```

ولنقوم باستبدال العنصر في الصف الثالث والعمود الأول إلى الرقم صفر

```
>> B(3,1)=0

B =

   1   3   7   8
   2   6   5   11
   0   14  15  13
```

وإذا أردنا استبدال عدة عناصر, يمكن ذلك بعمل مثال بسيط, لنقل أننا نريد أن نستبدل الصف الأول والثاني والعمودين من الأول إلى الثالث بقيمة صفر

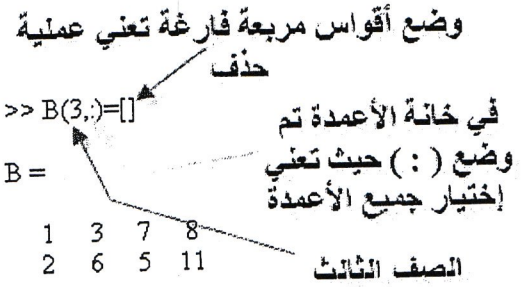
```
>> B(1:2,1:3)=0

B =

   0   0   0   8
   0   0   0   11
   12  14  15  13
```

حذف أكثر من عنصر :

لا يقوم الماتلاب بعملية حذف لعنصر واحد فقط في مصفوفة, حيث أنه من غير المعقول حذف عنصر من داخل المصفوفة, وبقية الصف والعمود بهم قيم, ولكن إذا أردت أن تقوم بحذف صف كامل أو عمود كامل فيمكن ذلك بعمل التالي نقوم أولاً بعمل مصفوفة للعمل عليها

```
>> B=[1 3 7 8; 2 6 5 11; 12 14 15 13]

B =

    1    3    7    8
    2    6    5   11
   12   14   15   13
```

لنقل اننا نريد حذف الصف الثالث كله



ولحذف العمود الرابع كله, قم بعمل التالي :

```
>> B(:,4)=[]

B =

    1    3    7
    2    6    5
   12   14   15
```

نداء عنصر :

عملية نداء عنصر من أكثر العمليات هامة جداً داخل الماتلاب, أي أنه نود الحصول على عنصر وحيد من المصفوفة, وذلك بذكر رقم الصف ورقم العمود الذي به هذا العنصر , ولتوضيح هذا الأمر, نقوم بعمل مثال بسيط, معتمدين على نفس المصفوفة التي تم ذكرها في المثال السابق

```
>> B=[1 3 7 8; 2 6 5 11; 12 14 15 13]

B =

   1   3   7   8
   2   6   5   11
  12   14   15   13
```

لنقل اننا نريد العنصر في الصف الأول والعمود الثالث

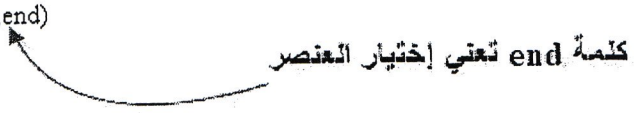
```
>> B(1,3)

ans =

   7
```

ولنداء أكثر من عنصر, نقوم مثلاً بنداء الصف الثاني ومن العمود الثاني إلى الرابع

```
>> B(2,2:4)

ans =

   6   5   11
```

هذا في حالة أننا نعرف حجم المصفوفة, ولكن ماذا إذا لم نكن نعرف حجمها, ونريد أن نحصل على العنصر الأخير مثلاً من الصف الثاني

```
>> B(2,end)
ans =
    11
```



إيجاد العنصر الأكبر :

يقوم الماتلاب بإيجاد العنصر الأكبر عن طريق العمل على المصفوفة بشكل مختلف, فكيف يبحث عن العنصر الأكبر في المصفوفة, يقوم الماتلاب بالبحث عن العنصر الأكبر في كل عمود في المصفوفة, وبعدما يقوم بعمل ذلك, يقوم بعمل متجه به الرقم الأكبر من كل عمود, أنظر المثال التالي للتوضيح لدينا الآن مصفوفة تم إنشائها على الماتلاب

```
A =
     1    15     2    11
    23     1     4     5
     3     1    15     7
     1     4     9    10
```

ولنقم بكتابة الأمر max كما ذكرنا مسبقاً

```
>> B=max(A)
B =
    23    15    15    11
```

كما تلاحظ فلقد قام الماتلاب باختيار العنصر الأكبر من كل عمود, ولاختيار الرقم الأكبر بينهم يجب كتابة نفس الأمر للناتج الخارج, وبالتالي نحصل على الرقم الأكبر في المصفوفة ككل

```
>> C=max(B)
C =
    23
```

إيجاد العنصر الأصغر :

هذه العملية أيضاً كثيرة الاستخدام في التطبيقات المختلفة, وهي نفس الخطوات السابق ذكرها في إيجاد العنصر الأكبر ولكن يتم استخدام الأمرmin وإليكم المثال التالي :

```
>> A=[1 15 2 11; 23 1 4 5; 3 1 15 7; 1 4 9 10]

A =

   1   15    2   11
  23    1    4    5
   3    1   15    7
   1    4    9   10

>> B=min(A)

B =

   1    1    2    5

>> C=min(B)

C =

   1
```

إيجاد مجموع العناصر :

لإيجاد المجموع كما تعلمنا نقوم باستخدام الأمر sum ولكن عملية الجمع يقوم الماتلاب بإيجاد جمع كل عمود على حدا وتوضع في صورة متجه, كما في المثال التالي

```
>> A=[1 15 2 11; 23 1 4 5; 3 1 15 7; 1 4 9 10]

A =

   1   15    2   11
  23    1    4    5
   3    1   15    7
   1    4    9   10

>> B=sum(A)

B =

   28   21   30   33

>> C=sum(B)

C =

  112
```

إيجاد حاصل ضرب العناصر :

يمكن ضرب عناصر المصفوفة, ولكن في الماتلاب عملية الضرب تكون لكل عمود على حدا ويتم وضع الناتج في متجه, وإذا تم استخدام الأمر مرة أخرى يتم ضرب عناصر المتجه جميعها ,لينتج حاصل الضرب المصفوفة جميعها, أنظر المثال التالي

```
>> A=[1 15 2 11; 23 1 4 5; 3 1 15 7; 1 4 9 10]

A =

   1   15    2   11
  23    1    4    5
   3    1   15    7
   1    4    9   10

>> B=prod(A)

B =

        69        60      1080      3850

>> C=prod(B)

C =

  1.7214e+010
```

إيجاد قطر المصفوفة :

هذه العملية قد تكون ذات استخدام أكاديمي, ولكنها هامة جداً, وخصوصاً أن تلك الخاصية تخدم المصفوفة المربعة (عدد الصفوف يساوي عدد الأعمدة), ويتم استخدام الأمر ,diag وهذا مثال لذلك

```
>> % By defining the Square Matrix A
>> A=[1 15 2 11; 23 1 4 5; 3 1 15 7; 1 4 9 10]

A =

   1  15   2  11
  23   1   4   5
   3   1  15   7
   1   4   9  10

>> % By Getting the Diagonal of the Matrix A
>> B=diag(A)

B =

   1
   1
  15
  10
```

يمكننا الآن عمل العديد من العمليات على قطر المصفوفة, فمثلاً نريد الحصول على عملية الجمع لعناصر المصفوفة

```
>> A=[1 15 2 11; 23 1 4 5; 3 1 15 7; 1 4 9 10]

A =

   1  15   2  11
  23   1   4   5
   3   1  15   7
   1   4   9  10

>> B=sum(diag(A))

B =

  27
```

أو أننا نريد الحصول على حاصل ضرب تلك العناصر

```
>> A=[1 15 2 11; 23 1 4 5; 3 1 15 7; 1 4 9 10]

A =

    1   15    2   11
   23    1    4    5
    3    1   15    7
    1    4    9   10

>> B=prod(diag(A))

B =

   150
```

العمليات الأساسية والدوال الخاصة بالمتجهات:

هناك العديد من الدوال التي يتم تنفيذها على المتجهات وتزيد من أهميتها واستخداماتها وسوف نقوم الآن بشرح معظم هذه العمليات والدوال من خلال الأمثلة التالية:

١. الدالة Length: تقوم بحساب عدد عناصر المتجه كما في المثال:

```
>> v=[2 5 0 1 4 -1]

v =
   2   5   0   1   4  -1
>> length(v)

ans =
   6
```

٢. الدالة Sum: تقوم هذه الدالة بإيجاد حاصل جمع عناصر المتجه كما في المثال:

```
>> w=sum(v)

w =
   11
```

٣. الدالة Max:تقوم هذه الدالة بإيجاد أكبر عناصر المتجه من حيث القيمة كما في المثال:

```
>> w=max(v)

w =
   5
```

٤. الدالة Min: تقوم هذه الدالة بإيجاد أصغر عناصر المتجه من حيث القيمة كما في المثال:

```
>> w=min(v)

w =
   -1
```

٥. الدالة Sort:تقوم هذه الدالة بترتيب عناصر المتجه ترتيباً تصاعدياً

```
>> r=[9 7 5 8 3]

r =
   9   7   5   8   3
>> s=sort(r)
```

```
s =
   3    5    7    8    9
```

٦. الدالة Range: تقوم هذه الدالة بحساب الفرق بين أكبر قيمة في المتجة وأصغر قيمة فيه

```
>> range(r)

ans =
   6
```

العمليات الحسابية التي يتم إجراؤها على المتجهات: وتشمل هذه العمليات الحسابية عمليات الجمع والطرح والضرب والرفع إلى أس ولكن يجب الإشارة هنا أن هذه العمليات تتبع جميعها ما يسمى بجبر المصفوفات.

بعض الأمثلة للتوضيح:

```
>> x=[1,3,5];
>> y=[2,4,6];
>> z=x+y
z =
   3    7   11
>> m=y-x
m =
   1    1    1
>> p=x.*y
p =
   2   12   30
>> p=x.^2
p =
   1    9   25
```

??

❖ **المصفوفات Matrices:**

المصفوفات هي عبارة عن ترتيب معين لبيانات معينه وعادة ما تكون هذه البيانات أرقاماً، والمصفوفة تتكون من صفوف وأعمدة وعادة ما نقول من النظام (mxn) حيث أن m هو عدد الصفوف و n هو عدد الأعمدة.

