

## المحاضرة الاولى

### مقدمة عن برنامج Matlab

برنامج MATLAB هو برنامج هندسي يقوم بعمليات تحليل وتمثيل البيانات من خلال معالجة تلك البيانات تبعاً لقاعدة البيانات الخاصة به، فمثلاً يستطيع البرنامج عمل التفاضل Differentiation والتكامل Integration وكذلك يقوم بحل المعادلات الجبرية Algebraic Equation والمعادلات التفاضلية Differential Equations ذات الرتب العليا والتي قد تصل مستوى عال من الصعوبة، ويستطيع البرنامج عمل التفاضل الجزئي وعمليات الكسور الجزئية Partial fraction بسهولة ويسر والتي تستلزم وقت كبير لعملها بالطرق التقليدية، هذا من الناحية الاكاديمية، أما من الناحية التطبيقية فيستطيع البرنامج العمل في جميع المجالات الهندسية.

بعض تطبيقات واستخدامات برنامج Matlab :

١. انظمة التحكم Control System
٢. معالجة الاشارة الرقمية Digital Signal Processing
٣. النظريات العددية التقريبية Numerical Application
٤. معالجة الصور Image Processing
٥. تطبيقات الرادار Radar Applications
٦. تطبيقات الروبوت Robots Applications
٧. التطبيقات الالكترونية Electronics Applications
٨. تطبيقات الاتصالات Communication Applications

برنامج الماتلاب يستخدم لإجراءات الحسابات التقنية المتقدمة ويتميز ماتلاب بكونه يمكنه التعامل مع المعادلات الرياضية والتكاملات والتفاضلات والمصفوفات المختلفة بسرعة ويمكن ماتلاب المستخدم من رسم المعادلات الرياضية في الاحداثيات المختلفة ويضم المئات من الدوال الجاهزة التي توفر للمبرمج وقتاً وجهداً عند انشاء البرنامج. وتعني كلمة MATLAB مختبر المصفوفات (MATrix LABoratory).

يؤمن برنامج MATLAB ادوات واجهة التخاطب الرسومية Graphical User Interface(GUI) التي تجعلك تتعامل مع البرنامج على انه اداة تطبيقية متطورة.

### تشغيل برنامج MATLAB

بعد تنصيب برنامج MATLAB على الحاسبة التي تعمل عليها. يتم اضافة رمز ايقونة البرنامج على سطح مكتب الحاسبة ويتم فتحه عند النقر على الايقونة بنقرتين مزدوجتين double click.

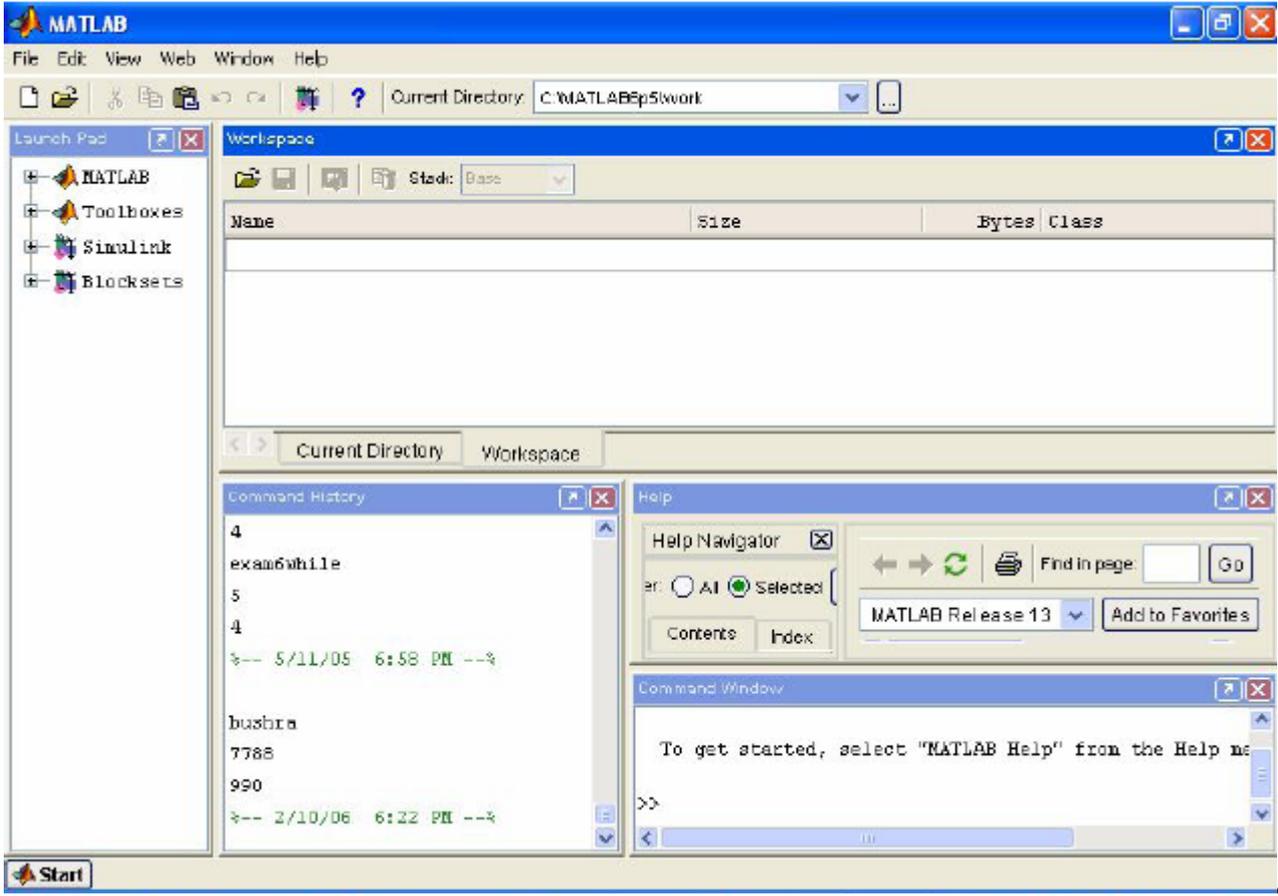
### واجهة برنامج Matlab

تقسم واجهة برنامج الماتلاب إلى اربع نوافذ رئيسية هي:

- المجلد الحالي Current Folder: يعرض محتويات المجلد الحالي.
- نافذة الاوامر Command Window: تستخدم لكتابة الاوامر البرمجية والمدخلات وعرض النتائج. حيث يظهر علامة الحث >> التي نكتب بعده الاوامر المطلوبة.

- نافذة منطقة العمل Workspace Window : تستخدم لرؤية متغيرات الادخال والايخارج وقيم تلك المتغيرات. حيث يظهر اسم المتغير Name، وقيمة المتغير Value، ونوع المتغير class، وحجم المتغير Size، ويمكن مسح محتويات هذه النافذة باستخدام الامر clear.
- نافذة تاريخ الاوامر Command History: ارشيف للاوامر التي تم ادخالها سابقاً في نافذة الاوامر.

كما في الشكل ادناه:



في

#### ملاحظات:

- كل متغير في MATLAB عبارة عن مصفوفة، لغة MATLAB موجهة بالمصفوفات حتى وان كانت المتغيرات اعداداً مفردة (scalar).
- الامر clear يستخدم لحذف المتغيرات والدوال من الذاكرة.
- الامر clc يستخدم لمسح نافذة الامر Command Window.
- ثلاث نقطة متتالية (...) في نهاية السطر مسبوقه بفراغ للدلالة على استمرار الابعاز في السطر التالي.
- فارزة منقوطة بعد الابعاز تمنع طباعة المتغير او الناتج في نافذة Command وكذلك في نافذة Editor.
- اشارة النسبة المئوية(%) تستخدم للتعليق فكل نص يأتي بعدها يعتبر نص تعليق، مثل:

% This Program Compute Area

- ملفات MATLAB تسمى M-files وتكون توسعها (.m)، مثلاً (example1.m).
- الاحتفاظ بكتابة الابعازات السابقة واللاحقة في نافذة Command بحركة السهم للاعلى والاسفل.
- نتيجة تنفيذ برنامج MATLAB (الناتج والايخارجات) تظهر في شاشة Command Window لذلك يجب الانتقال اليها بعد التنفيذ.
- لغة MATLAB لا تحتاج الى الاعلان عن المتغيرات والثوابت والانواع البيانية الاخرى المستخدمة بالبرنامج.

- لإنشاء ملف نصي M-file، انقر على ايقونة الصفحة الفارغة (البيضاء) الموجودة ضمن شريط ادوات سطح مكتب MATLAB او اختيار open لفتح ملف موجود مسبقاً. يستدعي هذه الاوامر نافذة محرر النصوص التي يمكنك في كتابة اوامر MATLAB (نافذة كتابة البرامج).
- يمكن تنفيذ الملف المخزون باختيار ايقونة Run الموجودة في شريط ادوات نافذة Editor او عبر ضغط المفتاح F5 او الاختيار Run من القائمة Debug، او كتابة اسم الملف المخزون امام علامة الحث >> في نافذة Command. بعد انتهاء كتابة البرنامج (الملف) يخزن هذا الملف كملف M-file باسم معين (مثلاً example1.m) على قرصك الصلب عبر اختيار الاختيار Save من القائمة File او الخزن ضمن شريط ادوات سطح مكتب MATLAB.
- العمليات الرياضية الاولية في Matlab:
- وهي عمليات اولية تستخدم لاجاد الجمع والطرح والضرب والقسمة والاس وكما في الجدول التالي:

| الرمز  | العمليات |
|--------|----------|
| +      | الجمع    |
| -      | الطرح    |
| *      | الضرب    |
| / او \ | القسمة   |
| ^      | الاس     |

- استخدام ماتلاب لاجراء العمليات الحسابية اعلاه لما يلي:

$$5/6, 6*888.5, 5*(\cos(a)/2)-3+\sin(90)$$

### رموز لغة MATLAB : MATLAB Symbols

تتكون لغة MATLAB من العناصر الاساسية التالية:

- أ- حروف ابجدية انكليزية: وهي A, B,...,Z a,b,...,z
- ب- ارقام حسابية: 0,1,2,...,9
- ت- رموز خاصة مثل: {, +, -, =, >, <, \*, (, ), ...}

### الثوابت Constants

يوجد في لغة MATLAB انواع متعددة من الثوابت اهمها :-

#### (أ) الثوابت العددية Numerical Constants

وتتكون من عدد من الارقام ولها عدة اشكال هي:

(١) الثوابت الصحيحة: مثل 18-, 427, 23+, 0,

ملاحظة: اكبر عدد صحيح مستخدم

```
>>bitmax
```

```
Ans=
```

```
9.00719925470991e+015
```

والتي تقابل  $2^{53}-1$

(٢) الثوابت الحقيقية: مثل 18.0-, 472.5, 51.8, 0.0,

ملاحظة: اصغر عدد حقيقي مستخدم

```
>>realmin
```

```
Ans= 2.225073858507201e-308
```

&gt;&gt;realmax

Ans=1.797693134862316e+308

&gt;&gt;pi

Ans=3.1416

(٣) الثوابت الحقيقية المدونة تدويناً يانياً: حيث تحول الصيغة الجبرية  $10^N$  إلى صيغة MATLAB يائية EN فمثلاً تصبح  $2.0 \times 10^3$  في الجبر  $2.0E+3$  بالتدوين اليائي في MATLAB وكذلك تصبح  $1.7 \times 10^{-2}$  في الجبر :  $-1.7E2$  في التدوين اليائي وكذلك تصبح  $0.0032$  :  $3.2E-3 = 3.2 \times 10^{-3}$

(٤) الثوابت العقدية: مثل:  $1-2i$  ,  $6-9i$  ,  $6+\sin(0.5)*j$  ,  $\sqrt{-2}$

حيث:  $i = \sqrt{-1}$

مثال ١: اذا كان  $c = -7.7782 - 4.9497i$

فلاستخراج الجزء الحقيقي  $cr = \text{real}(c)$   $cr = -7.7782$

ولاستخراج الجزء التخيلي  $ci = \text{imag}(c)$   $ci = -4.9497$

### (ب) الثوابت الرمزية String Constants

يسمى هذا النوع من "ثوابت" مجازاً لأن الثابت هذا يتكون من حروف وأرقام ورموز توضع بين علامتي اقتباس quotations مفردة اي ' ' ويستخدم عادة كعناوين توضح القيم الناتجة من الحسابات ووحداتها، تسمى العبارات التالية والموجودة بين الحاصرات العليا ثوابت رمزية.

'The speed of wind='

'I love Basrah'

'My birthday =1970'

كل الثوابت الرمزية أعلاه، وان استخدمت ارقاماً حسابية داخلها، فهي لاتحمل معنى حسابي، ومن الجدير بالذكر اثناء استعمال الثوابت الرمزية انه لايجوز استخدام حاصرات علوية داخل حاصراتها، كما ينبغي التنبيه اي ان هناك قيماً رمزية للحروف يعتبر الحرف A اقل من الحرف B ويمكن كتابة ذلك بالصورة:

'A' &lt; 'B'

### (ج) الثوابت المنطقية Boolean Constants

وهي الثوابت التي قيمتها العددية (1) في حالة true و (0) في حالة false .

