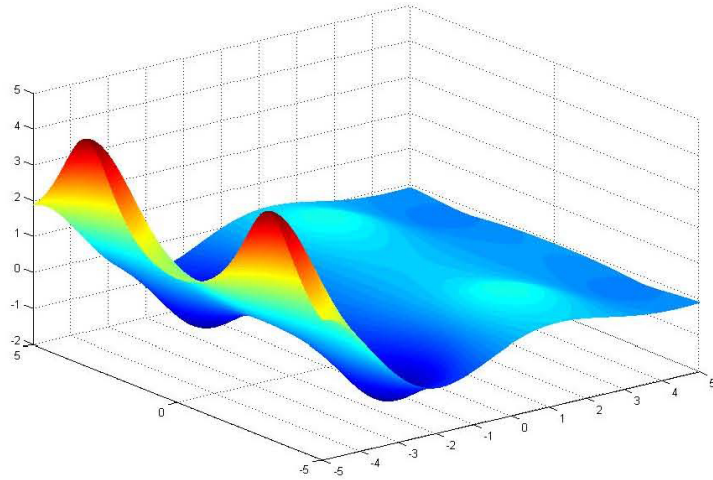


دورة الماتلاب خطوة بخطوة

المقدمة في
ملتقى المهندسين العرب



يقوم بشرح الدورة
المهندس أحمد عفيفي سلامة



جميع الحقوق محفوظة لدى ملتقى المهندسين العرب ©
٢٠٠٦

شكر خاص من موقع اليوسوليد

www.yousolid.com

السلام عليكم ورحمة الله وبركاته



تحية طيبة وبعد ..

أخواني الكرام توكلنا على الله في بدء شرح برنامج الماتلاب

سنتناول بإذن الله في مقدمة في الماتلاب التالي:

- 1-تعريف برنامج الماتلاب ومؤسسه
- 2- واجهة البرنامج
- 3- بعض الأوامر الأساسية
- 4- تعريف العمليات الأساسية
- 5- تعريف بعض المتغيرات
- 6- الأعداد المركبة
- 7- المصفوفات
- 8- الرسم في نظام الإحداثي الثاني $D Plotting$
- 9- الرسم في نظام الإحداثي الثلاثي الأبعاد $D Plotting$

وسنتناول في هذا الأسبوع (1/04/2006) إلى (7/04/2006) بإذن الله التالي

- 1-التعريف ببرنامج الماتلاب ومؤسسه
- 2- واجهة البرنامج
- 3- تعريف العمليات الأساسية
- 4- بعض الأوامر الأساسية
- 5- تعريف بعض المتغيرات
- 6- الأعداد المركبة

أولاً: تعريف برنامج الماتلاب

برنامج الماتلاب هو برنامج هندسي (وله مجالات أخرى) 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 $Algebraic Equations$ وكذلك المعادلات التفاضلية $Differential Equations$ ذات الرتب العليا والتي قد تصل من الصعوبة ما تصل، ليس فقط ذلك بل يستطيع البرنامج عمل التفاضل الجزئي، ويقوم بعمل عمليات الكسر الجزئي $Partial fraction$ بسهولة ويسر والتي تستلزم وقتاً كبيراً لعملها بالطرق التقليدية، هذا من الناحية الأكاديمية، أما من الناحية التطبيقية فيستطيع البرنامج العمل في جميع المجالات الهندسية مثل أنظمة التحكم $Control System$ وفي مجال الميكانيكا $Mechanical Field$ وكذلك محاكاة الإلكترونيات $Electronics$ وصناعة السيارات $Automotive Industry$ وكذلك مجال الطيران والدفاع الجوي $Aerospace and Defense$ والكثير من التطبيقات الهندسية. وحتى أؤكد للجميع ذلك، قامت شركة السيارات المرموقة نيسان $Nissan$ بتخفيض وقت التطوير إلى ٥٠% عندما قامت باستبدال التصميم على الأوراق $Paper Model Based Design$ إلى الأداة المتطورة في برنامج الماتلاب وهو $Model Based Design$ يقول المدير المساعد شيجاياكي كاكيزاكي في مجموعة هندسة إدارة نظام المحرك لشركة نيسان (شركة محدودة)

Without MathWorks tools for Model-Based Design, Nissan would not have become the first company to meet the CARB PZEV standard

CARB= California Air Resources Board

PZEV= Partial Zero Emission Vehicle

يمكنكم متابعة هذا التقرير من خلال الرابط

<http://www.mathworks.com/company/use...ml?by=industry>

فمع التقدم السريع في التكنولوجيا أصبحت الحاجة ملحة على تعلم مثل هذا البرنامج حتى نصبح في سباق التنافس

الصناعي.

التعريف بمؤسس برنامج الماتلاب
قام بتأسيس البرنامج شخصان، الأول هو كليف مولر والثاني جاك ليتل

كليف مولر

هو استاذ الرياضيات وعلوم الحاسب **Computer Science** لأكثر من عشرين عاماً في جامعة متشيجين وجامعة ستانفورد وجامعة نيو مكسيكو.

أمضى خمس سنوات عند إثنين من مصنعي الـ **Hardware** وهما **Intel Hypercube organization** و **Ardent Computer** قبل أن يقوم بالانتقال إلى شركة **MathWorks** الشركة الأم لبرنامج الماتلاب، كما أنه هو المؤلف لأول برنامج للماتلاب.



كليف مولر

جاك ليتل

هو المؤسس لشركة **Mathworks** كما أنه المساعد في وضع تخطيط برنامج الماتلاب.
جاك حاصل على بكالوريوس الهندسة الكهربائية وعلوم الحاسب من جامعة **MIT** عام ١٩٧٨
كما أنه حاصل على شهادة **M.S.E.E** من جامعة ستانفورد عام ١٩٨٠



جاك ليتل

يستكمل...

السلام عليكم ورحمة الله وبركاته



تحية طيبة وبعد ..

الحمد لله رب العالمين، والصلاة والسلام على خير المرسلين، سيدنا محمد عليه أفضل الصلاة والتسليم
أخواني نبدأ معاً تعليم الماتلاب خطوة بخطوة، والذي أسأل الله أن يكون خير أداة للجميع، وأن يكون تعلمهم لهذا
البرنامج من أجل خدمة هذا الدين الحنيف، ومن أجل رفع راية الإسلام.

سيتم تقسيم تعليم برنامج الماتلاب إلى أقسام، تبعاً لنوع التطبيق، وسيتم التطرق إلى البرنامج تبعاً للترتيب التالي (قد
يختلف هذا الترتيب تبعاً لرغبات الأعضاء)

1-مقدمة في الماتلاب

2-مقدمة في المحاكاة Simulink

3-أنظمة التحكم باستخدام الماتلاب Control System Using the Matlab

4-تطبيقات الإشارة باستخدام الماتلاب Signal Application Using Matlab

5-عمليات الإشارة الرقمية باستخدام الماتلاب Digital Signal Processing Using Matlab

6-النظريات الرياضية التقريبية باستخدام الماتلاب Numerical Application Using Matlab

7-تطبيقات معالجة الصور باستخدام الماتلاب Image Processing Applications Using Matlab

8-تطبيقات موجات الراديو باستخدام الماتلاب Radio Frequency Applications Using Matlab

9-التطبيقات الميكانيكية باستخدام الماتلاب Mechanical Applications Using Matlab

10-تطبيقات الرادار باستخدام الماتلاب Radar Applications Using Matlab

11-تطبيقات الروبوت باستخدام الماتلاب Robots Applications Using Matlab

12-التطبيقات الإلكترونية باستخدام الماتلاب Electronics Applications Using Matlab

13-تطبيقات أشباه الموصلات باستخدام الماتلاب Semiconductors Applications Using Matlab

14-التطبيقات المستخدمة في صناعة السيارات باستخدام الماتلاب Automotive Applications Using Matlab

Matlab

15-التطبيقات المستخدمة في علوم الفضاء والدفاع الجوي باستخدام الماتلاب Aerospace and Defense Applications Using Matlab

Applications Using Matlab

16-تطبيقات الاتصالات باستخدام الماتلاب Communication Applications Using Matlab

كما تروا إخواني الكرام فإن المشوار طويل، وأسأل الله أن يعيننا على إستكمال هذا المشوار إلى آخر

وسكون هنالك ثلاثة مواضع لا بد من أخذه قبل الشروع في التطبيق وهما

1-مقدمة في الماتلاب

2-مقدمة في المحاكاة

3-أنظمة التحكم باستخدام الماتلاب

أسأل الله ان يعيننا جميعاً في النهوض بهذه الأمة الكريمة

فقد حان وقت التقدم

وسنبدأ اليوم الموافق ٢٠٠٦/٠٣/٣١ بشرح

مقدمة في الماتلاب

السلام عليكم ورحمة الله وبركاته



تحية طيبة وبعد ..

أخواني الكرام، لابد من الإتفاق على بعض المبادئ الهامة في هذا الموضوع

- 1- يتم شرح جزء معين يتم تحديده في الأسبوع
 - 2- يتم تقديم الشرح في أول كل أسبوع حتى يتسنى للأعضاء الإستفسار بقدر المستطاع خلال فترة الأسبوع وكذلك الرد على إستفساراتهم
 - 3- يجب التقيد بما سيتم شرحه من خلال ترتيب معين، حتى نصل إلى درجة عالية من الفهم والتطبيق للبرنامج.
 - 4- ضرورة تحميل كتاب تعليم الماتلاب وكذلك كتاب المحاكاة باستخدام الماتلاب **simulink** من خلال موضوع مكتبة الميكاترونكس!
 - 5- تحميل برنامج الماتلاب، ويمكن للأعضاء تحميل البرنامج من خلال موضوع **برامج هندسة ميكاترونكس!**
 - 6- يمكن للأعضاء المشاركة في شرح البرنامج، مع الإلتزام بالترتيب المقرر للشرح.
 - 7- في حالة أن كان للعضو إستفسار حول موضوع قد سبق شرحه، فما عليه إلا أن يرسل المشرف حول الإستفسار، حتى يقوم المشرف بوضع إستفساره مع إستفسارات ذلك الجزء المشروح مسبقاً.
 - 8- بالنسبة للأعضاء الجدد الذين نقل مشاركاتهم عن ٢٥ مشاركة لبدء الرسائل الخاصة، يمكنهم بعث إستفساراتهم على بريدي الإلكتروني، ولقد وضعت بريدي الإلكتروني في الملف الشخصي الخاص بي.
 - 9- ضرورة توضيح ما يتم شرحه بالصور إذا إقتدى الأمر ذلك.
 - 10- يقيم الأعضاء إستفساراتهم حول برنامج الماتلاب في موضوع إستفسارات موضوع الماتلاب **خطوة بخطوة!** ، وفي حالة أن قام العضو بوضع إستفساره في موضوع الماتلاب **خطوة بخطوة** يقوم المشرف بنقل إستفساره إلى موضوع إستفسارات موضوع الماتلاب **خطوة بخطوة!** .
 - 11- تقدم الإستفسارات والطلبات في موضوع إستفسارات موضوع الماتلاب **خطوة بخطوة** ، وذلك حتى لا يمل الأعضاء بسبب كثرة الردود، في حالة قام العضو بوضع رده في موضوع **الماتلاب خطوة بخطوة** سهواً، يقوم مشرف القسم بنقل رده إلى موضوع إستفسارات موضوع الماتلاب **خطوة بخطوة**.
 - 12- تقدم المداخلات والشكر في موضوع مداخلات موضوع الماتلاب **خطوة بخطوة** ، وذلك حتى لا يمل الأعضاء بسبب كثرة الردود، في حالة قام العضو بوضع رده في موضوع **الماتلاب خطوة بخطوة** سهواً، يقوم مشرف القسم بنقل رده إلى موضوع مداخلات موضوع الماتلاب **خطوة بخطوة**.
- أعتذر على الإطالة، ولكنها أساسيات لموضوع وشرح مميزين
أسأل الله التوفيق

السلام عليكم ورحمة الله وبركاته



تحية طيبة وبعد ..

أخواني الكرام، وأيضاً قبل البدء، لابد من ذكر المواقع الهامة لخدمة برنامج الماتلاب، والتي من خلالها تستطيع أن تصل إلى التطبيق المطلوب بإذن الله

موقع الشركة المصنعة لبرنامج الماتلاب

<http://www.mathworks.com>

حيث ستجد في هذا الموقع آخر الإصدارات لبرنامج الماتلاب، وكذلك التحديثات الخاصة بالبرنامج، كما يوفر شرحاً باللغة الإنجليزية لبرنامج الماتلاب.

University of Utah

<http://www.math.utah.edu/lab/ms/matlab/matlab.html>

حيث يوفر موقع الجامعة مقدمة سريعة ومبسطة لبرنامج الماتلاب

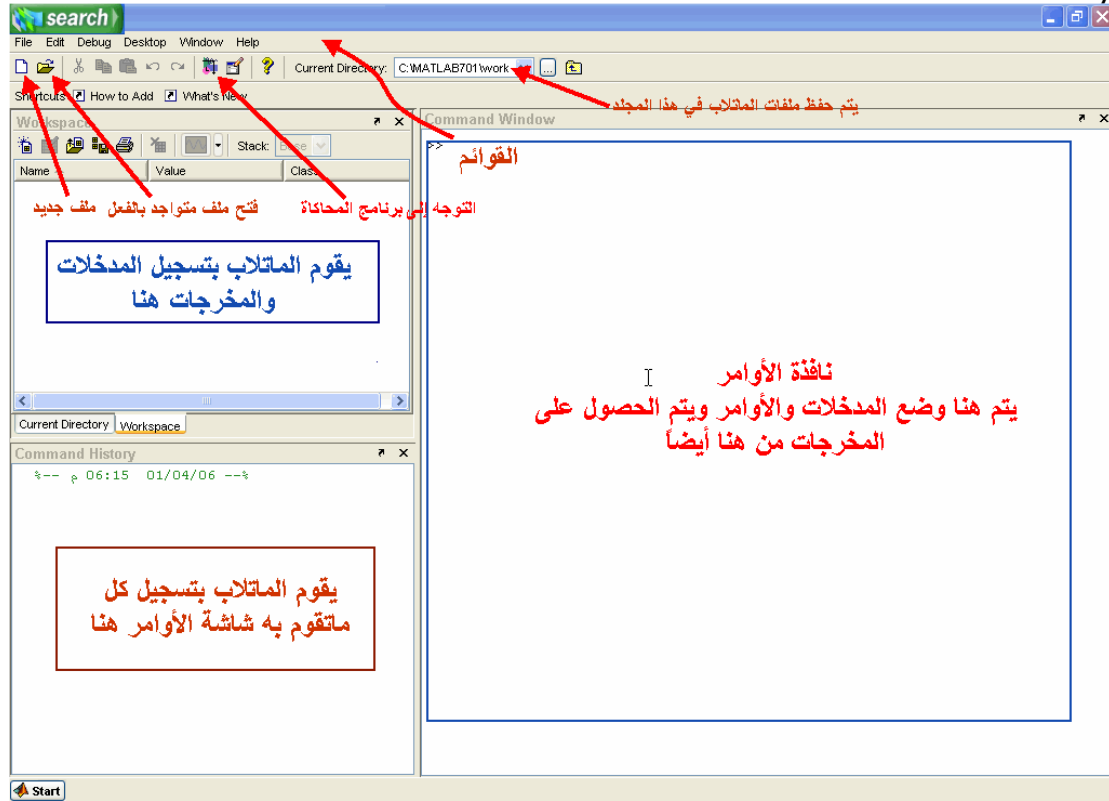
Indiana University

<http://www.indiana.edu/~statmath/math/matlab/>

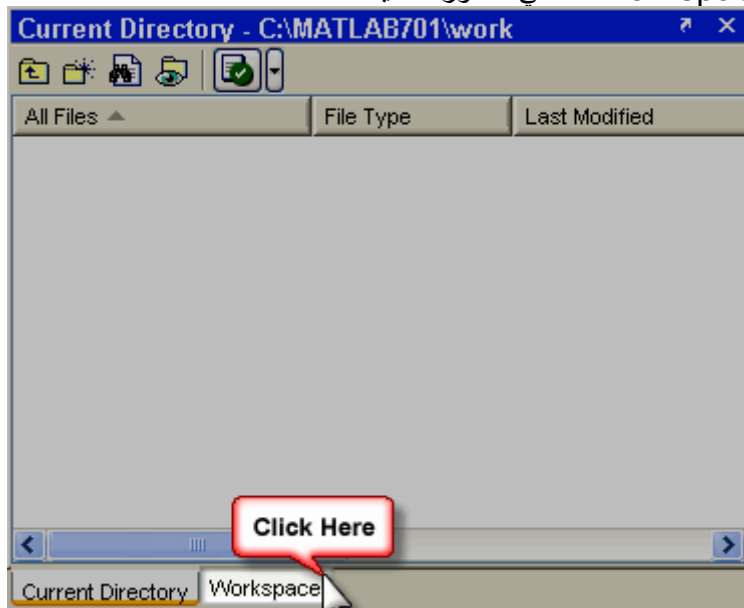
يقدم أيضاً موقع الجامعة لمحة سريعة حول برنامج الماتلاب

ملاحظة: يتم تجديد هذه الروابط بشكل إسبوعي، حتى نصل إلى أعلى إستفادة ممكنة من البرنامج.
نسأل الله التوفيق

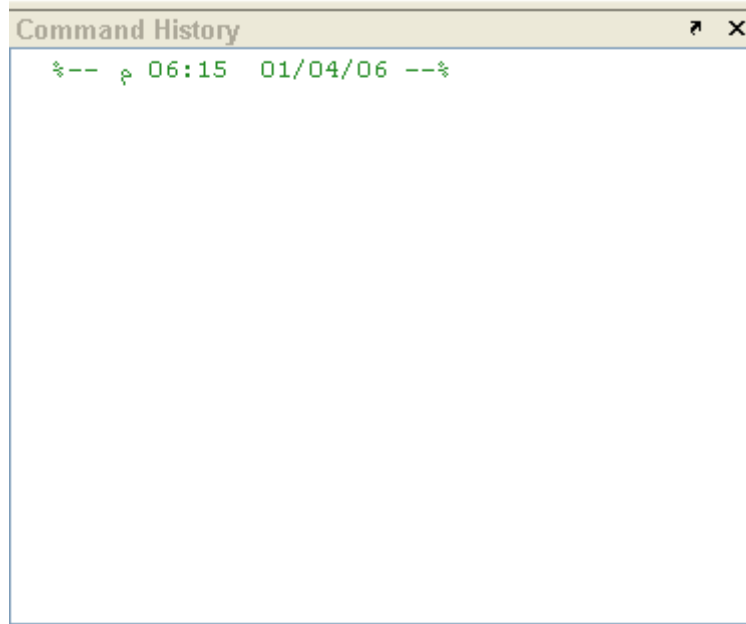
ثانياً: واجهة البرنامج وتنقسم واجهة البرنامج بالسهولة في التعامل معها، حيث يتم تقسيم مناطق العمل بها إلى ثلاث مناطق رئيسية، وهي كالتالي نافذة الأوامر Command Window و منطقة العمل Workspace و تاريخ الأوامر Command History، إنظر الصورة التالية.



نافذة الأوامر Command Window: حيث يتم إدخال المدخلات Inputs والأوامر Commands، ويعمل الماتلاب على تحليل تلك البيانات ومدى مطابقتها للمدخلات 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، لا تظهر نافذة Workspace، وحتى تظهر اضغط 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



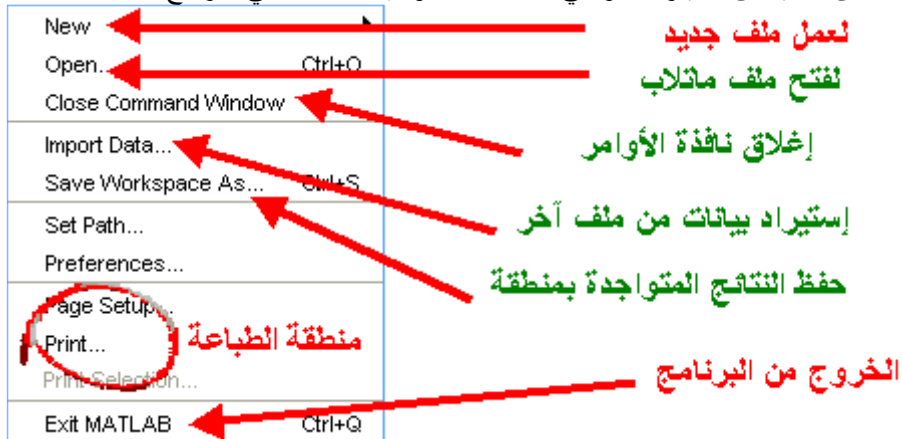
قائمة إبدأ Start: تستخدم هذه القائمة للوصول إلى التطبيق المراد تنفيذه، تستخدم هذه القائمة في المراحل المتقدمة في برنامج الماتلاب

صورة ٤

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

Undo	Ctrl+Z
Redo	Ctrl+Y
Cut	Ctrl+X
Copy	Ctrl+C
Paste	Ctrl+V
Paste Special...	
Select All	
Delete	
Find...	
Find Files...	
Clear Command Window	
Clear Command History	
Clear Workspace	

مسح قائمة الأوامر

مسح مسجل المدخلات
والمخرجات

مسح منطقة العمل

قائمة Debug

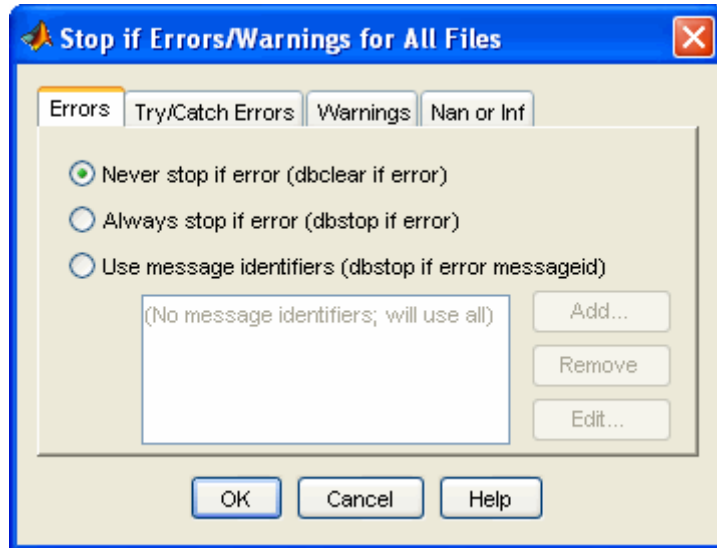
هذه القائمة خاصة بمعالجة البيانات، والطريقة المتبعة من قبل برنامج الماتلاب في مواجهة الأخطاء. أنظر الصورة التالية

<input checked="" type="checkbox"/> Open M-Files when Debugging	
Step	F10
Step In	F11
Step Out	Shift+F11
Continue	F5
Clear Breakpoints in All Files	
Stop if Errors/Warnings...	
Exit Debug Mode	

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

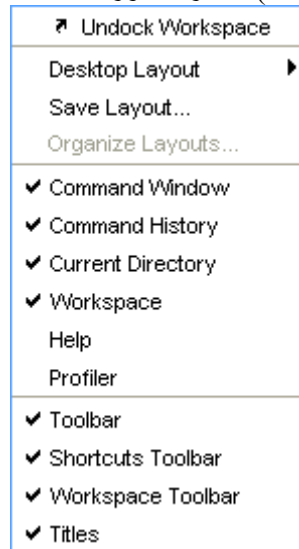
فمثلاً قم بإختيار Stop If Errors/Warnings...

ستلاحظ ظهور نافذة، تعطيك حرية الإختيار في تصرف برنامج الماتلاب عند حدوث أخطاء أو تحذيرات ملاحظة: يرجى ترك هذه النافذة دون تغيير، فلننا 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

- ١- Docked: حيث تكون النافذة غير قابلة للتحريك من مكانها.
- ٢- Undocked: حيث تكون النافذة قابلة للتحريك وتعديل مقاسها أيضاً



Docked Window

أي لا يمكن تحريكها



Undocked Window

أي يمكن تحريكها وتعديل مقاسها

يتبقى لدينا قائمتان هما

قائمة Window:

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

Close All Documents	
0 Command Window	Ctrl+0
1 Command History	Ctrl+1
2 Current Directory	Ctrl+2
3 Workspace	Ctrl+3

قائمة Help:

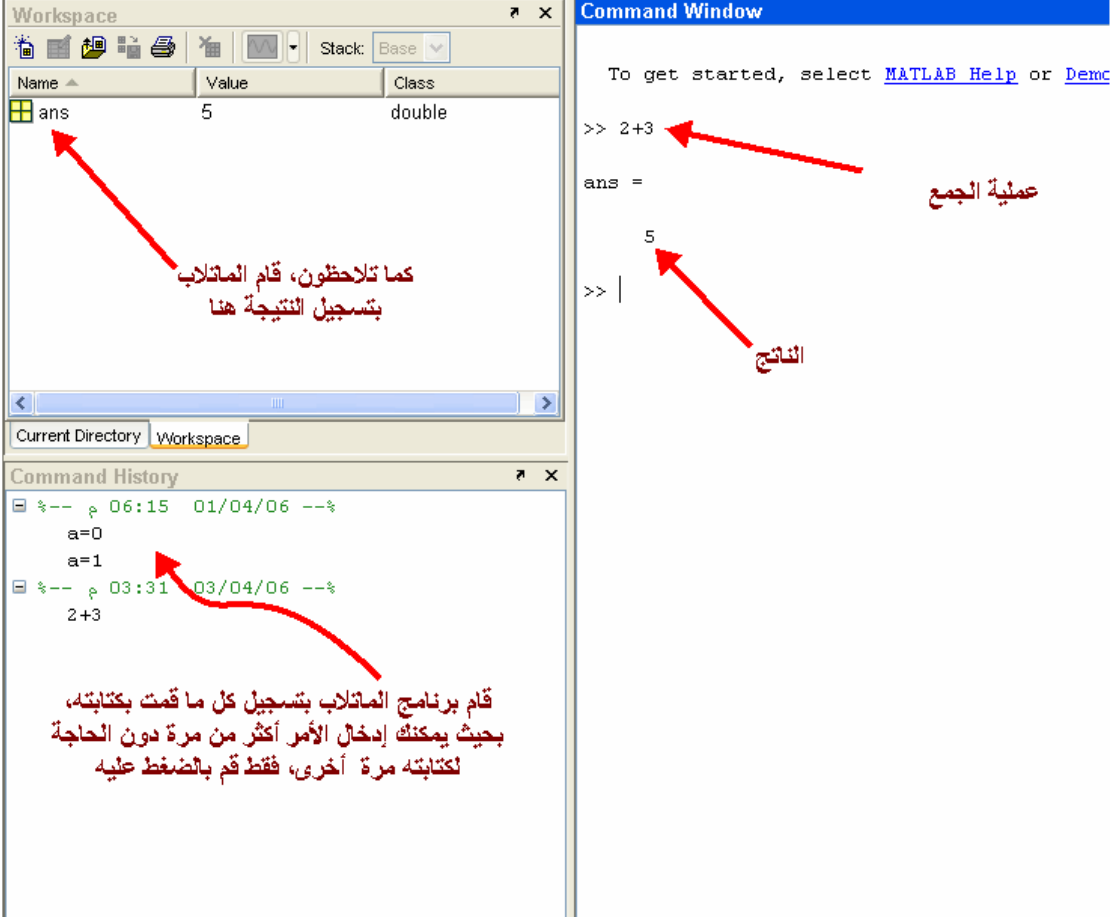
حيث تقوم تلك القائمة، بتوفير المساعدات الضرورية في البرنامج، ووسائل الإتصال بالشركة المصنعة، وآخر التحديثات، وكذلك تعلم الماتلاب باللغة الإنجليزية

Full Product Family Help	
MATLAB Help	F1
Using the Desktop	
Using the Command Window	
Web Resources	▶
Check for Updates	
Demos	
About MATLAB	

أخواني الكرام، نستكمل برنامج الماتلاب ونتناول اليوم بإذن الله العمليات الأساسية (الجمع و الطرح والضرب والقسمة) وبعض العمليات الهامة مثل وضع الأس لعدد، كما سنتعرف على بعض الأوامر الهامة.

عملية الجمع

تأخذ علامة الجمع في الماتلاب الرمز المعروف للجمع وهو "+"
فمثلاً إذا قمنا بجمع ٢+٣ سيقوم الماتلاب بوضع الإجابة في صورة أرقام وهو ٥، أنظر الصورة التالية



The screenshot displays the MATLAB environment with three windows:

- Workspace:** A table with columns 'Name', 'Value', and 'Class'. It contains one entry: 'ans' with a value of 5 and a class of 'double'. A red arrow points from the 'ans' variable to the text: "كما تلاحظون، قام الماتلاب بتسجيل النتيجة هنا".
- Command Window:** Shows the command prompt sequence: '>> 2+3', followed by 'ans =', and then '5'. A red arrow points from the '2+3' command to the text "عملية الجمع", and another red arrow points from the '5' output to the text "النتيجة".
- Command History:** Shows a list of commands. The second entry is '2+3', with a red arrow pointing from it to the text: "قام برنامج الماتلاب بتسجيل كل ما قمت بكتابته، بحيث يمكنك إدخال الأمر أكثر من مرة دون الحاجة لكتابته مرة أخرى، فقط قم بالضغط عليه".

إذهب إلى نافذة Workspace وقم بالنقر بالماوس بقرة مزدوجة، ستلاحظ ظهور نافذة حلت محل نافذة الأوامر وأصبحت نافذة الأوامر في الأسفل، أنظر الصورة

The image shows the MATLAB interface with three main windows:

- Workspace:** Displays a table with columns 'Name', 'Value', and 'Class'. A variable 'ans' is listed with a value of 5 and a class of 'double'. A red box with the text 'Click Here' points to the '+' icon next to 'ans'.
- Array Editor - ans:** A grid editor for the 'ans' variable. It shows a 1x4 array with the first cell containing the value 5. A red arrow points to the 'Edit' icon in the toolbar. Below the grid, there is Arabic text: 'نافذة تعديل وإضافة النتائج، يمكن إضافة النتائج في صورة عمودية أو أفقية، كما سيتم شرحه لاحقاً في المصفوفات'.
- Command Window:** Shows the execution of the command '2+3'. The output is 'ans = 5'. A red arrow points to the output '5' with the text 'نافذة الأوامر'.

لنفترض أننا قمنا بتغيير الناتج ٥ إلى ٣، قم بإغلاق نافذة تعديل النتائج، كما في الصورة التالية

Array Editor - ans

	1	2	3	4	5	6	7	8
1	3							
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								

Command Window

```
>> 2+3
ans =
     5
>>
```

تم تغيير النتيجة من 5 إلى 3

بعد عملية التعديل أو الإضافة قم بإغلاق هذه النافذة من هنا

ستلاحظ عودة نافذة الأوامر لوضعها الأساسي، قم بكتابة `ans` في نافذة الأوامر، ستلاحظ ظهور الناتج بالقيمة الجديدة وهي 3، أنظر الصورة

Command Window

```
>> 2+3
ans =
     5
>> ans
ans =
     3
>>
```

القيمة المعدلة

عملية الطرح:

تأخذ عملية الطرح رمز (-) في الماتلاب، فمثلاً $3-2=1$ ، أنظر الصورة

```
>> 3-2  
  
ans =  
  
    1
```

عملية الضرب

تأخذ عملية الضرب رمز (*)، فمثلاً $12*15=180$ ، أنظر الصورة

The screenshot displays the MATLAB environment. The 'Workspace' window shows a variable 'ans' with a value of 180 and a class of 'double'. The 'Command Window' shows the command '12*15' and the output 'ans = 180'. The 'Command History' window shows the command '12*15'.

Name	Value	Class
ans	180	double

```
>> 12*15  
  
ans =  
  
    180  
  
>> |
```

Command History

```
%-- م 03:58 03/04/06 --%  
12*15
```

عملية القسمة:

تأخذ عملية القسمة رمز (/)، فمثلاً 12 على 3 تساوي 4، أنظر الصورة للتأكد

The screenshot shows a software interface with three main panels:

- Workspace:** A table with columns 'Name', 'Value', and 'Class'. It contains one row: 'ans' with a value of '4' and a class of 'double'.
- Command Window:** A blue headered area containing the text:


```
>> 12/3
ans =
    4
>> |
```
- Command History:** A panel at the bottom showing the command '12/3' and the time '04:02' on '03/04/06'.