```
>> Matrix=[1,2,3,;4,5,6;7,8,9]

Matrix =

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

كذلك إذا كان لدينا مصفوفة فأننا نستطيع إيجاد الصف الثاني أو الثالث من المصفوفة.

```
>> Matrix(2,:)

ans =
```

```
   4    5    6
```

وكذلك نستطيع إيجاد العمود الثاني أو الثالث من المصفوفة.

```
>> Matrix(:,2)

ans =
    2
    5
    8
```

إذ أردنا جميع عناصر المصفوفة بترتيب الأعمدة

```
>> Matrix(:)

ans =
    1
    4
    7
    2
    5
    8
    3
    6
    9
```

أما إذا أردنا العنصر الواقع في الصف الأول والعامود الثاني:

```
>> Matrix(1,2)

ans =
    2
```

ونحذف صف أو عمود من المصفوفة:

```
>> Matrix(:,2) = [ ]
Matrix =
    1    3
    4    6
    7    9
>> Matrix(2,:) = [ ]
Matrix =
    1    2    3
    7    8    9
```

ونضيف صف أو عمود للمصفوفة:

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

```
   7    8    9
  10   11   12
```

ونجد قطر المصفوفة:

```
>> diag(Matrix)
ans =
   1
   5
   9
```

➢ منقول المصفوفة (Transpose):

لتكن $A=[a_{ij}]$ مصفوفة من الدرجة $n\times m$ يعرف المنقول للمصفوفة A بأنه المصفوفة من الدرجة $m\times n$ التي نحصل عليها من A بحيث تكون صفوفها هي أعمدة A وأعمدتها هي صفوف A على التوالي نرمز للمنقول A بالرمز A^T .

```
>> A=[1 3 5; 2 4 6]

A =

   1    3    5
   2    4    6

>> A'

ans =

   1    2
   3    4
   5    6
```

➢ المحددات: لتكن $A=[a_{ij}]$ مصفوفة مربعة من الدرجة n يعرف محدد المصفوفة ويرمز له بالرمز $\det(A)$ استقرائياً كالتالي:

١. إذا كان $n=1 \Leftarrow \det(A)=a_{11}$

٢. إذا كان $n=2 \Leftarrow \det(A)=a_{11}a_{22}-a_{12}a_{21}$

٣. إذا كان $n>2 \Leftarrow \det(A)=\sum_{j=1}^{n}(-1)^{j+1}a_{1j}\det A_{1j}$

مثال يوضح المحددات:

```
>> A=[1 0 3 ; 4 5 0; 7 8 9]

A =

   1    0    3
   4    5    0
   7    8    9
```

```
>> det(A)
ans =
   36
```

وهنا يجب الإشارة إلى بعض أنواع المصفوفات ذات الحالات الخاصة التي سوف نوضحها فيما يلي:

١. المصفوفة الصفرية: وهي التي تكون كل عناصرها عبارة عن أصفار وتعتبر هذه المصفوفة هي المحايد الجمعي للمصفوفات.

```
>> x=zeros(3,2)

x =

    0    0
    0    0
    0    0
```

٢. مصفوفة التي جميع عناصرها الواحد الصحيح: وهي المصفوفة التي تتكون جميع عناصرها من الرقم واحد.

```
>> x=ones(3,2)

x =

    1    1
    1    1
    1    1
```

٣. مصفوفة الوحدة : وهي مصفوفة مربعة تكون جميع عناصر القطر الرئيسي لها الواحد الصحيح وباقي عناصرها الأخرى أصفار.

```
>> id=eye(4)

id =

    1    0    0    0
    0    1    0    0
    0    0    1    0
    0    0    0    1
```

(٤-١)جبر المصفوفات Matrix Algebra :

يعتمد جبر المصفوفات على قواعد غير القواعد المعهودة في العمليات الحسابية العادية التي يتم تطبيقها على الأعداد، وسوف نحاول فيما يلي توضيح هذه القواعد بقدر الإمكان:

- الدوال الخاصة بالمصفوفات:

١. دالة Sum: وهي تقوم بجمع عناصر كل عمود من أعمدة المصفوفة كل على حدة كما في المثال:

```
>> x=[1,2,3;4,5,6;7,8,9]

x =

    1    2    3
    4    5    6
    7    8    9

>> A=sum(x)

A =

   12   15   18

>> A=sum(x')

A =

    6   15   24
```

٢.الدالة Max: وهي تقوم بعرض أكبر رقم موجود في كل عمود من أعمدة المصفوفة كما في المثال:

```
>> B=max(x)

B =

    7    8    9

>> B=max(x')

B =
    3    6    9
```

٣.الدالة Size: تقوم هذه الدالة بعرض أبعاد المصفوفة كما في المثال :

```
>> [C,D]=size(x)

C =

    3


D =

    3
```

➢ إجراء العمليات الحسابية على المصفوفات:

١. الجمع:تتم عملية الجمع بجمع كل عنصر من عناصر المصفوفة الأولى مع العنصر المناظر له من عناصر المصفوفة الثانية كما في المثال:

```
>> A=[1,3;5,7];
>> B=[2,4;6,8];
>> C=A+B

C =

   3   7
  11  15

>> C=A+3

C =

   4   6
   8  10
```

٢. الطرح: تتم عملية الطرح بطرح كل عنصر من عناصر المصفوفة الأولى مع العنصر المناظر له من عناصر المصفوفة الثانية كما في المثال:

```
>> C=A-B

C =

  -1  -1
  -1  -1
```

٣. الضرب: تتم عملية الضرب بضرب عناصر المصفوفة ببعض كما في المثال:

```
>> C=A*B

C =

  20  28
  52  76
```

٤. رفع المصفوفة إلى قوة(أس): كما يمكننا رفع المصفوفة المربعة إلى أس أو قوة كما في المثال:

```
>> C=A^2

C =

  16  24
  40  64

>> C=A.^2
```

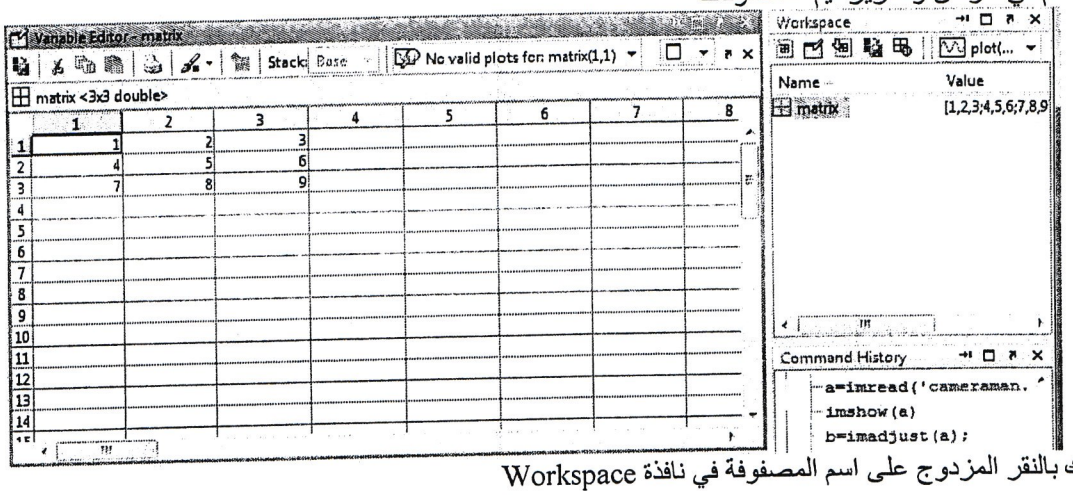
C =

1 9
25 49

??

محرر المصفوفة: array editor

يستخدم في عرض وتحرير قيم المصفوفات



وذلك بالنقر المزدوج على اسم المصفوفة في نافذة Workspace

أوامر الإدخال والإخراج:

١ - أمر الإدخال input

النوع الأول

```
x=input('enter your age:')
enter your age:38
x=
٣٨
```

النوع الثاني:

```
y=input('enter your name: ','s')
enter your name: Mohamed beelo
y =
Mohamed beelo
```

٢ - أوامر الإخراج disp , display

مثال

```
>> x=9;
>> disp(x)
    9

>> display(x)
x =
    9
```

(50)

٤,١ عمليات أخرى على المصفوفات

لتكن المصفوفة D :

$$\begin{matrix} 3 & 4 \\ 2 & 0 \end{matrix}$$

- إيجاد منقول مصفوفة (استبدال الأعمدة مع الأسطر)...