مثال:

 $3 > 2 \longrightarrow 1$ 
 $0 > 5 \longrightarrow 0$

## المتغيرات Variables

هناك بعض القواعد الواجب مراعاتها عند كتابة اسم المتغير وهي:

١. لا يمكن استخدام الكلمات المفتاحية (الكلمات المحجوزة) او الدوال التي توفرها اللغة كأسماء متغيرات، مثال:

If, end, for, break, else, global, return, function, sin, log

٢. أسماء المتغيرات حساسة لحالة الاحرف (A, a وكذلك COST, Cost, cost, CoST متغيرات مختلفة، وكذلك A, a).

٣. حرف l (small letter) في لغة MATLAB يشبه رقم 1 .

٤. يمكن لاسماء المتغيرات ان تحوي 63 رمزاً وسيهمل اي رمز زائد عن 63.

٥. يجب ان تبدأ اسماء المتغيرات بحرف متبوعاً بأي عدد من الارقام او الاحرف او underscore . ولا يجوز استخدام الرموز الخاصة او الفراغ.

٦. جميع اوامر MATLAB تكتب بالحروف الصغيرة (... if, while, input).

## (أ) المتغيرات العددية Numerical Variables

تتكون من حرف واحد او مجموعة من الحروف من A إلى Z إلى a إلى z ويمكن ان يحتوي على ارقام من 0 إلى 9 ويمكن ان تكون سلسلة من الارقام والحروف بشرط ان يبدأ بحرف (خليط من ارقام وحروف مبدوءة بحرف) ويمكن كذلك ان يحتوي المتغير على underscore حتى 63 رمزاً. وتكون قيمة المتغير عددية (صحيح، حقيقي، عقدي او اسي).

Ali\_Ahmed, X2, S2, ks, K

## التعبير الحسابي

يتكون التعبير الحسابي من مجموعة من الثوابت والمتغيرات تجمع بينهما عمليات حسابية ويستخدم فيها الرموز الحسابية مثل +, -, /, \*, والامثلة الاتية عن تعابير جبرية صيغت بلغة MATLAB .

### التعبير بلغة MATLAB

### التعبير الجبري

$$a-3*b$$

$$a-3b$$

$$c^2-10$$

$$C^2-10$$

$$(a^2+b^2)/12$$

$$a^2+b^2/12$$

$$m*(7*d-8*g)$$

$$m(7d-8g)$$

## قاعدة الاسبقية (الأولوية) Rule of Precedence

وهذه القاعدة مهمة في فهم وترتيب اولويات العمليات الحسابية في التعابير والمعاملات الحسابية، كما يجريها وينفذه الحاسب، وتنص القاعدة على ان الأولوية الاولى تعطى للعمليات الموجودة بين القوسين ومن اليسار إلى اليمين، وبالنسبة للعمليات الحسابية فالرفع إلى الأس أولاً، والضرب (أو القسمة) ثانياً، والجمع (او الطرح) اخيراً والمثال التالي يوضح هذه القاعدة:

$$\frac{A}{B} + c \quad \text{يكافئ في الجبر} \quad A/B+C$$

التعبير

1  
2

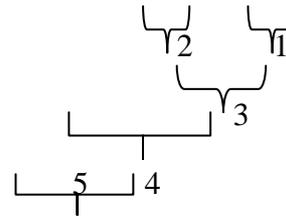
$$\frac{A}{B+C} \quad \text{يكافئ في الجبر} \quad A/(B+C)$$

بينما يكافئ التعبير

1  
2

لأن الجميع داخل الاقواس يجري أولاً حسب الأولوية ثم يقسم A على نتيجة القوس.

مثال التعبير

$$A - B / ( K * F - X ^ M )$$


تنفيذ العمليات حسب الخطوات التالية:

تأخذ الاقواس الأولوية، وتنفذ العمليات داخلها حسب الأولوية أيضاً.

العملية الاولى : رفع X إلى الاس M لتصبح كمية واحدة.

العملية الثانية : ضرب K في F لتصبح كمية واحدة.

العملية الثالثة : طرح نتيجة العملية الأولى من نتيجة العملية الثانية وتصبح النتيجة كمية واحدة.

العملية الرابعة : تقسم B على نتيجة العملية الثالثة وتصبح النتيجة كمية واحدة.

العملية الخامسة : تطرح نتيجة العملية الرابعة من A وتصبح النتيجة كمية واحدة.

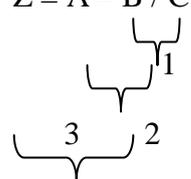
### الجملة الحسابية Arithmetic Statement

الجملة الحسابية في MATLAB تكافئ المعادلة الحسابية في الجبر إلا ان MATLAB تشترط ان يكون اسم المتغير المراد حساب قيمته في الطرف الايسر وحده بدون اشارة بينما يكون التعبير الحسابي (بقية المعادلة) في الطرف الايمن، كما في الامثلة التالية:

1)y=A\*X+B  
2)A=3.14\*R^2

مثال:

أولوية العمليات الحسابية في الجمل الحسابية

$$Z = A - B / C$$


يمكن ملاحظة ان إشارة المساواة تمثل اخر اولوية حسابية بعد انتهاء جميع العمليات الحسابية في الطرف الايمن.

### (ب) المتغيرات الرمزية String Variables

تشبه في تركيبها المتغيرات العددية والفرق الوحيد بينهما هو ان قيمة المتغير الرمزي تكون رمزية (محصورة بين علامتي اقتباس).

### الجملة الرمزية String Statement

تشبه في تركيبها الجملة الحسابية والفرق الوحيد بينهما هو ان المتغير في طرفها الايمن يكون رمزياً (محصورة بين علامتي اقتباس) والتعبير في طرفها الايسر يكون متغير.

والامثلة التالية توضح ذلك:

A='Hameed Abdul-Kareem';

N='Number of Student';

Dept='Computer Science';

ملاحظة: التعابير في الطرف الايمن لا يكون لها قيم حسابية لو استخدمت في عمليات حسابية لانها موضوعة داخل ' ' .

## الاقتارات المكتبية Library Functions

يتوفر في معظم الحاسبات باستخدام لغة MATLAB اقتارات رياضية يكثر استعمالنا لها، مثل الدوال والاقتارات المثلثية واللوغارتمية وغيرها ويمكن استدعائها في اي وقت، ومنها:

| الاقتاران         | المعنى                                 |
|-------------------|--|
| sqrt              | الجذر التربيعي                         |
| abs               | القيمة المطلقة                         |
| exp               | المرفوع إلى قوة بأساس 10               |
| log               | اللوغارتم الطبيعي                      |
| log <sub>10</sub> | اللوغارتم العشري                       |
| log <sub>2</sub>  | اللوغارتم ذو الاساس 2                  |
| sin               | جيب الزاوية                            |
| cos               | جيب تمام الزاوية                       |
| tan               | ظل الزاوية                             |
| atan              | ظل معكوس الزاوية                       |
| fix               | التدوير باتجاه الصفر                   |
| floor             | التدوير باتجاه اللانهاية السالبة       |
| ceil              | التدوير باتجاه اللانهاية الموجبة       |
| round             | التدوير باتجاه اقرب عدد صحيح           |
| mod               | الجزء الصحيح من حاصل القسمة            |
| rem               | بقية القسمة                            |
| sign              | اشارة العدد اذا كانت موجبة، سالبة، صفر |
| imag              | القسم التخيلي                          |
| real              | القسم الحقيقي                          |
| factor            | العوامل الاولية                        |
| isprime           | يعيد true اذا كان العدد اولياً         |
| primes            | ينشئ قائمة بالاعداد الاولية            |
| gcd               | القاسم المشترك الاعظم                  |
| lcm               | المضاعف المشترك الاصغر                 |

مثال:

```
>>x=2.6;
```

```
>>y1=fix(x); y2=floor(x); y3=ceil(x); y4=round(x);
```

```
y1=2
```

```
y2=2
```

```
y3=3
```

```
y4=3
```

س\ ما الفرق بين الدوال الاربعة أعلاه؟

ملاحظة: تأخذ الاقتارات المكتبية اولوية بعد الاقواس عند تنفيذ العمليات الحسابية.

$$\sin(a + b) - m / \sqrt{d}$$

The diagram illustrates the order of operations for the expression  $\sin(a + b) - m / \sqrt{d}$ . It shows that the addition  $a + b$  is performed first, followed by the square root of  $d$ . Then,  $m$  is divided by the result of the square root. Finally, the sine function is applied to the result of  $a + b$ , and the result of the division is subtracted from it.

يكون تنفيذ العمليات الحسابية كما يلي:

العملية الاولى: ايجاد قيمة جمع a مع b .

العملية الثانية: ايجاد قيمة جيب الزاوية لناتج العملية (١).

العملية الثالثة: ايجاد قيمة الجذر التربيعي لـ d .

العملية الرابعة: ايجاد ناتج قيمة ناتج قسمة m على ناتج العملية (٣).

العملية الخامسة: طرح ناتج العملية (٤) من ناتج العملية (٢) وتصبح النتيجة النهائية كمية واحدة (عدداً واحداً).

مثال: تمثل الجمل التالية اقتترانات مكتوبة في الجبر وإزائها قيمتها في MATLAB :

$$b = \text{sqrt}(a^2 + 10) \quad \longleftarrow \quad b = \sqrt{a^2 + 10}$$

$$z = \text{log}(c * x + n * y) \quad \longleftarrow \quad z = \ln(cx + ny)$$

$$y = (\sin(x + n * k))^3 \quad \longleftarrow \quad y = \sin^3(x + nk)$$

$$s = \text{atan}(y / x) \quad \longleftarrow \quad y = \tan^{-1}(y / x)$$

$$r = 2 * \text{sqrt}(\exp(x-5)) \quad \longleftarrow \quad r = 2\sqrt{e^{x-5}}$$

$$t = \text{abs}(x - \text{sqrt}(y)) / (a + m) \quad \longleftarrow \quad t = |x - \sqrt{y}| / (a + m)$$

$$g = p^{3/2} + (a * b / c)^{1/5} \quad \longleftarrow \quad g = p^{3/2} + \sqrt[5]{ab/c}$$

## المحاضرة الثانية

### MATLAB Vectors المتجهات في ماتلاب

هو مصفوفة من الاعداد ذات بعد واحد. يسمح MATLAB بإنشاء نوعين من المتجهات:

- متجهات صفوف Rows vectors
- متجهات اعمدة Columns vectors

متجهات الصفوف يمكن انشائها من خلال وضع عناصر المتجه بأقواس مربعة، نضع فراغ او فارزة بين العناصر.

```
>>R=[7 8 9 10 11]
```

سوف ينفذ MATLAB التعليمة السابقة ويرجع النتيجة التالية:

```
>> R= 7 8 9 10 11
```

المتجهات الاعمدة يمكن انشائها بوضع العناصر بأقواس مربعة، ويفصل عناصر المتجه الفارزة المنقوطة.

```
>> C=[7 ;8 ; 9; 10 ;11]
```

ونائج تنفيذ هذه التعليمة التالي:

```
C= 7
```

```
8
```

```
9
```

```
10
```

```
11
```

#### عنونة عناصر المتجه

يمكن عنونة او فهرسة عناصر المتجه بعدة طرق. العنصر  $i^{\text{th}}$  من المتجه  $V$  يمكن الاشارة له كـ  $V(i)$ .

مثال:

```
>> V=[1 ;2 ;3 ;4 ;5 ;6] ; %Creating a column vector of 6 elements
```

```
>> V(3)
```

سوف ينفذ الماتلاب التعليمة ويرجع النتيجة ادناه

```
ans= 3
```

عندما نعنون المتجه باستخدام النقطتين المتعامدتين (: colon) مثل  $V(:)$  كل عناصر المتجه سوف تطبع على شكل قائمة.

```
>> V(:)
```

```
ans=1
```

```
2
```

```
3
```

```
4
```

```
5
```

ويمكننا MATLAB كذلك من اختيار مدى من العناصر نختارها من المتجه وكالتالي:

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

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

```
>>sub_rv=rv(3:7)
```

```
sub_rv= 3 4 5 6 7
```

```
>>rv(4:end)
```

```
rv= 4 5 6 7 8 9
```

```
>>rv(3.5)
```

```
Error
```

بسبب ان قيمة المتجه قيمة كسرية

```
>>rv(11)
```

```
Error
```

بسبب تجاوز قيمة دليل المتجه (طول المتجه)

### العمليات على المتجهات Operations of Vectors

☒ **جمع وطرح المتجهات:** يمكن جمع وطرح متجهين. ويجب ان يكون كلا المتجهين بنفس الطول وعناصرهما من نفس النوع البياني. مثال:

```
>>A=[7 , 11, 15, 23, 9];
```

```
>>B=[2 , 5, 13, 16 ,20];
```

```
>>C=A+B;
```

```
>>D=A-B;
```

```
>>disp ( C );
```

```
>>disp(D);
```

```
9 16 28 39 29
```

```
5 6 2 7 -11
```

☒ **ضرب المتجه بقيمة عددية:** عند ضرب المتجه بقيمة عددية تسمى scalar multiplication . سوف ينتج متجه جديد من نفس النوع وكل عنصر من المتجه الاصيلي ضرب بالعدد.