عملية وضع الأس:

يأخذ رمز الأس (^)، يمكن الحصول على هذا الرمز من خلال الضغط على 6 + Shift في لوحة المفاتيح، فمثلاً $2^8=144$ ، أنظر الصورة

The screenshot shows a software interface with three main panels:

- Workspace:** A table with columns 'Name', 'Value', and 'Class'. It contains one row: 'ans' with a value of '144' and a class of 'double'.
- Command Window:** A blue headered area containing the text:


```
>> 12^2
ans =
    144
>> |
```
- Command History:** A panel at the bottom showing the command '12^2' and the time '04:05' on '03/04/06'.

أخذ الجذر التربيعي:

يتم أخذ الجذر التربيعي لأي رقم عن طريق كتابة الأمر `sqrt`، أنظر الصورة التالية

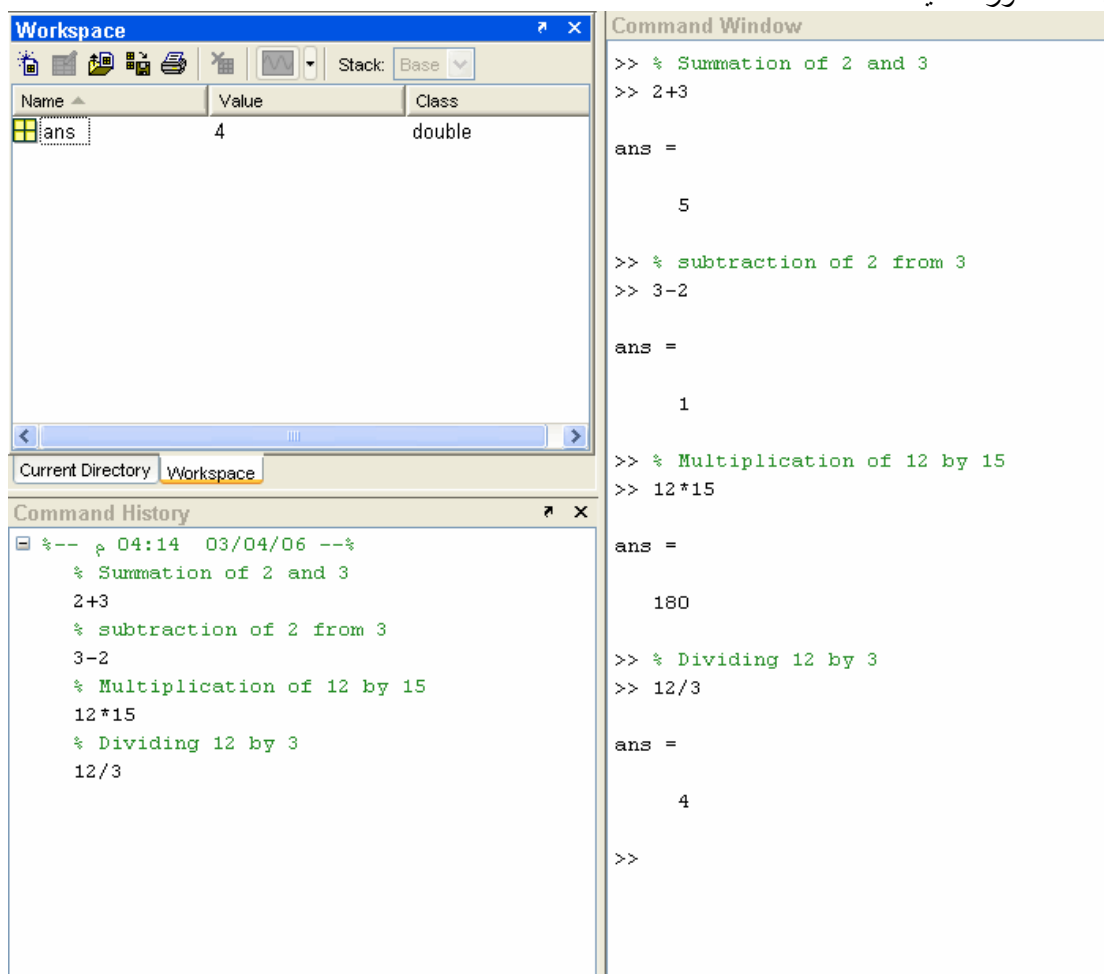
```
>> sqrt(144)
```

```
ans =
```

```
12
```

وضع عناوين أثناء البرمجة

كما تعودنا في برامج Qbasic و C++ وغيرها الكثير من برامج البرمجة، فيتم وضع عناوين لما نقوم به حيث تكون مثل المرجع لنا في معرفة ما نقوم به في جزء ما من البرنامج. ففي برنامج الماتلاب لوضع عنوان ما، لا بد من أن نبدأ بوضع علامة مئوية (%)، ثم نكتب ما نريده بعدها، لاحظ الصورة التالية



The screenshot shows a MATLAB workspace and command window. The workspace window displays a table with the following data:

Name	Value	Class
ans	4	double

The command window shows the following commands and outputs:

```
>> % Summation of 2 and 3
>> 2+3
ans =
    5
>> % subtraction of 2 from 3
>> 3-2
ans =
    1
>> % Multiplication of 12 by 15
>> 12*15
ans =
   180
>> % Dividing 12 by 3
>> 12/3
ans =
    4
>>
```

The Command History window shows the following commands:

```
%-- م 04:14 03/04/06 --%
% Summation of 2 and 3
2+3
% subtraction of 2 from 3
3-2
% Multiplication of 12 by 15
12*15
% Dividing 12 by 3
12/3
```

ولكن كما تلاحظون فهناك مشكلة في نافذة Workspace، حيث أنها سجلت آخر قيمة فقط، وذلك لأن كل النتائج الأربعة تأخذ رمز ans حيث اننا لم نجعل لها رمزا، أنظر الصورة

The screenshot shows the MATLAB environment with the following components:

- Workspace:** A table with columns 'Name', 'Value', and 'Class'. It contains one entry: 'ans' with a value of 4 and a class of 'double'. Below the table, there is Arabic text: "تم تسجيل آخر نتيجة فقط ، فما العمل؟".
- Command Window:** A list of commands and their outputs:


```
>> % Summation of 2 and 3
>> 2+3
ans =
    5
>> % subtraction of 2 from 3
>> 3-2
ans =
    1
>> % Multiplication of 12 by 15
>> 12*15
ans =
   180
>> % Dividing 12 by 3
>> 12/3
ans =
    4
>>
```
- Command History:** A list of the commands entered in the Command Window, including the same commands as above.

An orange arrow points from the 'ans' variable in the workspace to the 'ans = 4' output in the Command Window.

يتم تعريف النتائج بحروف، بحيث يأخذ الحرف القيمة التي يدخلها المستخدم له، أنظر الصورة

The screenshot shows the MATLAB workspace and command window. The workspace contains variables a, b, c, and d with values 2, 3, 5, and 1 respectively. The command window shows the following code and output:

```

>> % By defining the Inputs
>> a=2
a =
    2
>> b=3
b =
    3
>> % By Making summation of a & b
>> % Denoting the result of (a & b) as c
>> c=a+b
c =
    5
>> % By making subtraction of (a) from (b)
>> % Denoting the result of subtraction as (d)
>> d=b-a
d =
    1
>> |

```

Arrows indicate the flow of data from the workspace to the command window and back.

كما ترون فالمشكلة قد إنتهت، حيث ظهرت قيمة كل عملية بشكل منفرد

كما ترون فالمشكلة قد إنتهت تماماً، حيث تأخذ كل قيمة حرف معين. المشكلة التالية، هو أننا كلما أدخلنا قيمة، أو حصلنا على نتيجة تكون هنالك مشكلة، هو أن الماتلاب يقوم بإظهار القيمة المدخلة وكذلك النتيجة في نافذة الأوامر، مما يؤدي إلى كبر البرنامج المكتوب في حين أنه يؤدي شئ بسيط، أنظر الصورة التالية

The screenshot shows the command window with the following code and output:

```

>> a=2
a =
    2
>> b=3
b =
    3
>> % By Making summation of a & b
>> % Denoting the result of (a & b) as c
>> c=a+b
c =
    5

```

Annotations in Arabic explain the output format:

- القيمة المدخلة** (Input value): Points to the input command `a=2`.
- يقوم الماتلاب بإظهار القيمة المدخلة** (MATLAB displays the input value): Points to the output `a = 2`.
- المشكلة أننا كلما أدخلنا قيمة ما، يقوم الماتلاب بإظهار القيمة المدخلة أو حتى النتيجة وهذا بالتالي يأخذ من مساحة الكتابة كما يبين أن البرنامج كبير جداً،** (The problem is that whenever we enter a value, MATLAB displays the input value or even the result, and this takes up writing space, as it shows that the program is very large.)
- يقوم الماتلاب أيضاً بإظهار النتائج بشكل مباشر** (MATLAB also displays the results directly): Points to the output `c = 5`.

يتم أخفاء القيمة المدخلة وكذلك النتيجة من الظهور (ولكن عملية إدخال النتيجة والجمع مثلاً تتم بشكل طبيعي ويقوم الماتلاب بتنفيذ ما يأمره المستخدم) عن طريق وضع علامة (;) بعد كل قيمة مدخلة أو بعد طلب نتيجة

ما (الجمع مثلاً) ويتم إظهار النتيجة أو القيم المدخلة إذا طلب المستخدم ذلك ، عن طريق وضع حرف المدخلات أو النتيجة المطلوبة دون إستخدام الرمز المذكور (;) انظر الصورة التالية

The screenshot shows the MATLAB environment. The **Workspace** window on the left lists variables `a`, `b`, and `c` with values 2, 3, and 5 respectively, all of type `double`. The **Command Window** on the right shows the following commands and outputs:

```

>> % By defining the inputs
>> a=2;
>> b=3;
>> % By Making summation of (a) & (b)
>> % Bt denoting the result of summation by (c)
>> c=a+b;
>> % By acquiring the Inputs and result
>> a
a =
    2
>> b
b =
    3
>> c
c =
    5
>> |
  
```

Arabic annotations explain the output: "فمما بإستخدام الفصلة المتقطعة، وبالتالي فإن قيم المدخلات والنتائج لا تظهر في نافذة الأوامر، ولكنها تظهر في نافذة Workspace" (When using the semicolon, the input and output values do not appear in the command window, but they appear in the Workspace window). Another annotation says: "وإذا قمت بكتابة الحرف (إما يمثل مدخلات أو نتائج) دون الفصلة المتقطعة، فإن ذلك سيؤدي إلى ظهور القيم مباشرة" (If you write the letter (representing inputs or outputs) without the semicolon, it will lead to the direct appearance of the values).

بعض المتغيرات المعرفة مسبقاً في برنامج الماتلاب والمعروفة:

Predefined Variable	Stands For
pi	$\pi = 3.1416$
Inf	$\infty \equiv$ Infinity
NaN	Not a Number
i	The complex variable $\sqrt{-1}$
j	The complex variable $\sqrt{-1}$

يتم كتابة تلك المتغيرات المعرفة في برنامج الماتلاب

انظر الصورة التالية

Command Window

```
>> % The Following Command will show up the value of (pi)
>> pi

ans =

    3.1416

>> % The following command will show up the vlaue of (2*pi)
>> 2*pi

ans =

    6.2832

>> % the following Command will show up the value of square root of pi
>> sqrt(pi)

ans =

    1.7725
```

Command Window

```
>> % the following process will show the infinity
>> 1/0
Warning: Divide by zero.

ans =

    Inf

>> % the following command will show Not A Number
>> 0/0
Warning: Divide by zero.

ans =

    NaN

>> % the following command will show the complex number
>> i

ans =

    0 + 1.0000i

>> % the following command will show the complex number
>> j

ans =

    0 + 1.0000i
```

الكتابة فوق قيمة العدد المركب

تعلمنا أنه إذا كتبنا (i) في نافذة الأوامر يظهر التالي

```
>> % the following command will show the complex number
>> i

ans =

    0 + 1.0000i
```

كما يمكننا الكتابة فوق هذه القيمة، أي تغيير قيمته، حيث سنقوم بوضع قيمة لهذا الرمز، أنظر الصورة التالية

```

Command Window
>> % Overwriting the complex variable i
>> i=3;
>> a=1+3*i

a =

    10

>> % Notice that the presence of (*) has dealt (i) not complex but the value
>> % by the user
>> % If the multiplication sign has been removed so (i) represents complex No.
>> b=1+3i

b =

    1.0000 + 3.0000i

```

إلغاء القيم المدخلة والنتائج

يمكن للماتلاب مسح القيم المدخلة والنتائج (والتي تسجل في نافذة تسجيل النتائج)، دون مسح ما قمت بكتابته، وذلك بإستخدام أمر **Clear**، أنظر الصورة التالية

The screenshot shows the MATLAB interface with three windows: Workspace, Command Window, and Command History.

Workspace: A table showing the current state of variables in the workspace.

Name	Value	Class
a	10	double
b	12	double
c	120	double

Command Window: Shows the execution of MATLAB code. The output for 'c' is 120.

```

>> % By defining the inputs
>> a=10;
>> b=12;
>> % By multiplying (a) by (b)
>> % By denoting the result of mutiplication as (c)
>> c=a*b

c =

    120

>> |

```

Command History: Shows the commands entered in the Command Window.

```

%-- م 05:33 03/04/06 --%
% By defining the inputs
a=10;
b=12;
% By multiplying (a) by (b)
% By denoting the result of mutiplication
c=a*b

```

كما ترى فجميع القيم مسجلة في نافذة workspace

ولكن بعد تنفيذ أمر Clear


```

Workspace
Name Value Class
Current Directory Workspace

Command Window
>> % By defining the inputs
>> a=10;
>> b=12;
>> % By multiplying (a) by (b)
>> % By denoting the result of mutiplication as (c)
>> c=a*b

c =

    120

>> clear
>> |

Command History
%-- م 05:33 03/04/06 --%
% By defining the inputs
a=10;
b=12;
% By multiplying (a) by (b)
% By denoting the result of mutiplication
c=a*b
clear

```

وللتأكد قم بوضع أي حرف من الحروف التي قمت بتعريفها مسبقاً للماتلاب، ستلاحظ ان الماتلاب لا يتعرف عليها الآن، أنظر الصورة

```

Command Window
>> % By defining the inputs
>> a=10;
>> b=12;
>> % By multiplying (a) by (b)
>> % By denoting the result of mutiplication as (c)
>> c=a*b

c =

    120

>> clear
>> a
??? Undefined function or variable 'a'.

>> |

```

الماتلاب لم يعد يتعرف على المتغير (a) بعد تنفيذ أمر Clear

عملية المسح الجزى للمتغيرات:

ليس شرطاً أن نقوم بعملية مسح كلي لكل البرنامج، بل من الممكن عمل مسح لمتغير واحد فقط، عن طريق كتابة أمر Clear ثم إسم المتغير، ففي المثال السابق لدينا قيم لكلاً من (a) & (b) كما في الصورة التالية

The screenshot shows the MATLAB workspace and command window. The workspace contains three variables: 'a' with value 10, 'b' with value 12, and 'c' with value 120. The command window shows the following code:

```
>> % By defining the inputs
a=10;
b=12;
% By multiplying (a) by (b)
% By denoting the result of multiplication as (c)
c=a*b

c =

    120

>> |
```

Arrows indicate the flow of data: a green arrow from 'a' to the command window, a pink arrow from 'b' to the command window, and a cyan arrow from 'c' to the command window.

ثم سنقوم بمسح قيمة (a) فقط، أنظر الصورة تالية

The screenshot shows the MATLAB workspace and command window after clearing variable 'a'. The workspace now contains 'b' (12) and 'c' (120). The command window shows the following code:

```
>> % By defining the inputs
a=10;
b=12;
% By multiplying (a) by (b)
% By denoting the result of multiplication as (c)
c=a*b

c =

    120

>> % By clearing the value of a
>> clear a
>> a
??? Undefined function or variable 'a'.

>> b
b =

    12

>>
```

Arrows indicate the flow of data: a green arrow from 'b' to the command window and a blue arrow from 'b' to the command window.

Text annotations in Arabic:

- لا توجد قيمة للمتغير (a)، بعد تنفيذ أمر (Clear a)
- قمنا بمسح قيمة (a) فقط، لاحظ اختفاء قيمة (a) من نافذة Workspace
- لا تزال قيمة المتغير (b) موجودة، بينما لا توجد قيمة للمتغير (a)

The Command History window shows the following commands:

```
% -- م 05:43 03/04/06 --%
% By defining the inputs
a=10;
b=12;
% By multiplying (a) by (b)
% By denoting the result of multiplication
c=a*b
% By clearing the value of a
clear a
a
b
```

أكتفي هنا بهذا الكم اليوم
ونستكمل بإذن الله غداً
أسألكم الدعاء

السلام عليكم ورحمة الله وبركاته



تحية طيبة وبعد ..

إخواني الكرام، نستكمل بعض الأوامر الخاصة ببرنامج الماتلاب، وسنتناول بإذن الله

الدوال المثلثية Trigonometric functions
الدوال المثلثية العكسية Inverse Trigonometric functions
الدوال الزائدية Hyperbolic Functions
الدوال الزائدية العكسية Inverse Hyperbolic functions

أولاً: الدوال المثلثية Trigonometric Functions

Built In Function	Trigonometric Function
sin	Sine
cos	Cosine
tan	Tangent
sec	Secant
csc	Cosecant
cot	Cotangent

ملاحظة: يقوم الماتلاب بقياس الزوايا بالتقدير الدائري Radian
لاحظ الصورة التالية

The screenshot shows the MATLAB interface. The workspace window displays three variables: v, x, and y, all with a value of 1 and a class of double. The command window shows the following commands and outputs:

```
>> % Defining the Sine function
>> x=sin(pi/2)
x =
    1

>> % Defining the Cosine Function
>> y=cos(2*pi)
y =
    1

>> % Defining the Tangent Function
>> v=tan(pi/4)
v =
    1.0000

>> |
```

يتم التعويض بقيم مختلفة للزوايا في الدوال المثلثية

```
>> % By defining the secant function
>> a=sec(2*pi)

a =

    1

>> % By defining the cosecant function
>> b= csc(pi/2)

b =

    1

>> % By defining the cotangent function
>> c= cot(pi/4)

c =

    1.0000
```

وسيتم شرح هذا الجزء بالتفصيل أكثر في الجزء الخاص بـ Plotting 2D

الدوال المثلثية العكسية:

<u>Built In function</u>	<u>Inverse Trigonometric Function</u>
asin	<u>Inverse Sine</u>
acos	<u>Inverse Cosine</u>
atan	<u>Inverse tangent</u>
asec	<u>Inverse Secant</u>
acsc	<u>Inverse Cosecant</u>
acot	<u>Inverse Cotangent</u>

أنظر الصورة التالية لترى مدى قابلية الماتلاب على حل تلك الأجزاء بسهولة تامة

```
>> % By defining the Inverse sine function
```

```
>> a=asin(1)
```

```
a =
```

```
1.5708
```

يمكننا تعريف الدوال المثلثية العكسية بالطريقة التالية، ما هي قيمة الزاوية التي إذا أخذنا لها Sine نحصل على العدد 1 بالتأكيد ستكون $(\pi/2)=1.5708$

بنفس الطريقة لكل الدوال المثلثية العكسية

```
>> % By defining the Inverse Cosine Function
```

```
>> b=acos(1)
```

```
b =
```

```
0
```

نحصل على زاوية مقدارها صفر أو $\pi*2$ ، إذا أخذنا Inverse Cosine للعدد 1

```
>> % By defining the Inverse Tangent function
```

```
>> c=atan(1)
```

```
c =
```

```
0.7854
```

الزاوية المناظرة لدالة المماسية العكسية للعدد واحد هي $\pi/4=0.7854$

```
>> % By applying the Inverse secant function
```

```
>> d=asec(1)
```

```
d =
```

```
0
```

قيمة الزاوية التي تجعل دالة القاطع تساوي واحد هي صفر أو $\pi*2$

```
>> % By applying the Inverse Cosecant function
```

```
>> e=acsc(1)
```

```
e =
```

```
1.5708
```

قيمة الزاوية التي تجعل دالة تمام القاطع تساوي 1 هي $\pi/2=1.5708$

```
>> % By applying the Inverse cotan function
>> f=acot(1)
```

```
f =
0.7854
```

قيمة الزاوية التي تجعل قيمة تمام التماس يساوي واحد هي $\text{Pi}/4=0.7854$

الدوال الزائدية Hyperbolic functions

Built in functions	Inverse Hyperbolic functions
sinh	Hyperbolic Sine
Cosh	Hyperbolic Cosine
Tanh	Hyperbolic Tangent
Sech	Hyperbolic Secant
Csch	Hyperbolic Cosecant
Coth	Hyperbolic Cotangent

بعض العلاقات الهامة بالنسبة للدوال الزائدية

$$\sinh(z) = \frac{e^z - e^{-z}}{2}$$

أنظر الصورة التالية للتحقق من النتيجة باستخدام الماتلاب

```
>> % Comparing the result of (sinh) and the value of (exp(x)-exp(-x))/2
>> x=1
```

```
x =
1
```

```
>> a=sinh(x)
```

```
a =
1.1752
```

فكما هو واضح فإن قيمة **a** و **b** متساويتين، وهذا يحقق العلاقة

```
>> b=(exp(1)-exp(-1))/2
```

```
b =
1.1752
```

$$\cosh(z) = \frac{e^z + e^{-z}}{2}$$

أنظر الصورة التالية للتحقق من النتيجة باستخدام الماتلاب

```

>> % Comparing result of (cosh) and the value of (exp(x)+exp(-x))/2
>> x=1

x =

    1

>> a=cosh(1)
a =

    1.5431

>> b=(exp(x)+exp(-x))/2
b =

    1.5431

```

تلاحظ أن القيم قد تساوت لكلاً من (a) و (b) وهذا يحقق العلاقة

$$\tanh(z) = \frac{\sinh(z)}{\cosh(z)}$$

```

>> % By getting (sinh) function
>> x=1;
>> a=sinh(x)

```

```

a =

    1.1752

```

```

>> % By getting (cosh) function
>> b=cosh(x)

```

```

b =

    1.5431

```

```

>> c=a/b

```

```

c =

    0.7616

```

```

>> % By getting (tanh) function
>> d=tanh(x)

```

```

d =

    0.7616

```

تساوت قيم كلاً من (c) و (d) وبالتالي فإن العلاقة المذكورة أعلاه صحيحة

$$\operatorname{sech}(z) = \frac{1}{\cosh(z)}$$

```
>> % By getting (cosh) function  
>> b=cosh(x)
```

```
b =  
  
1.5431
```

```
>> c=1/b
```

```
c =  
  
0.6481
```

تساوت قيمة (c) & (d) وهذا
يؤكد العلاقة

```
>> % By getting hyperbolic secant function  
>> d=sech(x)
```

```
d =  
  
0.6481
```

$$\operatorname{csch}(z) = \frac{1}{\sinh(z)}$$

```
>> % By getting (sinh) function  
>> x=1;  
>> a=sinh(x)
```

```
a =  
  
1.1752
```

```
>> c=1/a
```

```
c =  
  
0.8509
```

تلاحظ تساوي قيمة (c) & (d) وهذا
بحقق العلاقة السابقة

```
>> % By getting hyperbolic cosecant function  
>> d=csch(x)
```

```
d =  
  
0.8509
```


$$\coth(z) = \frac{1}{\tanh(z)}$$

```
>> % By getting (tanh) function
>> x=1;
>> d=tanh(x)

d =

    0.7616

>> e=1/d
e =

    1.3130

>> % By getting the hyperbolic cotangent function
>> f=coth(x)

f =

    1.3130
```

تلاحظ تساوي قيمة (e) & (f) وهذا يحقق العلاقة

الدوال الزائدية العكسية

Built in function	Inverse Hyperbolic Functions
Asinh	Inverse hyperbolic Sine
Acosh	Inverse hyperbolic Cosine
Atanh	Inverse hyperbolic tangent
Asec	Inverse hyperbolic secant
Acsc	Inverse hyperbolic cosecant
Acot	Inverse hyperbolic cotangent

بعض القوانين الهامة للدوال الزائدية العكسية

$$\coth^{-1}(z) = \tanh^{-1}\left(\frac{1}{z}\right)$$

$$\sinh^{-1}(z) = \log \left[z + (z^2 + 1)^{\frac{1}{2}} \right]$$

$$\cosh^{-1}(z) = \log \left[z + (z^2 - 1)^{\frac{1}{2}} \right]$$

$$\tanh^{-1}(z) = \frac{1}{2} \log\left(\frac{1+z}{1-z}\right)$$

$$\operatorname{sech}^{-1}(z) = \cosh^{-1}\left(\frac{1}{z}\right)$$

$$\operatorname{csch}^{-1}(z) = \sinh^{-1}\left(\frac{1}{z}\right)$$

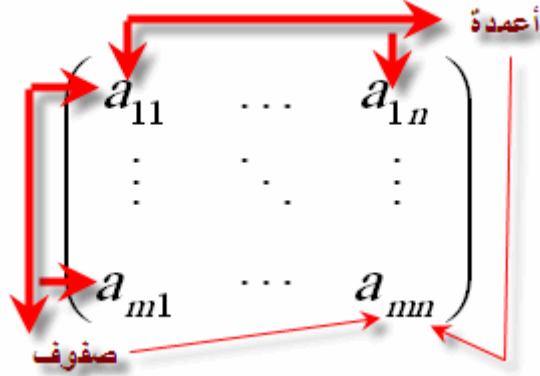
يقوم الماتلاب من خلال التعويض بالمتغير (**z**) في المعادلات الموضحة الحصول على الدوال الزائدية العكسية.



أخواني الكرام، نستكمل معاً الأسبوع الثاني برنامج الماتلاب، وسنتناول بإذن الله التالي
المصفوفات Matrices

ونتناول المواضيع كالتالي
ماهي المصفوفات
كيفية كتابة المصفوفات في برنامج الماتلاب
العمليات الحسابية في المصفوفات
مصفوفات خاصة
إستخراج عنصر محدد من المصفوفة
تغيير عنصر ما في المصفوفة

ماهي 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 في هذا الأسبوع بإذن الله.

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 (,) 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 (;)، أنظر الصورة التالية

```
>> % Enterring the value of matrix in different trends
>> % By defining the Matrix A
>> A=[1,3;6,4]
```

```
A =
     1     3
     6     4
```

ضرورة تواجد القوسين

تم استخدام الفاصلة، للفصل بين عناصر قيم الصف الواحد

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

```
A =
     1     3
     6     4
```

كما تم إدخال الفاصلة المنقوطة، لدلالة على إنتهاء قيم الصف المدخل، وإدخل قيم الصف الذي

```
>> A=[1 3
6 4]
```

```
A =
     1     3
     6     4
```

لم نستخدم هنا الفاصلة، وإكتفينا بعمل مسافة بين قيم الصف الواحد، وهذا طبعاً أفضل للسرعة

```
>>
```

لم نستخدم الفاصلة المنقوطة للفصل بين قيم الصفوف، وإكتفينا بالضغط على مفتاح Enter لإدخال قيم الصف التالي، وهذا طبعاً أفضل للسرعة

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

يستكمل....

أخواني الأعزاء، أعتذر في بادئ الأمر على طول فترة الغياب عن الشرح، والسبب كان بسبب ظروف الإمتحانات، أعاننا الله جميعاً إلى التفوق إنه ولي ذلك والقادر عليه نستكمل معكم اليوم شرح الجزء الخاص بالعمليات الحسابية على المصفوفات

فما هي العمليات الأساسية التي تتم على المصفوفات؟

- ١- الجمع
- ٢- الطرح
- ٣- الضرب
- ٤- القسمة
- ٥- المصفوفة الأسية

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

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix} \quad \begin{array}{l} \text{كما ترى فإن عدد} \\ \text{الصفوف في} \\ \text{المصفوفة الأولى} \\ \text{مساوياً لعدد الصفوف} \\ \text{في المصفوفة الثانية،} \\ \text{وكذلك عدد الأعمدة} \\ \text{لكننا المصفوفتين} \end{array}$$

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

كيف تتم عملية جمع مصفوفتين:

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

الجمع
1+7

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

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

هكذا يكون شكل

لتحصل على هذه النتيجة

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

الجمع في الماتلاب

يجب أولاً كتابة المصفوفتين B&A، كما تعلمنا سابقاً
ثم استخدام رمز الجمع (+) للتمم 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
```

.....يستكمل

طرح المصفوفات

فما هو شرط طرح المصفوفات؟

حقيقة هي 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 هي عدد الأعمدة
أنظر الصورة التالية

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

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

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

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

لنقم الآن بعمل نفس المثال على برنامج الماتلاب
أنظر الصورة التالية

```
Command Window
>> % 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
```

ملتقى المهندسين العرب

كما ترى فلقد حصلنا
على نفس الناتج السابق

ضرب المصفوفات



ما هو شرط ضرب المصفوفات؟

شرط ضرب أي مصفوفتين هو أن يكون عدد أعمدة المصفوفة الأولى $n1$ مساوياً لعدد الصفوف في المصفوفة الثانية $m2$

أنظر الصورة التالية

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

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

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

$$C = \begin{pmatrix} (1 \times 0) + (2 \times 4) & (1 \times 3) + (2 \times 9) & (1 \times 3) + (2 \times 7) \\ (4 \times 0) + (6 \times 4) & (4 \times 3) + (6 \times 9) & (4 \times 3) + (6 \times 7) \\ (9 \times 0) + (8 \times 4) & (9 \times 3) + (8 \times 9) & (9 \times 3) + (8 \times 7) \end{pmatrix}_{3 \times 3}$$

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

ملتقى المهندسين العرب

هذه هي عملية ضرب
المصفوفات بالطريقة اليدوية

لنقوم الآن بإدخال نفس المثال على الماتلاب



أنظر الصورة التالية

```
Command Window
>> % 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 من وجود كلمة القسمة للمصفوفات، ولكن الحقيقة أنها موجودة ومستخدمة بكثيرة ولكننا لا ننتبه لوجودها، فبهذه القسمة نقوم بحل المعادلات والتي سيتم شرحها لاحقاً بإذن الله وقبل أن أشرح لكم كيفية عمل القسمة، لابد من شرح كيفية حل المعادلات كثيرة الحدود لنفترض أن لدينا معادلتان كالآتي

$$3X + 3Y = 3$$

$$2X + 3Y = 5$$

وكلتا المعادلتان يمكن حلها 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 Multipliyin g by $(\frac{3}{2} \times R_2 - R_1)$

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

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

$$\therefore 1.5Y = 4.5$$

$$\therefore Y = 3$$

$$\therefore 3X + 3Y = 3$$

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

$$\therefore X = -2$$

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 A على B, كما في الصورة التالية

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

$$\frac{1}{A}$$

ولكن ماذا تعني من ناحية المصفوفات وليست الأعداد؟

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

Where $\text{inv}()$ 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

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

2x2 2x1

يجب الإنتباه لشرط عملية ضرب المصفوفة

فإذا قمنا بكتابة المعادلتين في الماتلاب كما في الصورة السابقة

```
Command Window
>> % By defining the Coefficient Terms
>> A=[3 3;2 3];
>> % By Defining the Absolute Terms
>> B=[3;5];
>> C=inv(A)*B

C =
-2
 3
>>
```

كما ترى فلقد حصلنا على نفس القيم التي حصلنا عليها باستخدام طريقة الحذف

X=-2

Y=3

الدرس القادم بإذن الله هو

العمليات على المصفوفات

العمليات على المصفوفات والمتجهات



المتجهات هي مصفوفة ولكن بعמוד واحد
مثلا المتجه العمودي
المتجهات هي مصفوفة ولكن بعמוד واحد
مثلا المتجه العمودي

```
>> 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 وأعمدها عن صف واحد أو عمود واحد
وسنتناول العمليات التي تتم على المتجهات أولاً ثم المصفوفات

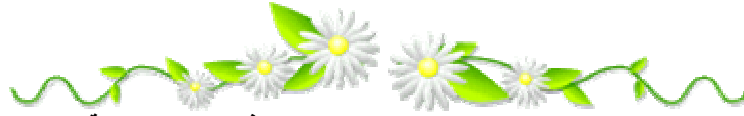


ماهي العمليات الشائعة على المتجهات؟

- 1- طول المتجه
- 2- إضافة عنصر
- 3- إستبدال عنصر
- 4- عملية حذف عنصر
- 5- نداء عنصر
- 6- نداء عدد عناصر
- 7- إيجاد العنصر الأكبر
- 8- إيجاد العنصر الأصغر
- 9- إيجاد حاصل ضرب العناصر

هذه هي العمليات الشائعة في الماتلاب وسنتناول كل منها بالتفصيل بإذن الله

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
```

تعريف متجه صفي

التي الأولى وهي
صفوفة

!