T = D′ فتكون المصفوفة الناتجة T

$$\begin{matrix} 3 & 2 \\ 4 & 0 \end{matrix}$$

- إيجاد مقلوب مصفوفة ... S = inv(D) فتكون المصفوفة الناتجة S

$$\begin{matrix} 0 & 0.5 \\ 0.25 & -0.375 \end{matrix}$$

- إيجاد رتبة مصفوفة ... V = rank(D)

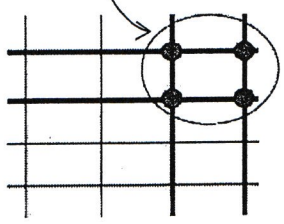
ans = 2

٥,١ عمليات متقدمة على المصفوفات

١,٥,١ استعمال مؤشر المصفوفة مع إحداثيات من عناصر المصفوفة

لتكن المصفوفة C

$$\begin{matrix} 1 & 2 & 5 & 6 \\ 1 & 6 & 0 & 9 \\ 4 & 7 & 8 & 10 \\ 2 & 4 & 5 & 6 \end{matrix}$$



- استدعاء مصفوفة جزئية من المصفوفة C (كما هو مبين بالشكل)...

c(1:2,3:4)

ans =
5 6
0 9

c(3:-1:1,:)

ans =
4 7 8 10
1 6 0 9
1 2 5 6

- استبدال ترتيب أعمدة (أو أسطر) المصفوفة مع قيم عناصرها ...

c(:,[3,4,3,4])

ans =
5 6 5 6
0 9 0 9
8 10 8 10
5 6 5 6

c([3,1,3,2],:)

ans =
4 7 8 10
1 2 5 6
4 7 8 10
1 6 0 9

٢,٥,١ التعامل مع المصفوفة وتغيير شكلها وتعيين حجمها

لتكن المصفوفة k

1 3 12 10
5 4 3 4
7 0 10 8

- k(:)'

ans = 1 5 7 3 4 0 12 3 10 10 4 8

- إعادة تشكيل المصفوفة بحيث عدد الأسطر (m) وعدد الأعمدة (n) .

reshape (matrix name ,m,n)

مثال :

reshape(k,4,3)

ans =
1 4 10
5 0 10
7 12 4
3 3 8

reshape(k,2,6) ans = 1 7 4 12 10 4
5 3 0 3 10 8

- تدوير مصفوفة ...

rot90(matrix name)

مثال :

rot90(k) ans = 10 4 8
12 3 10
3 4 0
1 5 7

- تعيين حجم مصفوفة ...

size (matrix name)
length(matrix name)

مثال :

size(k) ans = 3 4
length(k) ans = 4

٣,٥,١ أنواع من ضرب المصفوفات

1
2 1 2 3
3

لتكن المصفوفتان a السطرية وb العمودية

a*b ans = 14

حيث تم ضرب العمود الأول مع السطر الأول والعمود الثاني مع السطر الثاني وهكذا ...

b*a ans = 1 2 3
2 4 6
3 6 9

حيث تم ضرب عناصر كل سطر من b مع سطر a ...

ملاحظة:

لجعل العملية عملية ترتيبية أي معاملة كل عنصر بالعنصر المناظر له فقط ، يجب وضع نقطة قبل رمز العملية المراد إجراؤها سواء كانت قسمة أو ضرب أو رفع لقوة.

لتكن المصفوفتان : $a = \begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix}$ $b = \begin{matrix} 3 & 5 \\ 2 & 1 \end{matrix}$

- الضرب

a.*b $\quad ans = \begin{matrix} 3 & 10 \\ 6 & 4 \end{matrix}$

- القسمة

a./b $\quad ans = \begin{matrix} 0.3333 & 0.4000 \\ 1.5000 & 4.0000 \end{matrix}$

- الرفع لقوة

a.^b $\quad ans = \begin{matrix} 1 & 32 \\ 9 & 4 \end{matrix}$

ملاحظات:

- إنشاء مصفوفة متزايدة خطياً معلوم عدد عناصرها ...

linspace(a,b,c)

حيث : a القيمة البدائية b القيمة النهائية c عدد عناصر المصفوفة

- إنشاء مصفوفة متزايدة لوغاريتمياً معلوم عدد عناصرها ...

logspace(a,b,c)

حيث : a القيمة البدائية b القيمة النهائية c عدد عناصر المصفوفة

الاوامر التي تستخدم للتحكم بسير البرنامج:

عبارة if If statement

تستخدم للتحكم بسير البرنامج بالاعتماد على نتيجة فحص شرط او مجموعة شروط وهي تكون بثلاث صيغ:

1 - عبارة if البسيطة:

```
>>If expression 1
>>Statement group 1
>>end
```

مثال:

```
clc;
clear all;
a=6;
b=6;
if a>b
disp('a greater than b')
end
if a<b
disp('a less than b')
end
if a==b
disp('a equal to b')
end
```

2 - عبارة if المتداخلة :

```
clc;
clear all;
a=6;
b=6;
c=5;
d=7;
if a==b
if c<d
disp('a equal to b and c less than d')
end
end
```

3- عبارة elseif :

```
clc;
clear all;
a=8;
b=16;
c=a^2-b^2;
if a==b
disp('c equal to zero')
elseif a>b
disp('c greater than zero')
else
disp('c less than zero')
end
```

عبارة التكرار for statement

تستخدم لتنفيذ امر او مجموعة من الاوامر مرات متعددة ويتم التحكم بعدد مرات تنفيذ الامر اومجموعة الاوامر باستخدام عداد تحدد له قيمتي البداية والنهاية وكذلك (الخطوة step) وتكون الصيغة العامة لهذا الامر هي:

```
for i=k:m:n
Statement(s)
end
```

حيث: i يمثل العداد

k قيمة بداية العداد

m مقدار الخطوة

n قيمة نهاية العداد

طريقة العمل لهذا الامر تكون بان تعطى قيمة البداية k للعداد i وينفذ الامر او مجموعة الاوامر ثم بعد ذلك تزداد قيمة العداد بمقدار الخطوة m ثم ينفذ الامر او مجموعة الاوامر مرة اخرى، وتستمر العملية الى القيمة النهائية n والتي عندها يستمر عندها بالتنفيذ ثم يتوقف تنفيذ البرنامج. اذا كانت الخطوة تساري 1، فيمكن عدم كتابة الخطوة ونكتفي بتحديد قيمتي البداية والنهاية للعداد وتكون صورة الامر لهذه الحالة بالشكل التالي:

```
for i=k:n
Statement(s)
end
```

مثال: استخدم m-file لكتابة البرنامج التالي

```
s=0;
for i=1:1:10
    s=s+1
end
```

الناتج هوطباعة الارقام من 0-10 في نافذة الاوامر.

ملاحظة: بالامكان استخدام العد التنازلي مع وضع الخطوة (1-) كما في المثال التالي

```
s=0;
for i=10:-1:1
    s=s+1
end
```

مثال: استخدم m-file لكتابة البرنامج التالي

```
s=0;
for i=1:1:10
    s=s+i
end
```

الناتج هوطباعة الارقام الناتجة من جمع قيم s مع قيم i في نافذة الاوامر.

s=1, s=3 , s=6, ...s=66

مثال استخدام حلقتين متداخلتين

في هذه الحالة يتم استخدام الحلقة الخارجية للعداد i والحلقة الداخلية للعداد j .

المثال التالي يوضح طباعة عدة مصفوفات تبدأ ب 1 وتنتهي ب مصفوفة تمثل جدول الضرب .

```
for i=1:5
    for j=1:5
     A(i,j)=i*j
    end
end
```

```
 1   2   3   4   5
 2   4   6   8  10
 3   6   9  12  15
 4   8  12  16  20
 5  10  15  20  25
```

مثال:

```
for i=1:5
    for j=1:5
     A=i*j
    end
end
```

الناتج يظهر بشكل عمود لجميع القيم ابتداء من ضرب 1 في جميع قيم j وانتهاءا بضرب 5 في جميع قيم j.

البرمجة في برنامج MATLAB.

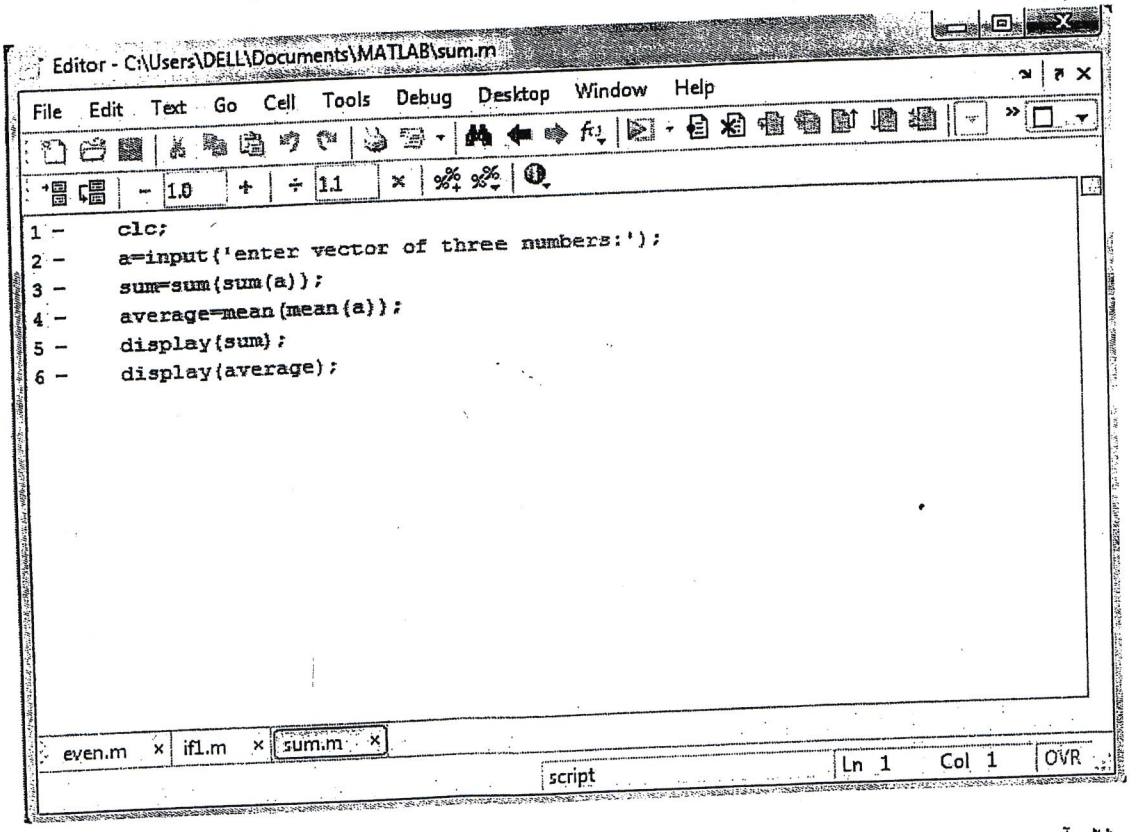
ملف m-file

الملفات التي تحتوي على كود حاسوبي (Computer code) تسمى m-files. هنالك نوعين من m-files: ملفات النص المكتوب، وملفات الدوال.