مثال:

```
>>v=[12 34 10 8];
```

```
>>m=5 * v
```

```
m=60 170 50 40
```

ملاحظة: يمكن اجراء كل العمليات على المتجه مثل الجمع والطرح والقسمة والاس مع العدد المفرد.

☒ تدوير المتجه Vector transpose : تحول المتجه العمودي إلى متجه افقي وبالعكس باستخدام علامة الاقتباس المفردة (\*)

مثال:

```
>>r=[1 2 3 4]
```

```
r= 1 2 3 4
```

```
>>tr = r '
```

```
tr= 1
```

```
2
```

```
3
```

```
4
```

```
>>v=[1 ;2 ;3 ;4]
```

```
v=1
```

```
2
```

```
3
```

```
4
```

```
>> tv = v '
```

```
tv= 1 2 3 4
```

```
>>disp(tr);
```

```
1
```

```
2
```

```
3
```

```
4
```

```
>>disp(tv);
```

```
1 2 3 4
```

## المحاضرة الثالثة

☒ إضافة (دمج) المتجهات **Appending vectors**: يمكننا MATLAB من إضافة متجهين لإنشاء متجه جديد.

مثال:

```
>>r1=[1 2 3 4];
```

```
>>r2=[5 6 7 8];
```

```
>>r=[r1,r2] %append the vector r1 to vector r2 to create new vector r with two different number of elements n,m.
```

```
r= 1 2 3 4 5 6 7 8
```

```
>>rMat= [r1;r2]
```

```
rMat= 1 2 3 4
```

```
5 6 7 8
```

مثال:

```
>>c1=[1 ;2 ;3 ;4];
```

```
>>c2=[5 ;6 ;7 ;8];
```

```
>>c=[c1;c2]
```

```
c= 1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
6
```

```
7
```

```
8
```

```
>>cMat=[c1,c2]
```

```
cMat=1 5
```

```
2 6
```

```
3 7
```

```
4 8
```

☒ **ضرب المتجهات : Vector Dot Product** ضرب متجهين  $a=(a_1, a_2, \dots, a_n)$  والمتجه  $b=(b_1, b_2, \dots, b_n)$  يعطى بالمعادلة :

$$a \cdot b = \sum (a_i \cdot b_i)$$

```
>>v1=[2 3 4];
```

```
>>v2=[1 2 3];
```

```
>>dp=dot(v1,v2);
```

```
>>disp(' Dot product:');
```

```
>>disp(dp);
```

Dot product:

20

☒ **انشاء متجه بعناصر ذا مدى متساوي Vectors with uniformly Spaced Elements** يمكن انشاء متجه  $V$  واول عنصر به  $F$  واخر عنصر به  $L$ ، والفرق بين عناصره اي رقم حقيقي  $N$ ، وكالتالي:

$$V=[F : N: L]$$

مثال:

```
>>V=[1 : 2: 20];
```

```
>>sqv=V.^2;
```

```
>>disp(V);
```

```
>>disp(sqv);
```

1 3 5 7 9 11 13 15 17 19

1 9 25 49 81 121 169 225 289 361

## المحاضرة الرابعة

### MATLAB Matrix مصفوفات في ماتلاب

مصفوفة ذات بعدين من الاعداد. يمكن انشاء المصفوفة بإدخال عناصر المصفوفة في كل سطر بينها فراغ او فارزة واستخدام الفارزة المنقوطة عند نهاية كل سطر.

مثال:

```
>>a=[1 2 3 4 5; 2 3 4 5 6; 3 4 5 6 7]
```

```
1 2 3 4 5
a=2 3 4 5 6
3 4 5 6 7
```

عنونة عناصر المصفوفة: لعنونة عنصر السطر  $m^{\text{th}}$  والعمود  $n^{\text{th}}$  للمصفوفة  $M_x$  نكتب :  $M_x(m,n)$

مثال:

```
>>a=[1 2 3 4 5; 2 3 4 5 6; 3 4 5 6 7; 4 5 6 7 8];
```

```
>>a(2,5)
```

```
a=6
```

لعنونة كل العناصر في العمود  $m^{\text{th}}$  نكتب  $A(:,m)$  ، مثلاً لننشئ متجه عمودي  $v$  من عناصر العمود الرابع من المصفوفة  $a$  وكالتالي:

```
>>a=[1 2 3 4 5; 2 3 4 5 6; 3 4 5 6 7; 4 5 6 7 8];
```

```
>>v= a(:,4)
```

```
v=4
```

```
5
```

```
6
```

```
7
```

كذلك يمكن اختيار الاعمدة من  $m^{\text{th}}$  إلى  $n^{\text{th}}$  وكالتالي :  $A(:,m:n)$

مثال:

```
>>a=[1 2 3 4 5; 2 3 4 5 6; 3 4 5 6 7];
```

```
>>a(:,2:3)
```

```
2 3
ans=3 4
4 5
```

بنفس الطريقة يمكننا إنشاء مصفوفة فرعية بأخذ جزء من المصفوفة الرئيسية.

مثال:

```
>>a=[1 2 3 4 5; 2 3 4 5 6; 3 4 5 6 7; 4 5 6 7 8];
```

```
>>sa= a(2:3,2:4)
```

```
sa= 3 4 5
     4 5 6
```

حذف سطر او عمود من المصفوفة: يمكن حذف صف او عمود من المصفوفة بالتنسيق المجموعة الخالية [] لذلك السطر او العمود.

مثال: لحذف السطر الرابع من المصفوفة a .

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

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

```
 1 2 3
a=2 3 4
 3 4 5
```

مثال: جعل جميع عناصر العمود الثالث تكون 3

```
>>a(:, 3)=3
```

```
 1 2 3
a=2 3 3
 3 4 3
```

مثال: ولحذف العمود الثالث من المصفوفة a.

```
>>a=[1 2 3; 2 3 4; 3 4 5];
```

```
>>a(:, 3)=[]
```

```
 1 2
a=2 3
 3 4
```

مثال: طباعة العنصر في الموقع (3, 3) a(3, 3)

```
>>a(3, 3)
```

```
ans = 5
```

ملاحظة: تقدم لغة MATLAB طريقة اخرى للإشارة إلى عناصر المصفوفة باستخدام رقم واحد فقط، ولفهم هذه الطريقة يجب التخيل بأن جميع عناصر المصفوفة مرتبة بشكل عمود واحد مكون من اعمدة المصفوفة من الاعلى إلى الاسفل (اي عناصر العمود الاول ثم الثاني ثم الثالث وهكذا).

مثال:

```
>>a(6)
```

```
ans= 4
```

مثال:

```
>>h=[1,2,3;4,5,6,7];
```

```
Error لان الاعمدة غير متساوية
```

ملاحظة: لاحظ الفرق بين:

عنصر → half= a(2,2);

مصفوفة → full=a;

## العمليات على المصفوفات Matrix Operations

- جمع وطرح المصفوفات: يجب ان تكون المصفوفتان بنفس عدد الاسطر والاعمدة.

مثال:

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

```
>>b=[7 5 6; 2 0 8; 5 7 1];
```

```
>>c=a+b
```

```
>>d=a-b
```

```
 8  7  9
c= 6  5 14
 12 15 10
```

```
 -6  -3  -3
d=  2   5  -2
   2   1   8
```

- العمليات المفردة على المصفوفة Scalar Operation: جمع، طرح، ضرب، قسمة المصفوفة بعدد مفرد يسمى scalar operation.

مثال:

```
>>a=[10 12 23; 14 8 6; 27 8 9];
```

```
>>b=2;
```

```
>>c=a+b
```

```
>>d=a-b
```

```
>>e=a*b
```

```
>>f=a/b
```

```
 12  14  25
c=16 10  8
 29 10 11
```

```
 8  10  21
d=12 6  4
 25 6  7
```

```
 20  24  46
e=28 16 12
 54 16 18
```

```
 5.0000  6.0000 11.5000
f= 7.0000  4.0000  3.0000
 13.5000  4.0000  4.5000
```

سؤال واجب....أوجد  $(b - a) * 2$  ؟

- تدوير المصفوفة Transpose of Matrix : يتم تبديل الصفوف بالأعمدة في المصفوفة باستخدام علامة الاقتباس المفردة.

مثال:

```
>>a=[10 12 23; 14 8 6; 27 8 9]
```

```
>> b=a'
```

```
10 12 23
a=14 8 6
27 8 9
```

```
10 14 27
b=12 8 8
23 6 9
```

▪ **دمج المصفوفات Concatenating Matrix:** يمكن دمج مصفوفتان لإنشاء مصفوفة أكبر باستخدام الأقواس المربعة [] كعامل للدمج.

مثال:

```
>>a=[10 12 ; 14 8 ]
```

```
>>b=[12 31; 8 0]
```

```
>>c=[a, b] %Horizontal Concatenation
```

```
>>d=[a; b] %Vertical Concatenation
```

```
a=10 12
14 8
```

```
b=12 31
8 0
```

```
c=10 12 12 31
14 8 8 0
```

```
d=10 12
14 8
12 31
```

```
8 0
```

## المحاضرة الخامسة

### ضرب المصفوفات : Matrix multiplication

لو كان لدينا المصفوفتان A و B وإذا كان ابعاد A هي  $n*m$  و B ابعادها  $m*p$  يمكن ان ينتج من حاصل ضربيهما المصفوفة C ذات البعد  $n*p$  ويجب ان يكون عدد الاعمدة m في A يساوي عدد الصفوف m في B.

يمكن شرح السابق كالتالي:

إن شرط ضرب المصفوفات رياضياً هو ان تكون **اعمدة المصفوفة الاولى تساوي صفوف المصفوفة الثانية** فيكون الناتج ابعاد المصفوفة الجديدة هي **صفوف المصفوفة الاولى × اعمدة المصفوفة الثانية**.

**الطريقة:**

1. نضرب الصف الاول في المصفوفة الاولى × العمود الاول في المصفوفة الثانية مجموع الضرب يكون قيمة عنصر الصف الاول العمود الاول
2. نضرب الصف الاول في المصفوفة الاولى × العمود الثاني في المصفوفة الثانية مجموع الضرب يكون قيمة عنصر الصف الاول العمود الثاني
3. نضرب الصف الاول في المصفوفة الاولى × العمود الثالث في المصفوفة الثالثة مجموع الضرب يكون قيمة عنصر الصف الاول العمود الثالث.
4. نضرب الصف الثاني في المصفوفة الاولى × العمود الاول في المصفوفة الثانية مجموع الضرب يكون قيمة عنصر الصف الثاني العمود الاول.
5. نضرب الصف الثاني في المصفوفة الاولى × العمود الثاني في المصفوفة الثانية مجموع الضرب يكون قيمة عنصر الصف الثاني العمود الثاني.

وهكذا.....

**مثال:**

```
>>a=[1 2 3;2 3 4;1 2 5]
```

```
>>b=[2 1 3;5 0 -2;2 3 -1]
```

```
>>res=a*b
```

|      |   |   |     |   |    |        |    |    |
|------|---|---|-----|---|----|--------|----|----|
| 1    | 2 | 3 | 2   | 1 | 3  | 18     | 10 | -4 |
| a= 2 | 3 | 4 | b=5 | 0 | -2 | res=27 | 14 | -4 |
| 1    | 2 | 5 | 2   | 3 | -1 | 22     | 16 | -6 |

**ملاحظة:** يمكن ضرب كل عنصر بالعنصر المناظر له من المصفوفة الاخرى او قسمته شرط ان تُسبق إشارة الضرب او القسمة بنقطة كما في الشكل:

```
>>g=[1 2 3;5 6 7;9 10 11];
```

```
>>h=[1 1 1;2 2 2;3 3 3];
```

```
>>g.*h
```

```
    1  2  3  
ans=10 12 14  
    27 30 33
```

قسمة المصفوفتين:

```
>>g./h
```

```
    1.000  2.000  3.000  
ans=2.500  3.000  3.500  
    3.000  3.333  3.666
```

وكذلك الرفع للاس كما في المثال التالي:

```
>>g.^2
```

```
    1  4  9  
ans=25 36 49  
    81 100 121
```

```
>>g*h
```

ERROR

لان اعمدة المصفوفة الاولى لا تساوي عدد اسطر المصفوفة الثانية

**محدد المصفوفة Determinant of a Matrix** : باستخدام الدالة det يمكن حساب محدد المصفوفة وكالتالي:

$$\begin{array}{l} \text{محدد المصفوفة الثنائية: } \begin{array}{cc} a & b \\ c & d \end{array} \text{ - } ad-bc \\ \text{محدد المصفوفة الثلاثية: } \begin{array}{ccc} a & d & g \\ b & e & h \\ c & f & i \end{array} \text{ - } (aei+dhc+gbf)-(dbi+ahf+gec) \end{array}$$

مثال:

```
>> a=[1 2 3;2 3 4;1 2 5];
```

```
>> det(a)
```

```
ans = -2
```

**معكوس المصفوفة The Inverse of a Matrix**: يشار إلى معكوس المصفوفة بالرمز  $A^{-1}$  والعلاقة بين المصفوفة والمعكوس كالتالي (  $AA^{-1}=A^{-1}A=1$  ) ولحساب معكوس المصفوفة نستخدم الدالة  $\text{inv}(A)$  ويجب ان تكون المصفوفة مربعة.

```
>>a=[1 2 3;2 3 4;1 2 5];
```

```
>> inv(a)
```

ans=

|        |        |        |
|--------|--------|--------|
| -3.500 | 2.000  | 0.500  |
| 3.000  | -1.000 | -1.000 |
| -0.500 | 0      | 0.500  |

## المحاضرة السادسة

### المصفوفات القياسية

يمكنك برنامج matlab من إنشاء مصفوفات قياسية وذلك لتمتع تلك المصفوفات بخواص وميزات خاصة وتتضمن ايضاً المصفوفات التي جميع عناصرها صفرية او مساوية للواحد ومصفوفات الاعداد العشوائية والمصفوفات القطرية والمصفوفات التي عناصرها اعداد ثابتة.