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 10

يمكن عمل نفس العملية على متجه عمودي
وننتقل الآن إلى العملية التالية

إضافة عنصر

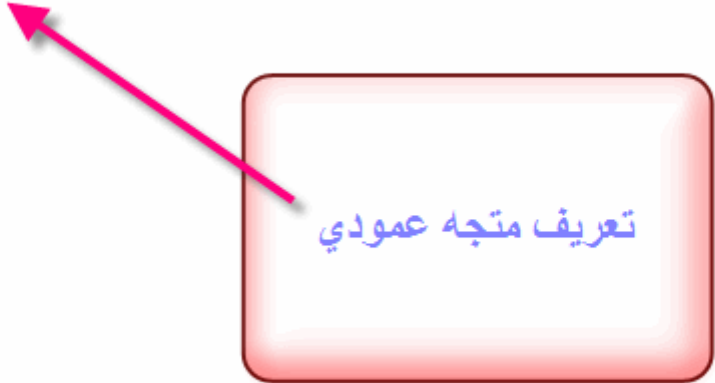


لنقوم بوضع متجه عمودي في الماتلاب، كما في الصورة التالية

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

A =

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



كما هو واضح، أن عدد العناصر الموجودة في هذا المتجه هو ١٠، 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 =

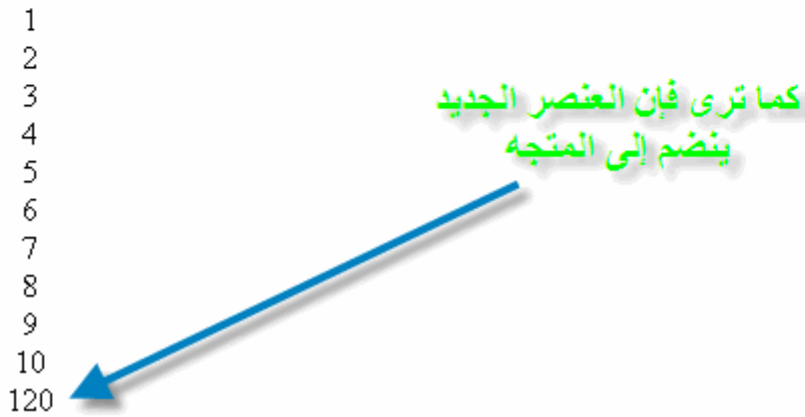
    10
```

لنقل أننا نريد إضافة الرقم ١٢٠ في الخانة الحادية عشرة، أي الخانة التالية للخانة العاشرة، أنظر الصورة التالية

A =



A =



ملاحظة: في المثال السابق تمت إضافة الرقم ١٢٠ إلى الخانة ١١، فماذا إذا قمنا بإضافة رقم جديد ولكن في الخانة رقم ١٣، فماذا ستكون قيمة الخانة ١٢ التي لم يتم إضافة أي عنصر لها، أنظر الصورة التالية

Command Window

1
2
3
4
5
6
7
8
9
10
120

تمت إضافة العنصر ١٤٠ إلى
الخانة رقم ١٣

>> A(13)=140

A =

1
2
3
4
5
6
7
8
9
10
120
0
140

كما ترى فإن الماتلاب يفترض
قيمة الخانة ١٢ بصفر، وعلى
الرغم من عدم إدخالنا لقيمتها،
لذلك نستنتج أن أي خانة تقوم
بتخطيها يقوم الماتلاب بفرض
قيمتها بصفر

وبهذا نكون قد شرحنا الجزء المتعلق بإضافة عنصر

إضافة أكثر من عنصر متتالي



لنفترض أننا نريد إضافة مجموعة من العناصر المتتالية في الخانات ١١ و 12 و ١٣ 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 يبدو ذلك مستنفذاً للوقت، إذا تم إدخال ١٠٠ 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 ١٠ نكتب التالي

1:10

وعندما نريد كتابة مجموعة من الأرقام المتتالية من 10 إلى ١٢٠٠ نكتب

10:1200

وبالتالي إذا أردنا كتابة مجموعة من الأرقام المتتالية من ١١ إلى ١٣ 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

يتم تحديد قيم الخانات
بشرط أن يتم وضعها في
قوسين

[قيم الخانات]

تم تحديد الخانات المتتالية
من ١١ إلى ١٣

وبذلك نكون قد شرحنا كيفية إضافة مجموعة من العناصر المتتالية

إستبدال عنصر



عملية إستبدال عنصر تتطلب عدة شروط

1- أن يكون العنصر موجوداً بالفعل

2- أن تحدد مكان هذا العنصر

ففي المثال التالي أردنا أن نستبدل 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

Command Window

To get started, select MATLAB Help or Demos from the Help menu.

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

```
A =
```

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

قيمة العنصر الثالث قبل التغيير

```
>> A(3)=15
```

```
A =
```

```
1  
2  
15  
4  
5  
6  
7  
8  
9  
10
```

قيمة العنصر الثالث بعد التغيير

وعلى هذا المنوال تستطيع أن تغير أي عنصر في المتجه
وبهذا نكون قد إنتهينا من شرح عملية إستبدال عنصر واحد في المتجه

إستبدال مجموعة عناصر متتالية



كما شرحنا كيفية إضافة مجموعة عناصر متتالية، سنقوم بإستبدال مجموعة عناصر متتالية 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

>> A(6:10)=[0 0 0 0 0]

A =

1
2
3
4
5
0
0
0
0
0
```

مجموعة العناصر في المتجه

تم تحديد مجموعة العناصر التي سيتم تغييرها

مجموعة العناصر بعد

وبهذا نكون قد إنتهينا من شرح عملية إستبدال مجموعة عناصر متتالية

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

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

A =

1
2
3
4
5
6
7
8
9
10

>> A(10)=[]

A =

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

تم تحديد العنصر العاشر لحذفه

يتم وضع قوس مربع فارغ ليبدل على أن هذه عملية حذف للعنصر

كما ترى إختفاء العنصر العاشر

وبهذا نكون قد أتمنا شرح الجزء المتعلق بحذف عنصر وحيد من المتجه

حذف مجموعة عناصر متتالية



لحذف مجموعة عناصر متتالية، 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

>> A(6:10)=[]

A =
     1
     2
     3
     4
     5
```

تم تحديد مجموعة العناصر
المطلوب حذفها

كما تلاحظ اختفاء مجموعة
العناصر التي تم تحديدها

وبهذا أكون قد أتممت شرح الجزء المتعلق بحذف مجموعة عناصر متتالية

نداء عنصر



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

Command Window

To get started, select MATLAB Help or Demos from the Help menu.

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

```
A =
```

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

نداء العنصر رقم ٥ وقيمته
٥ كما هو واضح

```
>> A(5)
```

```
ans =
```

```
5
```

وبهذا نكون قد أتممنا شرح نداء عنصر

نداء أكثر من عنصر



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

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

A =

     1
     2
     3
     4
     5
     6
     7
     8
     9
    10

>> A(6:10)

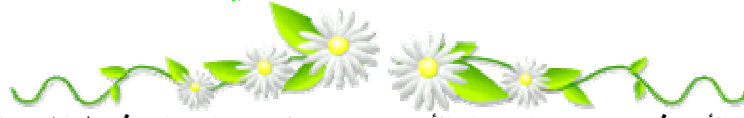
ans =

     6
     7
     8
     9
    10
```

تم تحديد مجموعة
العناصر الذين نريد
الحصول على قيمهم
داخل المتجه

وبهذا نكون قد شرحنا الجزء الخاص بنداء أكثر من عنصر

إيجاد العنصر الأكبر في المتجه



لإيجاد العنصر الأكبر في متجه، يتم استخدام الأمر `max`، حيث يمكن إيتخدامه في الماتلاب بالشكل التالي

Command Window

```
>> A=[10 22 36 41 44 59 61 73];
```

```
>> max(A)
```

```
ans =
```

```
73
```

```
>>
```

١- يجب عند إيجاد الرقم الأكبر

داخل المتجه كتابة الأمر `max`

ويجب أن يأخذ الصورة التالية

(إسم المتجه) `max`

٢- وهذا هو الرقم الأكبر داخل

المتجه

وبهذا نكون قد إنتهينا من شرح كيفية إيجاد الرقم الأكبر في المتجه

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

```
Command Window
>> A=[10 22 36 41 44 59 61 73];
>> min(A)
ans =
    10
>> |
```

١ - لإيجاد العنصر الأصغر في المتجه،
قم باستخدام الأمر **min**
حيث يأخذ الصورة التالية
min(اسم المتجه)

٢ - كما ترى فإن العنصر
الأصغر في هذا المتجه هو

أعتقد أننا الآن بدأنا في فهم كيفية الماتلاب بشكل جيد، وسنبدأ في التفاصيل بشكل تدريجي قريباً بإذن الله

إيجاد مجموع عناصر المتجه

يمكن جمع جميع عناصر المتجه، باستخدام الأمر **sum** حيث أن هذا الأمر لابد أن يأخذ طريقة في تنفيذه فيجب أن ينفذ بالصورة التالية

(إسم المتجه) **Sum**

وفالآن نقوم بعمل مثال في الماتلاب الآن

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

ans =

6
```

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

(إسم المتجه) **prod**

والآن لناخذ مثالاً تطبيقياً في الماتلاب

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

ans =

24
```

العمليات على المصفوفات

أولاً يجب تعريف أنواع المصفوفات، فهناك نوعان من المصفوفات

1- مصفوفة غير منتظمة

2- مصفوفة منتظمة أو مربعة

أما العمليات التي سوف تتم على المصفوفات فهي

1- طول المتجه

2- إضافة عنصر

3- استبدال عنصر

4- عملية حذف صف أو عمود بأكمله

5- نداء عنصر

6- نداء عدد عناصر

7- إيجاد العنصر الأكبر

8- إيجاد العنصر الأصغر

9- إيجاد مجموع عناصر المصفوفة

10- إيجاد حاصل ضرب العناصر

11- إيجاد قطر المصفوفة Diagonal

وهذه هي العمليات الشائع استخدامها في الماتلاب

وسيتم شرحها بالتفصيل بإذن الله

إيجاد حجم المصفوفة

لإيجاد حجم المصفوفة أو دعونا نقول لإيجاد عدد الصفوف والأعمدة لمصفوفة، يجب استخدام الأمر `size`، حيث لا يصلح استخدام الأمر `length`، فأمر `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
```

والآن 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
```

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 ٢٤ 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 بإفترضهما صفرًا.

فماذا إذا أردنا إضافة عدة عناصر في المصفوفة؟ يمكن إيضاح ذلك باستخدام المثال التالي
لنقوم أننا نريد إضافة الأعداد ٣١ و ٥٤ و ١٣ و 11 في الصف الرابع والعمود الأول والثاني والثالث والرابع على

التوالي، يمكن ذلك من خلال الماتلاب بالشكل التالي

```
>> B(4,1:4)=[31 54 13 11]
```

الأعمدة من الأول إلى الرابع

الصف الرابع

```
B =
```

1	3	7	8
2	6	5	11
12	14	15	13
31	54	13	11

العناصر الجديدة

إستبدال عنصر

قد تكون هذه العملية نادراً ما يتم إستخدامها، ولكنها هامة جداً، حيث توفر إمكانية إستبدال 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 بإستبدال 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 إستبدال 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

لايقوم الماتلاب بعملية حذف لعنصر واحد فقط في مصفوفة، حيث أنه من غير المعقول حذف عنصر من داخل المصفوفة، وبقيّة 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

وضع أقواس مربعة فارغة تعني عملية حذف

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

في خانة الأعمدة تم وضع (:) حيث تعني إختيار جميع الأعمدة

```
B =
```

```
1 3 7 8
2 6 5 11
```

الصف الثالث

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 بندااء 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
```

كلمة **end** تعني إختيار العنصر

وسنقوم بعد عملية الشرح تماماً بالعديد والعديد من الأمثلة التي تزيد من سرعتك ومهارتك في الماتلاب

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، وعندما 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، ولإختيار الرقم الأكبر بينهم يجب كتابة نفس الأمر للنتائج 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 الأمر `min` وإليك 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

حدي 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 حدى ويتم وضع الناتج في متجه، وإذا تم إستخدام 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
```

M-File

هي وسيلة لإدخال الأوامر ولكن ليس من خلال نافذة الأوامر، ولكن ماذا قد يختلف في هذه الوسيلة الجديدة في إدخال الأوامر؟

1- في عملية إدخال الأوامر التي كنا نستخدمها، إذا أردنا تعديل عنصر أو أكثر كان يجب إعادة إدخال الأمر من جديد.

2- إذا وجد خطأ، فيجب كتابة الأمر من جديد

3- إذا كتبنا برنامج كبير، وأردنا إعادة العملية مرة أخرى يجب إدخال جميع الأوامر من جديد وبنفس الترتيب.

4- إذا حدث خطأ في ترتيب الأوامر لهذا البرنامج الكبير ستقوم بإعادة الإدخال الأوامر من البداية مرة أخرى.

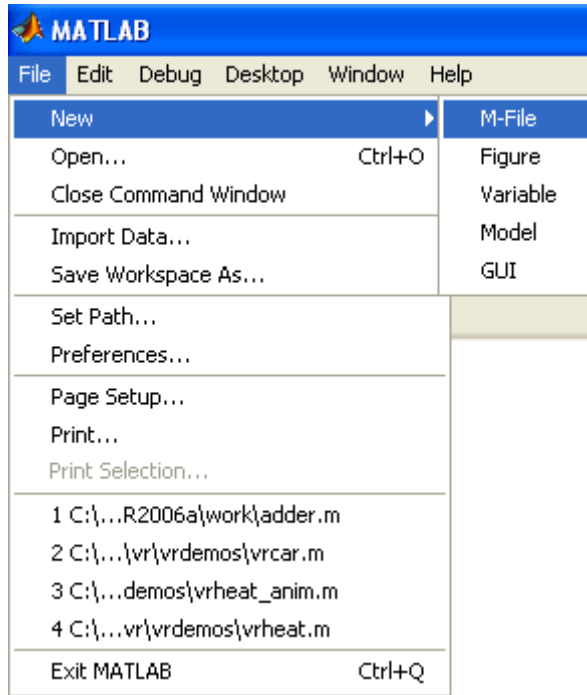
5- يصعب عمل عملية تصحيح للأخطاء Debugging

وهذا بالطبع يستغرق وقتاً كبيراً هذا بالإضافة إلى الملل الذي يحدث للمستخدم

وطبعاً حلاً لهذه المشكلة، تم عمل بما يسمى M-File والتي تعطي القدرة على كتابة البرنامج كاملاً أولاً بدون تشغيل، وبعد الإنتهاء منه يتم تشغيله، هذه الخاصية تعطي القدرة على تعديل القيم دون الحاجة إلى كتابتها مرة أخرى، أو إعادة

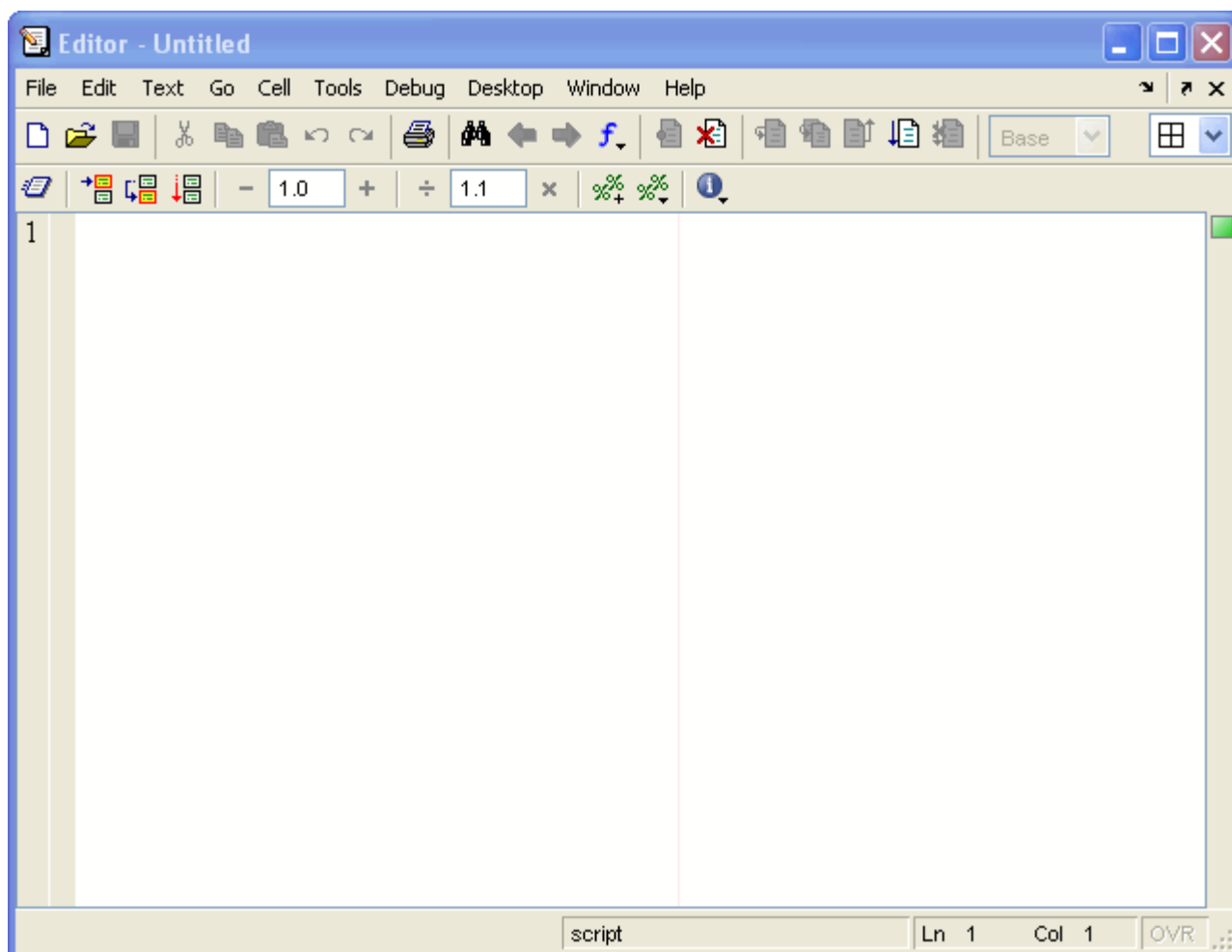
إدخال الأوامر التي تعتمد على هذا الأمر.

فكيف يتم تشغيل تلك الخاصية؟ إتبع الصورة التالية



وبالتالي ستظهر نافذه جديدة، تأخذ الشكل التالي

!Error

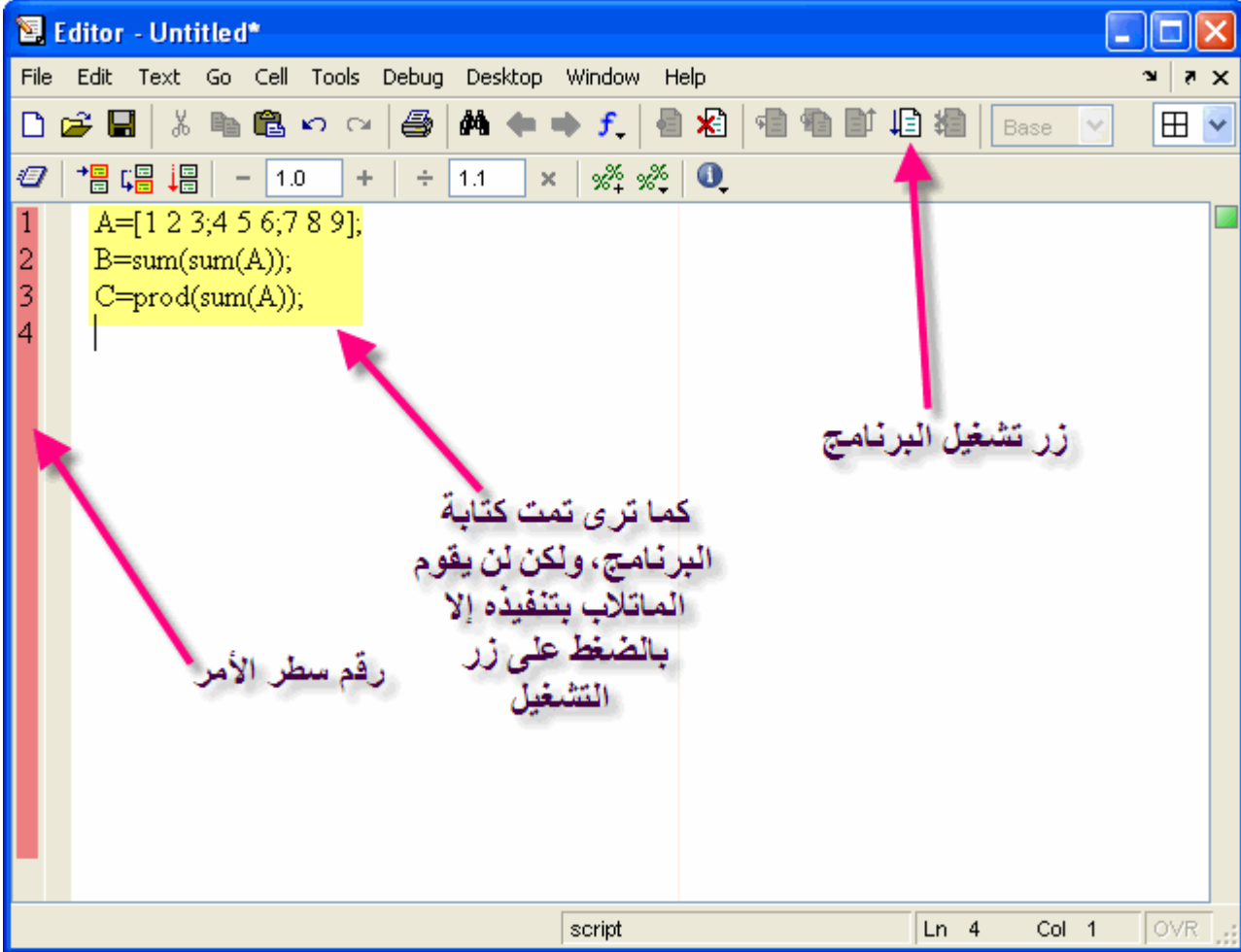


وستعرف على نافذة **M-File** بالتفصيل الممل بإذن الله في الدرس القادم

نافذة M-File

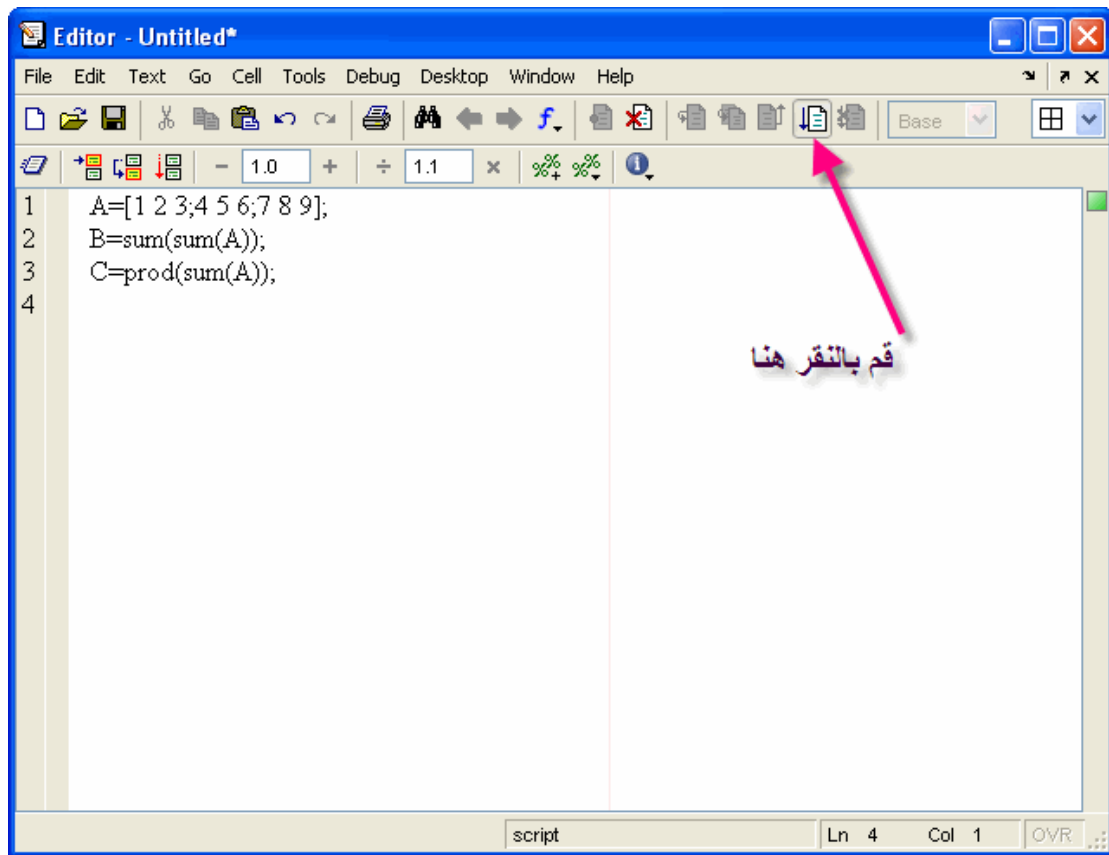
سنقوم الآن بالتعرف على نافذة M-File، أنظر الصورة التالية

!Error

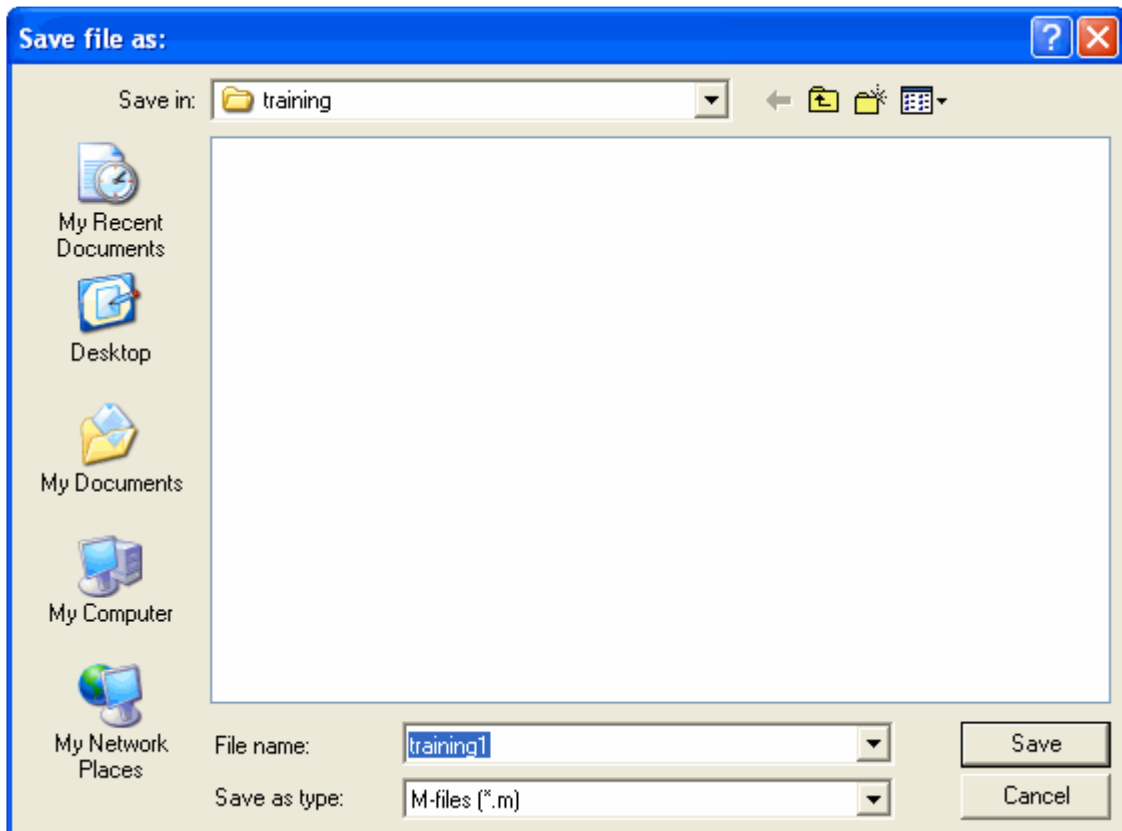


ولكن عند الضغط على زر التشغيل، سيطلبك الماتلاب بحفظ البرنامج، ولكن يشترط الآتي عند حفظ البرنامج

- 1- أن لا يبدأ بأرقام
 - 2- أن لا يكون أمراً معروفاً في الماتلاب
 - 3- أن لا يحتوي الاسم على مسافات فاصلة
 - 4- أن لا يحتوي على رموز خاصة مثل + ، - ، & ، *
- يجب مراعاة تلك الشروط وإلا لن يقوم الماتلاب بتنفيذ البرنامج
فالنقم بتنفيذ المثال المكتوب الآن في النافذة السابقة
1- يتم الضغط على زر التشغيل كما هو واضح في الصورة التالية



2- سيطلبنا الماتلاب بحفظ البرنامج أولاً، ولنسميه training1



3- ستظهر القيم في كلاً من Command Window and Workspace

The screenshot displays the MATLAB environment with the following components:

- Workspace:** A table showing variables A, B, and C. A is a 3x3 double matrix with values [1 2 3; 4 5 6; 7 8 9]. B is a double scalar with value 45. C is a double scalar with value 3240.
- Editor:** Contains the following MATLAB code:

```
1 - A=[1 2 3;4 5 6;7 8 9]
2 - B=sum(sum(A))
3 - C=prod(sum(A))
4
```
- Command History:** Lists the executed commands:

```
B=sum(diag(A))
clc
A=[1 15 2 11; 23 1 4 5; 3 1 15 7; 1 4 9 10]
B=prod(diag(A))
clc
clear
clc
help magic
magic(3)
magic(9)
clc
A=magic(3)
B=magic(9)
clc
```
- Command Window:** Displays the output of the commands:

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

B =
    45

C =
    3240
```

4- لنعود إلى M-File ونقوم بتغيير بعض القيم للمصفوفة، كما في الشكل التالي

```
1 - A=[11 22 33;41 51 61;78 88 98]
2 - B=sum(sum(A))
3 - C=prod(sum(A))
4 - |
```

5- سنقوم الآن بتشغيل البرنامج، وسيقوم الماتلاب الآن بالحفظ تلقائياً دون الحاجة لإعادة التسمية، ثم شاهد نافذة الأوامر **Command Window**

```
Command Window

1  2  3
4  5  6
7  8  9

B =
45

C =
3240

A =
11  22  33
41  51  61
78  88  98

B =
483

C =
4018560
```

هذه قيم البرامج التي قد حصلنا عليها منذ قليل

وهذه قيم البرنامج بعد عمل التعديلات عليه

وكما تلاحظ فإنه في كل عملية تحديث للبرنامج ستظل قيم البرنامج القديم موجودة, فحلاً لهذه المشكلة، يتم وضع الأمر **CLC** في أول كل برنامج، وهذا يكون مبدأ في جميع البرامج التي نقوم بعملها لا بد من أن تبدأ بهذا الأمر. ودعونا نقوم بمثال يوضح لنا ذلك

```
1 - clc
2 - A=[1 2 3;4 5 6;7 8 9]
3 - B=sum(sum(A))
4 - C=prod(sum(A))
5
```

Command Window

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

B =
45

C =
3240
```

سنقوم الآن بتعديل المثال، وحتى نتأكد أن أمر **CLC** يعمل، ستختفي القيم من **Command Window** وتظهر القيم الجديدة

Editor - C:\Documents and Settings\Eng.Ahmed Afifi\Desktop\training\training1.m

```

1 - clc
2 - A=[11 21 31;42 52 62;73 38 39]
3 - B=sum(sum(A))
4 - C=prod(sum(A))
5