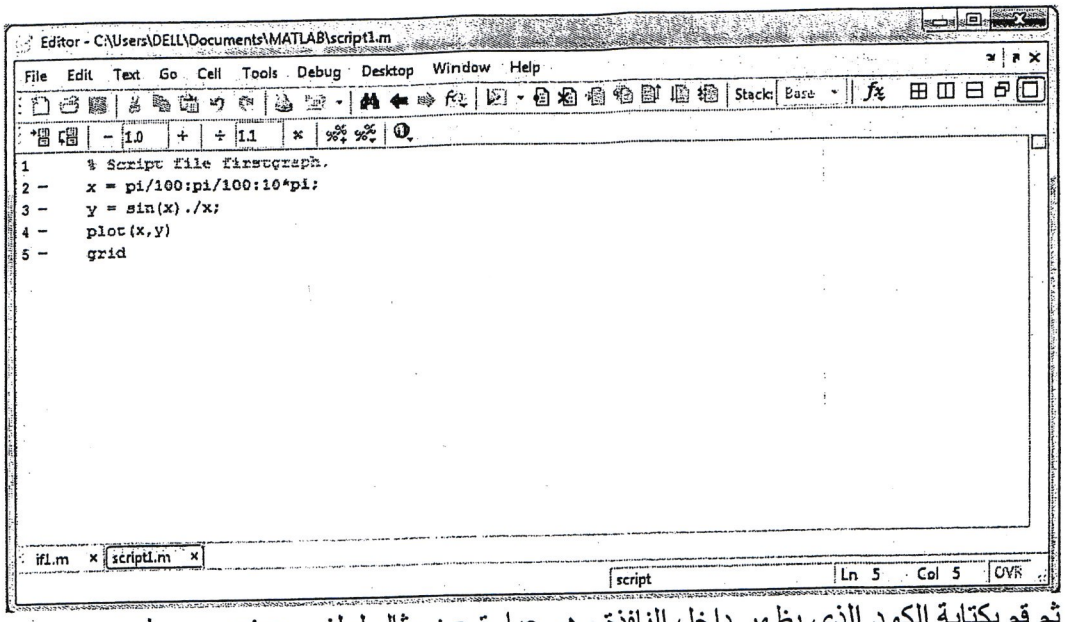
وملف الـ m-file هو وسيلة لإدخال الأوامر بدون استخدام نافذة الأوامر، والتي تعطي القدرة على كتابة البرنامج كاملاً بدون تشغيل، وبعد الانتهاء منه يتم تشغيله، حيث تعطي هذه الطريقة بعض المميزات أهمها:

- التعديل على قيم العناصر دون الحاجة لإدخال الأوامر من جديد.
- التعديل على الأخطاء دون الحاجة لكتابة الأوامر من جديد.
- في حالة حدوث أخطاء في برنامج كبير لا نحتاج لإعادة كتابة الأوامر من جديد.

لإنشاء ملف m-files أنقر على قائمة file ثم اختر new ثم اختر script سوف تظهر النافذة التالية:



مثال آخر:



ثم قم بكتابة الكود الذي يظهر داخل النافذة وهو عبارة عن مثال لملف script بسيط.

دعنا نقوم بشرح أو تحليل محتوى الملف.

أولاً: السطر الأول يبدأ بالرمز % وهو عبارة عن تعليق، كل التعليقات يتم تجاهلها بواسطة برنامج MATLAB أي لا تنفذ، فقط تضاف لتحسين قراءة الكود البرمجي.

ثانياً: الأسطر الاثنين التالية مصفوفات x و y . حيث ينتهي كل سطر بفاصلة منقوطة والذي يمنع من عرض المتجهين على الشاشة.

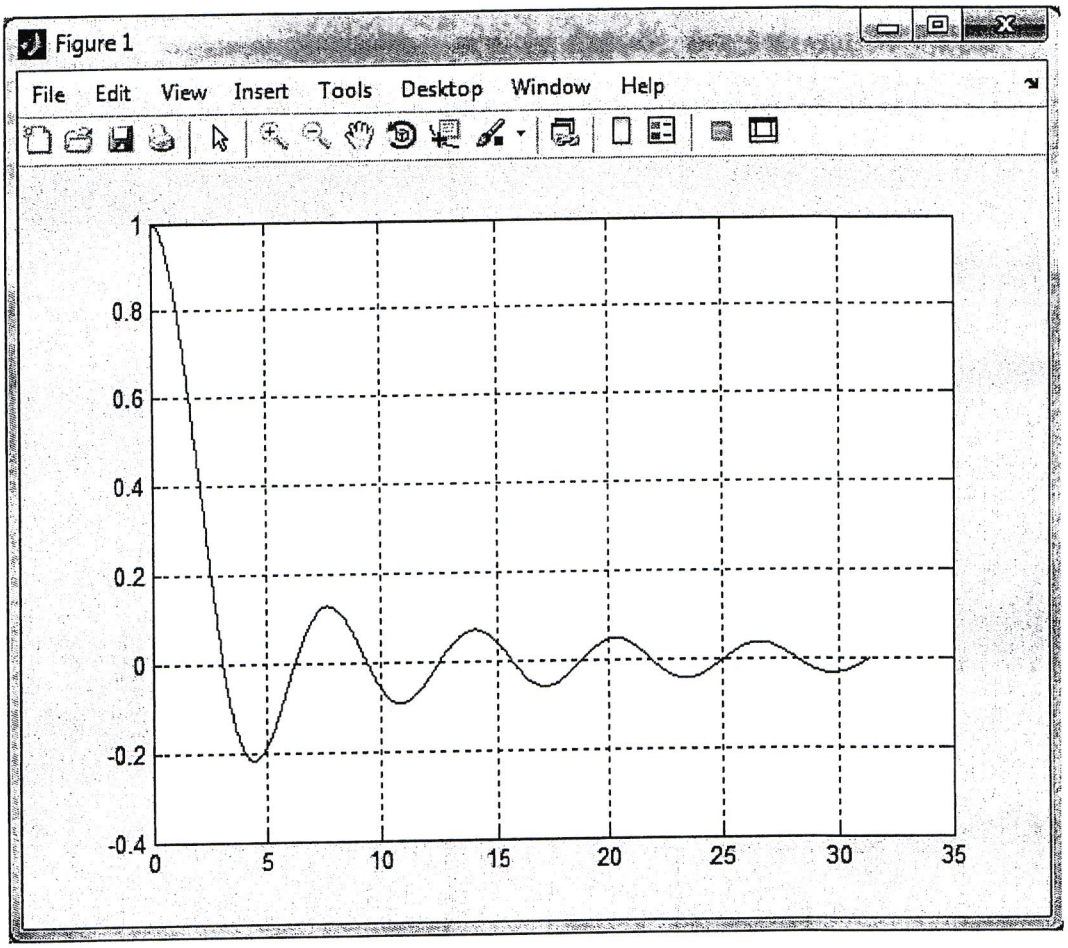
ثالثاً: الأمر plot يقوم بإنشاء رسم بياني للدالة sin(x) باستخدام النقاط التي تم توليدها من خلال التعريفات في السطرين السابقين

رابعاً: الأمر grid لإضافة خطوط الشبكة للمخطط البياني.

حفظ وتنفيذ ملف m-file:

لحفظ ملف m-file من قائمة file اختر save as ثم حدد مكان الحفظ والاسم المطلوب للحفظ وليكن first.

ولتنفيذ الملف من قائمة Debug اختر run first.m أو أضغط على مفتاح F5 لتظهر النافذة التالية:



جمل التحكم Control Statements

سنتعرف في هذا الفصل على كيفية التحكم في سير تنفيذ أوامر البرنامج Program Flow Executions وذلك من خلال استخدام جمل التحكم Control Statements، والتي تعد احد أساسيات لغات البرمجة بصفة عامة.