الدالة `zeros()` تولد مصفوفة من الازرار

مثال:

```
>> zeros(5,3)
```

```
    0 0 0
ans= 0 0 0
    0 0 0
```

الدالة `ones()` تولد مصفوفة من الواحدات

مثال:

```
>> ones(3,4)
```

```
    1 1 1 1
ans= 1 1 1 1
    1 1 1 1
```

الدالة `eye()` تولد مصفوفة الوحدة (`identity`) والتي تكون جميع عناصرها ازرار ما عدا القطر الرئيسي يكون واحداً

مثال:

```
>> eye(3)
```

```
    1 0 0
ans= 0 1 0
    0 0 1
```

الدالة `rand()` تولد مصفوفة عناصرها اعداد عشوائية بين (0-1)

مثال:

```
>> rand(3,2)
```

```
    0.8147 0.9134
ans= 0.9058 0.6324
    0.1270 0.0975
```

**ملاحظة:** عند كتابة معامل واحد للدالة تكون المصفوفة مربعة (عدد الصفوف يساوي عدد الاعمدة) بقيمة المعامل المعطى اما عند كتابة معاملين اثنين للدالة يكون المعامل الاول هو عدد الصفوف والمعامل الثاني هو عدد الاعمدة.

### مصفوفة المربع السحري Magic Square Array

هي مصفوفة عندما يتم جمع عناصر الصف او جمع عناصر العمود او القطر تعطي نفس الناتج، ودالة magic() تولد مصفوفة النربع السحري بشرط ان يكون المعامل قيم مفردة اكبر او تساوي 3.

مثال:

```
>> magic(4)
```

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

دالة diag() تطبع عناصر القطر الرئيسي

دالة eig() تستخدم لحساب القيم الذاتية للمصفوفة

دالة trace() تستخدم لحساب مجموع عناصر القطر الرئيسي للمصفوفة

دالة triu() تستخدم لاستخلاص جزء المثلثية العليا (upper) من المصفوفة

دالة tril() تستخدم لاستخلاص جزء المثلثية السفلى (lower) من المصفوفة

### دوال المصفوفات والمتجهات Arrays and Vectors Functions

| Functions | Purpose  |
|-----------|--|
| length    | length of vector or largest array dimension<br>تعيد هذه الدالة في المتجهات طول المتجه (عدد العناصر) اما في المصفوفات فأنها تعيد اكبر بعد بين البعدين |
| ndims     | number of array dimensions<br>تعيد عدد ابعاد المصفوفة  |
| numel     | number of array elements<br>تعيد عدد عناصر المصفوفة (اي حاصل ضرب ناتج دالة size)   |
| size      | array dimension<br>قيم ابعاد المصفوفة (اي عدد الصفوف وعدد الاعمدة)   |
| diag      | diagonal matrices and diagonals of matrix  |

| Functions | Purpose  |
|-----------|--|
| circshift | shifts array circularly<br>دوال الازاحة الحلقية  |
| fliplr    | flips matrix from left to right<br>تقلب صفوف المصفوفة (تقلب المصفوفة من اليسار إلى اليمين) |
| flipud    | flips matrix from up to down<br>تقلب اعمدة المصفوفة (تقلب المصفوفة من الاعلى إلى الاسفل)   |
| reshape   | reshape array  |
| sort      | sorts array elements in ascending or descending order                                      |
| transpose | transpose<br>تستخدم لتدوير المتجه او المصفوفة مثل ايعاز ' كل صف يصبح عمود وكل عمود يصبح صف |
| max       | largest elements in array  |
| min       | smallest elements in array   |
| sum       | sum of array elements  |
| prod      | product of array elements  |

مثال:

```
>> x=[7.1, 3.4, 7.2, 28/4, 3.6, 17, 9.4, 8.9]
```

```
x=7.100 3.400 7.200 7.000 3.600 17.000 9.400 8.900
```

```
>> z=size(x)
```

```
z= 1 8 (ابعاد المصفوفة)
```

```
>> y=length(x)
```

```
y= 8 (طول المتجه)
```

```
>> w=ndims(x)
```

```
w= 2 (عدد ابعاد المصفوفة)
```

```
>> v=numel(x)
```

```
v=8 (عدد عناصر المصفوفة)
```

```
>> reshape(x,[4,2])
```

ans = 7.100 3.600

3.400 17.000

7.200 9.400

7.000 8.900

ملاحظة: كون عدد عناصر المصفوفة هو ثابت لا يمكن التلاعب به لذا عند استخدام دالة reshape لتغيير شكل المصفوفة يجب ان يكون المعاملات الجديدة المستخدمة تنتج نفس عدد عناصر المصفوفة الاصيلي اي مثلاً لا يمكن ان نكون مصفوفة جديدة بالمعاملات ٣:٢ والمصفوفة الاصلية معاملاتها ٢:٢ .

## المحاضرة السابعة

### دوال الازاحة الحلقية (الدائرية) لعناصر المصفوفة:

إن الإيعاز المستخدم لترتيب عناصر المصفوفة كإزاحة حلقية هو circshift ويجب أن نلاحظ التالي:

1. عند كتابة رقم فقط في المعامل الثاني للإيعاز يقوم بإنزال الصفوف حسب الرقم.
2. عند كتابة رقمين في المعامل الثاني للإيعاز يستخدم الرقم الأول لإنزال الصف حسب الرقم بينما الرقم الثاني يستخدم لتحريك عناصر الصف حسب الإشارة إذا كان سالب ندفع القيم من اليمين إلى اليسار كأنما تصبح القيم المدفوعة سالبة الإشارة وإذا كان موجب ندفع القيم من اليسار إلى اليمين كأنما تصبح القيم المدفوعة موجبة الإشارة.

ex:

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

```
>> b=circshift(a,1);
```

```
>> c=circshift(a,1,-10);
```

```
>> a
```

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

```
>> b
```

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

تم ازاحة الصف الاول بمقدار واحد وحسب المعامل الثاني بالايغاز والصف الاخير بالمصفوفة صعد كصف أول (ازاحة حلقية)

```
>> c
```

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

تم ازاحة الصف الاول بمقدار واحد وحسب الرقم الاول للمعامل الثاني بينما تم دفع الصفوف من اليمين إلى اليسار بمقدار واحد لوجود إشارة سالب بالرقم الثاني للمعامل الثاني.

### ايغاز الترتيب التصاعدي (sort):

إن إيعاز sort يستخدم لترتيب عناصر المصفوفة تصاعدياً وفي حالته الاعتيادية يتم كتابته كالتالي (a) sort فيقوم بترتيب عناصر كل عمود من المصفوفة a تصاعدياً وعند كتابة الإيعاز كالتالي (a,1) sort ايضاً يقوم بترتيب عناصر كل عمود بالمصفوفة تصاعدياً بينما اذا كتب الإيعاز كالتالي (a,2) sort اذن سيقوم بترتيب عناصر كل صف ترتيب تصاعدي.

```
>> m=[2 6 4;5 3 9;2 0 1]
```

```
 2 6 4
m=5 3 9
 2 0 1
```

```
>> s=sort(m,1)
```

```
 2 0 1
s=2 3 4
 5 6 9
```

```
>> t=sort(m,2)
```

```
 2 4 6
t=3 5 9
 0 1 2
```

>> k=flipr(m) "يقرب العناصر الصف" يقرب المصفوفة من اليسار إلى اليمين

```
 4 6 2
k=9 3 5
 1 0 2
```

>> l=flipud(m) "يقرب العناصر العمود" يقرب المصفوفة من الأعلى إلى الأسفل

```
 2 0 1
l=5 3 9
 2 6 4
```

### فوائد النقاط المتعامدة ( : )

a=1:10

١. تستخدم لإنشاء متجه

٢. تستخدم لإنشاء متجه ذو قفزات ثابتة "دورات" مثلاً يكون متجه ذو اعداد زوجية او فردية او مضاعفات عدد معين.

مثال: كون متجه لمضاعفات العدد ٣ للارقام المحصورة بين ١ و ١٠

```
>> a=3:3:10
```

```
a=3 6 9
```

```
>> b=10:-2:0
```

```
b=10 8 6 4 2 0
```

```
>> y=[2 4 9 5 0 3];
```

```
>> y([1 1 6 3 4])
```

ans=

تم وضع قيمة العنوان المعطى في السؤال لان عندما يكتب اقواس مربعة يعني عنوان 2 2 3 9 5

٣. عند كتابة هذا الابعاز باستخدام النقاط المتعامدة a(:,j) اي كل قيم الصفوف بالعمود j المعطى بينما الابعاز a(I,:) كل قيم الاعمدة المقابلة للصف j المعطى.

٤. الابعاز a(:,j) كل قيم الصفوف والاعمدة بينما الابعاز a(i,:) يعطي جميع عناصر المصفوفة وبشكل عمودي.

٥. الابعاز a(i:j:k) كل عناصر الصفوف والمقابلة للمنطقة المحصورة بين j و k من الاعمدة.

مثال:

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

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

```
>> a(:,2)
```

```
2
ans= 5
8
```

```
>> a(:,2:3)
```

```
2 3
ans= 5 6
8 9
```

```
>> a(2:3,2:3)
```

```
5 6
ans= 8 9
```

### المصفوفات المنطقية:

هناك المصفوفات المنطقية الناتجة عن العمليات المنطقية أي يكون عناصرها اما واحد اذا كان قيمة العملية true او عناصرها صفر اذا كان قيمة العملية false.

مثال:

```
>> abs(a) > 6
```

```
0 0 0 0
ans= 0 0 0 1
1 1 1 1
```

ويمكن تحديد موقع او دليل العناصر التي تحقق شرط معين والموجودة ضمن مصفوفة معينة من خلال الابعاز find والذي يعيد اليك موقع العناصر الذي تكون نتيجة تحقيقه لشرط ما true.

مثال:

```
>> [i,j]=find(a>6)
```

```
3
```

```
i=3
```

```
3
```

```
2
```

```
3
```

```
1
```

```
j=2
```

```
3
```

```
4
```

```
4
```

**دوال ارجاع العنصر الاكبر والاصغر في المصفوفة:**

إن إيعازي  $\max$  ,  $\min$  تعيد العنصر الاصغر والاكبر ضمن عناصر العمود الواحد بالمصفوفة وكالتالي  $\min(a)$  ولكن عند كتابة الإيعاز هكذا  $\max(a')$  فيقوم بإرجاع العنصر الاكبر ضمن كل صف من المصفوفة اما عند كتابة الإيعاز هكذا  $\min(\min(a))$  فسيقوم بإرجاع العنصر الاصغر بكل المصفوفة وعند كتابة الإيعاز هكذا  $[mm,r]=\min(a)$  سيقوم بإرجاع العنصر الاصغر لكل عمود وموقعه ضمن الصفوف.

مثال:

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

```
1 2 3
```

```
a=4 5 6
```

```
7 8 9
```

```
>>max(a)
```

```
ans= 7 8 9
```

```
>>min(a)
```

```
ans= 1 2 3
```

```
>> max(a')
```

```
ans= 3 6 9
```

```
>> [mm,r]=min(a)
```

```
ans=
```

mm= 1 2 3

r= 1 1 1

>> max(max(a))

ans= 9

\*\*إن إيعاز sum يقوم بإيجاد جمع عناصر كل عمود ضمن المصفوفة ولكن عند استخدام الإيعاز sum بهذه الطريقة (sum(a,2) او يكتب sum(a') يقوم بإيجاد جمع عناصر كل صف ضمن المصفوفة وعند استخدام الإيعاز sum(sum(a)) يقوم بإيجاد مجموع كل عناصر المصفوفة.

>> sum(a)

ans= 12 15 18

>> sum (a,2)

ans= 6 15 24

>> sum(sum(a))

ans= 45

\*\* إن إيعاز prod يعمل مثل إيعازي max , min ولكن بإيجاد حاصل ضرب كل عمود بالمصفوفة وبإضافة ' على اسم المصفوفة يمكن إيجاد حاصل ضرب عناصر كل صف وباستخدام الإيعاز مرتين يمكن إيجاد حاصل ضرب عناصر كل المصفوفة.