```

Command Window

```

11 21 31
42 52 62
73 38 39

B =

369

C =

1846152

```

كما ترى فإن القيم السابقة إختفت وظهرت القيم الجديدة

وبهذا نتأكد من أن الأمر CLC يعمل بكفاءة
ولكن دعونا نشاهد نافذة Workspace والتي تحتوى على قيم A,B,C

Workspace

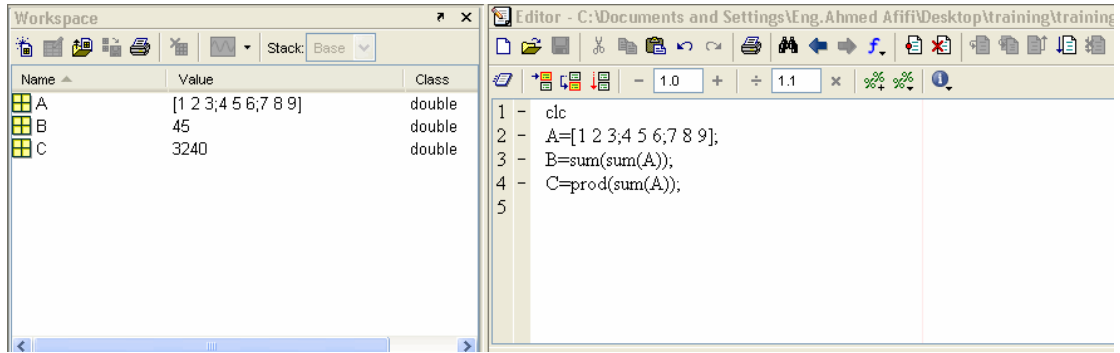
Name	Value	Class
A	[1 2 3;4 5 6;7 8 9]	double
B	45	double
C	3240	double

Editor - C:\Documents and Settings\Eng.Ahmed Afifi\Desktop\training\training1.m

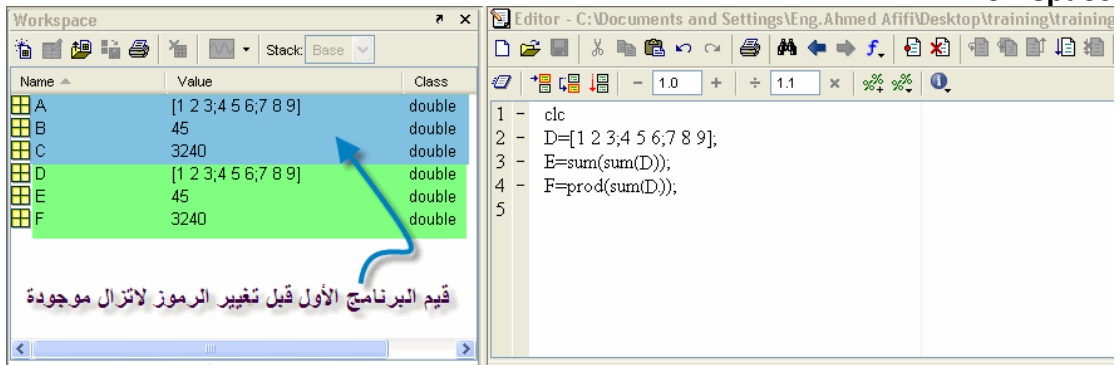
```

1 - clc
2 - A=[1 2 3;4 5 6;7 8 9];
3 - B=sum(sum(A));
4 - C=prod(sum(A));
5

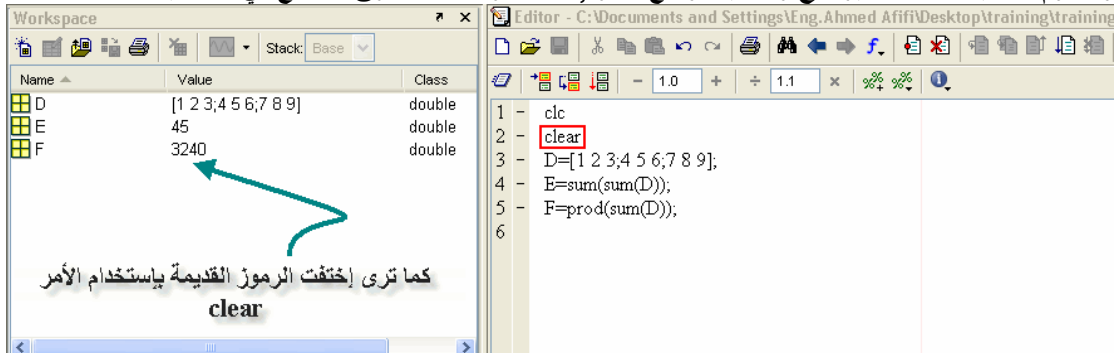
```



لنقم بتعديل بسيط في البرنامج عن طريق تغيير الرموز فقط من A,B,C إلى D,E,F ومشاهدة النافذة **Workspace**



ولتلافي هذه المشكلة، يجب وضع أمر **Clear** بعد الأمر **clc** بحيث يقوم بمسح أي قيمة سابقة من أي برنامج آخر في **Workspace**، ويجب تثبيت هذا الأمر أيضاً في جميع البرامج والتي سيتم عملها لاحقاً بإذن الله. وسنقوم الآن بتنفيذ نفس البرنامج ولكن بعد وضع الأمر **clear** وستلاحظ الفرق الشاسع في الماتلاب الآن

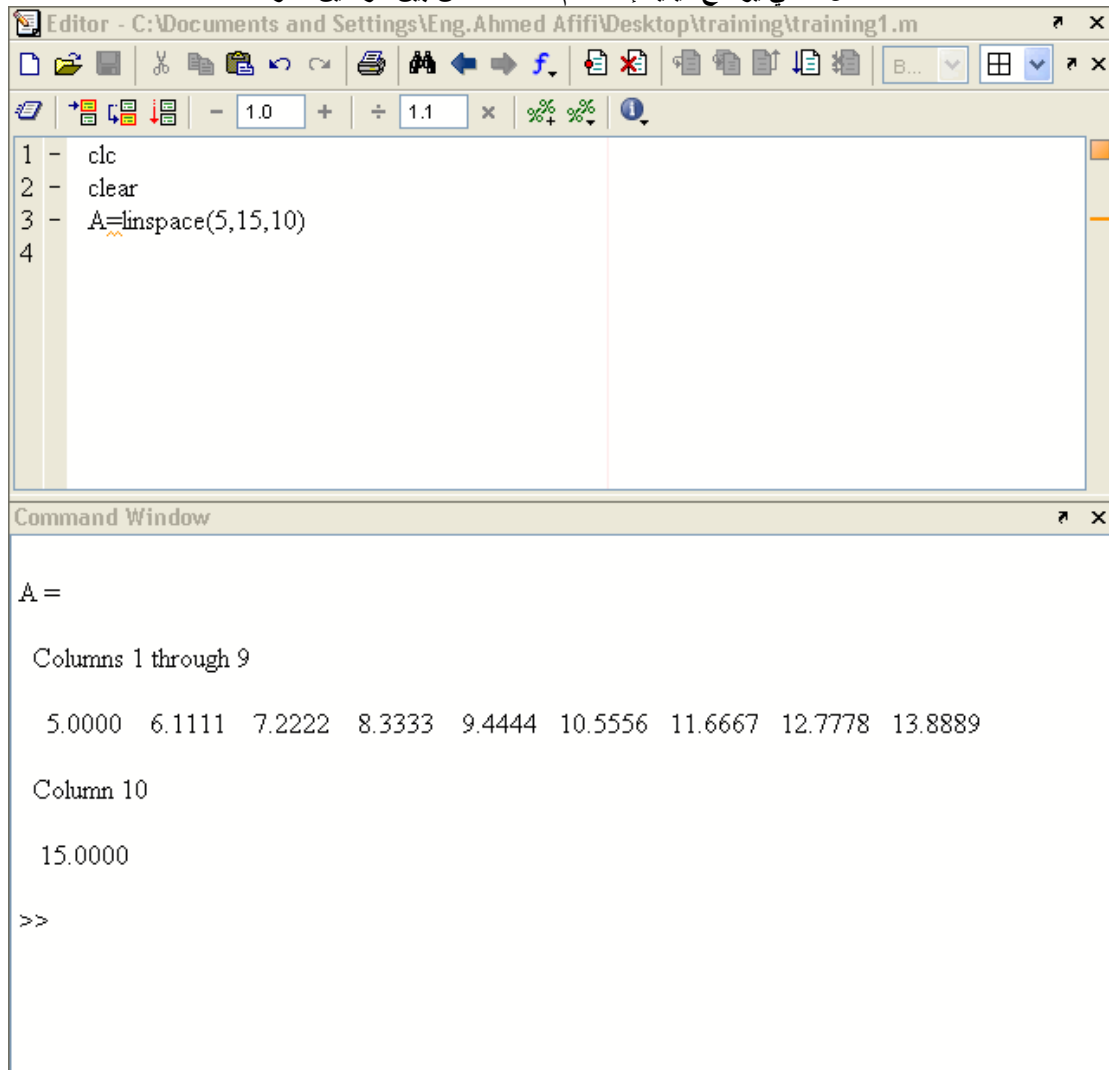


بعض الأوامر الهامة

قبل الإنتقال إلى الرسم **plotting** لابد من أخذ بعض الأوامر الهامة، والتي سيتم تداولها بشكل مستمر بإذن الله
linspace
هذا الأمر كثير الإستخدام، والذي يستخدم في عملية إنتاج متجه، عن طريق تحديد الرقم الأصغر والرقم الأكبر، وعدد
النقط المرغوبة بين هذين الرقمين
ويأخذ الصورة التالية

linspace(minimum number,maximum number,number of points in between)

فالمثال التالي يوضح كيفية إستخدام ١٠ نقاط من بين الرقمين ٥ و ١٥



The screenshot shows the MATLAB Editor window with the following code:

```
1 - clc
2 - clear
3 - A=linspace(5,15,10)
4
```

The Command Window displays the output of the command:

```
A =
Columns 1 through 9
 5.0000  6.1111  7.2222  8.3333  9.4444 10.5556 11.6667 12.7778 13.8889
Column 10
15.0000
>>
```

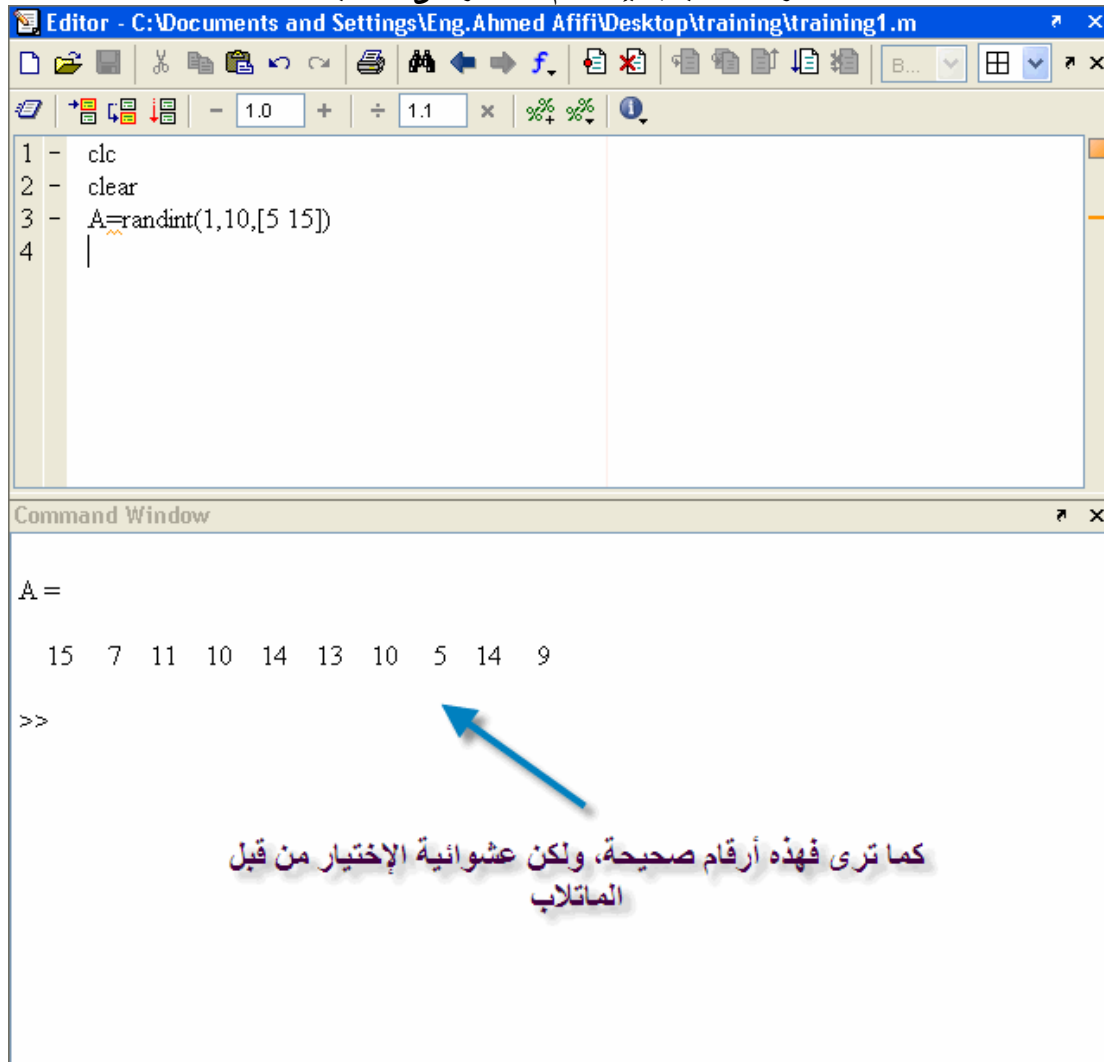
وهذا ما يسمى النظام العشوائي للماتلاب **Randomization System in Matlab**
والميزة في هذا النظام هو أن المتجه نظام عشوائي متزايد.

randint

هذا الأمر من ضمن الأوامر والتي تنشأ نظام عشوائي للأرقام، ولكن ليس نظام نظام عدد صحيح وليس على هيئة كسور مثل الأمر السابق، كما أن نظام الأرقام به ليس تزايدياً أو تناقصياً بل عشوائياً ويأخذ الصورة التالية

`randint(number of rows,number of column,[minimum number,maximum number])`

وهذا مثال بسيط باستخدام هذا الأمر على الماتلاب



The screenshot shows the MATLAB Editor window with the following code:

```
1 - clc
2 - clear
3 - A=randint(1,10,[5 15])
4 - |
```

The Command Window shows the output:

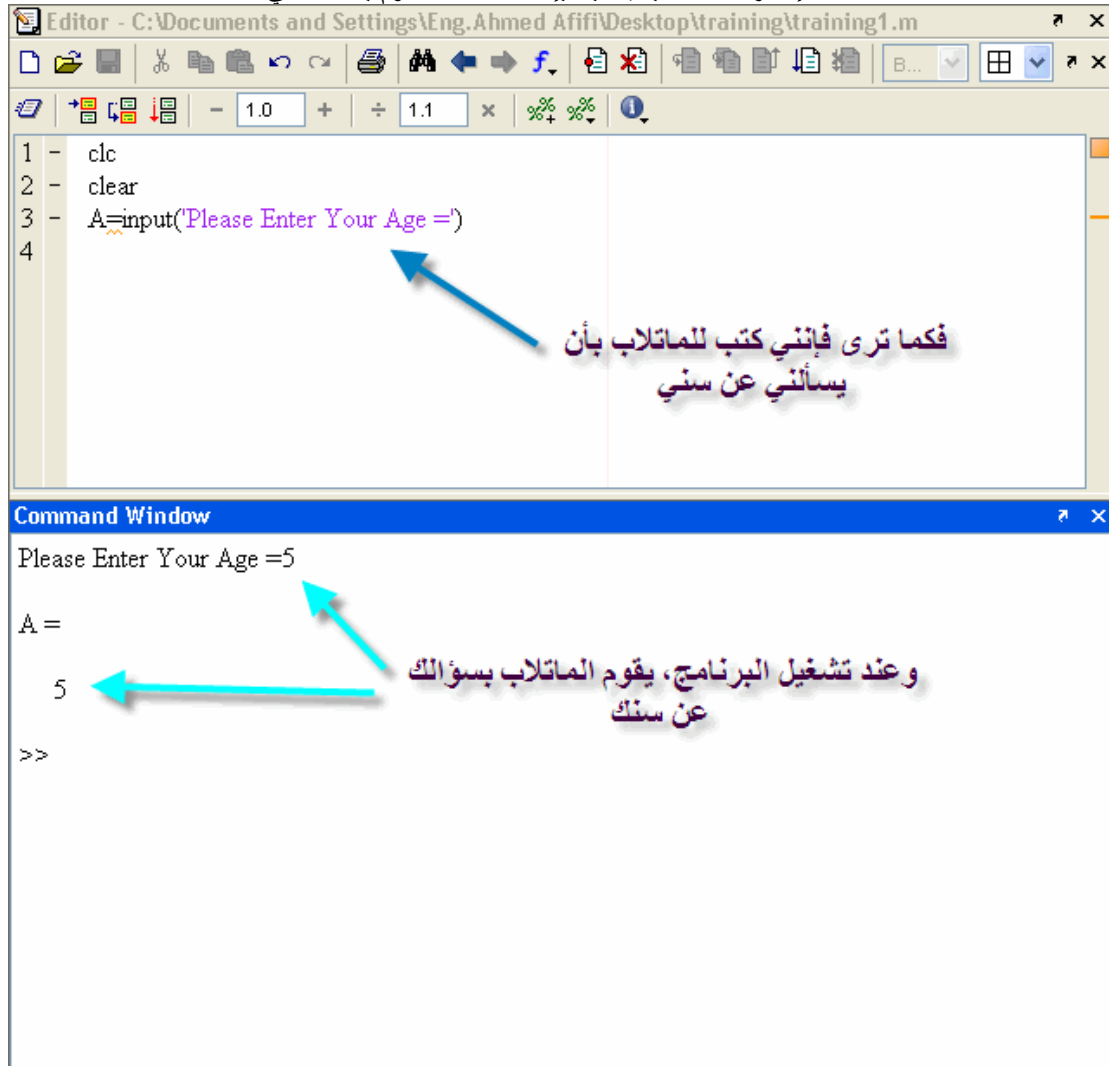
```
A =
15  7  11  10  14  13  10  5  14  9
>>
```

A blue arrow points to the number 5 in the output, with the following text below it:

كما ترى فهذه أرقام صحيحة، ولكن عشوائية الإختيار من قبل الماتلاب

Input

هذا الأمر هام جداً في الماتلاب، حيث يقوم المستخدم من خلال استخدامه يجعل الماتلاب يسأله عن متجه أو مصفوفة، أو حتى حروف وأسماء وذلك تبعاً لما يقوم المستخدم بتعريفه
فمثلاً إذا أردنا الماتلاب يطالبك بإدخال سنك، سنقوم بعمل التالي



The image shows a MATLAB Editor window with the following code:

```
1 - clc
2 - clear
3 - A=input('Please Enter Your Age =')
4
```

Below the code is the Command Window showing the execution:

```
Please Enter Your Age =5
A =
5
>>
```

Arabic annotations with arrows point to the code and the Command Window output:

- An arrow points from the text "فكما ترى فإنني كتب للماتلاب بأن يسألني عن سني" to the input function call in the code.
- An arrow points from the text "وعند تشغيل البرنامج، يقوم الماتلاب بسؤالك عن سنك" to the input prompt and the number 5 in the Command Window.

وإذا أردنا أن نعرف للماتلاب بأن ما سيتم إدخاله هو اسم وليس رقم، يتم كتابة البرنامج بالشكل التالي

```

1 - clc
2 - clear
3 - A=input('Please Enter Your Name : ','s')
4

```

فوجود حرف 'S' والتي تعني حرف أو string تعني أن الماتلاب مستعد لإن يتم إدخال إسم أو حرف كما هو واضح بالمثال

Command Window

```

Please Enter Your Name : Ahmed

A =

Ahmed

>>

```

Str2num & num2str

أولاً ما الفرق بين String and Character ؟

string هو حرف أو كلمة في الماتلاب

character هو عبارة عن رقم أو مجموعة من الأرقام

وهناك أمر يقوم بتحويل **string to Character** والعكس كذلك

وهما

num2str

str2num

ولكن فيما يفيدوا هذه الأوامر

سنقوم بمثال بسيط حتى تفهم المقصود من هذه الأمور، سنقوم باستخدام الأمر **input** في وضعية **string** وسنقوم بإدخال أرقام، ستبدأ تستغرب الآن، فهل الماتلاب حتى بعد إدخال الرقمة سيتعرف عليها كأنها أرقام أم أي شيء غير ذلك، شاهد الصورة التالية

The image shows a MATLAB environment with two windows: the Editor and the Command Window. The Editor window displays a script with three lines of code:

```
1 - clc
2 - clear
3 - age=input('Please Enter Your Age ','s')
```

A blue arrow points from the text below to the 's' character in the code. The text reads: "١- تم استخدام الأمر input في صورة string وليس character".

The Command Window shows the execution of the script. The prompt "Please Enter Your Age 5" is displayed, followed by "age = 5". A second blue arrow points from the text below to the number "5". The text reads: "٢- قمنا بإدخال رقماً للتعريف على أنه سن الإنسان، ولكن هل تعرف الماتلاب على أنه رقم".

Below that, the command ">> check=2*age" is entered, resulting in "check = 106". A third blue arrow points from the text below to the number "106". The text reads: "٣- في الحقيقة لم يتعرف الماتلاب على أنه رقم بل تعرف على أنه string وهذا نتيجة استخدام الأمر input في وضعية string".

سنقوم الآن باستخدام `str2num` لتحويل `string` إلى `character` أو رقم، شاهد الصورة التالية

```
1 - clc
2 - clear
3 - age=input('Please Enter Your Age ','s');
4 - age_modified=str2num(age)

Please Enter Your Age 5

age_modified =

     5

>> check=2*age_modified

check =

    10

>>
```

في هذه المرة تعرف الماتلاب على أن الذي يتم إدخاله هو رقم وهذا واضح من خلال عملية الضرب في 2 كعملية تأكيد، وذلك بسبب استخدام الأمر `str2num`

ماذا بخصوص الأمر الثاني `num2str` يستخدم هذا الأمر في تحويل الأرقام إلى `string` كتعريف لدى الماتلاب، ولكنه لا يغير من شكله، فالمثال التالي طلب منا إدخال الإسم والسن، ثم سنضع الإسم والسن في متجه، ولكن يجب أن نراعي كما ذكرنا مسبقاً أن المتجه إما يحتوي على أرقام أو `string` فقط وليس كلاهما


```
Editor - C:\Documents and Settings\Eng.Ahmed Afifi\Desktop\training\training1.m
1 - clc
2 - clear
3 - name=input('Please Enter Your Name ','s');
4 - age=input('Please Enter Your Age = ');
5 - answer=['Your name is ',name,' and you are ',num2str(age),' Years Old']

- ٢- وهذه هي صورة المنتج، ولكن كما ترى تم وضعها في صورة string
مما تطلب تحويل الرقم age إلى string باستخدام num2str

Command Window
Please Enter Your Name Ahmed
Please Enter Your Age = 22

answer =

Your name is Ahmed and you are 22 Years Old

>>

١- لإظهار الكتابة بهذا الشكل، لابد من وضعها في صورة منتج
```

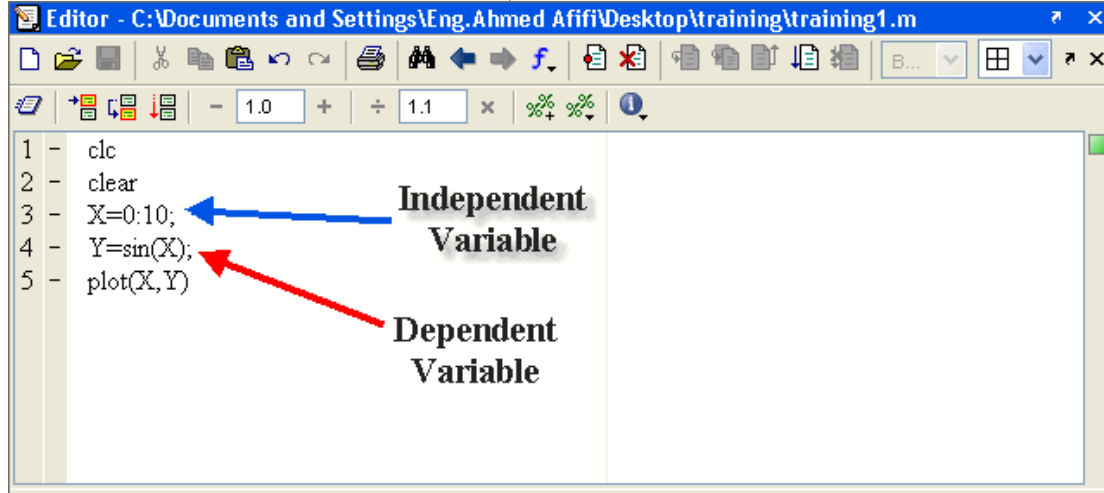
والآن نحن على إتم إستعداد للبدء في الجزء المتعلق بالرسم الثاني والثلاثي الأبعاد

الرسم ثنائي الأبعاد **2D Plotting**
المقصود بالرسم ثنائي الأبعاد هو ان تكون العلاقة التي تحكم عملية الرسم تكون بين متغيرين فقط أحدهما يسمى **independent** والآخر يسمى **dependent**

فما المقصود بـ **independent Variable** أي المتغير المستقل أي أن قيمه لا تحكمها علاقة بينما **dependent Variable** أي المتغير المعتمد حيث يعتمد قيمه على قيم المتغير المستقل
الآن وبعد شرح هذين العنصرين الهامين، سنقوم بشرح الأمر **plot** حيث يأخذ الصورة التالية

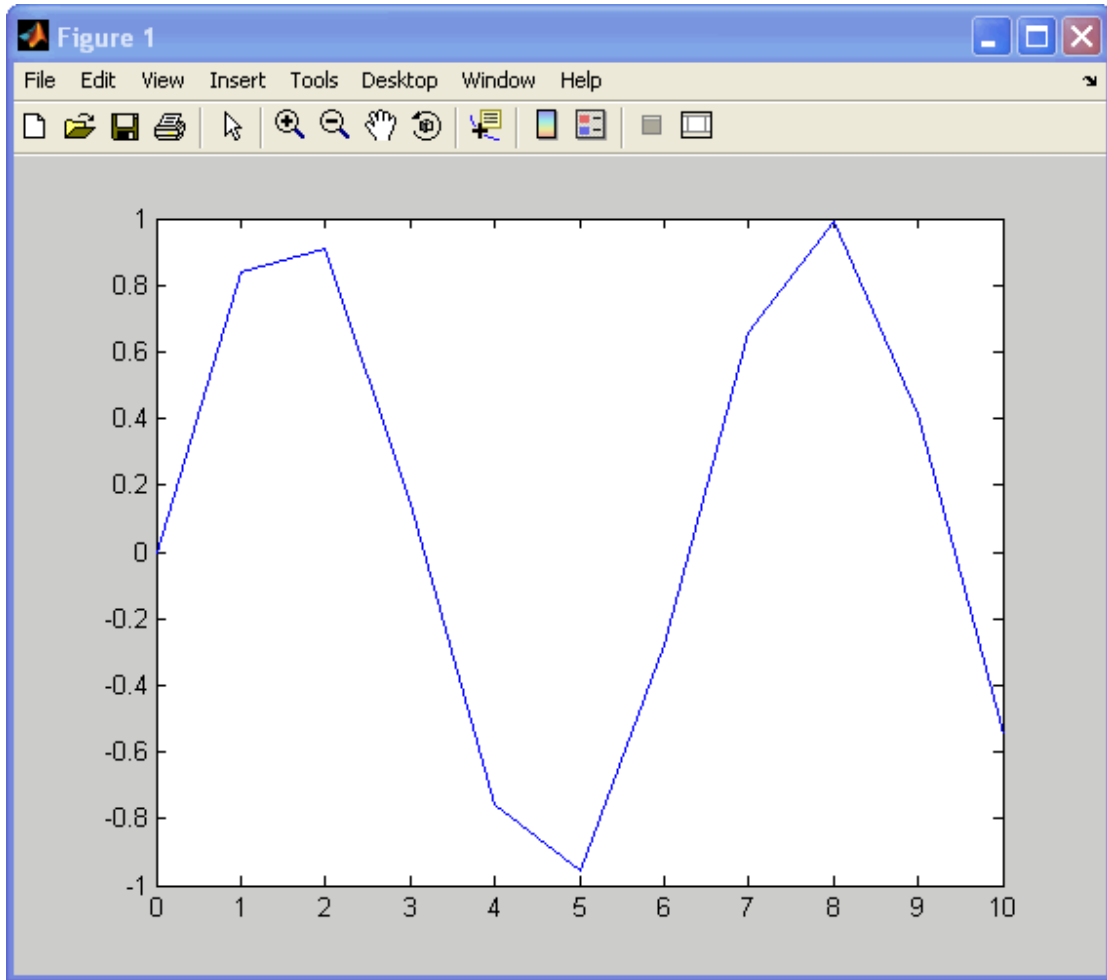
plot(independent variable, dependent variable)

وهذا مثال بسيط لكيفية رسم **sine Wave**



```
1 - clc
2 - clear
3 - X=0:10;
4 - Y=sin(X);
5 - plot(X,Y)
```

وتلاحظ أننا قد اخترنا ١٠ نقاط فقط لرسم **Sine Wave** , وهذا عدد قليل لرسم **Sine Wave** وتلاحظ ظهور الرسمة بالشكل التالي



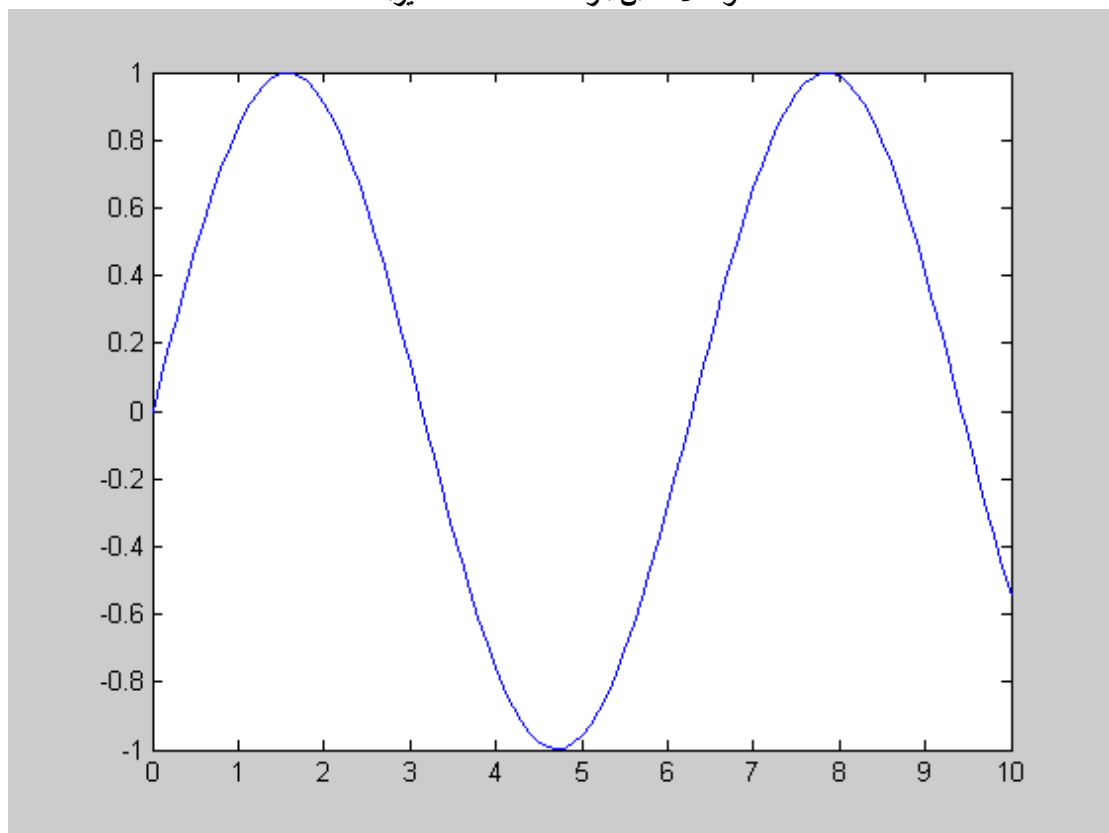
وحلاً لهذه المشكلة، لابد من زيادة عدد النقاط داخل المتجه، كما في الشكل التالي

```
Editor - C:\Documents and Settings\Eng.Ahmed Afifi\Desktop\training\training1.m*
File Edit Text Go Cell Tools Debug Desktop Window Help
+ - 1.0 + ÷ 1.1 x % %
1 - clc
2 - clear
3 - X=0:0.1:10;
4 - Y=sin(X);
5 - plot(X,Y)
6
```

يمكن وضع المتجه في هذه الصورة
minimum number : step : maximum number

script Ln 6 Col 1 OVR

وستلاحظ أن الرسمة قد تحسنت كثيراً



وسنأخذ في الدرس القادم كيفية وضع مسميات حول محور السينات **X-Axis** ومحور الصادات **Y-Axis** وعنوان للرسم، ووضع شبكة على الرسم، وكيفية تغيير لون الرسم، وكذلك وضع أكثر من رسم فوق بعضهما، وكيفية عمل كل رسم في نافذة منفصلة، وكيفية إنشاء عدة رسومات منفصلة في نافذة واحدة بإذن الله.