ويمكن تقسيم جمل التحكم في برنامج MATLAB إلى أربعة أنواع وهي:

١. الجمل الشرطية conditional Statement
   a. الجملة الشرطية if
   b. الجملة الشرطية switch
٢. الجمل التكرارية Looping Statements
   a. الجمل التكرارية for loops
   b. الجمل التكرارية المشروطة while loops
٣. جمل القفز Jumping Statements
   a. جملة Break
   b. جملة Continue
٤. جمل الاستثناء Exception Statements
   a. بنية Try- catch

الجمل الشرطية conditional Statement

الجملة الشرطية if

تستخدم لتنفيذ جملة (أو عدة جمل) أو تجاهلها بناء على شرط معين. ولها ثلاث صور:-

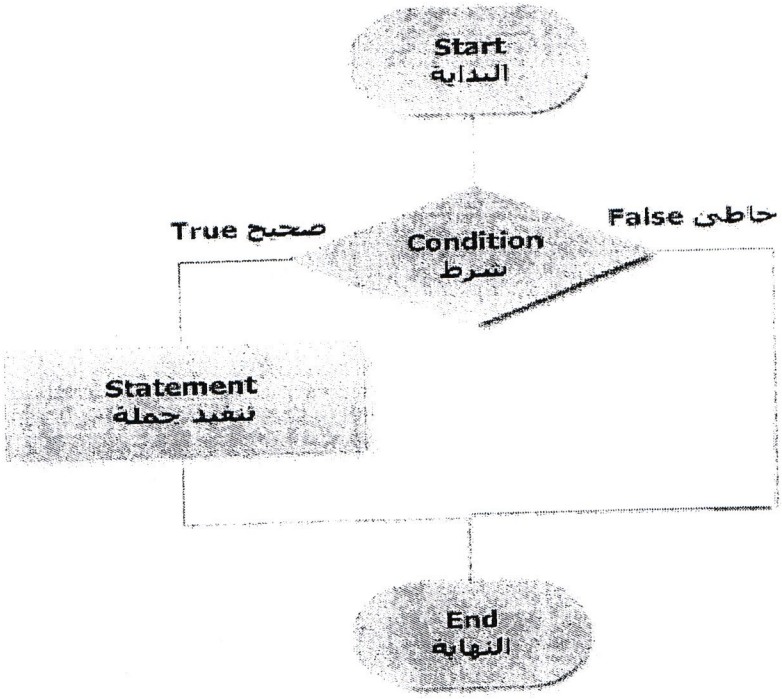
١. IF البسيطة:-Simple If

الجملة if هي جملة في ابسط أشكالها تختبر شرط ما إذا كان الشرط صحيح فإنه ينفذ الجملة التابعة لجملة if وإذا كانت خاطئة فيتجاوز ما يتبع جملة if ويتابع عمله...

والصيغة العامة تكون كالتالي:**كود:**

```
if condition

   statement1;
   statement2;
   statement3;
.
.
.
```

في هذه الصيغة نلاحظ أنه في حال تنفيذ الجملة فإذا تحقق الشرط فإنه يتابع لينفذ الجملة او البلوك التابع لجملة if وإذا لم يتحقق فإنه يتجاوز مباشرة عن هذه الجملة .... لاحظو الشكل التالي:



وهذه جملة الـ if في ابسط حالاتها

مثال١:

برنامج للتحقق من صحة كلمة المرور password فإذا كانت صحيحة يطبع ذلك وإلا ينهي البرنامج

```
clc;clear;close all;
```

```
str='demo';
password=input('Enter password:','s');
if strcmpi(str,password)==1
disp('password is right!');
end
```

الصورة الثانية: if else

الصيغة العامة لها

```
if condition

   statement1;
   statement2;
   ......
Else
statement3;
statement4;
......
end
```

مثال٢:

أنظر للبرنامج هو يقوم بقراءة رقم نحن أدخلناه خلال تنفيذ البرنامج ، يقرأ البرنامج قيمة المتغير age بعد الإدخال ومن ثم يرى هل هو اكبر من القيمة ٥٠ فإذا كان ذلك، يطبق الجملة التي داخل البلوك التابع له فإذا كان العمر المدخل اقل من ٥٠ يطبع!You are pretty young

```
clc;
x=input('enter your age : ');
if x>50
    disp('you are old');
else
    disp('you are pretty young');
end
```

الصورة الثالثة:if elseif

الصيغة العامة لها

```
if condition1
   statement1;
........
elseif condition2
   statement2;
   ......
elseif condition(n-1)

Statement(n-1);
else
```

```
statement(n);
end
```

مثال٣:

برنامج لحساب مجموع خمسة درجات ثم إيجاد التقدير

```
clc;
sum=0;
x=input('enter fife marks as vector: ');
sum=sum+x(1)+x(2)+x(3)+x(4)+x(5);
display(sum);
if sum<60
    disp('Fail');
elseif sum<70
    disp('Pass');
    elseif sum<80
    disp('Good');
    elseif sum<90
    disp('Very Good');
    elseif sum<100
    disp('Excellent');
else
    disp('Out of Range');
end
```

الجملة الشرطية switch – case

تتشابه مع الجملة الشرطية if – elseif بل إنها تعد بديل أفضل لجملة if – elseif في حالة وجود عدة إحتمالات.

الصيغة العامة لها:

```
Switch testvar
Case value1
Statements Block1
……..
Case {value2, value3, value4,…..}
Statements Block2
……..
Case value5
Statements Block3
……..
Otherwise
Statements Block4
……..
end
```

مثال١: التعامل مع القيم الرقمية المفردة

```
clc;
num=input('enter number : ');
switch num
    case 1
        disp('you entered one');
    case {2,3,4}
        disp('you entered two or three or four');
    case 5
        disp('you entered five');
    otherwise
        disp('you entered other value');
end
```

مثال٢: استخدامه مع السلاسل الحرفية

```
clc;
char=input('enter chracter : ','s');
switch char
    case 'A'
        disp('you entered A');
    case {'B','C','D'}
        disp('you entered A or B or C');
    case 'E'
        disp('you entered E');
    otherwise
        disp('you entered other chracter');
end
```

مثال٣: برنامج لتحويل عدد مدخل بوحدات(بوصة، قدم، متر، مليمتر) إلى سنتمترات

```
clc;
disp('Convert x to centimeters ');
x=input('enter the value of x : ');
units=input('enter the unit of x : ','s');
switch units
    case {'inch','in'}
        y=x*2.54;
    case {'feet','ft'}
        y=x*2.54*12;
    case {'meter','m'}
        y=x*100;
        case {'milimeter','mm'}
        y=x/10;
    otherwise
        disp('Unknown unit ');
end
display(y);
```

تمارين

١ -قم بتعريف متجه صفي في نافذة الأوامر يتكون من العناصر التالية

(2,4,6,8,10,12)

ثم قم بالإجابة على الاتي:

قم بتغيير العنصر الثالث في المتجه ليكن ٧ بدلا من ٦

قم بسحب العنصر الرابع من المتجه وإسناده للمتغير x

قم بأخذ العناصر من الثاني حتى الخامس واسناده لمتغير جديد وليكن k

قم باضافة العنصر ٥ وذلك بعد العنصر ٤ مباشرة

باستخدام الدوال أوجد مجموع عناصر المتجه

٢ -قم بتعريف المصفوفة التالية باسم matrix

1	2	3
4	5	6
7	8	9

قم بطباعة عناصر الصف الثالث

قم بطباعة عناصر العمود الأول

قم بطباعة عناصر قطر المصفوفة

قم بإيجاد محدد المصفوفة

قم بحذف الصف الثالث

٣ -باستخدام جملة if المتداخلة وباستخدام ملف m-file

قم بتصميم برنامج لحساب مجموع أربعة درجات يتم إدخالها من المستخدم في شكل متجه ثم إيجاد التقدير علماً بأن التقدير يكون إذا كان المجموع أكبر من ١٠٠

من ٩٠ إلى ١٠٠ التقدير"Excellent"

من ٨٠ وأقل من ٩٠ التقدير "Very Good"

من ٧٠ وأقل من ٨٠ التقدير "Good"

من ٦٠ إلى أقل من ٧٠ "Pass"

أقل من ٦٠ "Fail"

؟؟

٢- الجملة التكرارية Looping statements

أولاً: جملة التكرار **for**

تستخدم هذه الجملة التكرارية لتنفيذ مجموعة من الأوامر البرمجية لعدد معين من المرات، وتعد من أشهر وأبسط أنواع جمل التكرار في لغات البرمجة بصفة عامة.

بناء الجملة يكون بالشكل التالي:

```
For  k = start: step : last
```

```
Commands
end
```

حيث أن الأوامر التي تقع بين جملتي for و end تنفذ لكل القيم المخزنة في k والتي تبدأ ب start وتنتهي ب last ومقدار الخطوة step.

مثال:

للحصول على قيم الدالة sin عند النقاط pi*n/10 حيث n=0,1,.....,10 نكتب الآتي على برنامج MATLAB.

```
clc;clear;close all;
for n=1:10
    x(n)=(sin(n*pi/10));
end
display(x)
```

ويكن تحديد مقدار الخطوة ويمكن أن تكون الخطوة بالسالب

```
x=zeros(1,10);
for n=10:-2:2
    x(n)=(sin(n*pi/10));
end
display(x)
```

كما يمكن لقيم عداد جملة for أن تكون مخزنة كعناصر في متجه صفي

```
clc;clear;close all;
x=zeros(1,10);
for n=[1 3 5 7 9]
    x(n)=(sin(n*pi/10));
end
display(x)
```

مثال٢:

برنامج لتحديد هل العدد المدخل زوجي أم فردي مع تحديد عدد مرات تنفيذ هذا التحديد(عدد الأعداد التي سوف تدخلها).

```
clc;clear;close all;
t=input('enter the number of times :');
for i=1:t
    x=input('enter anumber :');
    if rem(x,2)==0
        display('the number you enterd is even ');
    else
```

```
        display('the number you enterd is odd ');
    end


end
```

الحلقات التكرارية المتداخلة: Nested For Loops

يقصد بها تداخل أكثر من حلقة for بداخل بعضها البعض في نفس الوقت.

مثال:

```
clc;clear;close all;
k=0;
for i=1:10
    for j=1:10
        k=k+1;
    end
        disp(k);
end
```

حيث أنه في كل مرة تعمل فيها الحلقة الخارجية i فإن الحلقة الداخلية j تعمل ١٠ مرات، ولتوضيح ذلك يمكننا طباعة قيم الحلقة الخارجية i مع المتغير k بتعديل أمر الإخراج إلى disp([I,k]) في البرنامج.