## المحاضرة الثامنة

### جمل الإدخال والإخراج

جمل الإدخال: هناك عدة صيغ للإدخال بالإضافة إلى عملية التنسيب ومنها تعليمة input :

مثال (للدلالة على ادخال)

```
>> x=input ('enter x:')
```

```
enter x
```

مثال (لإدخال أسماء رمزية)

```
clc;
```

```
clear;
```

```
z=input('enter name','s');
```

ملاحظة: يفضل كتابة البرامج في محرر برنامج ماتلاب (نافذة Editor)

جمل الإخراج: هناك عدة صيغ للإخراج منها:

١. تعليمة disp :

```
1)) >> d=15;
```

```
>> disp(d);
```

```
15
```

```
2)) >>a='ali';
```

```
>>disp(a);
```

```
ali
```

```
3)) >>sum=9.8;
```

```
>>disp(['sum=',num2str(sum)]);
```

```
sum=9.8
```

ملاحظة: يجب ان يكون محتويات disp قيمة ذات نوع بياني واحد ضمن الجملة الواحدة (كل جملة نوع بياني واحد)، وعندما تكون محتوياتها اكثر من قيمة ذات نوع بياني يجب ان تجمع القيم في قوسين كبيرين [ ] كما في المثال السابق.

٢. تعليمة msgbox :

```
>> msgbox('ok','result')
```

الشيء المطلوب طباعته (نوع بياني رمزي)

عنوان الصندوق

ملاحظة: يمكن طباعة الاعداد والاسماء والنتائج من خلال كتابة الايعازات بدون فارزة منقوطة وستظهر النتائج في نافذة الامر command window.

سؤال: ما الفرق بين disp, display؟؟ نفس الايعاز لا يوجد اي فرق.

## الجمل الشرطية

يدعم برنامج ماتلاب العمليات المنطقية والمقارنة مثلما يدعم العمليات الرياضية وتهدف العمليات والمعاملات المنطقية الحصول على اجوبة للأسئلة التي يجاب عنها بصح او خطأ (True/ False).

تعتبر لغة ماتلاب في تعاملها مع جميع التعبيرات المنطقية وعمليات المقارنة ان اي عدد غير صفري هو True ويعتبر الصفر False، وكما يكون اخراج جميع التعبيرات المنطقية وعمليات المقارنة عبارة عن مصفوفات منطقية تحوي العدد واحد من اجل True والغعد صفر من اجل False.

وتعتبر المصفوفات المنطقية نوعاً خاصاً من المصفوفات العددية كما يمكن عنونة المصفوفة المنطقية بنفس طريقة عنونة باقي المصفوفات التي استخدمها سابقاً ضمن التعبيرات العددية.

## معاملات المقارنة (العوامل العلائقية) : Relational Operators

تتضمن معاملات المقارنة كل اشارات المقارنة الشائعة والمدرجة في الجدول التالي:

| الوصف                           | معامل المقارنة |
|---------------------------------|----------------|
| اصغر من                         | <              |
| اصغر او يساوي                   | <=             |
| اكبر من                         | >              |
| اكبر او يساوي                   | >=             |
| اشارة المساواة(لكي نميزها عن =) | ==             |
| اشارة عدم المساواة              | ~=             |

يمكن استخدام معاملات المقارنة للمقارنة بين مصفوفتين لها نفس الحجم، او للمقارنة بين مصفوفة وعدد مفرد وتتم هذه الحالة مقارنة كل عنصر من المصفوفة مع العدد المفرد، وتكون المصفوفة الناتجة بنفس حجم المصفوفة التي تمت مقارنتها كما يبينه المثال التالي:

مثال (1) :

```
>> a=1; b=5;
```

```
>> x=a >b
```

```
x=0
```

```
>> A=1:9;B=9-A
```

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

```
B=8 7 6 5 4 3 2 1 0
```

```
>> tf=A>4
```

```
tf= 0 0 0 0 1 1 1 1 1
```

لقد اوجدنا العناصر من A التي هي اكبر من 4 وتظهر الازهار في المصفوفة الناتجة في مواقع العناصر عندما  $A \leq 4$  بينما يظهر الرقم 1 عندما  $A > 4$

```
>> tf=(A==B)
```

```
tf=0 0 0 0 0 0 0 0
```

لقد تم هنا إيجاد عناصر A التي تساوي العناصر في المصفوفة B .

ملاحظة: لاحظ بأن الاشارتين (=) و(==) تعنيان شيئاً مختلفاً حيث يقوم (==) بمقارنة متغيرين وتعيد العدد واحد اذا كانا متساويين وصفرًا اذا لم يكونا متساويين بينما تستخدم (=) لاسناد اخراج العملية إلى متغير.

مثال (1): لتوليد مصفوفة احادية منطقية عناصرها واحدات (في حالة اكبر من thr) واصفارةً (في حالة اصغر من او تساوي thr).

```
>> indent=[10 17 22 0 7 3 2];
```

```
>> thr=7;
```

```
>> y=(indent > thr)
```

```
y=1 1 1 0 0 0 0
```

مثال (2): لتوليد مصفوفة احادية عناصرها نفس العناصر (في حالة اكبر من thr) واصفارةً (في حالة اصغر من او تساوي thr).

```
>> z=indent.*(indent > thr)
```

```
z=10 17 22 0 0 0 0
```

## المعاملات المنطقية (العوامل المنطقية) :Logical Operators

توفر المعاملات المنطقية طريقة لدمج او نفي تعابير المقارنة، ويظهر الجدول التالي المعاملات المنطقية الموجودة في لغة ماتلاب:

|             |         |           |
|-------------|---------|-----------|
| ~ (نفي) NOT | (أو) OR | & (و) AND |
|-------------|---------|-----------|

| not | or | and | نتائج المقطع الثاني | نتائج المقطع الاول |
|-----|----|-----|---------------------|--------------------|
| 1   | 0  | 0   | 0                   | 0                  |
| 0   | 1  | 0   | 1                   | 0                  |
| 0   | 1  | 0   | 0                   | 1                  |
| 0   | 1  | 1   | 1                   | 1                  |

وسنقدم لك فيما يلي بعض الامثلة على استخدام المعاملات المنطقية:

```
>> a=1;
```

```
>> b=5;
```

```
>>x=a~=b
```

```
x=1
```

```
>>b=(1==1)&(2~=3)
```

```
b=1
```

```
>>b=(1==1)|(2~=3)
```

```
b=1
```

```
>>b=(1==1)&not((2~=3))
```

```
b=0
```

```
>>A=1:9;B=9-A;
```

```
>>tf=A>4
```

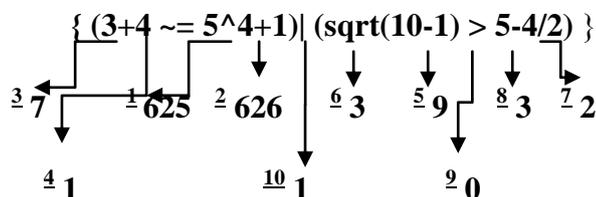
tf=0 0 0 0 1 1 1 1 1 1 حيث قام بإيجاد عناصر المتجه التي قيمها اكبر من 4

```
>>tf=~(A>4)
```

tf=1 1 1 1 0 0 0 0 0 0 لقد قام البرنامج بقلب النتيجة السابقة وتعني استبدال مواقع الازهار والواحدات

```
>>tf=(A>2)&(A<6)
```

tf=0 0 1 1 1 0 0 0 0 0 حيث تعيد هذه العبارة العدد واحد عندما يكون العنصر من المتجه اكبر من 2 واقل من 6



مثال: جد ناتج التعبير الحسابي التالي

## اسبقية المعامل

يقوم البرنامج بإيجاد قيمة تعبير مستنداً إلى مجموعة من القواعد الناظمة لاسبقية المعامل وتحسب المعاملات ذات الاسبقية العليا قبل المعاملات ذات الاسبقية العليا قبل المعاملات ذات الاسبقية الدنيا وتقيم المعاملات ذات الاسبقية المتساوية من اليسار إلى اليمين. ويشرح الجدول التالي قواعد اسبقية المعامل التي يعتدها برامج ماتلاب.

| مستوى الاسبقية | المعامل  |
|----------------|--|
| الأعلى         | الاقواس ( )  |
|                | المدور (')، القوة ( ^ ، .^ )   |
|                | إشارة النفي (~)  |
|                | الضرب (* ، .* )، القسمة (/ ، ./)   |
|                | الجمع (+)، والطرح (-)  |
|                | معامل النقطتين المتعامدتين ( : )   |
|                | اصغر من <، واصغر او يساوي <=، اكبر من >، اكبر من او يساوي >=،<br>المساواة ==، عدم المساواة ~ |
|                | الجمع المنطقي (&) AND  |
| الأدنى         | المعامل المنطقي ( ) OR   |

## الصيغة IF-ELSE-END

قد نحتاج إلى حساب مجموعة من اوامر استناداً إلى اخراج ناتج عن اختبار شرطي. وتنفذ هذه التعليمة في لغة ماتلاب عبر استخدام الصيغة if-else-end وكما يلي:

**if** expression

(commands)

**end**

وسينفذ الاوامر (commands) الواقعة بين العبارتين if و end إذا كانت قيمة التعبير (expression) تكون true واليك المثال التالي:

```
>> x=10;
```

```
>> if x==10
```

```
disp('ok')
```

```
end;
```

وإذا كان لدينا خياران، فتصبح الصيغة if-else-end كما يلي:

**if** expression

(commands evaluated if True)

**else**

(commands evaluated if False)

**end**

حيث ستنفذ المجموعة الأولى من الأوامر في حال امتلاك التعبير expression القيمة true بينما تنفذ المجموعة الثانية إذا امتلك التعبير expression القيمة false.

وإذا كانت هناك عدة حالات، فسنأخذ التعبير if-else-end الشكل التالي:

**if** expression1

(commands evaluated if expression1 is True)

**else if** expression2

(commands evaluated if expression2 is True)

**else if** expression3

(commands evaluated if expression3 is True)

**else if** expression4

(commands evaluated if expression4 is True)

.

**else**

(commands evaluated if no other expression is True)

**end**

مثال (١): برنامج لطباعة OK إذا كانت قيمة x المدخل هو ١٠ :

```
>> x=input('enter x:');
```

```
>> if x==10
```

```
msgbox('ok','result');
```

مثال (٢): برنامج لطباعة ok إذا كانت قيمة  $x=10$  والا لطباعة no:

```
>> x=input('enter x:');  
  
>> if x==10  
  
    msgbox('ok','result');  
  
else  
  
    msgbox('no','result');  
  
end;
```

مثال (٣): برنامج لقراءة عدد  $x$  وطباعة ١ إذا كانت  $x=1$ ، ٢ إذا كانت  $x=2$ ، ٣ إذا كانت  $x=3$ :

```
>> x=input('enter x:');  
  
>> if x==1  
  
    disp('1');  
  
else if x==2  
  
    disp('2');  
  
else  
  
    disp('3');  
  
end;  
  
end;
```

## الصيغة SWITCH-CASE

عندما يتوجب علينا تنفيذ اوامر اعتماداً على استخدام متكرر لاختيار كمي لوسط ما، عندها من السهل استخدام الصيغة switch-case التي لها الصيغة العامة التالية:

```
switch expression  
  
case test-expression1  
  
    (commands1)  
  
case test-expression2
```

(commands2)

**otherwise**

(commands3)

**end**

يجب ان يكون expression هنا اما عدداً مفرداً او سلسلة رمزية يقارن التعبير expression الموجود في الصيغة السابقة بالتعبير test-expression1 الموجود في عبارة case الاولى. وإذا تساوى التعبيران، سيتم تنفيذ الاوامر (commands1) وتخطي التعليمات الواقعة بعدها حتى العبارة end. اما اذا لم يتحقق الشرط الاول، فسيختبر الشرط الثاني حيث سيقارن expression في المثال السابق مع العبارات test-expression2 الموجودة في عبارة case الثانية. وإذا تساوى التعبيران، سيتم تنفيذ (commands2) وتهمل بقية العبارات حتى عبارة end. اذا لم تحقق اي عبارة case المساواة مع التعبير expression عندها ستنفذ الاوامر (commands3) التي تلي العبارة otherwise.

لاحظ من الشرح الذي اوردناه عن صيغة switch-case بأن سيتم تنفيذ احدى مجموعات الاوامر المكونة للصيغة switch-case واليك الامثلة التالية.