إضافة خصائص إلى الرسومات داخل الماتلاب

في بعض الأحيان يكون من الضروري جداً تغيير بعض الخواص لدى الرسومات التي نحصل عليها مثل تغيير الألوان، وتغيير الرسمة من خطوط متصلة إلى نجوم ونفاك وغيرها، وهذه هي مجموعة الخصائص التي تتم من خلال الماتلاب

b	blue	.	point	-	solid
g	green	o	circle	:	dotted
r	red	x	x-mark	-.	dashdot
c	cyan	+	plus	--	dashed
m	magenta	*	star	(none)	no line
y	yellow	s	square		
k	black	d	diamond		
		v	triangle (down)		
		^	triangle (up)		
		<	triangle (left)		
		>	triangle (right)		
		p	pentagram		
		h	hexagram		

فكيف يتم وضع تلك الخصائص داخل الماتلاب، تكون هذه الخصائص متضمنة في الأمر `plot` حيث تأخذ الصورة التالية

`plot(independent Variable, Dependent Variable, ' the property ')`

كما ترى فإن أي خاصية يتم وضعها
بعد **Dependent Variable**
ولكن يجب وضع الخاصية بين
فاصلتين ' الخاصية '

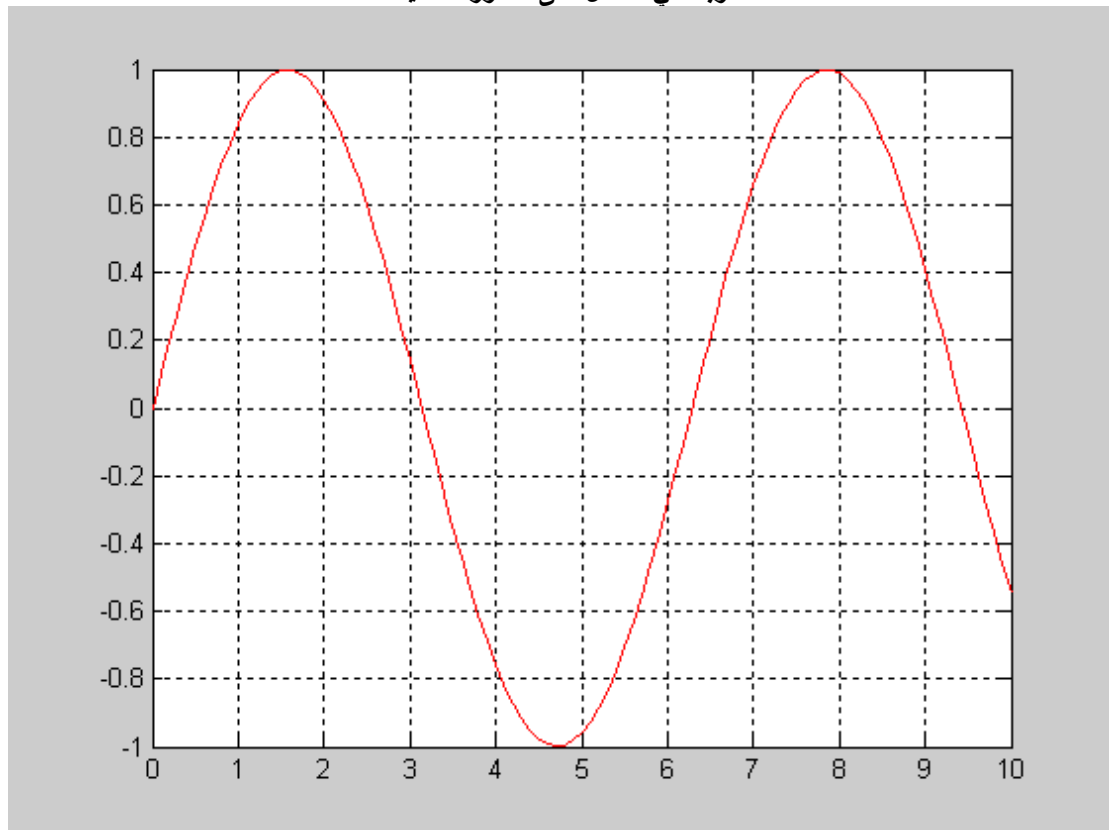
إعتماداً على المثال السابق أخذه سنقوم بتعديل بعض الخصائص
سنقوم مثلاً بتغيير لون الخط إلى الأحمر

```
Editor - C:\Documents and Settings\Eng.Ahmed Afifi\Desktop\training\making_plot_g...
File Edit Text Go Cell Tools Debug Desktop Window Help
+ - 1.0 + ÷ 1.1 x % %
1 - clc
2 - clear
3 - X=0:0.1:10;
4 - Y=sin(X);
5 - plot(X,Y,'r');
6 - grid
```

كما ترى فلقد تم وضع الخاصية

script Ln 6 Col 5 OVR

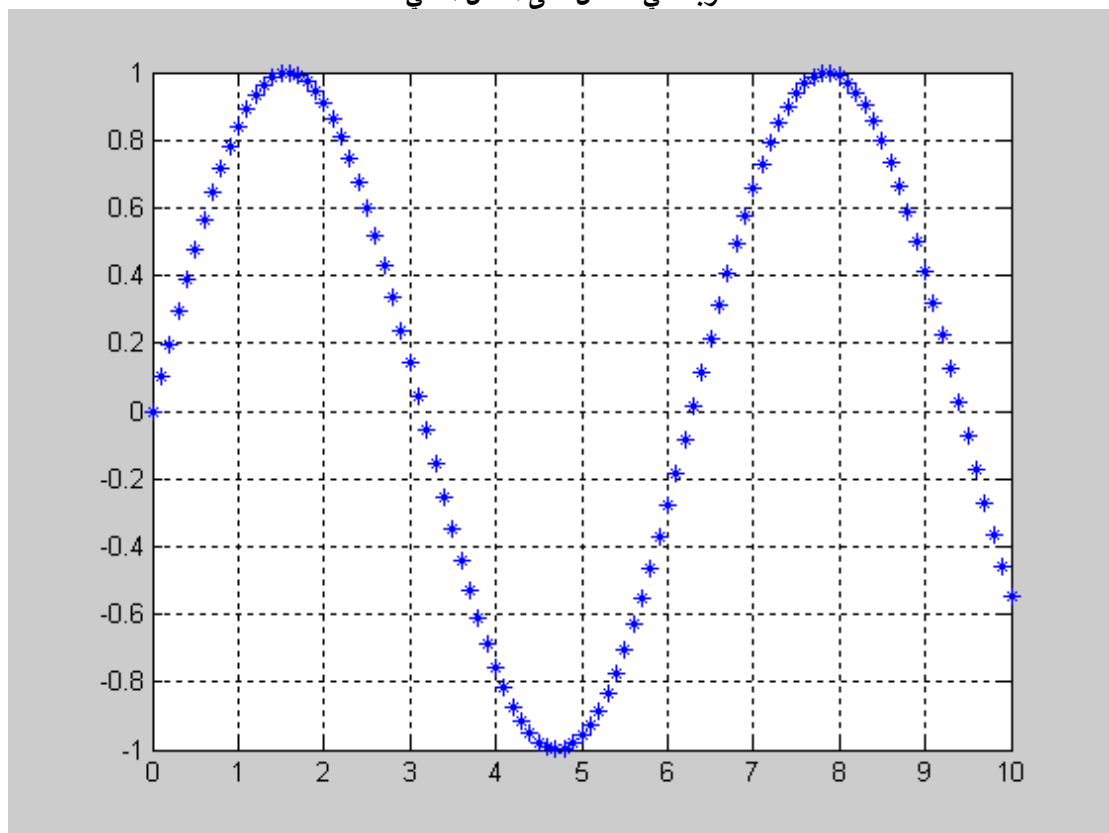
وبالتالي نحصل على الصورة التالية



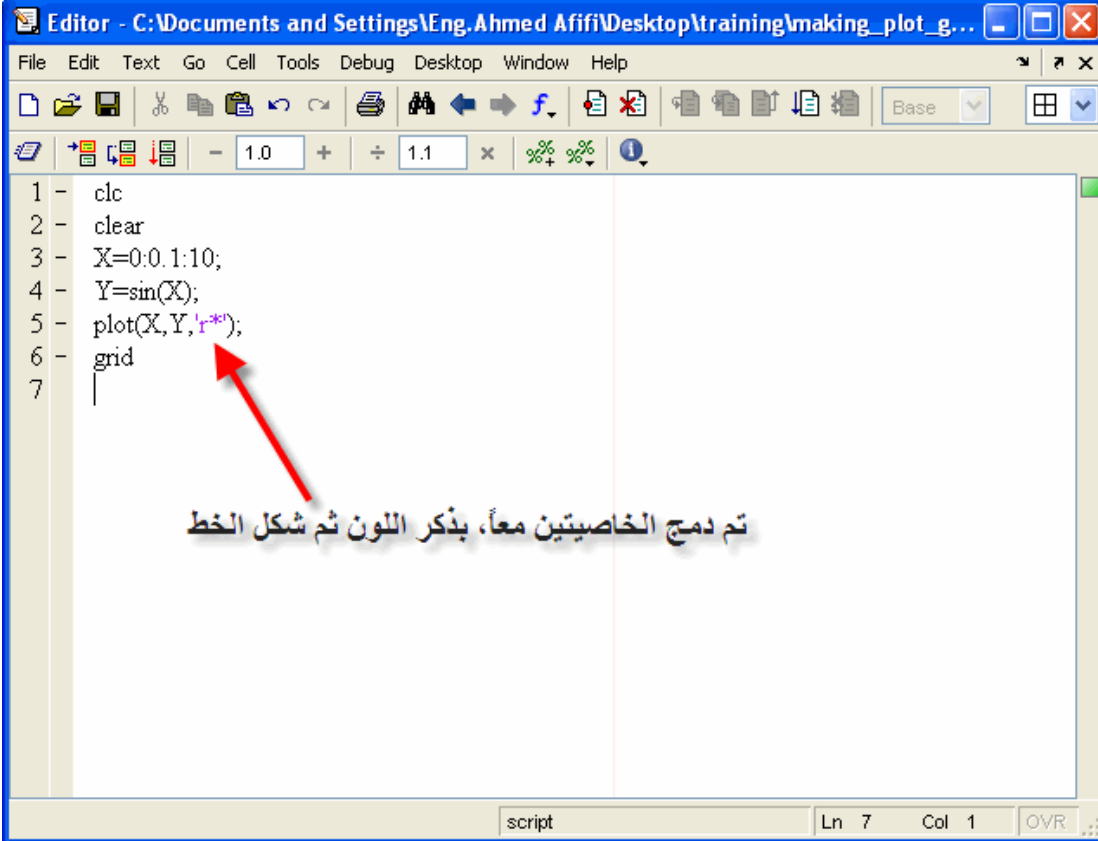
ولنقم الآن بإضافة خاصية جديدة بأن يكون الخط ليس خطأ متصل وإنما عبارة عن نجوم

```
Editor - C:\Documents and Settings\Eng.Ahmed Afifi\Desktop\training\making_plot_g...
File Edit Text Go Cell Tools Debug Desktop Window Help
+ - 1.0 + ÷ 1.1 x % %
1 - clc
2 - clear
3 - X=0:0.1:10;
4 - Y=sin(X);
5 - plot(X,Y,'*');
6 - grid
script Ln 5 Col 12 OVR
```

وبالتالي نحصل على الشكل التالي



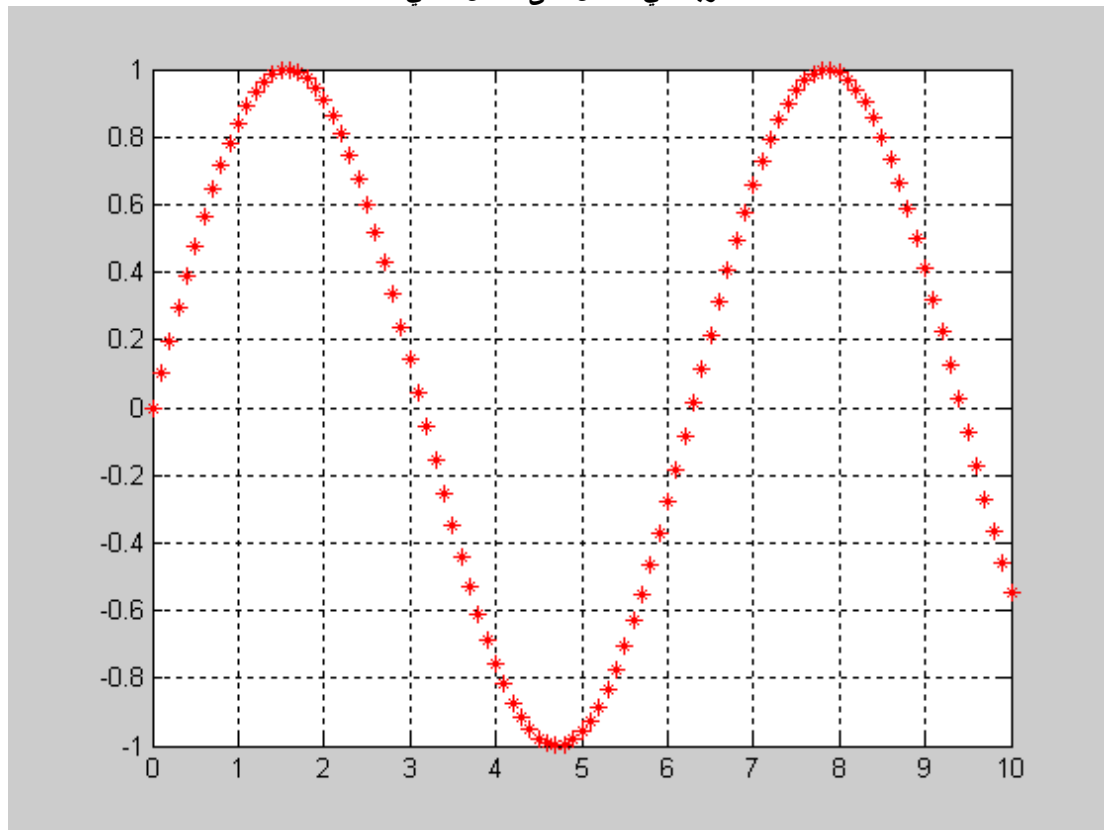
وإذا أردنا أن نحصل على نجوم حمراء (أي دمج الخاصيتين معاً)



```
1 - clc
2 - clear
3 - X=0:0.1:10;
4 - Y=sin(X);
5 - plot(X,Y,'r*');
6 - grid
7 -
```

تم دمج الخاصيتين معاً، بذكر اللون ثم شكل الخط

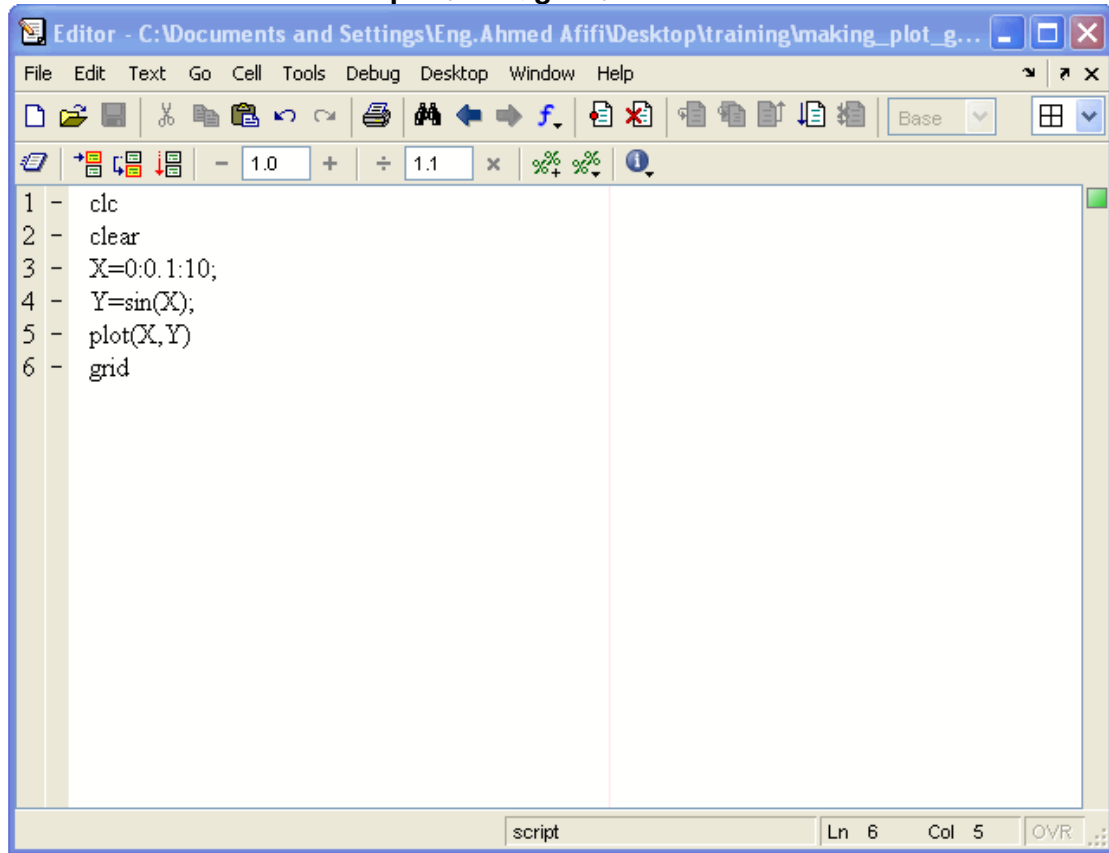
وبالتالي نحصل على الشكل التالي



وهنا نكون قد شرحنا خصائص الرسومات داخل الماتلاب

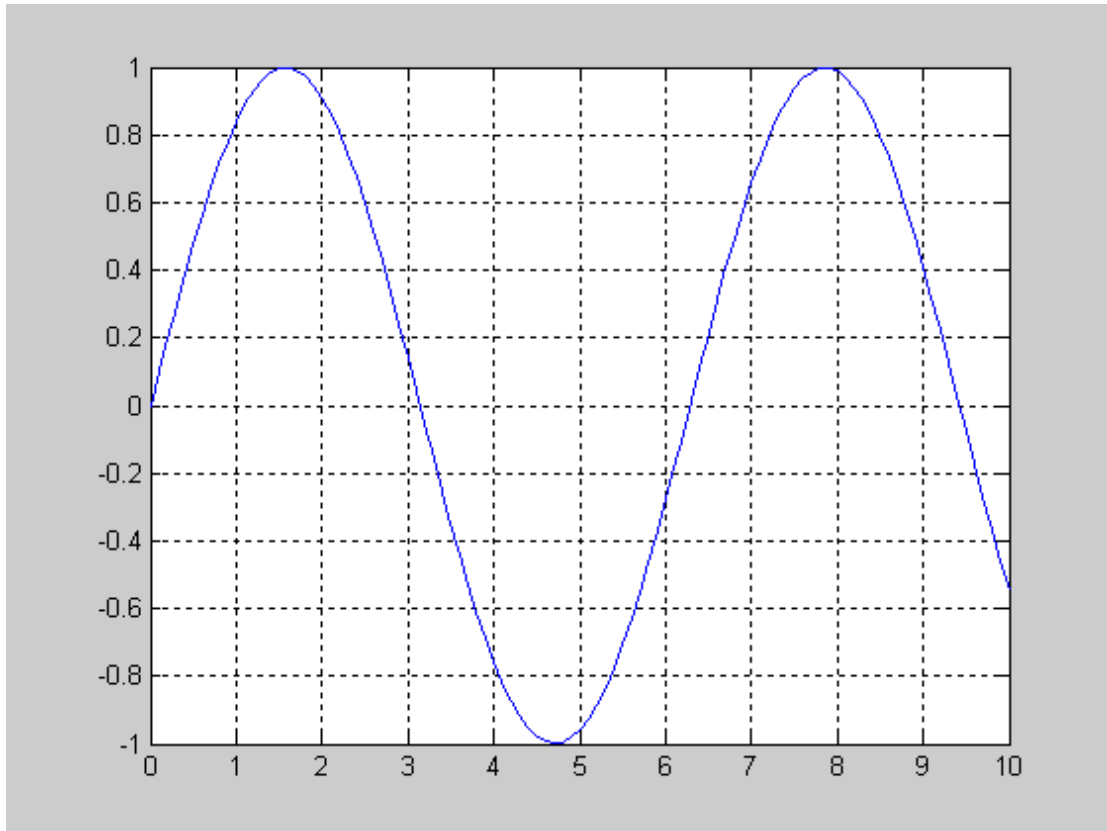
عملية وضع شبكة على الرسم

يقوم الماتلاب بوضع شبكة على الرسم، بحيث يكون من السهل تحديد القيم من على الرسم حيث تأخذ الأمر `grid` بعد الأمر `plot`



```
Editor - C:\Documents and Settings\Eng.Ahmed Afifi\Desktop\training\making_plot_g...
File Edit Text Go Cell Tools Debug Desktop Window Help
[Icons] Base [Grid Icon]
- 1.0 + ÷ 1.1 x % %
1 - clc
2 - clear
3 - X=0:0.1:10;
4 - Y=sin(X);
5 - plot(X,Y)
6 - grid
script Ln 6 Col 5 OVR
```

وسيكون شكل الرسم كالتالي



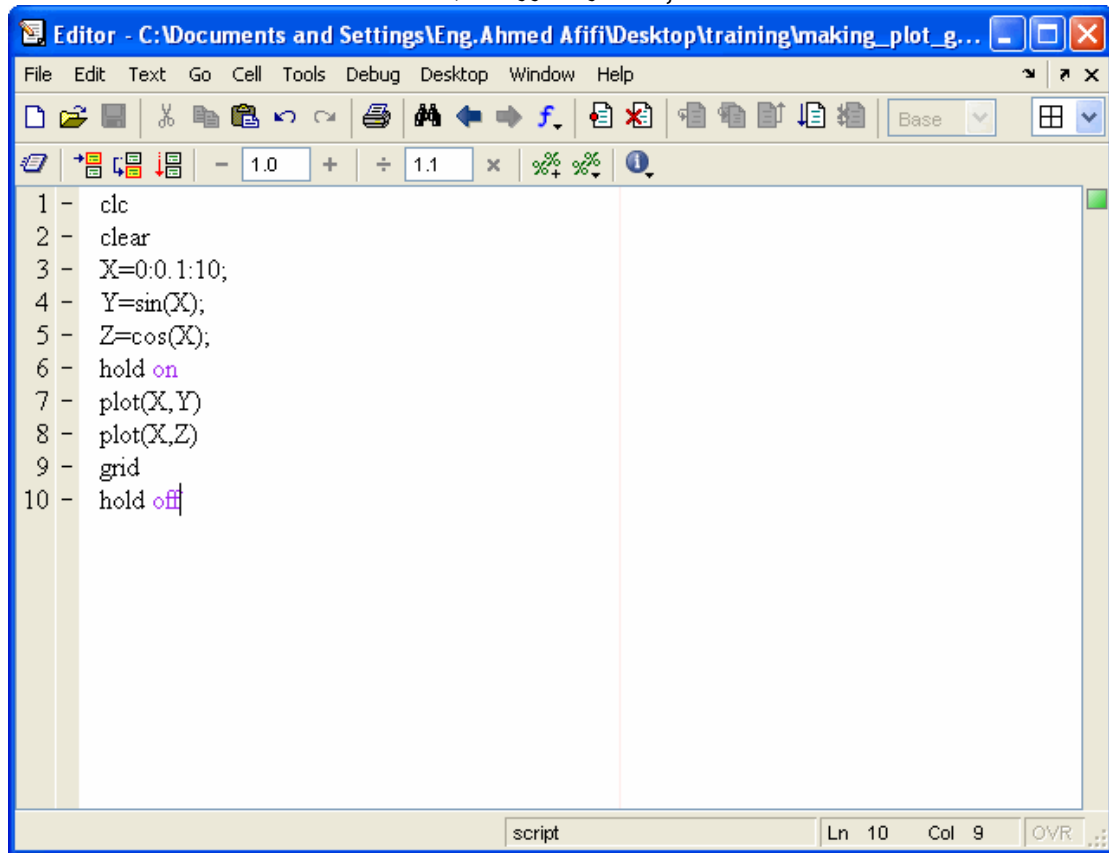
الآن سنقوم بعمل معادلة آخر بالإضافة إلى المعادلة المذكورة بحيث يكون لدينا رسمتان، بحيث تأخذ الشكل التالي

```

Editor - C:\Documents and Settings\Eng.Ahmed Afifi\Desktop\training\making_plot_g...
File Edit Text Go Cell Tools Debug Desktop Window Help
[Icons] Base [Grid]
- 1.0 + 1.1 x % %
1 - clc
2 - clear
3 - X=0:0.1:10;
4 - Y=sin(X);
5 - Z=cos(X);
6 - plot(X,Y)
7 - plot(X,Z)
8 - grid
script Ln 7 Col 10 OVR

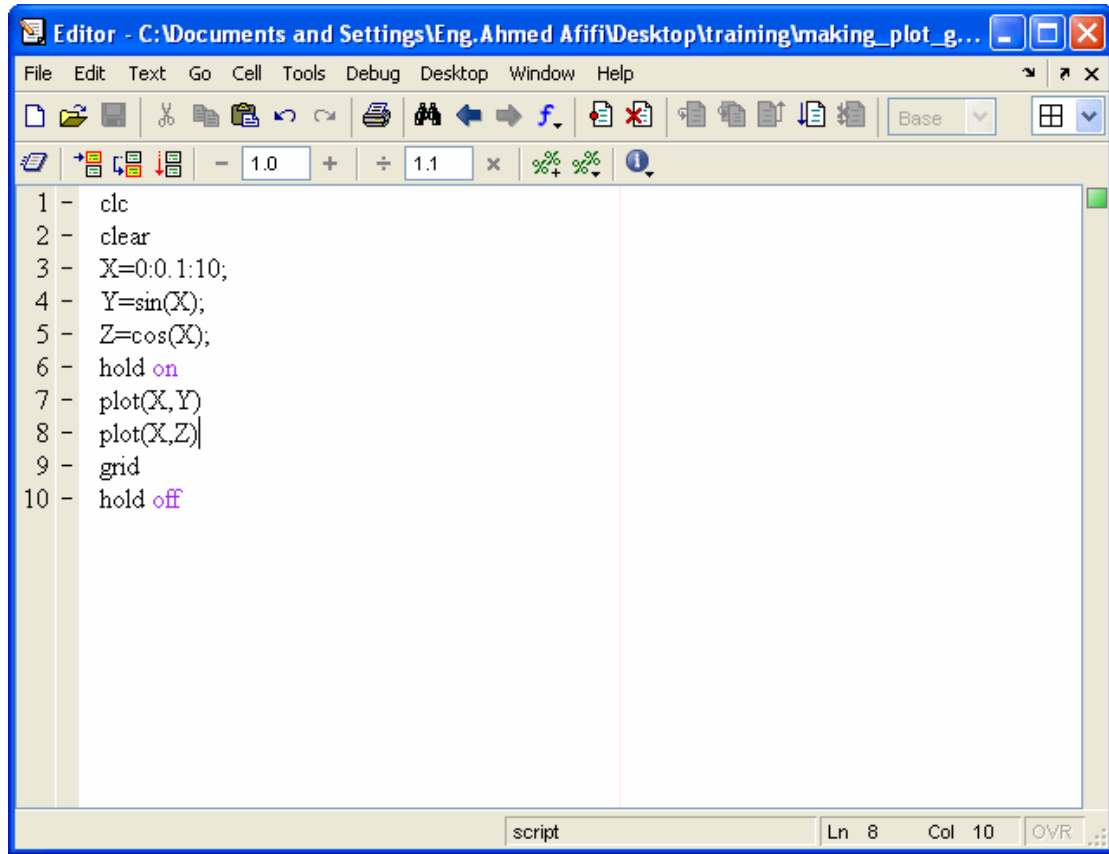
```

ولكن عند تشغيل البرنامج، سيقوم الماتلاب بإظهار الرسمة الأخيرة فقط، فكيف يتم إظهار الرسمتين، يتم ذلك باستخدام الأمر **Hold on** قبل الأمر **plot** لكي يتم وضع الرسمتين في نافذة واحدة، وفي نهاية الأمر يتم وضع الأمر **hold off**، أنظر الصورة التالية



```
1 - clc
2 - clear
3 - X=0:0.1:10;
4 - Y=sin(X);
5 - Z=cos(X);
6 - hold on
7 - plot(X,Y)
8 - plot(X,Z)
9 - grid
10 - hold off
```

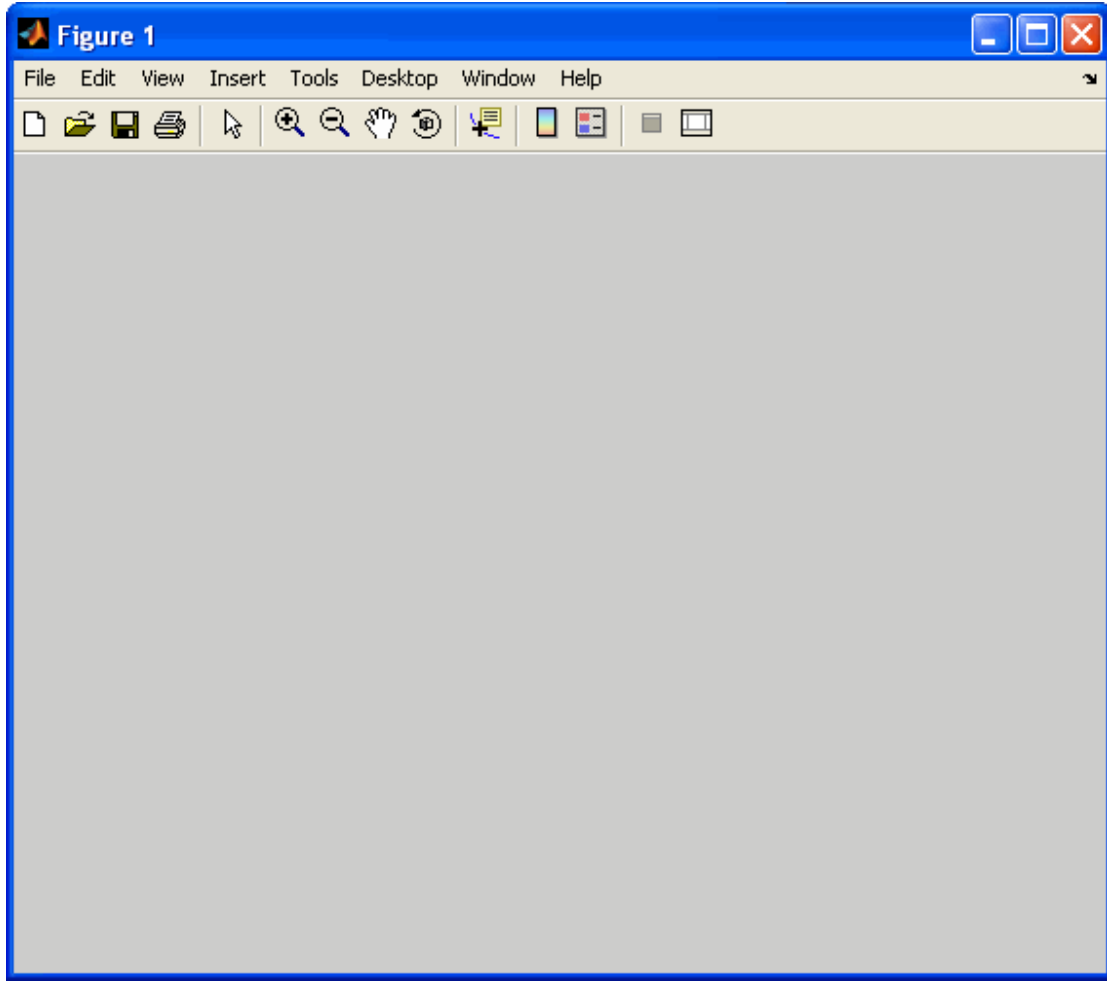
وبالتالي تكون الرسمتان كالتالي



```
1 - clc
2 - clear
3 - X=0:0.1:10;
4 - Y=sin(X);
5 - Z=cos(X);
6 - hold on
7 - plot(X,Y)
8 - plot(X,Z)
9 - grid
10 - hold off
```