إنشاء المصفوفات باستخدام الحلقات المتداخلة:

مثال:

صمم برنامج لتكوين مجموعة من القيم الرقمية يتم تخزينها في مصفوفة ثلاثية الأبعاد باستخدام الحلقات التكرارية المتداخلة.

```
clc;clear;close all;
x=zeros(3,3);
for m=1:3
    for n=1:3
        x(m,n)= n^2+ m^2;
    end
end
display(x);
```

مثال آخر:

برنامج لإيجاد مجموع عناصر مصفوفة يتم إدخالها من قبل المستخدم

```
clc;clear;close all;
matrix=input('enter the matrix :');
sum=0;
```

```
[m n]=size(matrix);
for i=1:m
    for j=1:n
        sum=sum + matrix(i,j);
    end
end
display(matrix);
display(sum);
```

؟؟

ثانياً: جملة التكرار while.

الصيغة العامة لجملة التكرار while هي كالتالي:

```
While expression
Statements
end
```

هذا التكرار يستخدم عندما يكون المبرمج لا يعلم عدد التكرارات

مثال:

```
clc;clear;close all;
x=1;
while x<20
    disp(x);
    x=2*x+1;
end
```

(١-٥)الدوال المخزنة على MATLAB :

الدوال هي عبارة عن أكواد برمجة سابقة الإعداد أو التجهيز تؤدي لنا وظائف متنوعة ولكل دالة اسم خاص بها لا يتشابه مع غيرها إلا أنه ينبغي التنوية إلى أنه يجب التمييز بين نوعين من الدوال:

١. الدوال التي نقوم بكتابتها بنفسنا من خلال ملف من النوع M-File وتخزينها باسم معين لاستخدامها فيما بعد.

فإن برنامج الـ Matlab يتيح لنا إمكانية كتابة وأضافة دوال إلى الدوال الأساسية الموجودة فيه، وذلك عن طريق إعداد هذه الدوال كملفات M-File من خلال النافذة وحفظها بإسم معين.

يتم حفظ الدالة في m-files ويجب تعريف الدالة في أول سطر مع مراعاة التالي :

- أن يكون اسم الدالة الموجود في تعريف الدالة هو نفسه الذي يتم به حفظ الدالة.
- أن يكون اسم الدالة مكون من مقطع واحد لا يفصل بينه مسافات .
- أن لا يتجاوز اسم الدالة ٣١ حرف .
- أن يبدأ اسم الدالة بحرف ويمكن إتباعه برمز .

وعند الحاجة للبرنامج يتم كتابة اسم البرنامج ثم استخدامه ، أو يمكن تشغيله من أمر Run الموجود على شاشة الملف مباشرة.

٢. الدوال المخزنة في برنامج الـ Matlab وهي دوال معدة بواسطة الشركة المنتجة للبرنامج ويمكننا استخدامها مباشرة دون الحاجة لمعرفة الكود المكتوب لها.

هناك الكثير من الدوال المخزنة على Matlab ويبين الجدول التالي بعضاً منها :

➢ دوال التقريب:

الدالة	الوظيفة
Rem	تقوم بإخراج الباقي الصحيح لعملية القسمة.
Round	تقريب الرقم العشري باتجاه أقرب رقم صحيح

الجدول (٢-١)

➢ الدوال المثلثية:

الدالة	الوظيفة
Sin	لحساب جيب الزاوية.
Cos	لحساب جيب التمام للزاوية.
Tan	لحساب ظل الزاوية.
Cot	لحساب ظل التمام للزاوية.

الجدول (٣-١)

➢ الدوال الحسابية الأولية:

الدالة	الوظيفة
Exp	e^x

Sqrt	لإيجاد الجذر التربيعي
Abs	لإيجاد القيمة المطلقة
Max	لإيجاد القيمة العظمى
Min	لإيجاد القيمة الصغرى
Mod	القيمة المطلقة للباقي الصحيح للقسمة.
Rem	لحساب الباقي الصحيح للقسمة.
Log	اللوغاريتم الطبيعي: ذو الأساس الطبيعي ٢,٧١٨٣=e
Factorial	لحساب المضروب.

الجدول (٤-١)

(٦-١)الرسم على MATLAB:

الرسم إما ثنائي و ثلاثي الأبعاد :

يمتلك برنامج Matlab قدرة كبيرة وإمكانيات عالية في عرض المتجهات والمصفوفات والدوال كرسومات بيانية، كما يمكنه من رسم الأشكال ثلاثية الأبعاد بالإضافة إلى تحريك تلك الأشكال الرسومية، وهذا بالإضافة إلى إمكانية إدراج أية تعليقات نصية على الرسومات وطباعتها، وبذلك تكون إمكانيات رسم المنحنيات الرياضية والمصفوفات في Matlab من أهم الإمكانيات المميزة فيه. ويقدم لنا برنامج Matlab وسائل تساعدنا على الرسم مثل تغير لون الخط، وتسمية المحاور، وتسمية الرسمة، وتسمية المتغيرات، وتقسيمها ومنها:

الدالة	الوظيفة
Plot	يستخدم للرسم الخطية ثنائية الأبعادD-2 .
Plot3	تستخدم للرسم ثلاثي الأبعاد.
Surf	مشابة لـmesh لكن مع تلوين الرسم وبالتالي تلوين الشكل كاملاً وهوللرسم ثلاثي الأبعاد D-3.
Surfc	مشابة لـ meshc لكن مع تلوين الرسم وبالتالي تلوين الشكل كاملاً وهو للرسم ثلاثي الأبعاد D-3.
Mesh	للرسم على المحاور الاحداثية الثلاثة D-3 على شكل شبكة.
Contour	لعمل تخطيط للرسم في بعدين او ثلاثة أبعاد.

الجدول (٥-١)

لرسم أكثر من دالة نستخدم الألوان التالية :

اللون	أحمر	أبيض	أسود	أصفر	أخضر	أرجواني	أزرق	أزرق داكن
الرمز	R	W	K	Y	G	M	C	B

مثال(١) :

قم برسم المخطط البياني للدالة sin(x)

```
clc;clear;close all;
x=0:1:10;
t=sin(x);
plot(x,t);
title('plot of pure sine wave ');
```

مثال(٢):

قم برسم مخطط للدالتين sin(x) و cos(x) في مخطط بياني واحد

```
clc;clear;close all;
x=0:0.1:10;
y=sin(x);
z=cos(x);
plot(x,y,x,z);
title('plot of sine and cosine waves');
```

تنسيقات الرسم البياني:

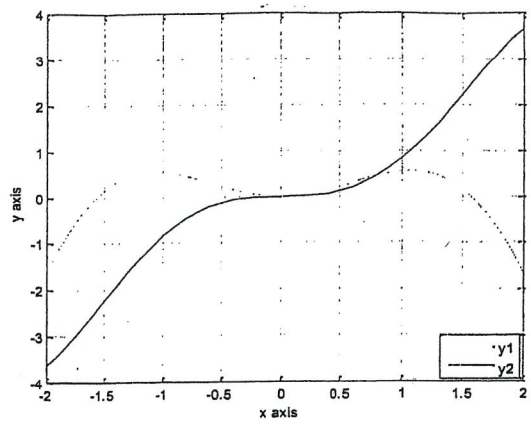
مثال(٣) ارسم الدالتين التالية بنفس الرسم $y_1 = x^2 \cos x, y_2 = x^2 \sin x, x = -2:0.1:2$

؟

الحل:

```
>> x=-2:0.1:2;
>> y1=x.^2.*cos(x);
>> y2=x.^2.*sin(x);
>> plot(x,y1);
>> hold on
>> plot(x,y2);
>> hold off
>> xlabel('x-axis')
>> ylabel('y-axis')
>> grid on
```

يظهر لنا الرسم التالي:



الشكل (٤-١): رسم الدالتين $y_1 = x^2 \cos x , y_2 = x^2 \sin x$

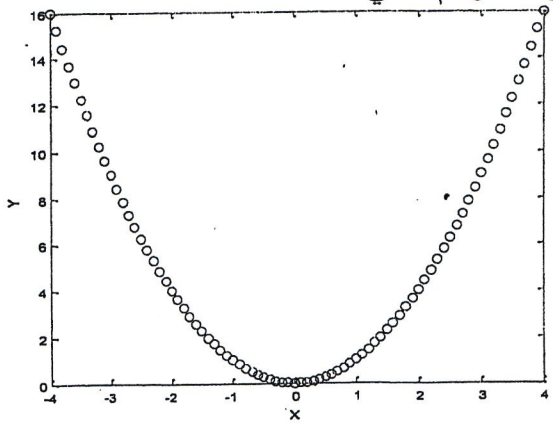
مثال(٢) :

ارسم الدالة y=x^2 , x=-4:0.1:4 ؟

الحل:

```
>> x=-4:.1:4;
>> y=x.^2;
>> plot(x,y,'o')
```

يظهر لنا الرسم التالي:



الشكل (٥-١) : رسم الدالة y=x.^2

أكثر من مخطط في نافذة واحدة:

حيث يمكنك عرض أكثر من مخطط بياني في نافذة واحدة وذلك باستخدام الدالة subplot.