مثال (١) اكتب برنامج بلغة ماتلاب لقراءة عدد x وطباعة :

'1..5' اذا كان العدد المقروء (المدخل) ١ أو ٢ أو ٣ أو ٤ أو ٥

'9..10' اذا كان العدد المقروء (المدخل) ٩ أو ١٠

'this is impossible' اذا كان العدد المقروء (المدخل) غير ذلك.

```
x=input('enter x:');
```

```
switch x
```

```
case {1,2,3,4,5}
```

```
disp('1..5');
```

```
case {9,10}
```

```
disp('9..10');
```

```
otherwise
```

```
disp('this is impossible');
```

```
end;
```

مثال (٢) اكتب برنامج بلغة ماتلاب لقراءة عدد  $n$  وحساب قيمة  $m$  وكالتالي:

$n=0$  اذا كانت قيمة  $m=n+3$

$n=2$  اذا كانت قيمة  $m='ali'$

$n=3$  اذا كانت قيمة  $m=magic(n)$

وطباعة error اذا كان العدد المدخل عكس ذلك.

```
clc;
clear;
n=input('enter n:');
switch n
case {0} /*يمكن ان تكتب بدون اقواس مجموعات لكونها قيمة واحدة*/
    m=n+3;
case {2}
    m='ali';
case {3}
    m=magic(n);
otherwise
    disp('error');
end;
disp(m);
```

مثال (٣) اكتب برنامج بلغة ماتلاب لتحويل عدد مدخل بوحدات (بوصة، قدم، متر، مليمتر) إلى سنتيمترات.

```
clc;
clear;
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 {'meter','m'}**

**y=x\*100;**

**case {'feet','ft'}**

**y=x\*2.54\*12;**

**case {'millimeter','mm'}**

**y=x/10;**

**case {'centimeter','cm'}**

**y=x;**

**otherwise**

**disp('error');**

**end;**

**display(y);**

## المحاضرة التاسعة

### جمل الدوران والتكرار

توفر لغة ماتلاب مجموعة من جمل الدوران والتكرار وهي:

#### جملة for

تقوم حلقات for بإعادة تنفيذ مجموعة من الأوامر لعدد معين من المرات وبخطوة معينة، وتعطى الصيغة العامة لحلقة for كما يلي:

```
for i=x1:x3:x2
```

```
(commands)
```

```
end;
```

حيث يعاد تنفيذ الأوامر (commands) الواقعة بين عبارتي for و end من القيمة الابتدائية x1 إلى القيمة النهائية x2 وبزيادة مقدارها x3 كما في المثال التالي:

مثال (١): طبق المعادلة  $x(n)=\sin(n*\pi/10)$  للأعداد من ١ إلى ١٠ :

```
>> for n=1:10
```

```
    x(n)=sin(n*pi/10);
```

```
end;
```

```
>> x
```

```
x = 0.3090 0.5878 0.8090 0.9511 1.0000 0.9511 0.8090 0.5878 0.3090 0.0000
```

يمكن تفسير الدوارة أعلاه كما يلي:

من اجل كل قيمة ل-n من ١ إلى ١٠ يجب حساب قيمة العبارة الموجودة حتى عبارة end التالية، تكون قيمة n في الدورة الأولى n=1، وتكون في الدورة الثانية n=2، وهكذا حتى تصل إلى n=10. ملاحظة: يمكن إنشاء عدة حلقات for متداخلة كما في المثال التالي:

```
>> for n=1:5
```

```
    for m=1:5
```

```
        A(n,m)=n^2+m^2;
```

```
    end;
```

```
    disp(n);
```

```
end;
```

الإخراج

1

2

3

4

5

```
>> A
```

```
A=
```

```
2 5 10 17 26
```

```
5 8 13 20 29
```

```
10 13 18 25 34
```

```
17 20 25 32 41
```

```
26 29 34 41 51
```

مثال (٢): لطباعة الاعداد من ١ إلى ١٠

```
>> for i=1:10
```

```
disp(i);
```

```
end;
```

الإخراج

```
1  
2  
3  
.  
.  
10
```

مثال (٣): لطباعة الاعداد التي تبدأ بـ ٠ وتنتهي بـ ١٠ وبزيادة ٢ :

```
>> for i=0:2:10
```

```
disp(i);
```

```
end;
```

الإخراج

```
0  
2  
4  
6  
8  
10
```

مثال (٤): لطباعة الاعداد التي تبدأ بـ ١٠ وتنتهي بـ ٢ وبتناقص ٢ :

```
>> for i=10:-2:1
```

```
disp(i);
```

```
end;
```

الإخراج

```
10  
8  
6  
4  
2
```

واجب: ما هو إخراج البرنامج التالي عندما  $m=3$  و  $n=3$  ؟

```
n=input('enter n:');  
m=input('enter m:');  
for i=1:n  
    for j=1:m  
        result(i,j)=i^j;  
    end;  
end;  
disp(result);
```

### جملة WHILE

تجري حلقات while عمليات الحساب عدداً غير محدد من المرات على عكس حلقات for التي تؤدي عدداً معيناً من التمريرات، ويمكن كتابة الصيغة العامة لحلقة while كما يلي:

```
while expression  
    (commands)  
end;
```

ستنفذ مجموعة الأوامر (commands) الواقعة بين العبارتين while و end طالما ان كل العناصر ضمن expression تمتلك قيمة صحيحة (true)، وعادة ما تكون نتيجة expression عدداً مفرداً.

مثال (٦): اطبع الأعداد من ١ إلى ٢٤ باستعمال تعليمة **while**:

```
>> x=1;  
>>while x< 25  
    disp(x);  
    x=x+1;  
end;
```

الإخراج

```
1  
2  
3  
.  
.  
24
```

مثال (٧):

```
>> num=0;eps=1;
>>while (1+eps) > 1
    eps=eps/2;
    num=num+1;
end;
>> num
num=53
```

ملاحظة: هناك طريقة قانونية للخروج من حلقة for و while وكالاتي:

( في حالة تحقق الشرط يتم الخروج من الدوارة for وكذلك while )

|   |  |
|---|--|
| <pre>s=0; for i=1:100     s=s+i;     if s&gt;250         break;     end; end;</pre> | <pre>s=0; x=1; while x&lt;100     s=s+x;     if s&gt;250         break;     end;     x=x+5; end;</pre> |
| <pre>الإخراج i=22 s=253</pre>   | <pre>الإخراج x=51 s=286</pre>  |

ملاحظة: إذا وجدت تعليمة break ضمن حلقة داخلية واقعة ضمن حلقات اكبر فإن البرنامج يخرج من الحلقة التي صادف فيها التعليمة ولا يخرج من الحلقات الأكبر.

س<sup>1</sup>: اكتب برنامج بلغة ماتلاب لجمع عشر أعداد

الحل:

```
sum=0;
for i=1:10
    a=input('enter a:');
    sum=sum+a;
end;
disp(sum);
```

س<sup>2</sup>: اكتب برنامج بلغة ماتلاب لجمع الأعداد الموجبة فقط ضمن الاعداد المدخلة وينتهي البرنامج عمله اذا العدد سالب

الحل:

```
sum=0;
n=input('enter the number of numbers:');
for i=1:n
    a=input('enter the number:');
    if a > 0
        sum=sum+a;
    else
        break;
    end;
end;
disp(sum);
```

س٣: اكتب برنامج بلغة ماتلاب لإيجاد معدل الأعداد التي تقبل القسمة على ٣ والأعداد (٩٩-٣)

الحل:

```
sum=0;
n=0;
for i=3:3:99
    sum=sum+i;
    n=n+1;
end;
avg=sum/n;
disp(avg);
```

## المحاضرة العاشرة

### الرسوم البيانية

يزودك برنامج ماتلاب بالعديد من الإيعازات التي تظهر البيانات ثنائية الأبعاد وثلاثية الأبعاد، حيث يرسم بعضها منحنيات ثنائية الأبعاد وثلاثية الأبعاد بينما يرسم بعضها سطوحاً وإطارات، كما يمكن استخدام اللون كبعد رابع.

### الإيعاز plot

يقوم هذا الإيعاز بإظهار البيانات على شكل ثنائي الأبعاد.

مثال(١):

```
x=[1:0.5:10];
```

لاحتساب مجموعة قيم للمصفوفة  $y=\exp(x)$ ;

لرسم قيم بيانية للمحورين  $\text{plot}(x,y)$

مثال(٢):

```
x=1:10;
```

```
plot(x)
```

ملاحظة: في حالة وجود إحداثي واحد (قائمة واحدة) يقوم الإيعاز  $\text{plot}$  برسم قيم بيانية متناظرة بالمحورين  $(x,x)$  لكل عناصر القائمة.

مثال(٣):