علمية وضع الرسومات في نوافذ منفصلة

سنقوم الآن بدلاً من وضع الرسومات في نفس النافذة سنقوم بوضعها في نوافذ مختلفة وعلى نحتاج إلى الأمر **figure** والذي يقوم بفتح نافذة فارغة إذا تم وضعه منفصلاً، جرب ذلك في نافذة الأوامر ستلاحظ أن الماتلاب قام بإظهار نافذة رمادية اللون فارغة شاهد الصورة التالية



حيث وجود تلك النافذة يعني انه سيتم تنفيذ أمر الرسم **plot** الذي بعد أمر **figure** علماً أنه بعد كل أمر **figure** يتم وضع الخصائص التي تختص بهذه الرسمة مثل أمر **grid** الذي سبق شرحه. وهذا مثال بسيط على ذلك

Editor - C:\Documents and Settings\Eng.Ahmed Afifi\Desktop\training\making_plot_g...

```

1 - clc
2 - clear
3 - X=0:0.1:10;
4 - Y=sin(X);
5 - Z=cos(X);
6 - plot(X,Y,'r*');
7 - grid
8 - figure
9 - plot(X,Z,'mo');
10 - grid
11 |

```

لم يتم وضع أمر **figure** في أول مرة سيتم عمل **plotting** حيث أن الماتلاب في جميع الظروف سيقوم برسم أول أمر بشكل طبيعي

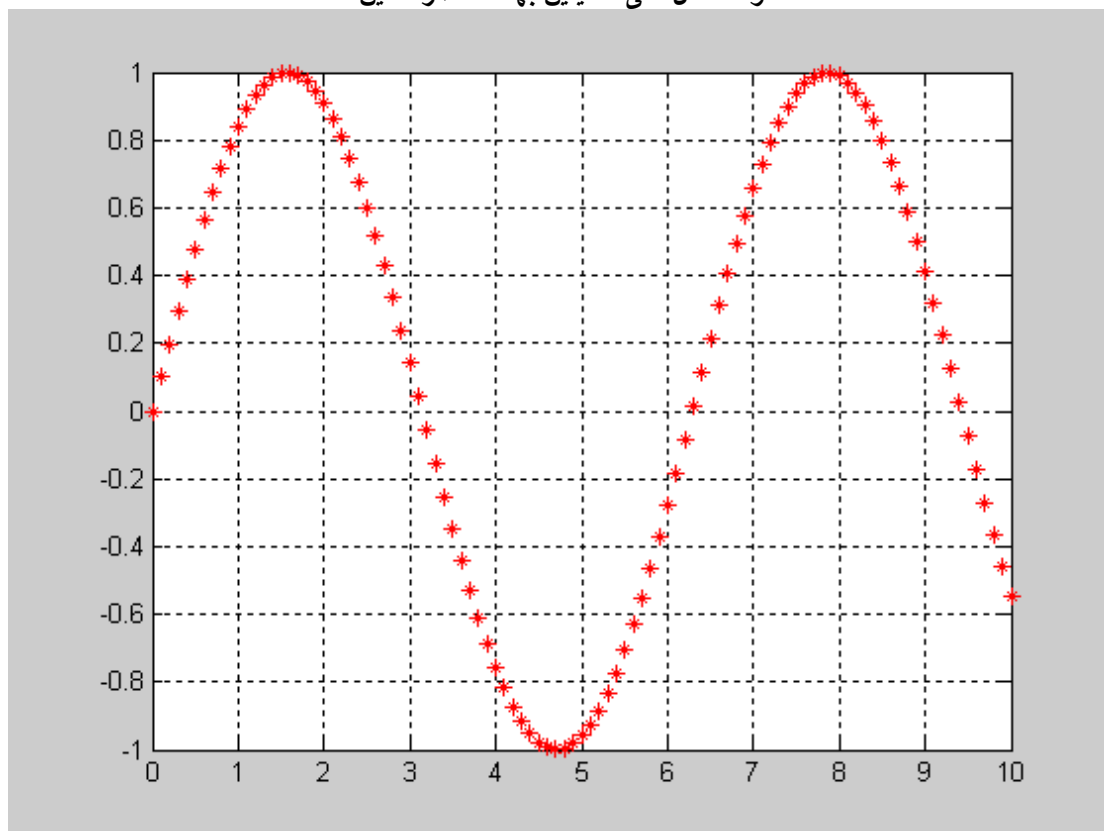
تم وضع الأمر **grid** حيث أنها خاصية لأمر **plot** كما سبق ذكره

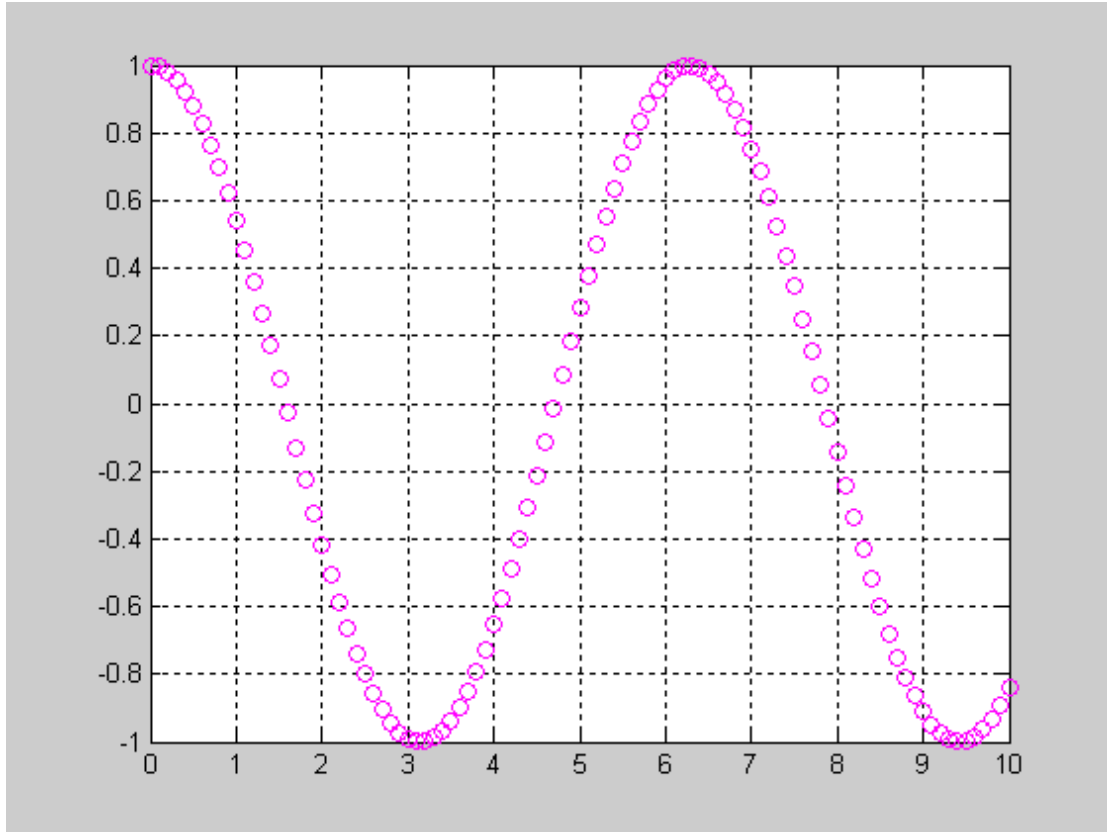
تم وضع الأمر **figure** وذلك لفتح نافذة مستقلة لتنفيذ الأمر **plot** الذي سوف يأتي مباشرة

هذا هو الأمر الثاني والذي سوف يتم رسمه في نافذة منفصلة

script Ln 11 Col 1 OVR

وستحصل على نافذتين بهما كلتا الرسمتين





والآن قم بتشغيل البرنامج مرة أخرى، ستلاحظ أن عدد النوافذ قد زاد نافذة واحدة، فكيف حدث هذا؟ يقوم الماتلاب برسم أول دالة على النافذة الأخيرة التي تم رسم الدالة الثانية بها، ثم يقوم برسم الدالة الثانية في نافذة جديدة بسبب وجود الأمر **figure** ولحل هذه المشكلة قم باستخدام الأمر **close all** بعد الأمر **clear** بحيث يتم إغلاق أي نوافذ كانت مفتوحة قبل ذلك عند تشغيل البرنامج كل مرة وبالتالي سيكون هنالك ثلاثة أوامر لا بد من استخدامها في كل مرة يتم عمل أي برنامج وهم

```
clc
clear
close all
```

وهذا هو المثال الذي تم عمله منذ قليل بعد التعديل

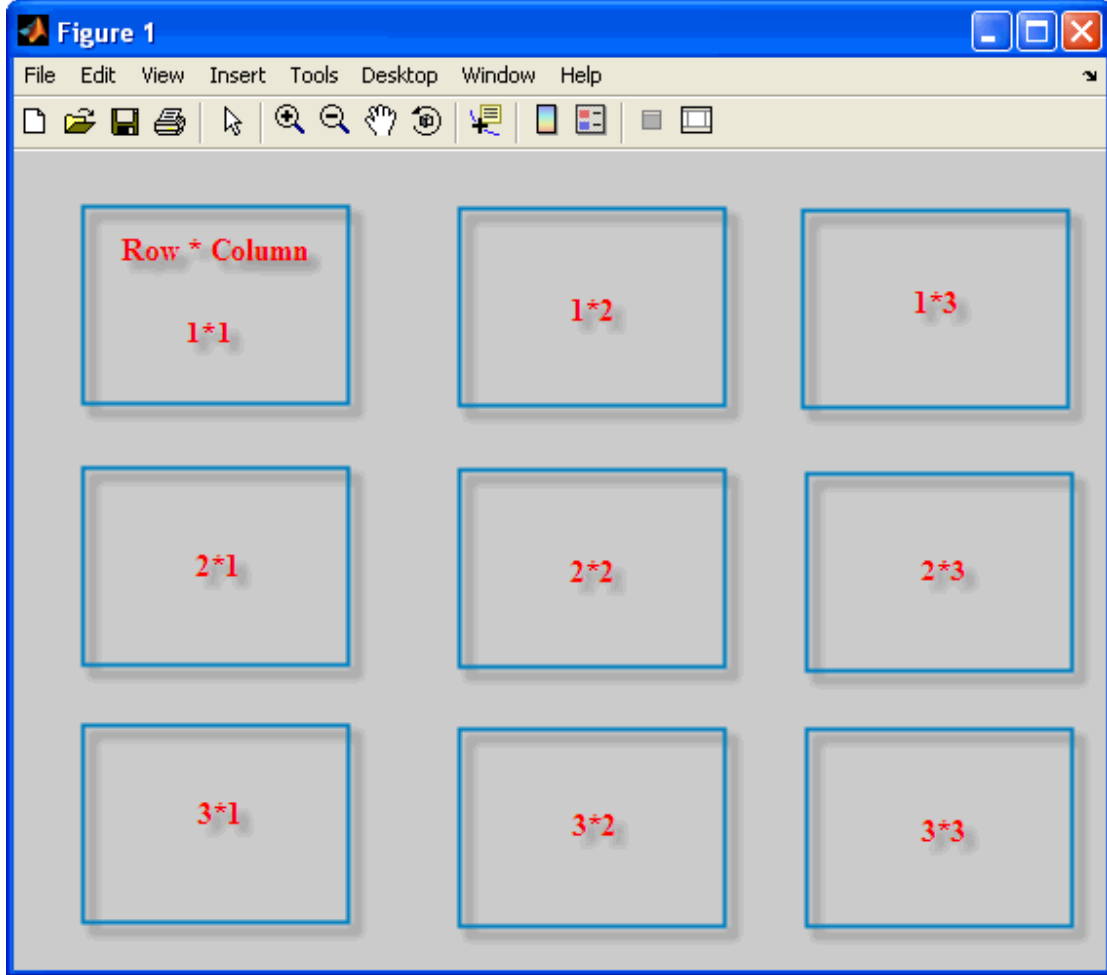
The image shows a screenshot of a MATLAB script editor window. The window title is "Editor - C:\Documents and Settings\Eng.Ahmed Afifi\Desktop\training\making_plot_g...". The menu bar includes File, Edit, Text, Go, Cell, Tools, Debug, Desktop, Window, and Help. The toolbar contains various icons for file operations and editing. Below the toolbar is a numeric keypad with buttons for minus, 1.0, plus, divide, 1.1, multiply, percent, and a help icon. The main editing area contains the following MATLAB code:

```
1 - clc
2 - clear
3 - close all
4 - X=0:0.1:10;
5 - Y=sin(X);
6 - Z=cos(X);
7 - plot(X,Y,'r*');
8 - grid
9 - figure
10 - plot(X,Z,'mo');
11 - grid
12
```

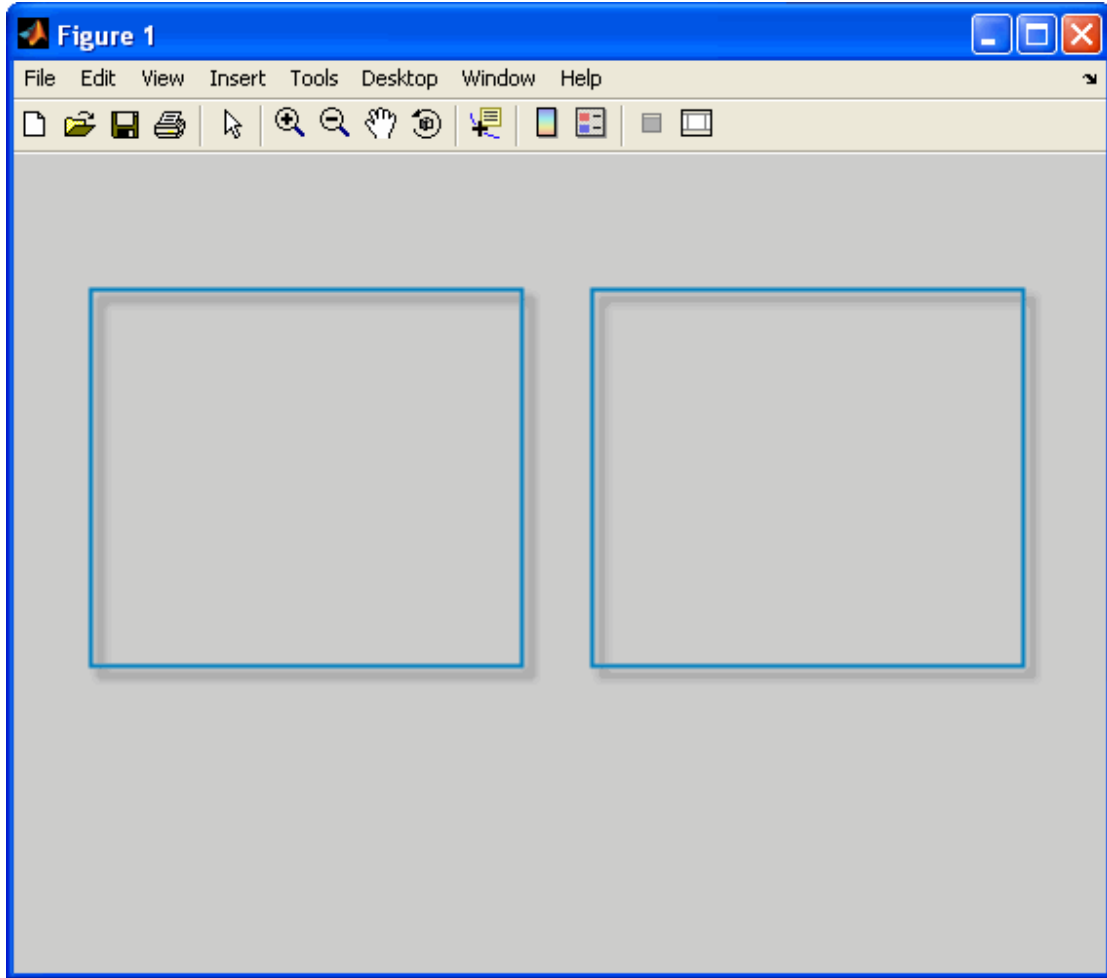
The status bar at the bottom indicates the current file is "script", the cursor is at line 12, column 1, and the window is in "OVR" (Overwrite) mode.

إنشاء رسومات منفصلة في نافذة واحدة

أخذنا أننا بإمكاننا أن نقوم بعمل أكثر من رسمة على نفس النافذة، ولكن هل تتخيل أن نقوم بوضع عدة رسومات منفصلة في نافذة واحدة، في الحقيقة يمكن ذلك باستخدام الأمر `subplot` قبل كل أمر `plot` يعمل الأمر `subplot` من خلال تحديد عدد الرسومات التي ستقوم بإظهارها، حيث يقوم الأمر `subplot` على وضع الصور وكأنها مصفوفة أو متجه، ويجب عند استخدام الأمر معرفة عدد الرسومات التي ستظهرها وكيفية وضعها، ويفضل استخدام الشكل التالية لتحديد الأماكن التي ستقوم بوضع الرسومات بها



سنأخذ مثلاً، لنقول أن لدينا معادلات يجب رسمهما، وسنقوم بوضعهما بجوار بعضهما كما في الشكل التالي



وبالتالي الرسمتان سيكون وكأتهما متجه عدد صفوفها ١ وعدد الأعمدة ٢ ، والرسم الأولى تأخذ الخانة الأولى ، والرسم الثانية ستأخذ الخانة الثانية

هذا ما يجب تحديده بالتفصيل عند استخدام الأمر **subplot** ثم نستخدم الصورة العامة لأمر **subplot** والتي تكون كالتالي

`plot(number of rows, number of column, the number of the matrix which occupy the figure)`

ولرسم الشكل الأول لابد من كتابة الأمر في الصورة التالية

`subplot(1,2,1)`

رغم الخانة التي تشغلها
عدد الأعمدة
عدد الصفوف

ولرسم الشكل الثاني لابد من كتابة الأمر في الصورة التالية

`subplot(1,2,2)`

رغم الخانة التي يشغلها الشكل
عدد الأعمدة
عدد الصفوف

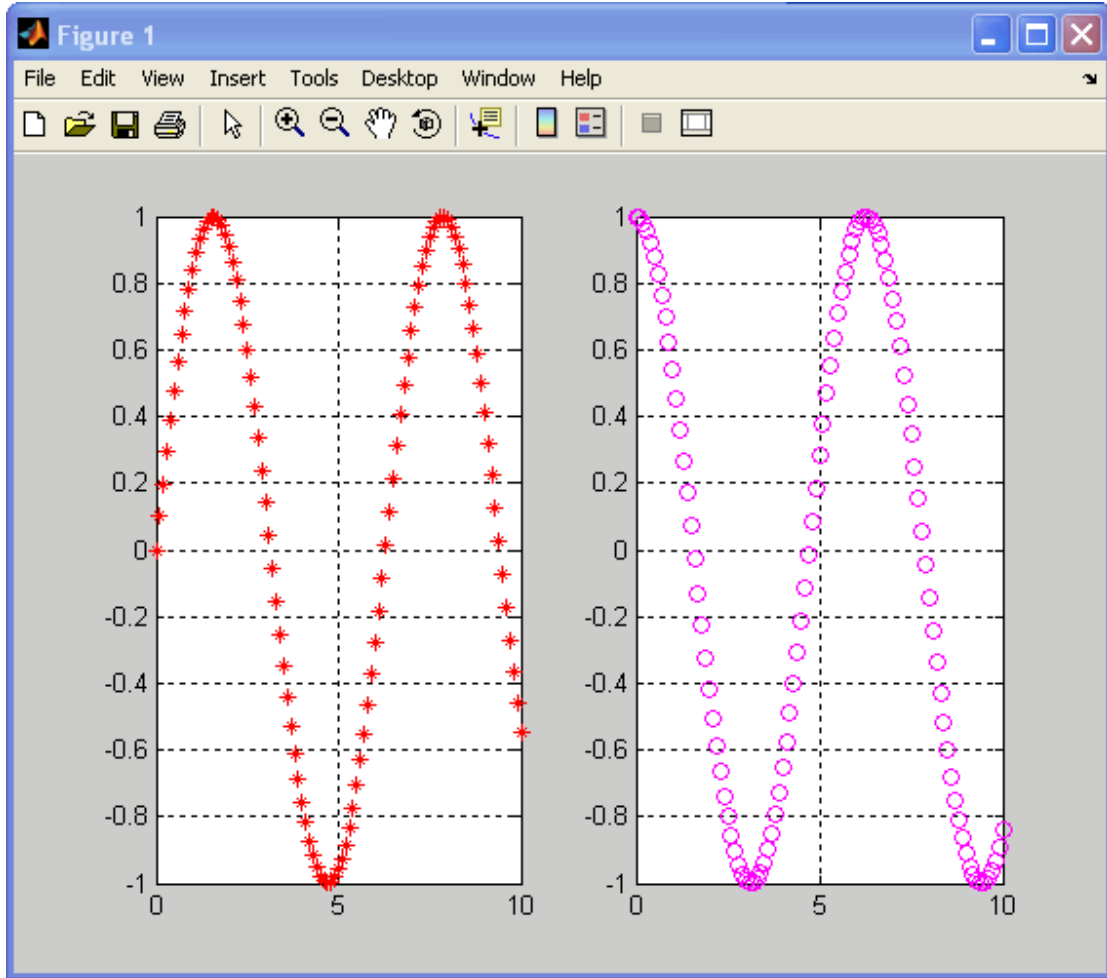
والآن سنقوم بوضع البرنامج كاملاً ليكون المعنى قد وضح تماماً

```
1 - clc
2 - clear
3 - close all
4 - X=0:0.1:10;
5 - Y=sin(X);
6 - Z=cos(X);
7 - subplot(1,2,1)
8 - plot(X,Y,'r*');
9 - grid
10 - subplot(1,2,2)
11 - plot(X,Z,'mo');
12 - grid
13
```

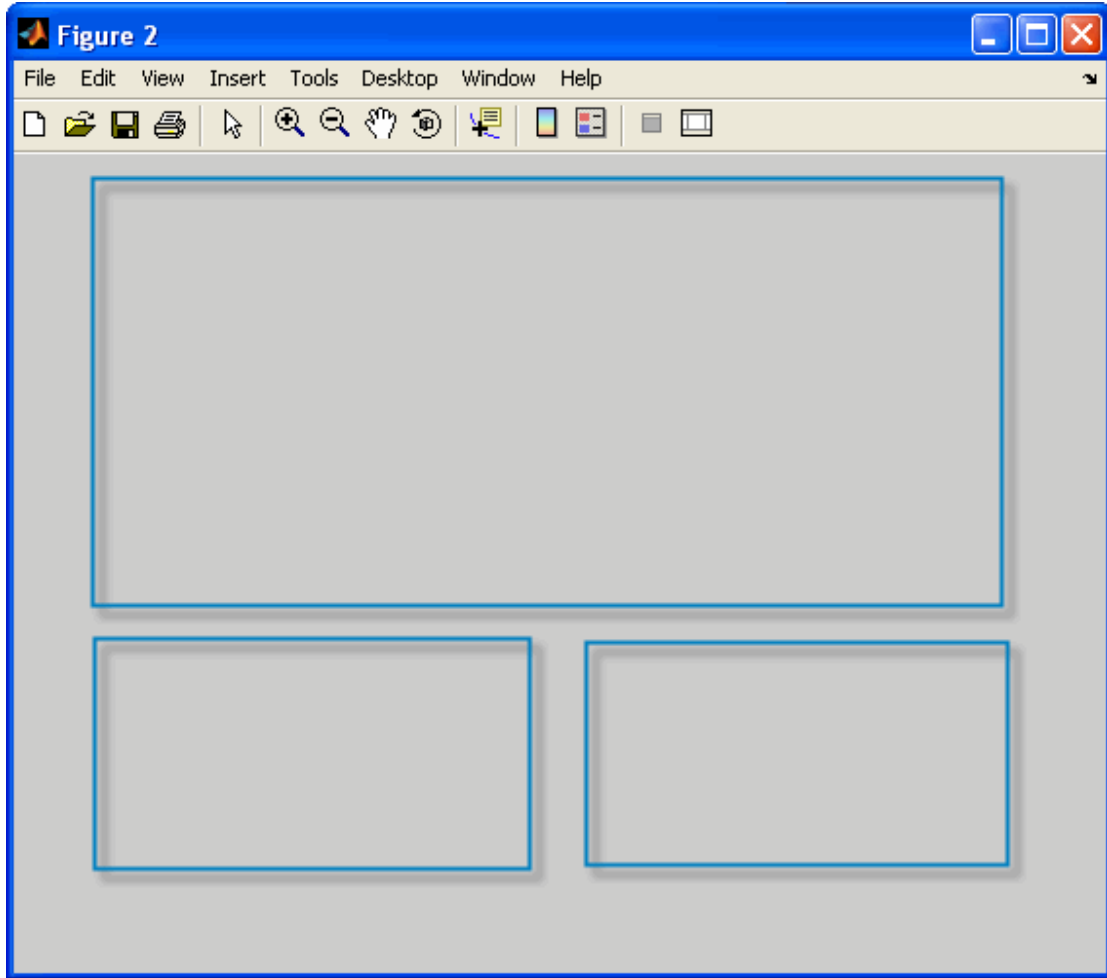
لا بد من استخدام الأمر **plot** بعد الأمر **subplot**

script Ln 13 Col 1 OVR

وستكون الرسمة النهائية كما في الشكل التالي



ملاحظة إذا كانت الرسمة تشغل أكثر من خانة يتم استخدام الأقواس المربعة، وتأخذ الشكل التالي
 [أرقام جميع الخانات التي تشغلها الرسمة]
 وسنقوم بإعطاء مثال
 نريد أن يكون الشكل الخارج على شكل الصورة التالية



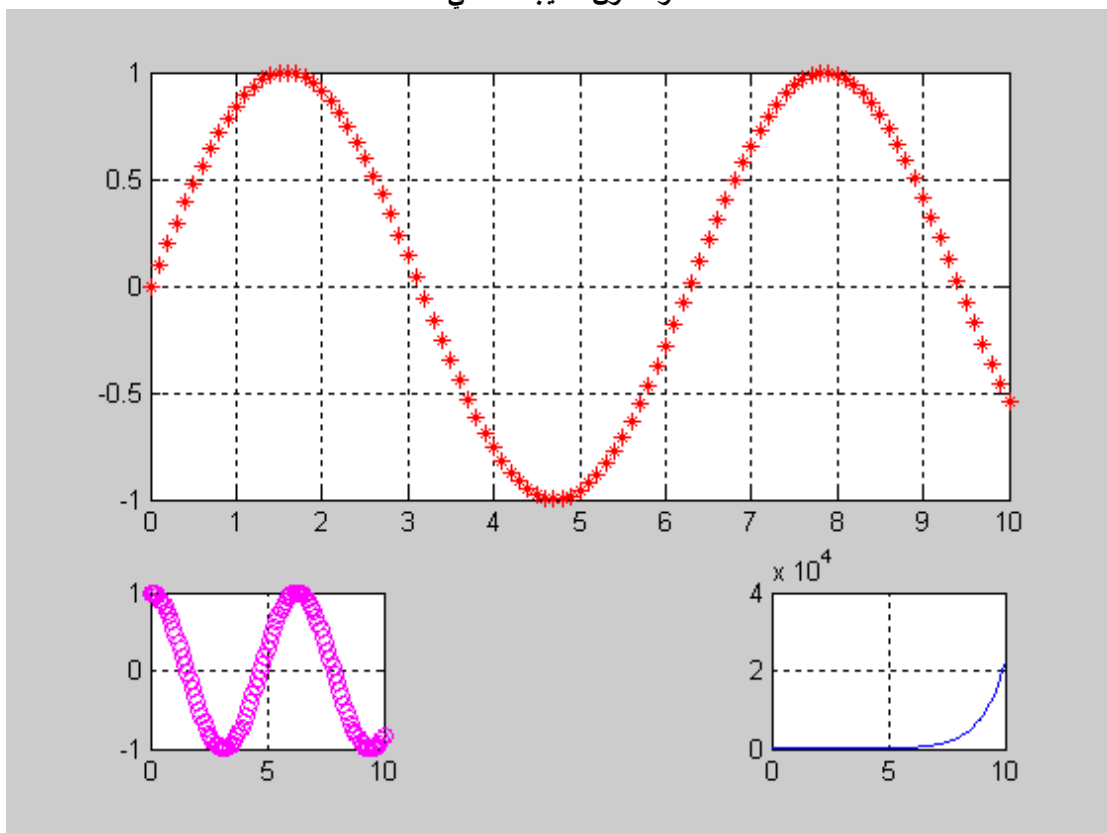
فإن عدد الصفوف ٣ وعدد الأعمدة ٣ وأرقام الخانات التي تشغلها الرسمة الأولى ١ و ٢ و ٣ و ٤ و ٥ و ٦ على التوالي،
وأرقام الخانات التي تشغل الرسمة الثانية ٧ وأرقام الخانات التي تشغل الرسمة الثالثة هي ٩
والبرنامج يكون بالشكل التالي

```

Editor - C:\Documents and Settings\Eng.Ahmed Afifi\Desktop\training\making_plot_g...
File Edit Text Go Cell Tools Debug Desktop Window Help
Base
- 1.0 + ÷ 1.1 x %>% %>%
1 - clc
2 - clear
3 - close all
4 - X=0:0.1:10;
5 - Y=sin(X);
6 - Z=cos(X);
7 - V=exp(X);
8 - subplot(3,3,[1 2 3 4 5 6])
9 - plot(X,Y,'r*');
10 - grid
11 - subplot(3,3,7)
12 - plot(X,Z,'mo');
13 - grid
14 - subplot(3,3,9)
15 - plot(X,V);
16 - grid
17 |
script Ln 17 Col 1 OVR