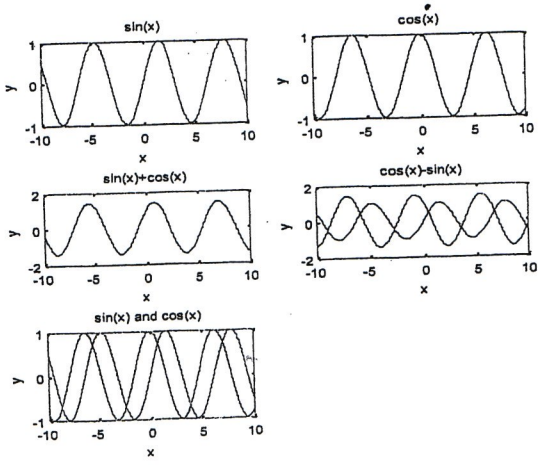
مثال:

قم بكتابة الأوامر التالية في ملف m-file ثم قم بحفظ وتنفيذ الملف

ارسم الدالة (cosn(x),sin(x وحاصل جمعهما وحاصل الطرح والدالتين مع بعضهما في نفس الرسم؟

الحل:

```
x=-10:.01:10;
y1=sin(x);
subplot(3,2,1)
plot(x,y1);xlabel('x');ylabel('y');title('sin(x)')
subplot(3,2,2)
y2=cos(x);
plot(x,y2,'r');xlabel('x');ylabel('y');title('cos(x)')
subplot(3,2,3)
plot(x,y1+y2,'k');xlabel('x');ylabel('y');title('sin(x)+cos(x)')
y4=y2-y1;
subplot(3,2,4)
plot(x,y1,x,y4);xlabel('x');ylabel('y');title('cos(x)-sin(x)')
y5=sin(x);
y6=cos(x);
subplot(3,2,5)
plot(x,y5,'r',x,y6,'k');xlabel('x');ylabel('y');title('sin(x) and cos(x)')
```



الشكل (٨-١): رسم للدالتين (cosn(x),sin(x

مخططات ثانوية:

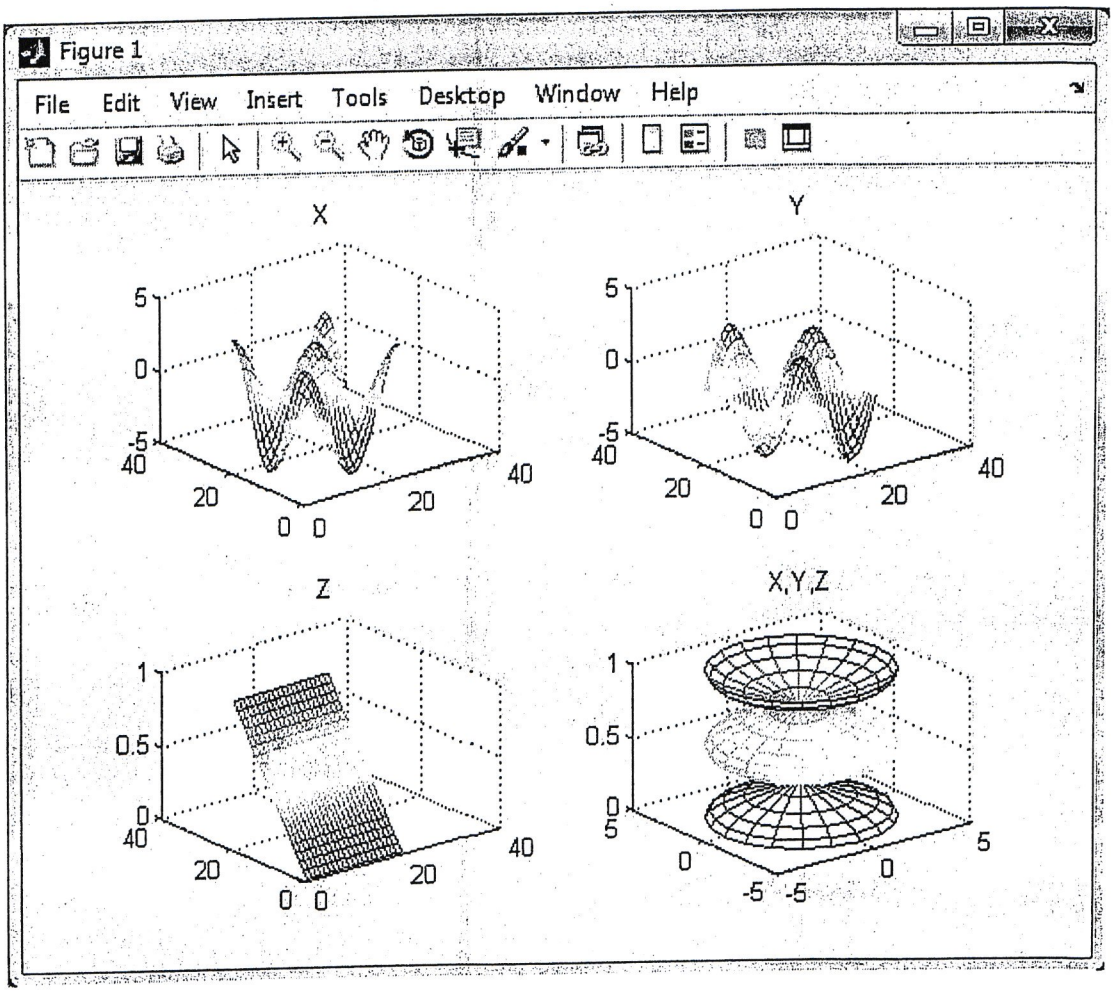
يمكنك عرض أكثر من مخطط بياني في نافذة واحدة وذلك باستخدام الدالة subplot.

مثال:

قم بكتابة الأوامر التالية في ملف m-file ثم قم بحفظ وتنفيذ الملف

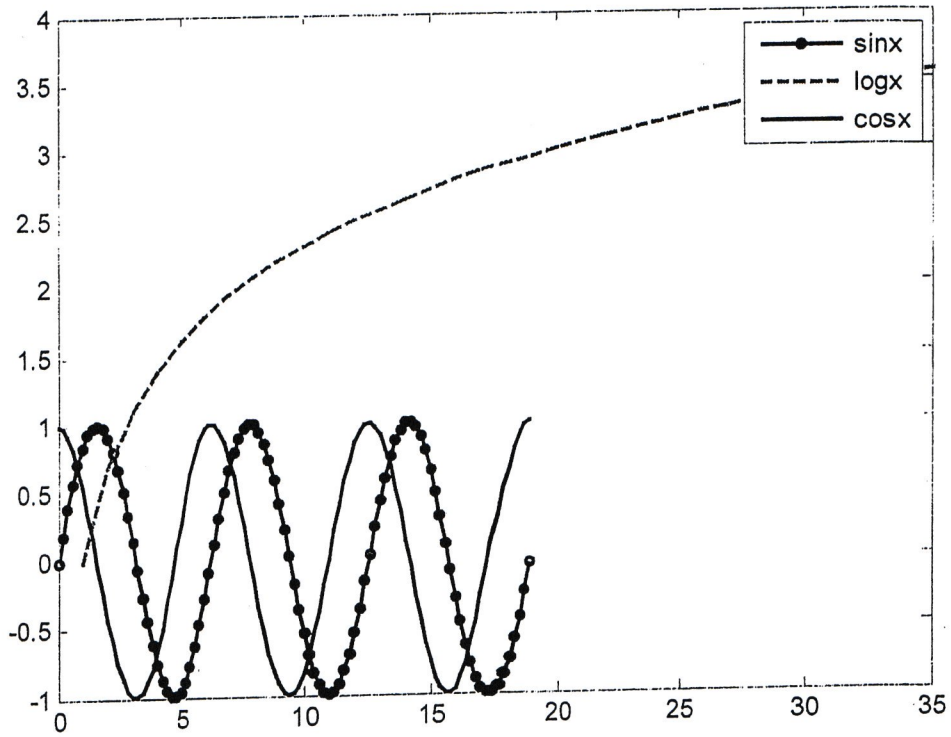
```
t=0:pi/10:2*pi;
[X,Y,Z]=cylinder(4*cos(t));
subplot(2,2,1); mesh(X); title('X');
subplot(2,2,2); mesh(Y); title('Y');
subplot(2,2,3); mesh(Z); title('Z');
subplot(2,2,4); mesh(X,Y,Z); title('X,Y,Z');
```

سوف تظهر لك الرسومات التالية في نافذة واحدة كما بالشكل التالي:

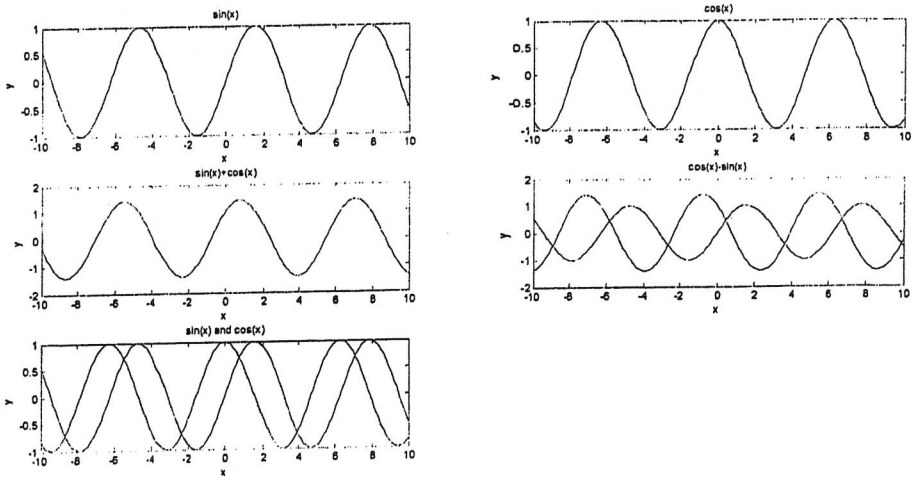


تمارين

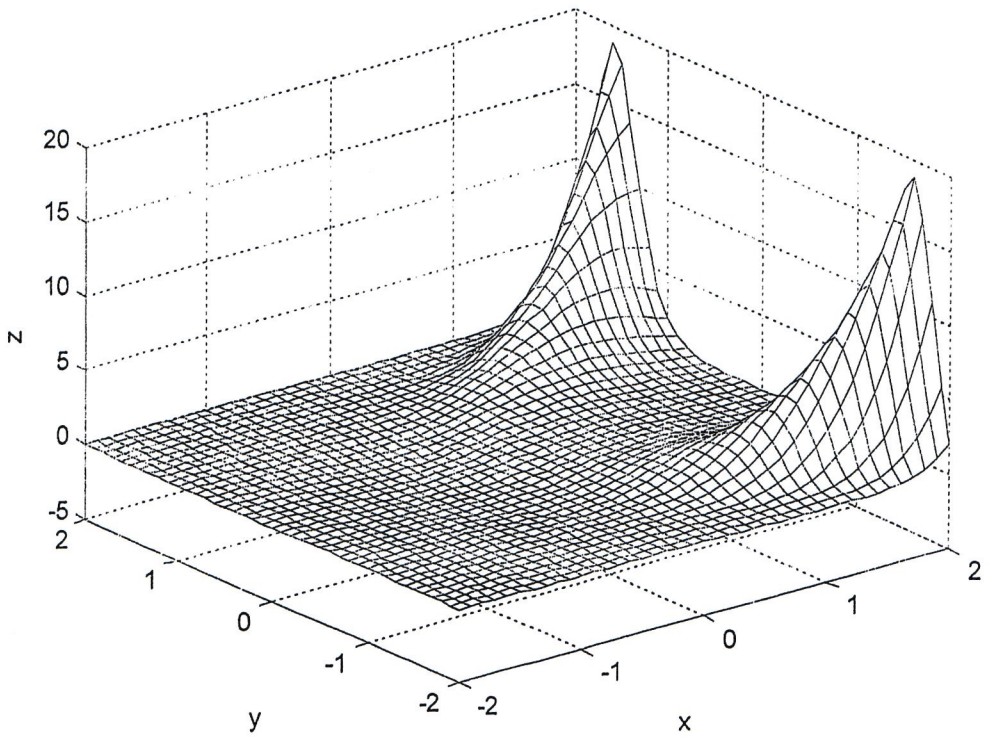
```
x=0:0.2:6*pi;
y=sin(x);
plot(x,y,'-ob','LineWidt',2,'MarkerSize',2)
z=1:1:35;
t=log(z);
hold on
plot(z,t,'--*r','linewidth',2,'markersize',2)
x=0:0.2:6*pi;
r=cos(x);
plot(x,r,'-+k','linewidth',2,'markersize',2)
legend('sinx','logx','cosx',1)
```



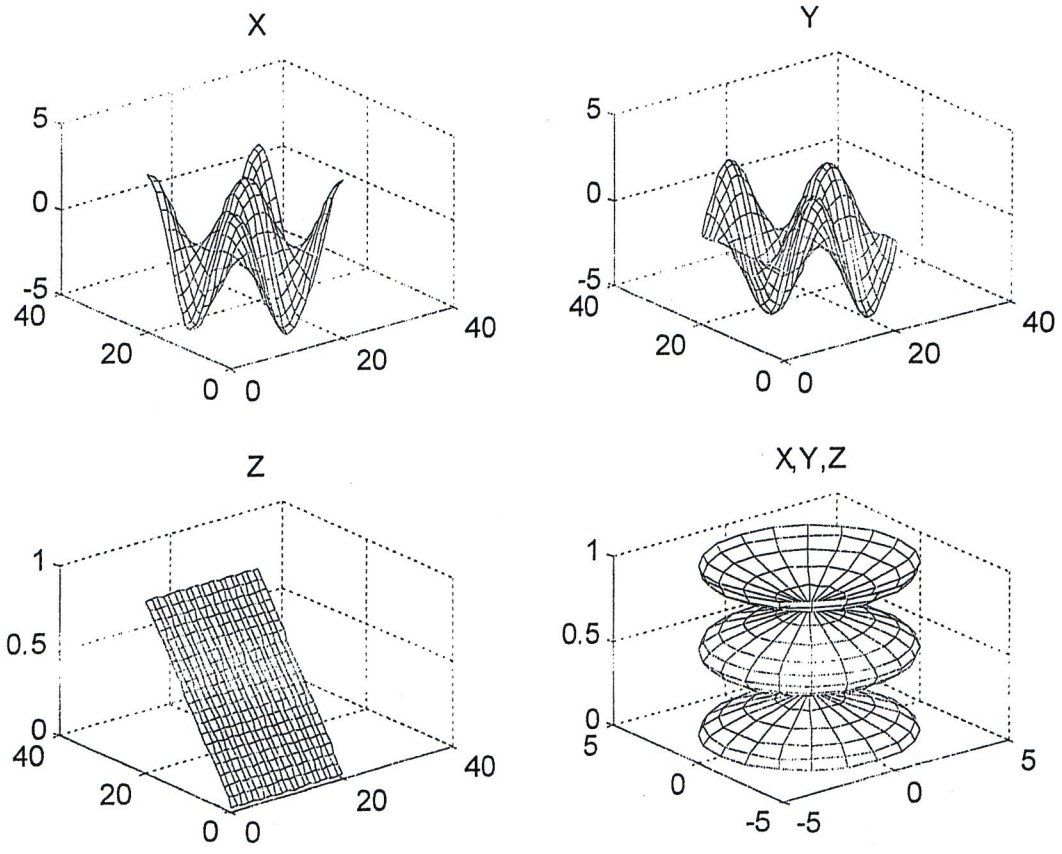
```
x=-10:0.1:10;
y1=sin(x);
subplot(3,2,1)
plot(x,y1);xlabel('x');ylabel('y');title('sin(x)')
subplot(3,2,2)
y2=cos(x);
plot(x,y2,'r');xlabel('x');ylabel('y');title('cos(x)')
subplot(3,2,3)
plot(x,y1+y2,'k');xlabel('x');ylabel('y');title('sin(x)+cos(x)'
)
y4=y2-y1;
subplot(3,2,4)
plot(x,y1,x,y4);xlabel('x');ylabel('y');title('cos(x)-sin(x)')
y5=sin(x);
y6=cos(x);
subplot(3,2,5)
plot(x,y5,'r',x,y6,'k');xlabel('x');ylabel('y');title('sin(x)
and cos(x)')
```



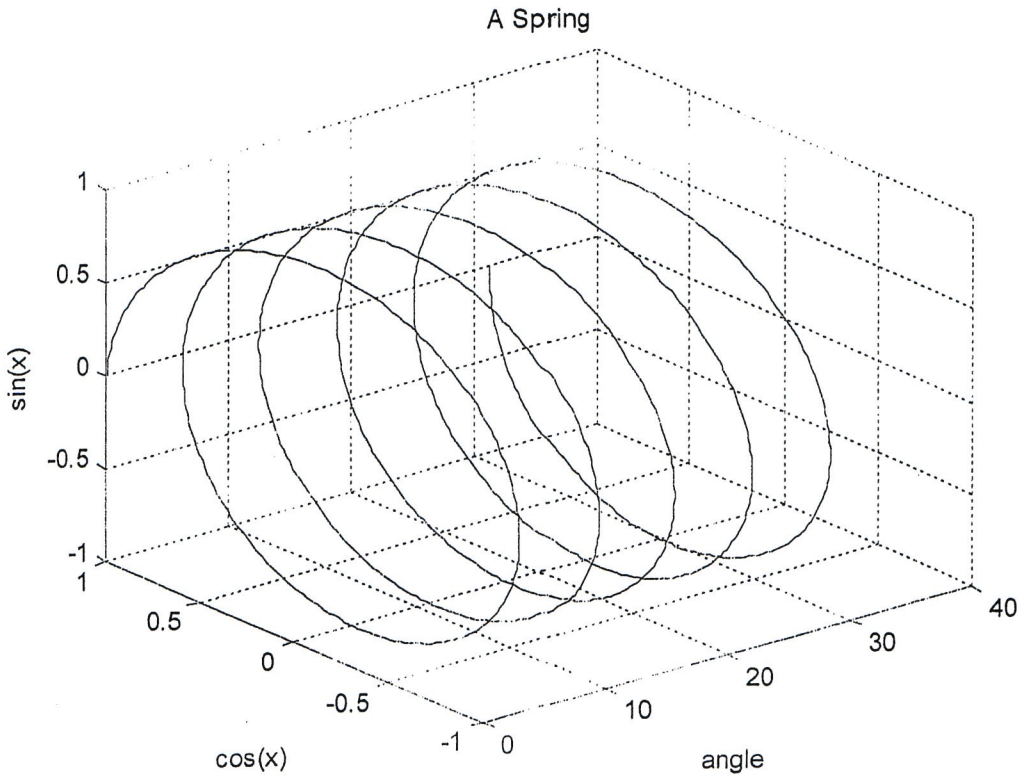
```
[X,Y]=meshgrid(-2:0.1:2);
Z = X.*exp(-((X-Y.^2).^2 - Y.^2));
mesh(X,Y,Z),xlabel('x'),ylabel('y'),zlabel('z')
```



```
t=0:pi/10:2*pi;
[X,Y,Z]=cylinder(4*cos(t));
subplot(2,2,1);
mesh(X); title('X');
subplot(2,2,2);
mesh(Y); title('Y');
subplot(2,2,3);
mesh(Z); title('Z');
subplot(2,2,4); mesh(X,Y,Z); title('X,Y,Z');
```



```
clear, clc
x = linspace(0,10*pi,1000);
y = cos(x);
z = sin(x);
plot3(x,y,z)
grid
xlabel('angle'), ylabel('cos(x)'),zlabel('sin(x)'),title('A
Spring')
```



```
[X,Y] = meshgrid(-8:.5:8);
R = sqrt(X.^2 + Y.^2) + eps;
Z = sin(R)./R;

figure
mesh(Z)
```