```
y=[ ];
```

```
for i=1:10
```

```
    y(i)=exp(i);
```

```
end;
```

```
plot(y);
```

```
y=[ ];
```

```
for i=1:10
```

```
    y=[y exp(i)];
```

```
end;
```

```
plot (y);
```

مثال(٤):

إيعازات الرسم:

`plot(x,y)` إيعاز للرسم الثنائي #

`legend('.....')` إيعاز لكتابة دليل المخطط #

`xlabel('.....')` إيعاز لكتابة عنوان المحور x #

`ylabel('.....')` إيعاز لكتابة عنوان المحور y #

`title('.....')` إيعاز لكتابة عنوان للمخطط #

`text(x,y,'string')` إيعاز لكتابة نص في المخطط #

`plot3(x,y,z)` إيعاز للرسم الثلاثي الأبعاد #

`surf(x,y,z)` إيعاز للرسم البياني السطحي #

`bar(x,y)` إيعاز لرسم bar chart #

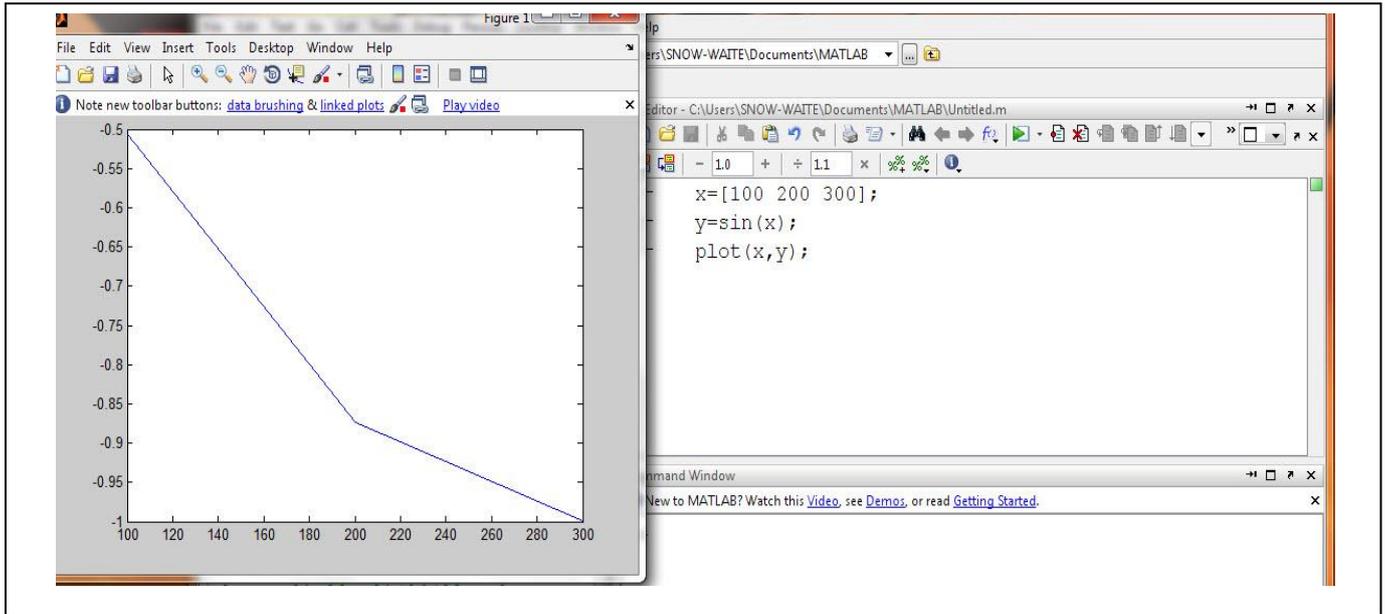
`hist(x,y)` إيعاز لرسم histogram #

`pie(x,y)` إيعاز للرسم الدائري pie chart #

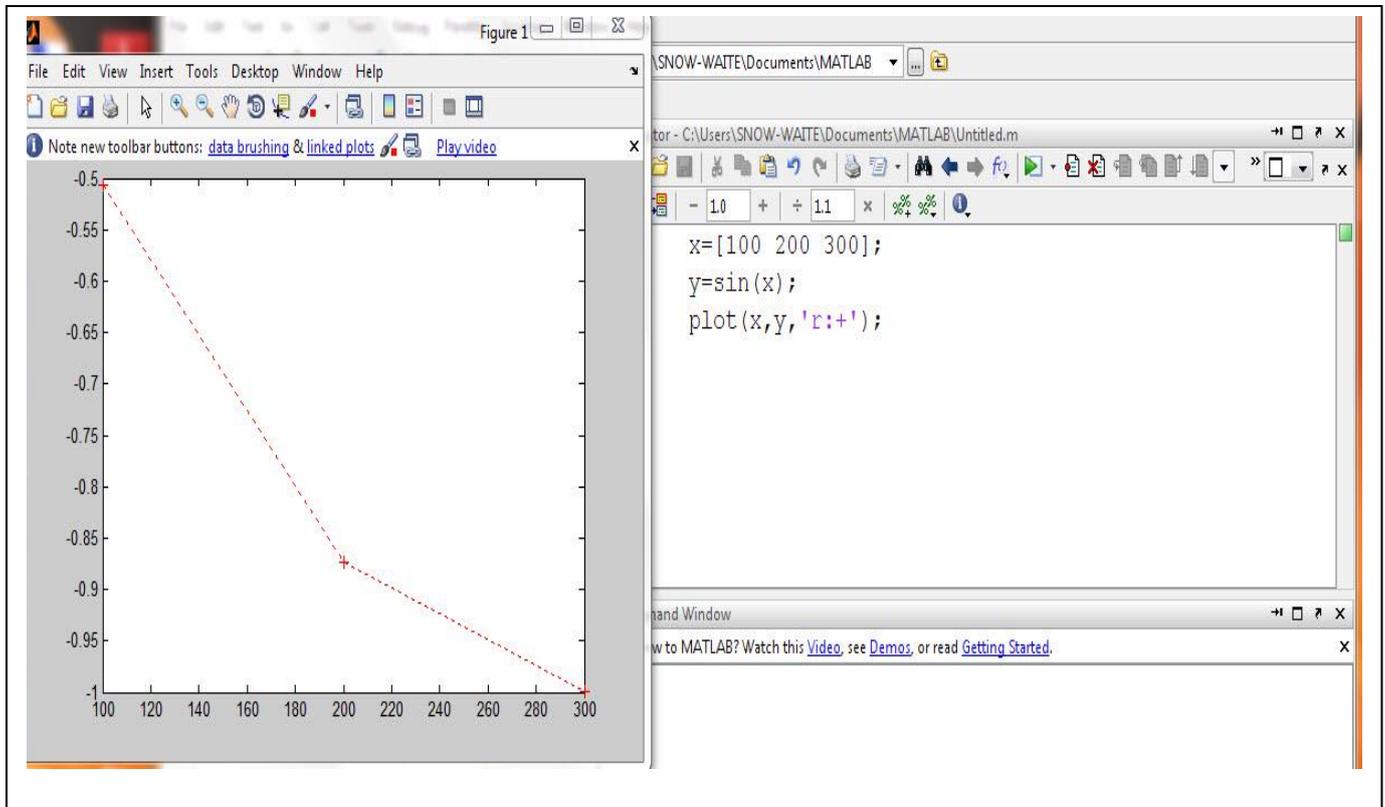
`plot(x,y,'r:+')` إيعاز لجعل رسم الخط بلون وشكل مغاير #

`subplot(2,2,1)` إيعاز لرسم أكثر من رسم على الشاشة #

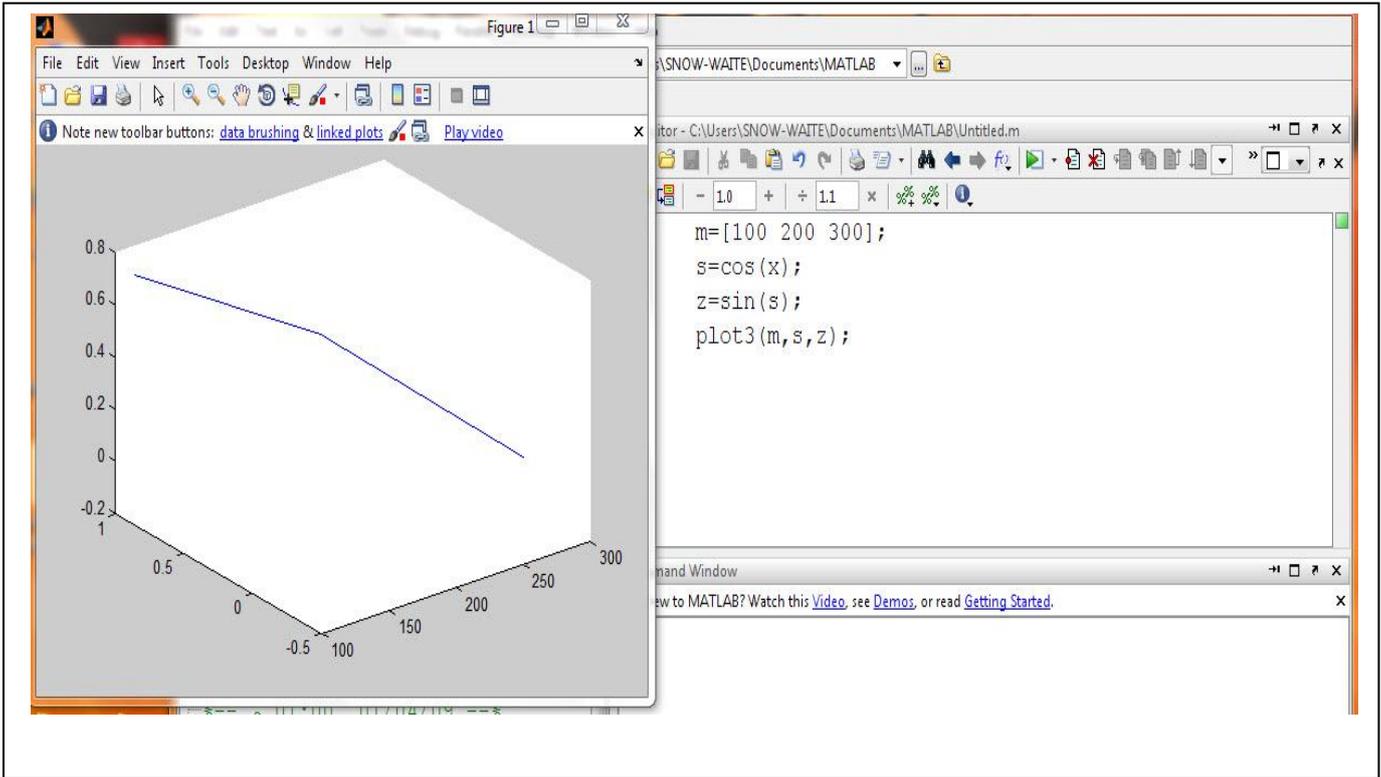
مثال (١) اكتب برنامج لرسم ثنائي الابعاد للمتجهين  $x=100,200,300$ ,  $y=\sin(x)$



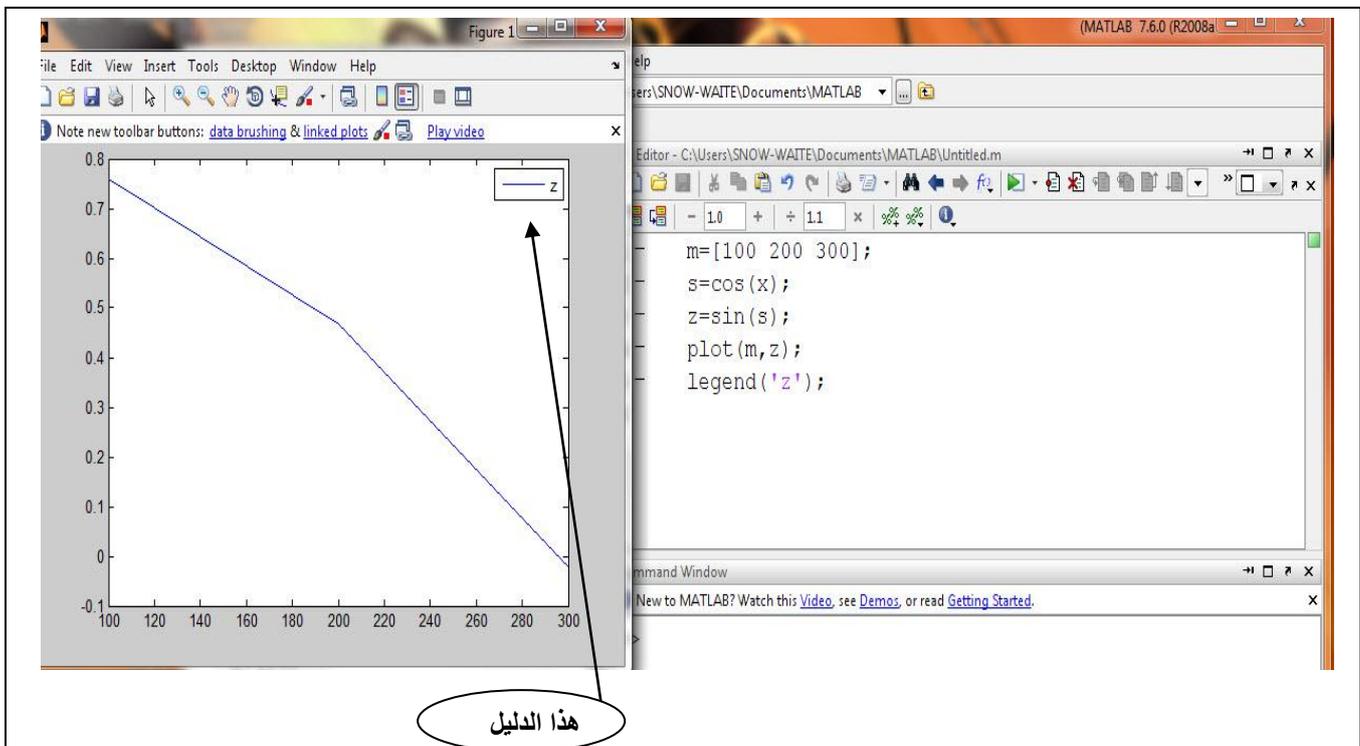
مثال (٢) اكتب برنامج لرسم ثنائي الابعاد للمتجهين  $x=100,200,300$ ,  $y=\sin(x)$  مع جعل خط الرسم باللون الاحمر وبشكل +