```

وستكون النتيجة كالتالي



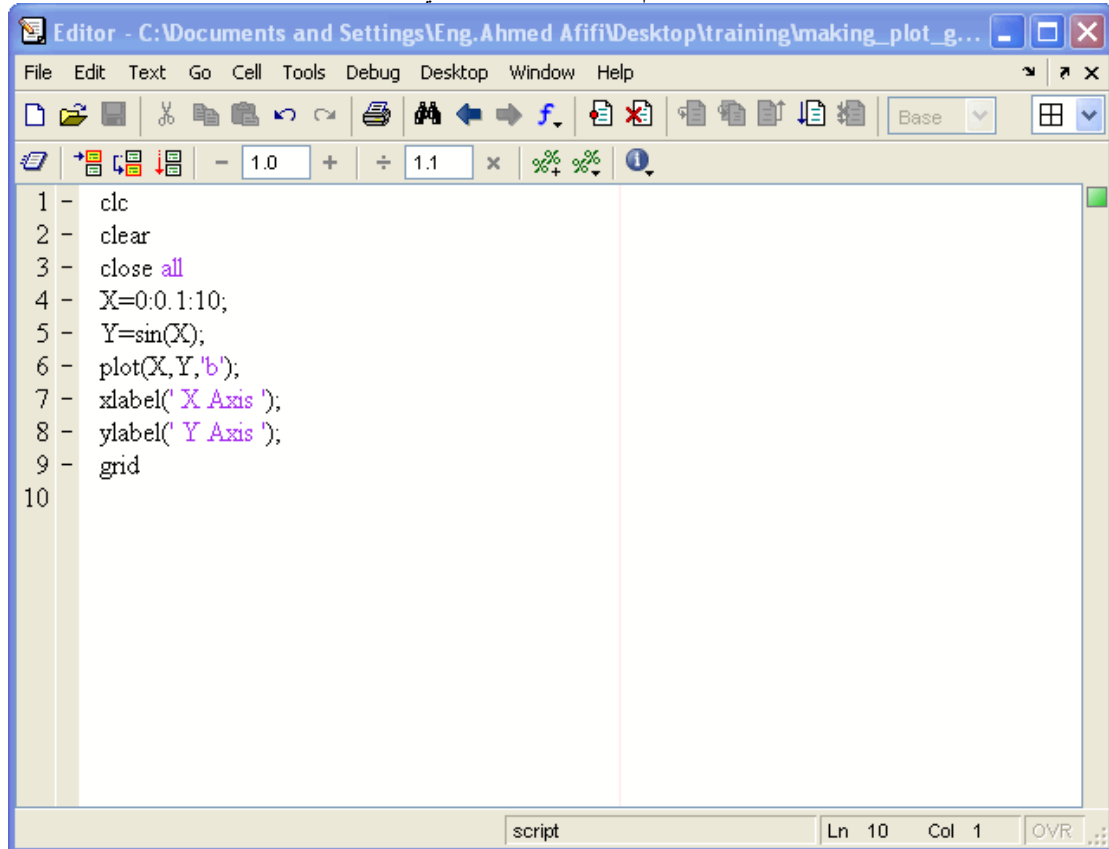
تسمية المحاور

سنقوم الآن بتنفيذ الجزء قبل الأخير من الدورة وهو تسمية المحاور، فمثلاً إذا أردنا أن نقوم بتسمية محور السينات-X Axis نقوم باستخدام الأمر xlabel وإذا أردنا أن نقوم بتسمية محور الصادات نقوم باستخدام الأمر ylabel حيث يأخذ كلا الأمرين صورة واحدة وهي كالتالي

```
xlabel( ' The name of the axis' )
```

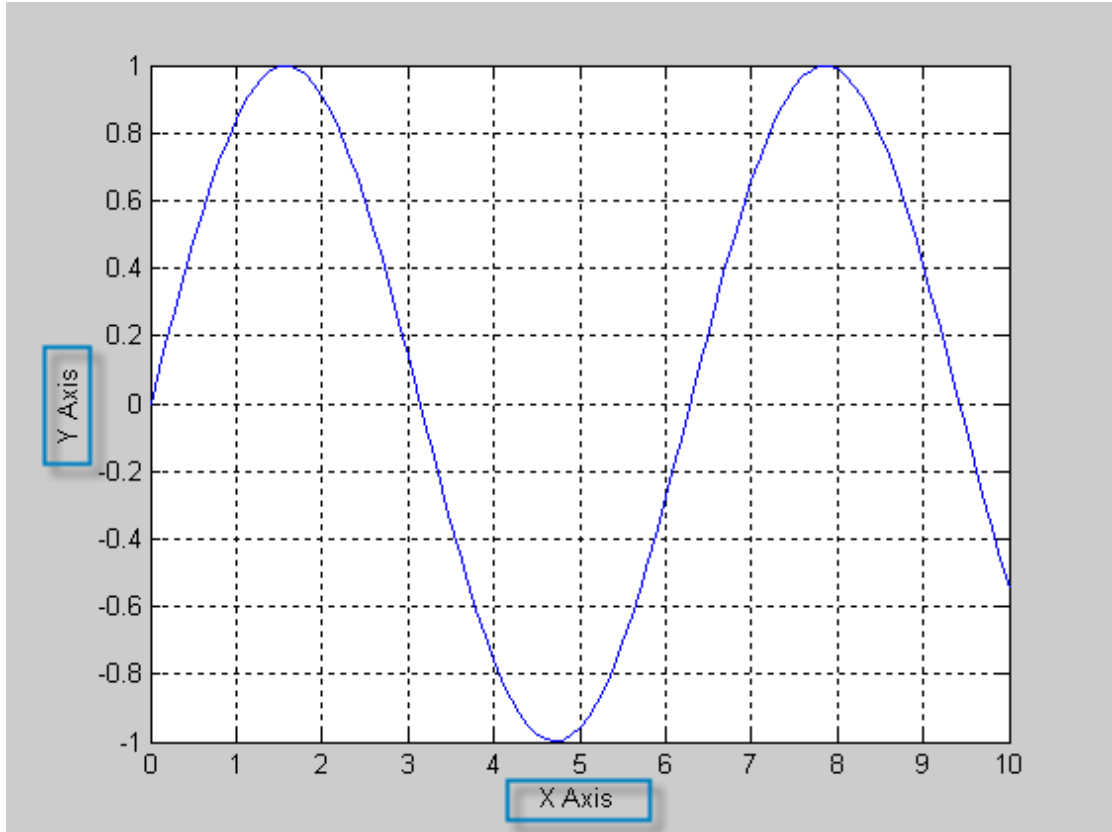
كما ترى لابد من أن يكون اسم المحور بين قوسين كما هو واضح بالشكل

نفس الشيء يتم تطبيقه على محور الصادات ylabel ولنقوم الآن بعمل مثال تطبيقي



```
Editor - C:\Documents and Settings\Eng.Ahmed Afifi\Desktop\training\making_plot_g...
File Edit Text Go Cell Tools Debug Desktop Window Help
[Icons] Base [Grid]
- 1.0 + 1.1 x % % ?
1 - clc
2 - clear
3 - close all
4 - X=0:0.1:10;
5 - Y=sin(X);
6 - plot(X,Y,'b');
7 - xlabel(' X Axis ');
8 - ylabel(' Y Axis ');
9 - grid
10
```

وبالتالي نحصل على الصورة التالية



وضع عنوان في أعلى الرسمة

يمكن وضع عنوان أعلى كل رسمة وذلك من خلال الأمر **title** حيث يكون هذا الأمر بالشكل التالي

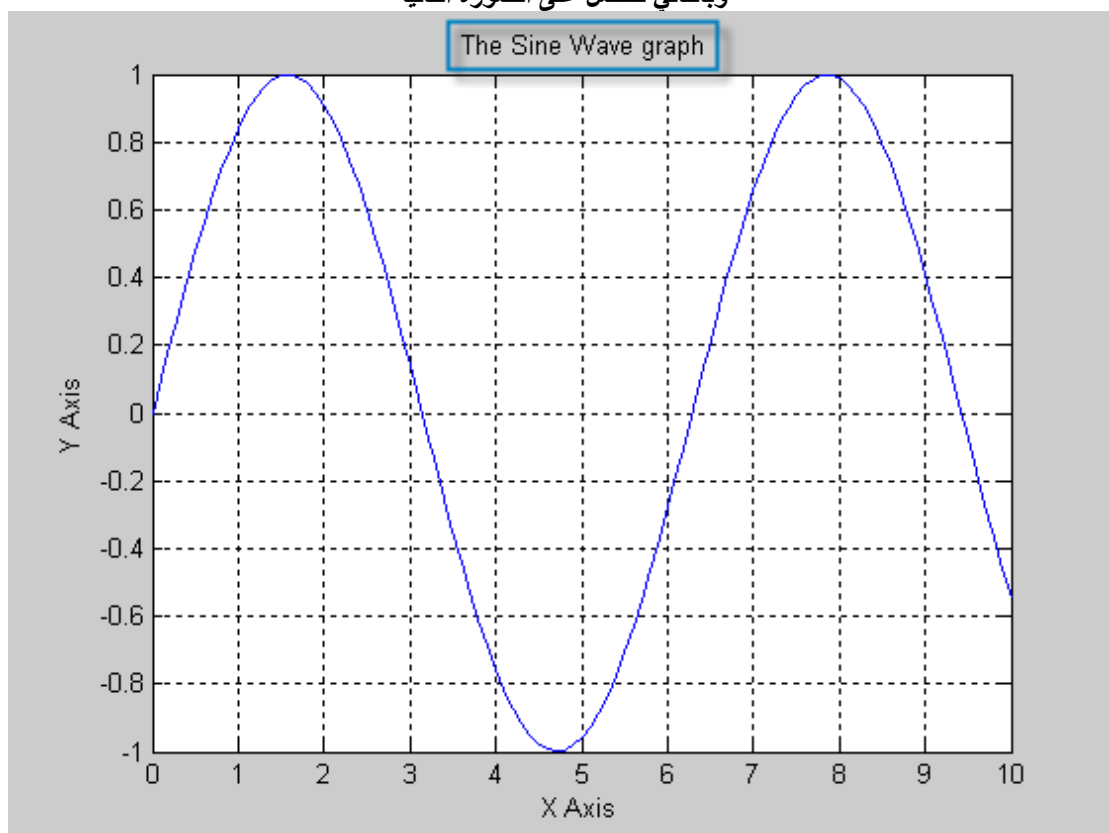
Title ('The title of the graph')

لا بد من وضع العنوان بين فاصلتين كما هو موضح بالرسم

وبالرجوع إلى المثال السابق ووضع التعديلات عليه كما هو موضح

```
Editor - C:\Documents and Settings\Eng.Ahmed Afifi\Desktop\training\making_plot_g...
File Edit Text Go Cell Tools Debug Desktop Window Help
Base
- 1.0 + ÷ 1.1 x % %
1 - clc
2 - clear
3 - close all
4 - X=0:0.1:10;
5 - Y=sin(X);
6 - plot(X,Y,'b');
7 - xlabel(' X Axis ');
8 - ylabel(' Y Axis ');
9 - title(' The Sine Wave graph ');
10 - grid
11
```

وبالتالي نحصل على الصورة التالية

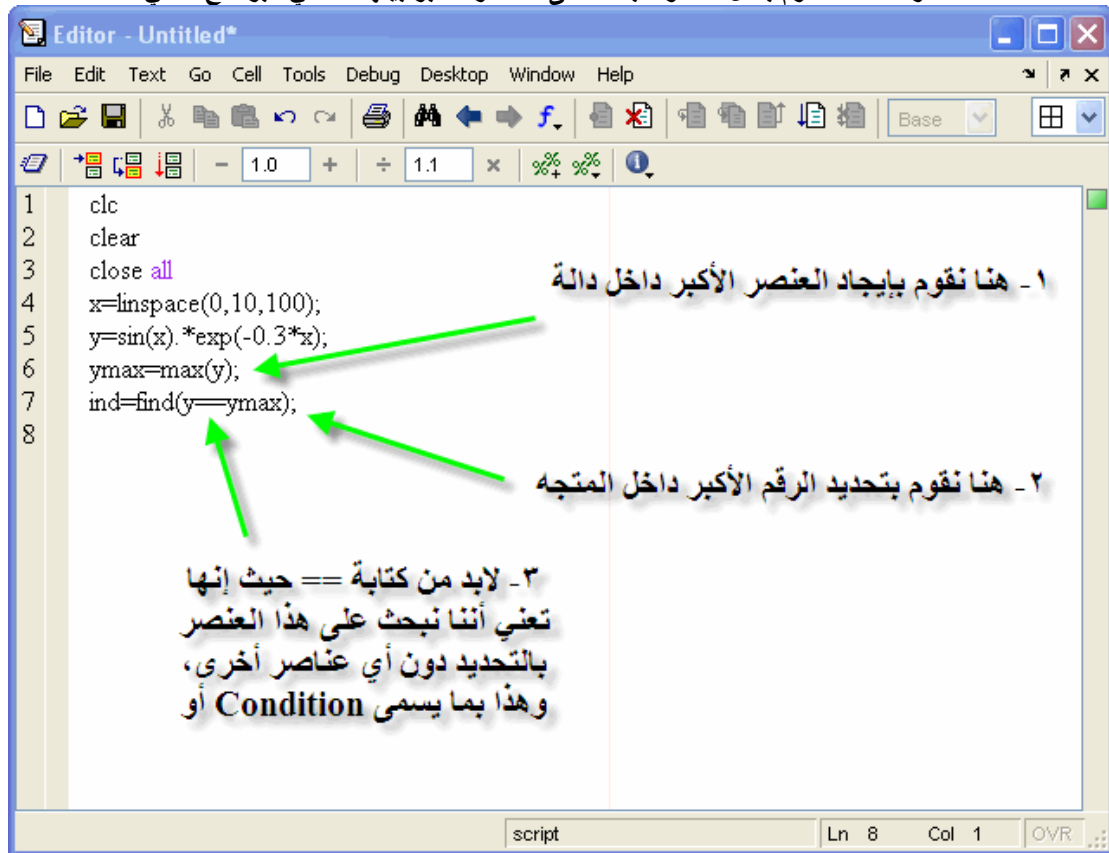


وضع نص على نقطة أو أكثر داخل الرسم

يمكن إضافة نص على نقطة أو أكثر على الرسم، وذلك باستخدام الأمر `text` ويأخذ الصورة التالية
('Text (position of the point at X-Axis, position of Y-Axis, 'The text on that point')

يجب وضع النص بين فاصلتين

وسنأخذ مثلاً بسيطاً في كيفية إيجاد الرقم الأكبر، ثم وضع دائرة حمراء حول النقطة العظمى ووضع كلمة **maximum point** ولكن دعونا نقوم بشرح الأمر الهام `find` هذا الأمر يقوم بإيجاد مكان العنصر داخل المتجه بمجرد تحديد خصائص هذا العنصر، فمثلاً سنقوم بعمل دالة وسنبحث على العنصر الأكبر بينها كما في البرنامج التالي



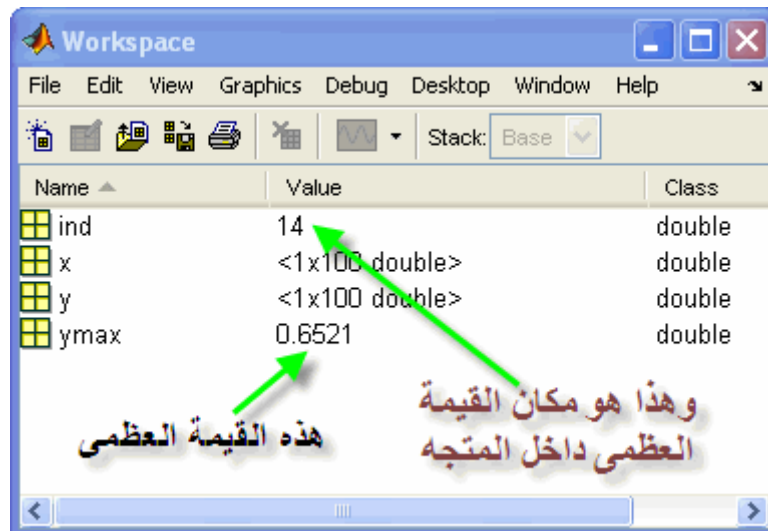
```
1 clc
2 clear
3 close all
4 x=linspace(0,10,100);
5 y=sin(x).*exp(-0.3*x);
6 ymax=max(y);
7 ind=find(y==ymax);
8
```

١ - هنا نقوم بإيجاد العنصر الأكبر داخل دالة

٢ - هنا نقوم بتحديد الرقم الأكبر داخل المتجه

٣ - لابد من كتابة == حيث إنها تعني أننا نبحث على هذا العنصر بالتحديد دون أي عناصر أخرى، وهذا بما يسمى **Condition** أو

وعند تشغيل البرنامج، نجد القيم كالتالي



وبالتالي إذا أردنا الحصول على قيمة X عند القيمة العظمى للـ Y , سنقوم بعمل التالي

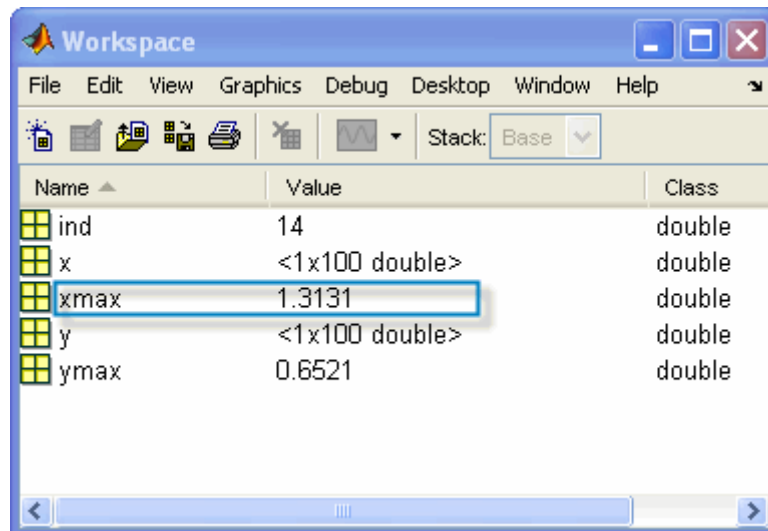
```

1 - clc
2 - clear
3 - close all
4 - x=linspace(0,10,100);
5 - y=sin(x).*exp(-0.3*x);
6 - ymax=max(y);
7 - ind=find(y==ymax);
8 - xmax=x(ind);
9 - |

```

قمنا بإيجاد قيمة X عند القيمة العظمى Y عن طريق وضع الصورة التالية $X(ind)$ أي إيجاد X عند الخانة التي تعطي Y

وكما تلاحظ فإن قيمة X والتي تعطي القيمة العظمى Y تظهر في **workspace**, أنظر الصورة التالية



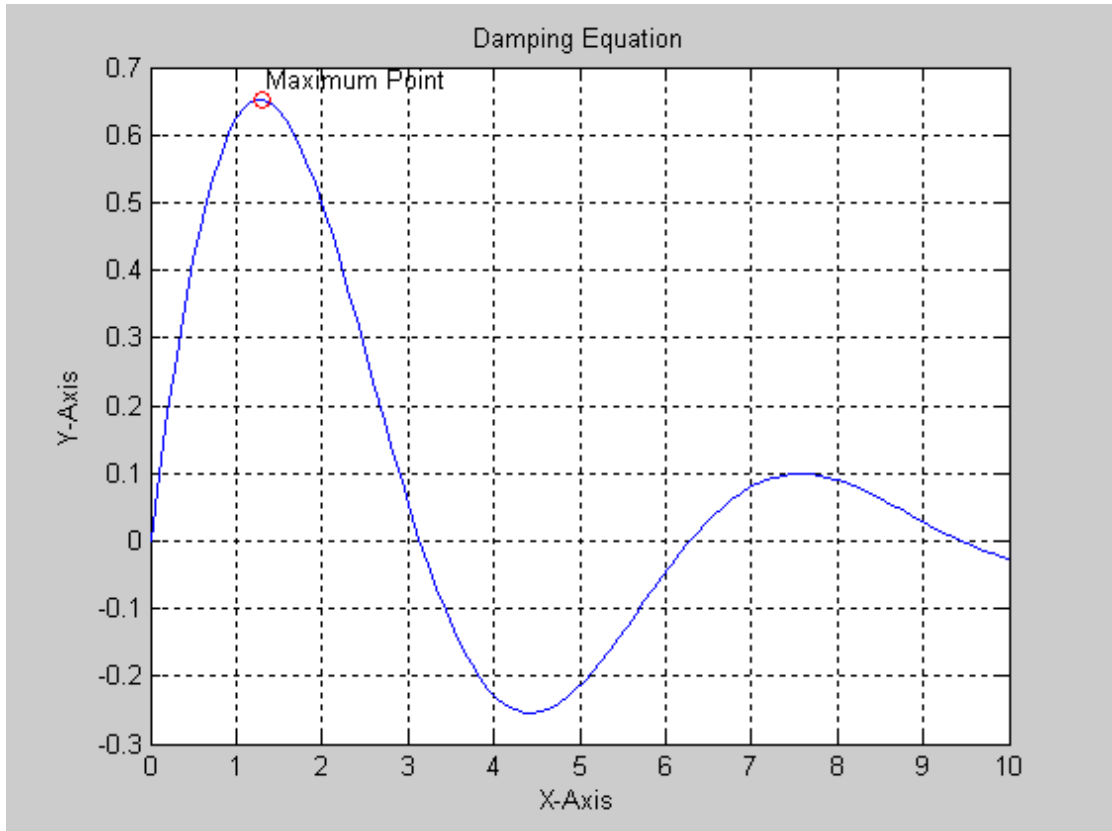
والآن سنقوم بتطبيق المثال ووضع كلمة النقطة العظمى عليها

```

1 - clc
2 - clear
3 - close all
4 - x=linspace(0,10,100);
5 - y=sin(x).*exp(-0.3*x);
6 - ymax=max(y);
7 - ind=find(y==ymax);
8 - xmax=x(ind);
9 - plot(x,y,xmax,ymax,'ro');
10 - title('Damping Equation');
11 - xlabel('X-Axis');
12 - ylabel('Y-Axis');
13 - grid
14 - text(xmax+0.03,ymax+0.03,'Maximum Point');

```

وبالتالي ستظهر الرسمة كالتالي

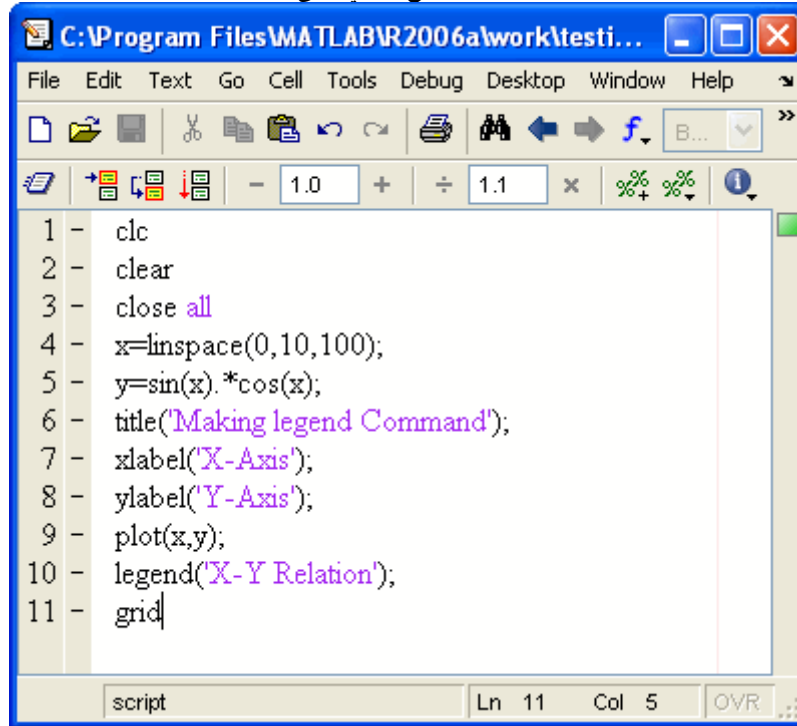


الأمر legend

يستخدم هذا الأمر في وضع دليل على صفحة الرسم ليبين ماذا يعني كل لون على الرسم، فمثلاً سنقوم بوضع الأمر legend في المثال التالي علماً أن هذا الأمر لابد من أن يأخذ الصورة التالية

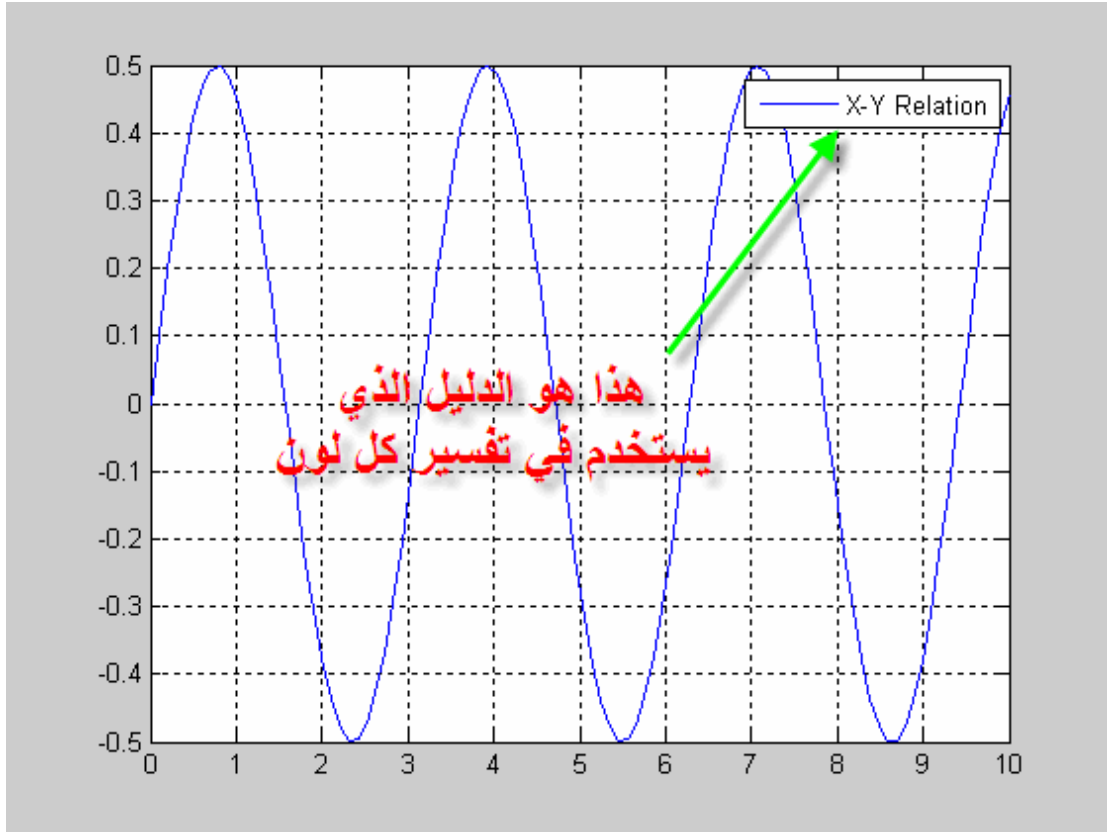
Legend ('the color reference')

ويمكن كتابة البرنامج التالي على الماتلاب



```
1 - clc
2 - clear
3 - close all
4 - x=linspace(0,10,100);
5 - y=sin(x).*cos(x);
6 - title('Making legend Command');
7 - xlabel('X-Axis');
8 - ylabel('Y-Axis');
9 - plot(x,y);
10 - legend('X-Y Relation');
11 - grid
```

وبالتالي ستجد الناتج كالتالي



كما ترى فإن الأمر **legend** يعتمد على عدد العلاقات المرسومة داخل الرسم، فمثلاً المثال الذي سبق أخذه كان يستخدم في رسم علاقة ثم إيجاد النقطة العظمى أي أن عدد العلاقات المرسومة إثنان، وبالتالي تتم برمجته بالشكل التالي


```

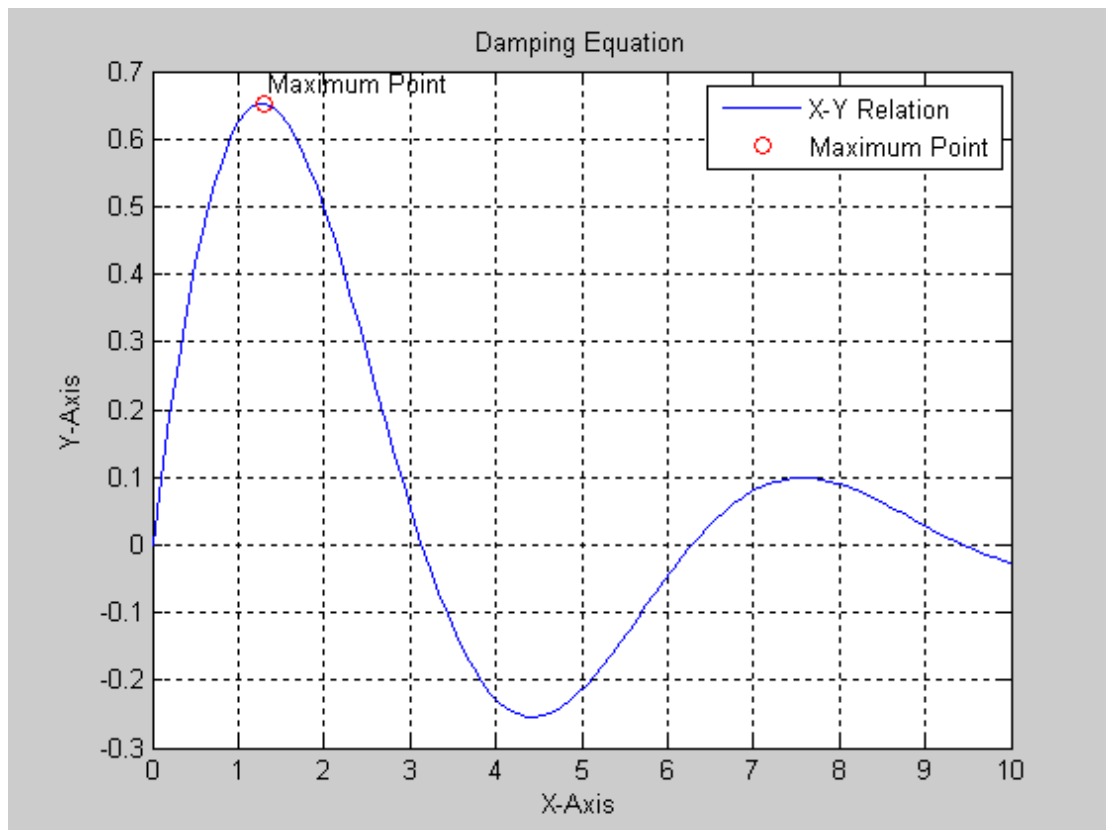
C:\Documents and Settings\Eng.Ahmed Afifi\Desktop\tr...
File Edit Text Go Cell Tools Debug Desktop Window Help
[Icons] B... >>
- 1.0 + ÷ 1.1 × [Zoom] [Info]
1 - clc
2 - clear
3 - close all
4 - x=linspace(0,10,100);
5 - y=sin(x).*exp(-0.3*x);
6 - ymax=max(y);
7 - ind=find(y==ymax);
8 - xmax=x(ind);
9 - plot(x,y,xmax,ymax,'ro');
10 - title('Damping Equation');
11 - xlabel('X-Axis');
12 - ylabel('Y-Axis');
13 - grid
14 - text(xmax+0.03,ymax+0.03,'Maximum Point');
15 - legend('X-Y Relation','Maximum Point');
16

```

← ↑ →
 للعلاقة الأولى فاصل للعلاقة الثانية

script Ln 16 Col 1 OVR

وبالتالي تكون الرسمة كالتالي



ويجب مراعاة أن يتم إستخدام الأمر **legend** بعد الأمر **plot** وليس العكس

فتح نافذة جديدة وتحديد دقتها
يعطي الماتلاب القدرة على فتح نافذة جديدة وتحديد القيم العظمى والصغرى لمحور السينات وكذلك بالنسبة لمحور
الصادات، وذلك بإستخدام الأمر **axis** والذي يأخذ الصورة التالية في كتابته

!Error

`axis([minimum value of X , Maximum value of X , Minimum value of Y , Maximum value of Y])`

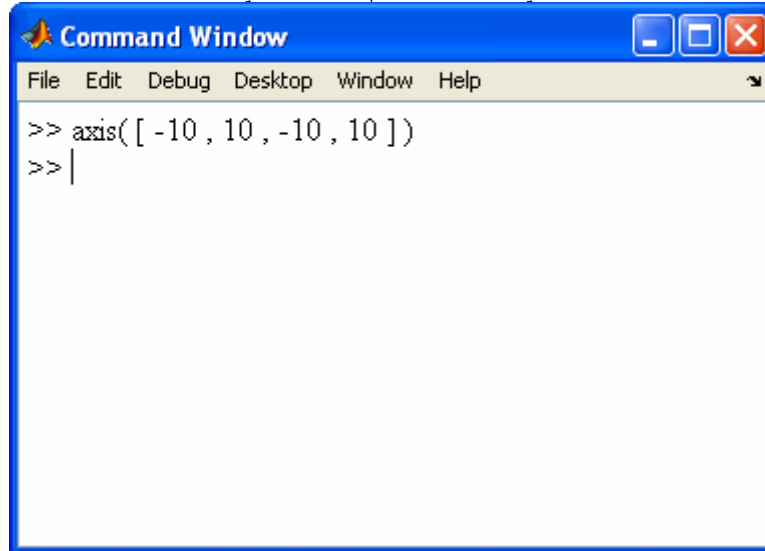
مثال تطبيقي

قم بفتح نافذة للرسم بحيث تكون بالمواصفات التالية

- ١- أقل قيمة لمحور السينات هي ١٠-
- ٢- أكبر قيمة لمحور السينات هي ١٠
- ٣- أقل قيمة لمحور الصادات -١٠
- ٤- أكبر قيمة لمحور الصادات ١٠