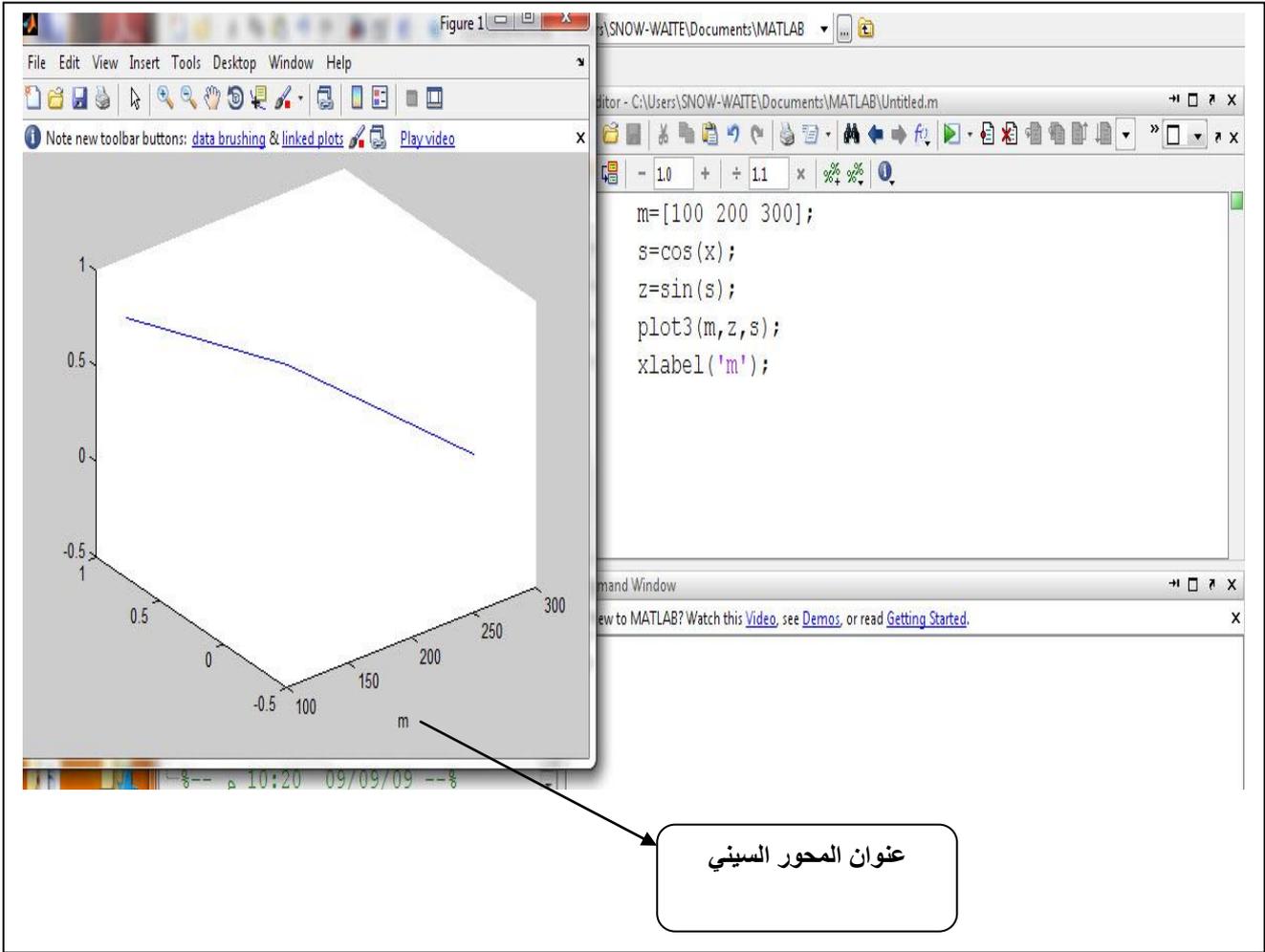
مثال (٣) اكتب برنامج لرسم ثلاثي الابعاد للمتجهات  $m=100,200,300$  ,  $s=\cos(m)$ ,  $z=\sin(s)$



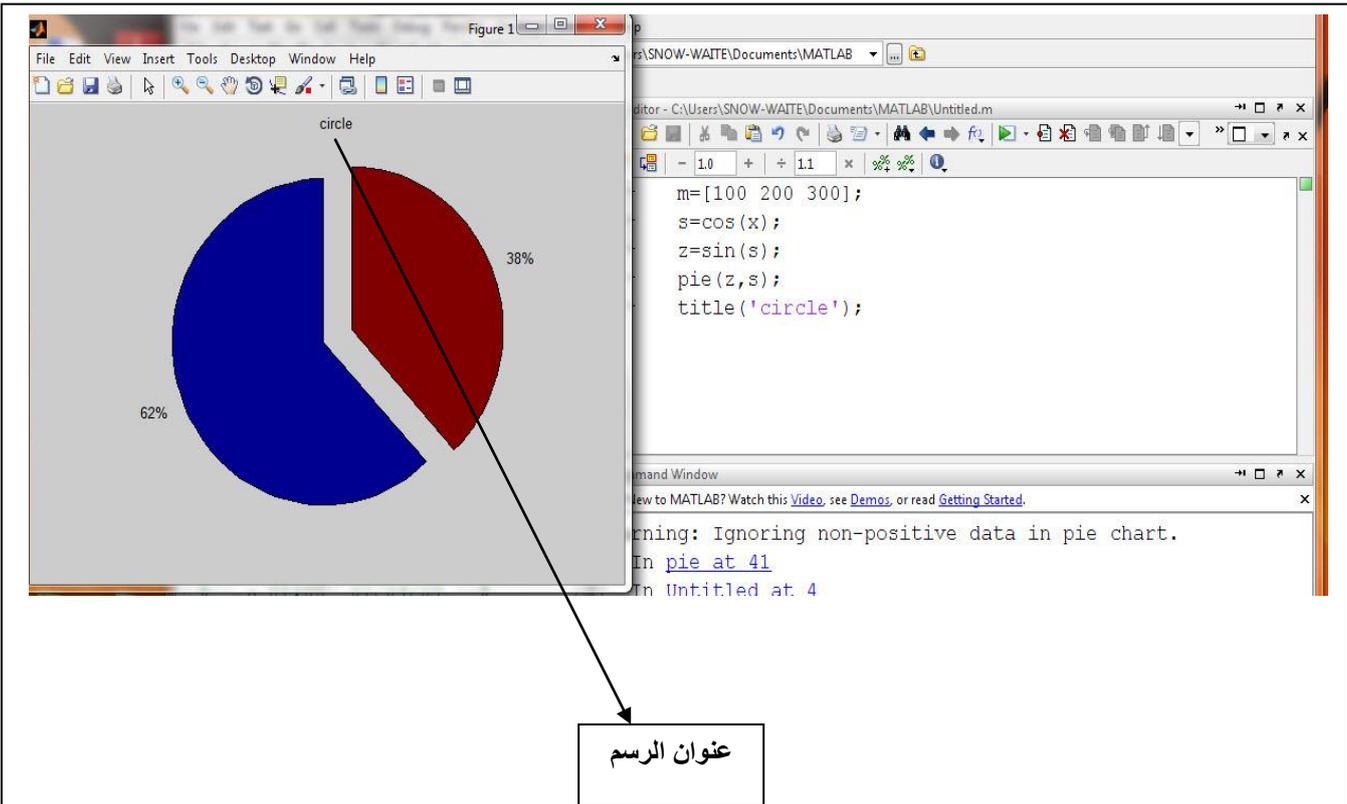
مثال (٤) اكتب برنامج لرسم ثنائي الابعاد للمتجهات  $(m,z)$  علماً إن المتجهات هي  $m=100,200,300$  ,  $s=\cos(m)$ ,  $z=\sin(s)$  واكتب دليل للمخطط هو  $(z)$ :



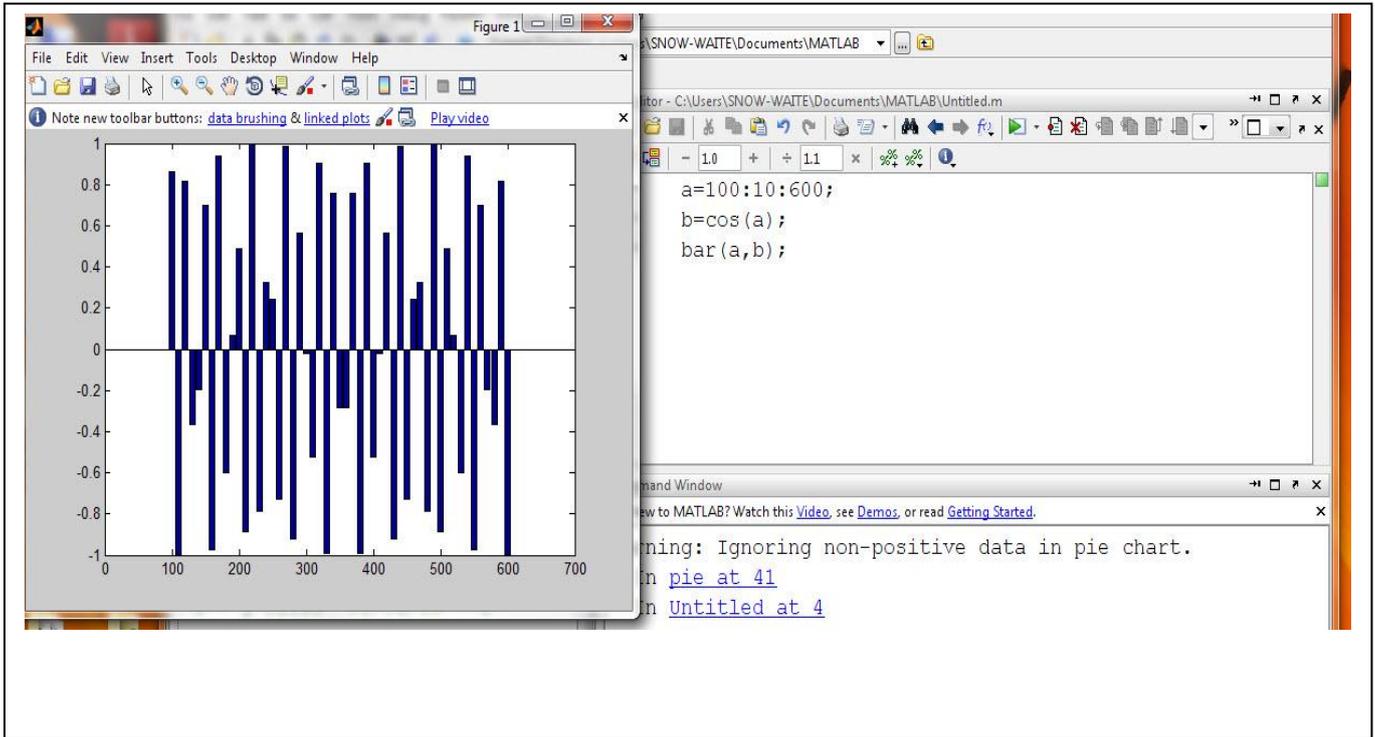
مثال (٥) اكتب برنامج لرسم ثلاثي الابعاد للمتجهات التالية مع كتاب عنوان للمحور السيني (m) والمتجهات هي  $m=100,200,300$  ,  $s=\cos(m)$ ,  $z=\sin(s)$



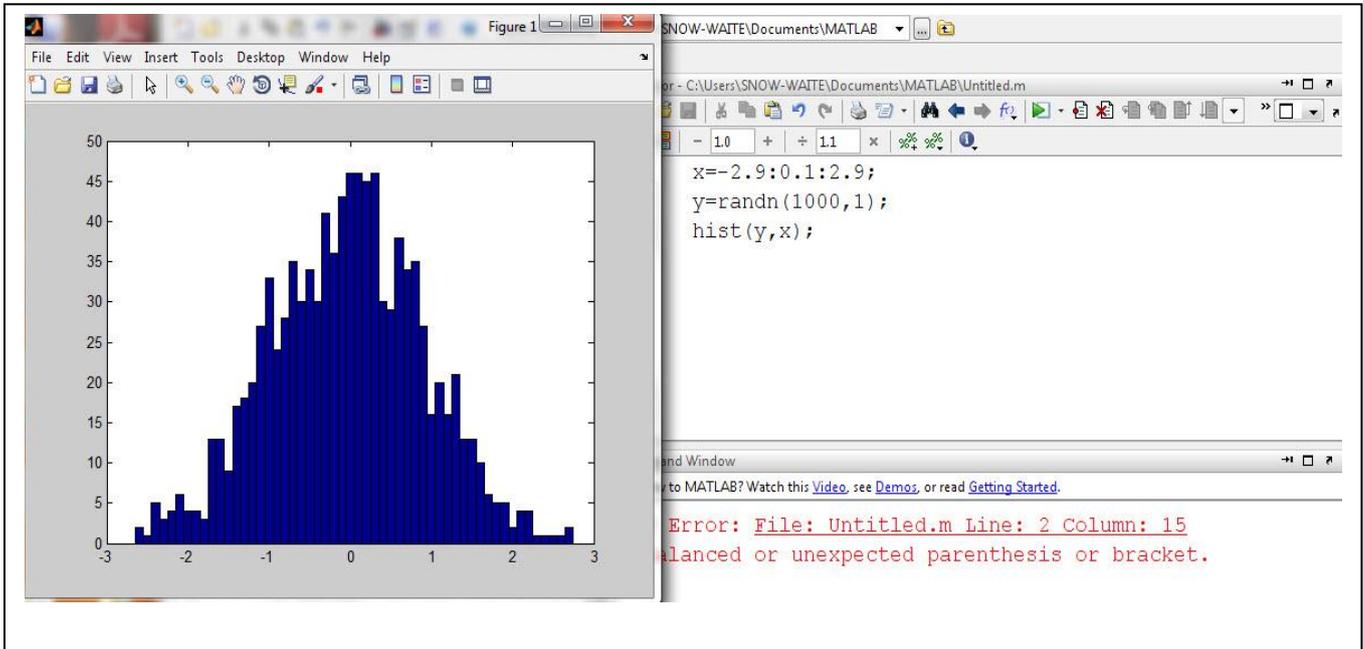
مثال (٦) اكتب برنامج لرسم دائري للمتجهات أعلاه مع كتابة عنوان للرسم هو (circle):



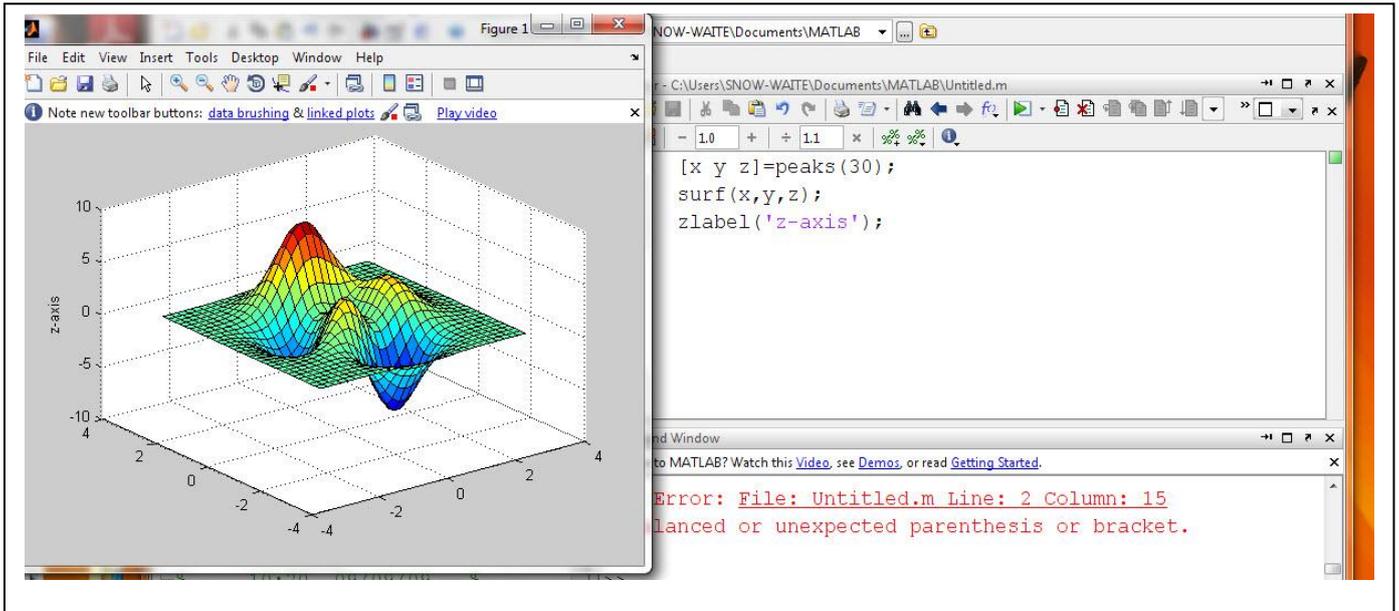
مثال (٧) اكتب برنامج لرسم خطوط عرضية للمتجهات  $a=100:10:600$ ,  $b=\cos(a)$



مثال (٨) اكتب برنامج لرسم مخطط histogram للمتجهات x,y والمدرجة ادناه



مثال (٩) اكتب برنامج بلغة ماتلاب لرسم سطحي للمتجهات  $x,y,z$  مع كتابة عنوان للمحور  $z$



مثال (١٠) اكتب برنامج بلغة ماتلاب لرسم عدة رسوم في الشاشة للمتجهات  $x,y,z$  والرسوم هي الرسم ثنائي الابعاد والثلاثي الابعاد والرسم الدائري ورسم الخطوط العريضة والمتجهات هي:  $x=0:2*\pi:30$  ,  $y=\sin(x)$  ,  $z=\cos(x)$