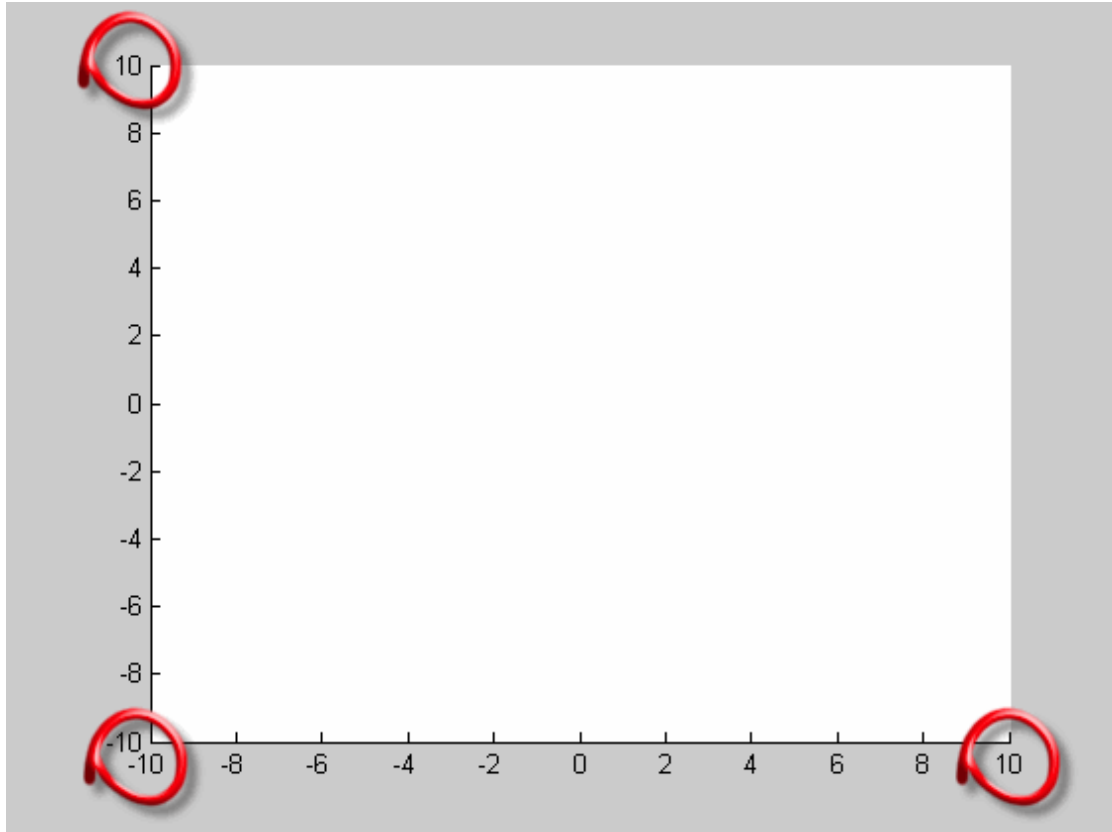
خطوات الحل

في نافذة الأوامر قم بإدخال التالي



```
Command Window
File Edit Debug Desktop Window Help
>> axis([-10, 10, -10, 10])
>> |
```

وستظهر لك النافذة التالية

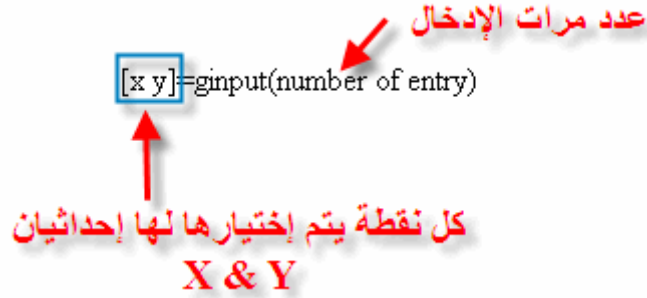


وبالتالي نكون قد أتممنا شرح كيفية فتح نافذة للرسم بنجاح
يمكنك الآن وضع الخصائص التي تريدها على تلك النافذة
أما الدرس القادم فهو مهم جداً وهو كيفية إدخال النقط على الرسم من خلال إستخدام الماوس
ونلقاكم في رعاية الله في الدرس القادم

كيفية إدخال النقاط من خلال الماوس

تعلمنا أنه يمكننا إدخال القيم باستخدام المتجهات أو المصفوفات، ولكن يوفر الماتلاب قدرة في إدخال النقاط من خلال الرسم باستخدام الماوس، ونظراً لأننا نقوم باختيار النقاط من على الرسم فهذا يعني أن النقاط التي يتم إختيارها يتم تمثيلها في قيمة في محور السينات وقيمة في محور الصادات، ويتم وضع قيم محاور السينات والصادات في صورة متجه.

يستخدم الأمر `ginput` في عملية إدخال النقاط باستخدام الماوس، ويتم كتابة ذلك الأمر في الصورة التالية



أما إذا أردنا إدخال عدد لا نهائي من النقاط يمكن ذلك بعدم ذكر عدد نقاط الإدخال, كما في الشكل التالي

`[x y]=ginput()`

وذلك لإدخال عدد لا نهائي من النقاط

وبعد الإنهاء من إدخال النقاط كل ما عليك هو الضغط على مفتاح `Enter` في لوحة المفاتيح.

مثال تطبيقي

سنقوم بفتح نافذة للرسم بها شبكة، وأقل قيمة لمحور السينات هي صفر وأكبر قيمة لمحور السينات هي ١٠ وكذلك بالنسبة لمحور الصادات، ثم إدخال عدد كبير من النقاط على الرسم باستخدام الأمر `ginput` وهذه النقاط يتم طباعتها على شكل دوائر حمراء. ويتم كتابة الأوامر بالشكل التالي

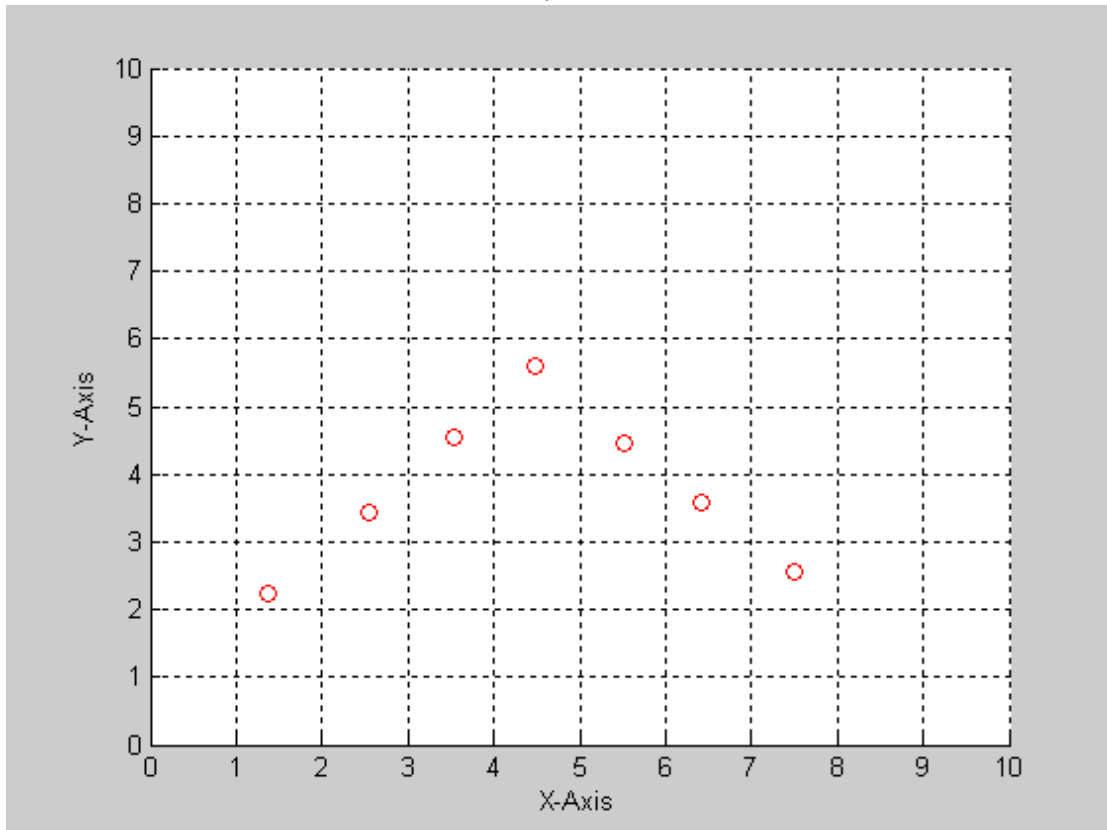
```

C:\Program Files\MATLAB\R2006a\work\week_2.m
File Edit Text Go Cell Tools Debug Desktop Window Help
[Icons] B...
- 1.0 + ÷ 1.1 × % % ?
1 - clc
2 - clear
3 - close all
4 - hold on
5 - axis([0 10 0 10]);
6 - xlabel('X-Axis');
7 - ylabel('Y-Axis');
8 - grid
9 - [x y]=ginput(3);
10 - plot(x,y,'ro')
11
script Ln 11 Col 1 OVR

```

لا بد من وضع الأمر **hold on** حتى يتم رسم النقاط المختلفة في نفس الرسمة

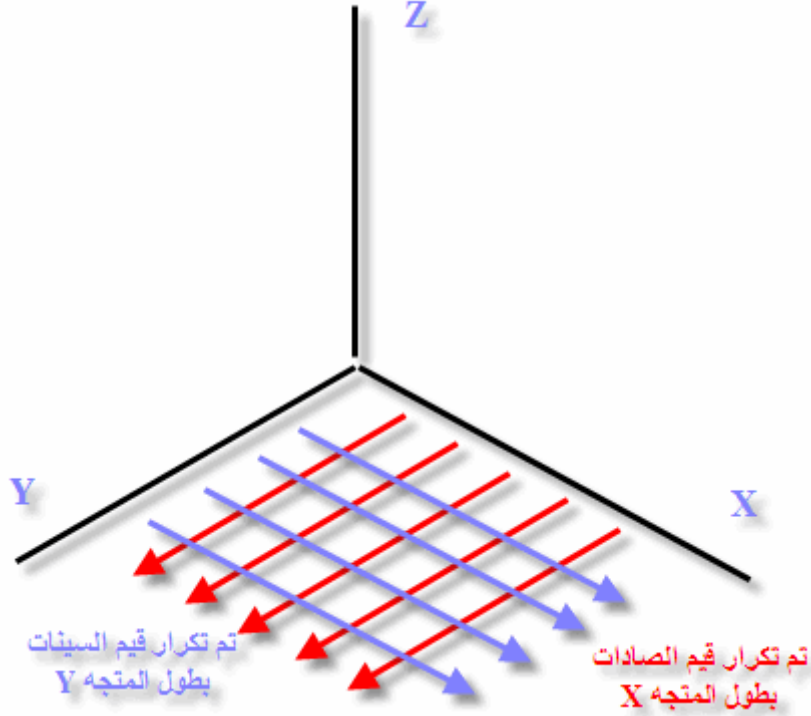
وستظهر لك نافذة لإدخال النقاط ، وبعد إتمام عملية الإدخال اضغط على **Enter** لإتمام الإدخال وستظهر لك النافذة التالية



وبالتالي نكون قد أتمنا شرح هذه الجزئية بنجاح، وسيتم التطرق في دورة البرمجة باستخدام الماتلاب إلى كيفية إظهار النقاط بمجرد الضغط عليها.

الرسم ثلاثي الأبعاد

كما تعلمنا أن الرسم ثلاثي الأبعاد يعتمد على ثلاثة محاور لرسمها، محور X , Y & Z ، وأن كلاً من X & Y يمثلان المستوى الأفقي، وأن المحور Z يمثل الارتفاع، ولكن تلك القيم هي قيم النقاط الموجودة المحاور، ولكن حتى يتم رسم أي نقطة في المستوى الأفقي يجب أن نقوم بتعريف ذلك للماتلاب وذلك باستخدام الأمر `meshgrid` حيث يقوم الماتلاب بإنتاج مصفوفة يتم تكرار قيم محور السينات X -Axis بنفس طول محور الصادات Y -Axis كما يقوم بتكرار قيم محور الصادات Y -Axis بنفس طول قيم السينات X -Axis وبهذا تكون المصفوفة المتكونة هي المستوى الأفقي كما هو واضح بالرسم التالي



علما أن الأمر `meshgrid` يأخذ الصورة التالية في كتابته

$$[x \ y] = \text{meshgrid}(x, y)$$

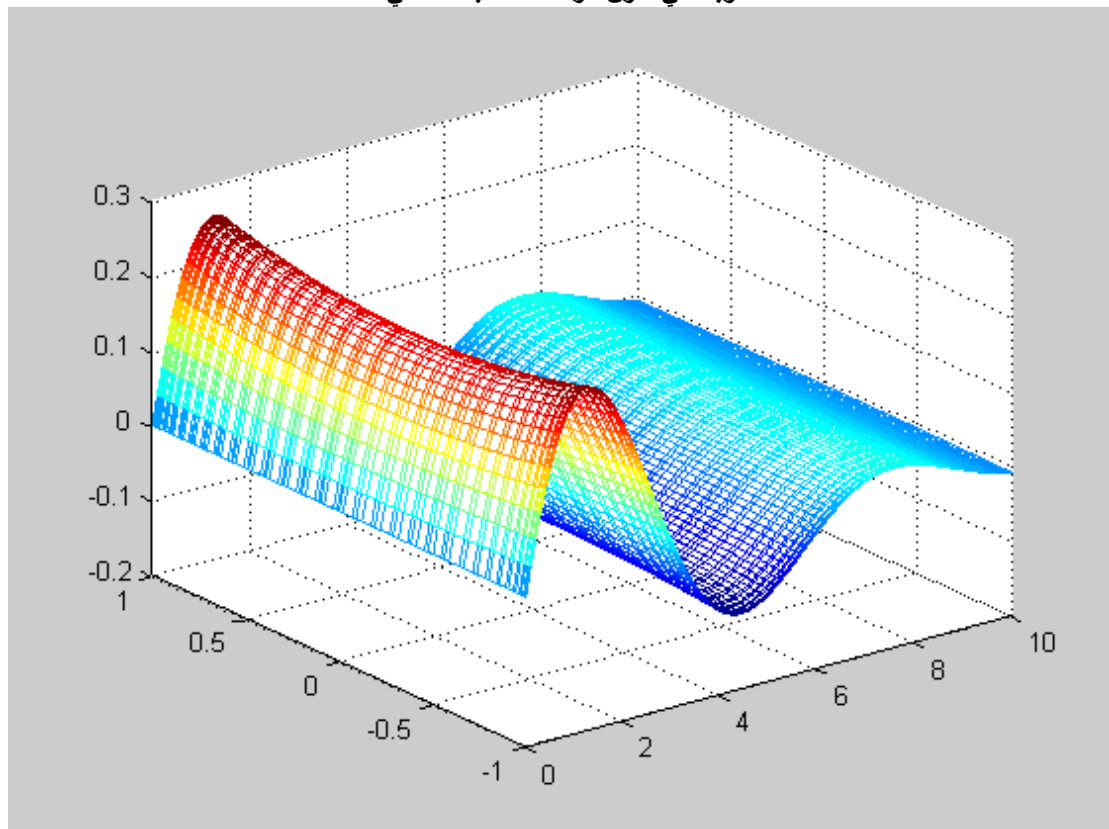
وبعد استخدام الأمر `meshgrid` يتم استخدام الأمر `mesh` والذي يستخدم كبديل الأمر `plot` ولكن في الرسم ثلاثي الأبعاد

مثال تطبيقي

نقوم في هذا المثال بتعريف قيم محور السينات X -Axis وسنقوم بوضع المعادلة التي تصف محور الصادات وعلاقته بمحور السينات، أخيراً وليس آخراً نقوم بوضع العلاقة التي تربط بين محور السينات والصادات.

```
C:\Program Files\MATLAB\R2006a\work\week_2_...
File Edit Text Go Cell Tools Debug Desktop Window Help
[Icons] [B...]
- 1.0 + ÷ 1.1 × %>% %>% ⓘ
1 - clc
2 - clear
3 - close all
4 - x=linspace(0,10,100);
5 - y=sin(x);
6 - [x y]=meshgrid(x,y);
7 - z=sin(x).*exp(-0.3*x)./(cos(y)+2);
8 - mesh(x,y,z);
9
script Lh 4 Col 20 OVR
```

وبالتالي تكون الرسمة الناتجة كالتالي

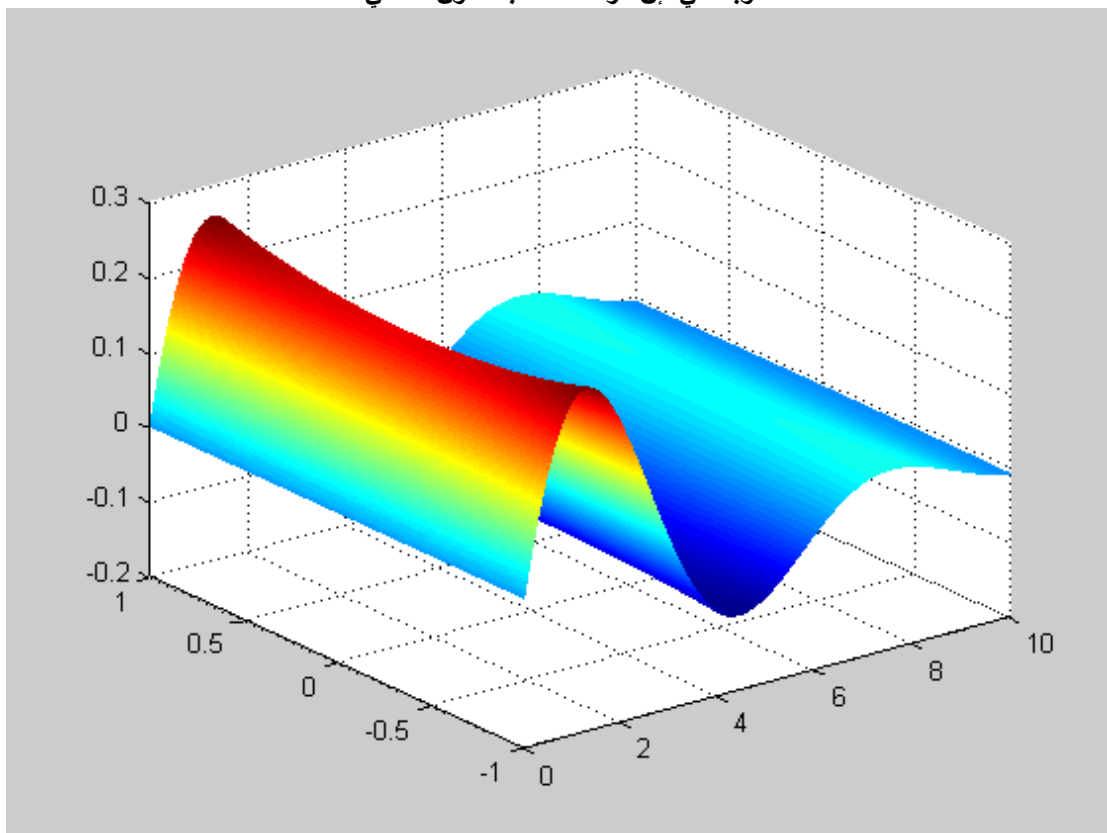


كما ترى فإن الرسة الناتجة عبارة عن شبكة تعتمد مجموعة النقاط لكلاً من X & Y فإذا أكثرنا عدد نقاط X وبالتالي تزداد قيمة Y كذلك

```
C:\Program Files\MATLAB\R2006a\work\week_2_...
File Edit Text Go Cell Tools Debug Desktop Window Help
[Icons] [B...]
- 1.0 + 1.1 x % % 1
1 - clc
2 - clear
3 - close all
4 - x=linspace(0,10,1000);
5 - y=sin(x);
6 - [x y]=meshgrid(x,y);
7 - z=sin(x).*exp(-0.3*x)./(cos(y)+2);
8 - mesh(x,y,z);
9 |
script Ln 9 Col 1 OVR
```

كما ترى فإن عدد النقاط تم زيادتها من 100 إلى 1000

وبالتالي فإن الرسة الناتجة تكون كالتالي



أعتقد أنك تلاحظ الفرق الآن
ملاحظة كلما زادت عدد النقاط كلما زاد الوقت المستغرق لإظهار النتائج في الماتلاب.

أخواني الكرام، نستكمل معكم دورة الماتلاب، وهي في إسبوعها الثالث، وسوف نتناول في هذا الإسبوع العديد من الأمثلة التطبيقية، وسيتم شرح مثال تطبيقي كل يوم، وهذا ملخص للأمثلة التطبيقية التي سيتم أخذها بإذن الله.

١- كيفية إدخال ثلاث معادلات وإيجاد قيم المتغيرات

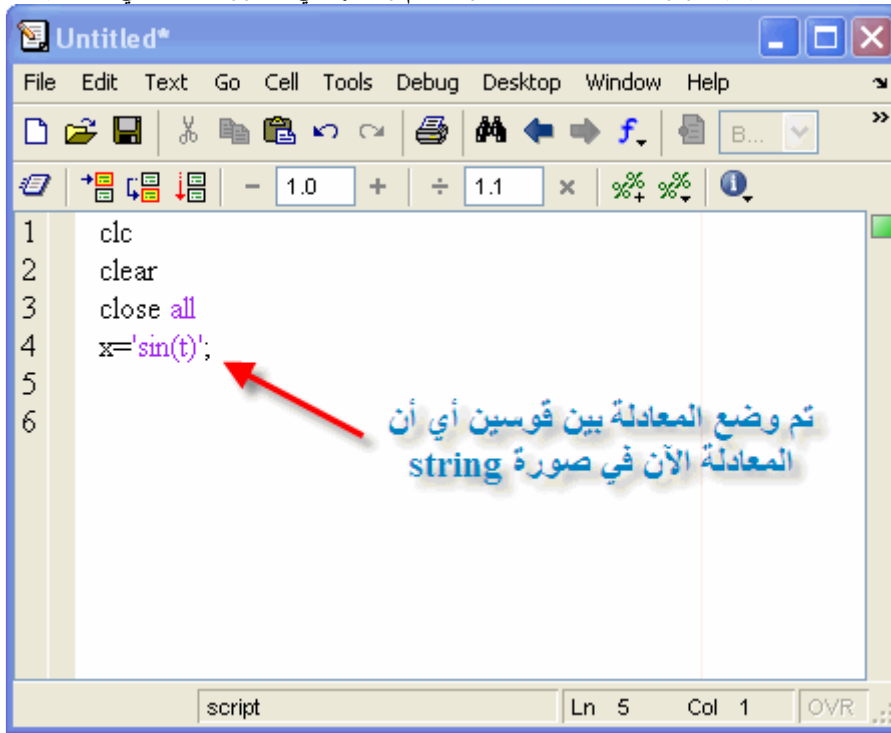
٢- إيجاد نقط تقاطع الرسمة مع محور السينات ووضع علامة عليها

٣- إختيار نقطتان من على الرسم، وإيجاد المساحة تحت المنحنى بين تلك النقطتين

٤- Curve Fitting

EVAL

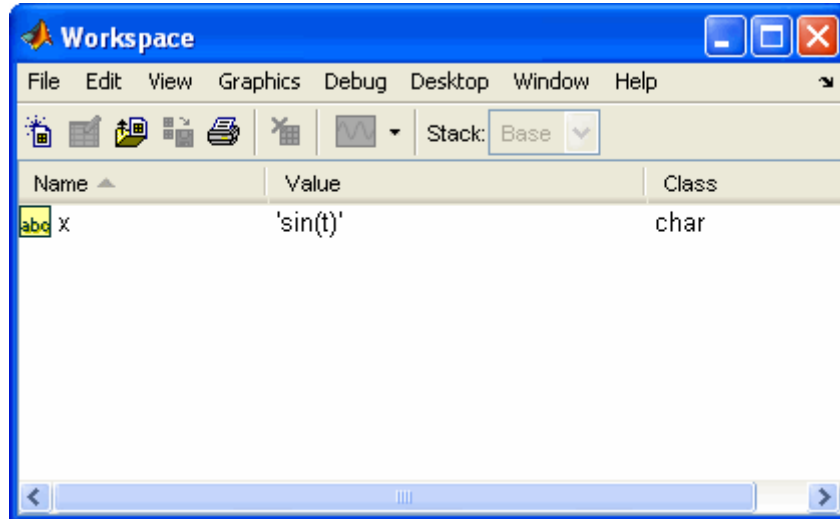
قبل البدء بالتطبيق الأول، لا بد من ذكر أمر هام وهو الأمر `eval` والذي له هدف أكثر من رائع سيتضح بمثال لنقل أن لدينا معادلة جيب الزاوية `Sine Wave` ولكن تم وضعها في الصورة التالية في الماتلاب



```
1 clc
2 clear
3 close all
4 x='sin(t)';
5
6
```

تم وضع المعادلة بين قوسين أي أن
المعادلة الآن في صورة string

وللتأكد أنها في صورة string يجب الذهاب إلى الـ `Workspace`

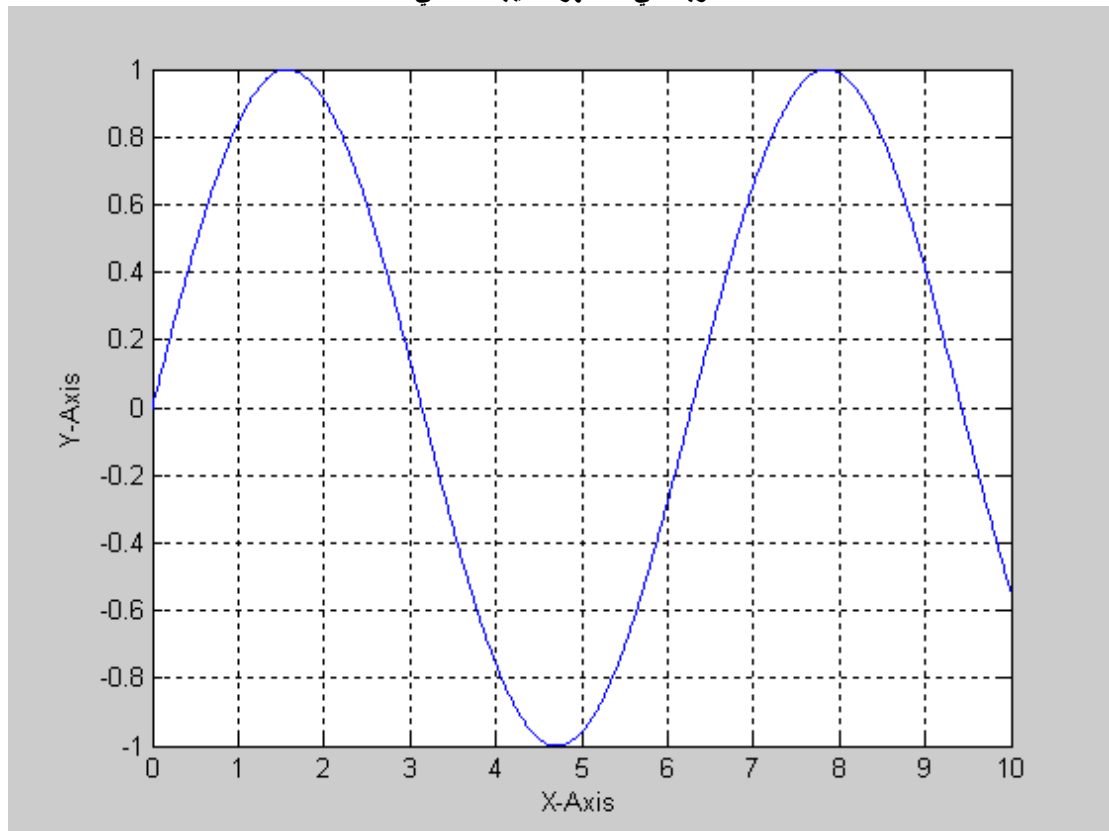


Name	Value	Class
x	'sin(t)'	char

ولرسم موجة جيب الزاوية لا بد من تعريف قيم `t` والتعويض فيها، ولكن كما ترون يصعب التعويض في المعادلة بسبب وجودها بين قوسين، وتلك الأقواس هي بمثابة حاجز للتعويض، وهنا يجب أن نخترق ذلك الحاجز وذلك باستخدام الأمر `eval`، بحيث يعمل الأمر على البحث عن المعادلة بين الأقواس، ثم يبحث عن القيم التي سوف يتم تعويضها في تلك المعادلة، وسوف يتم كتابة البرنامج كالتالي في الماتلاب

```
C:\Program Files\MATLAB\R2006a\work\Evalute_command.m
File Edit Text Go Cell Tools Debug Desktop Window Help
[Icons] Stack: Base
- 1.0 + ÷ 1.1 x %>% %>% %>%
1 - clc
2 - clear
3 - close all
4 - x='sin(t)'; % By Writing the equation in the string form
5 - t=linspace(0,10,1000); % By defining the inputs
6 - x_new=eval(x); % By using the Evaluate cmmand to get the value of the function x
7 - plot(t,x_new) % By Plotting the function
8 - grid
9 - xlabel('X-Axis');
10 - ylabel('Y-Axis');
11 - |
```

وبالتالي ستظهر النتيجة كالتالي



ونحن الآن على أتم إستعداد لتنفيذ التطبيق الأول

حل ثلاثة معادلات

كما تعلمنا أن الماتلاب له القدرة على إدخال حل المعادلات، فمثلاً المعادلتان التاليتين

$$X + 2Y = 5$$

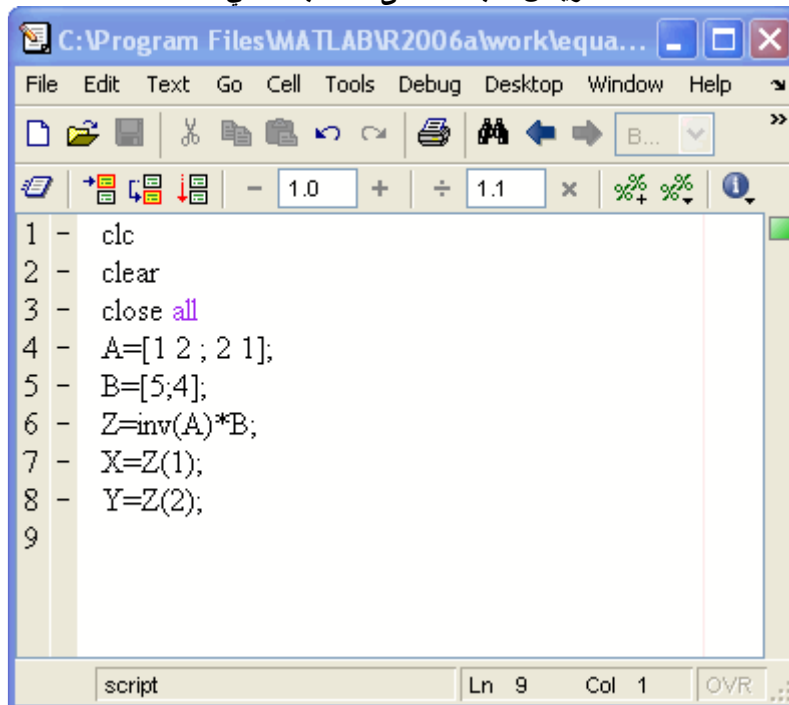
$$2X + Y = 4$$

$$\begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} 5 \\ 4 \end{pmatrix}$$

$$A \times Z = B$$

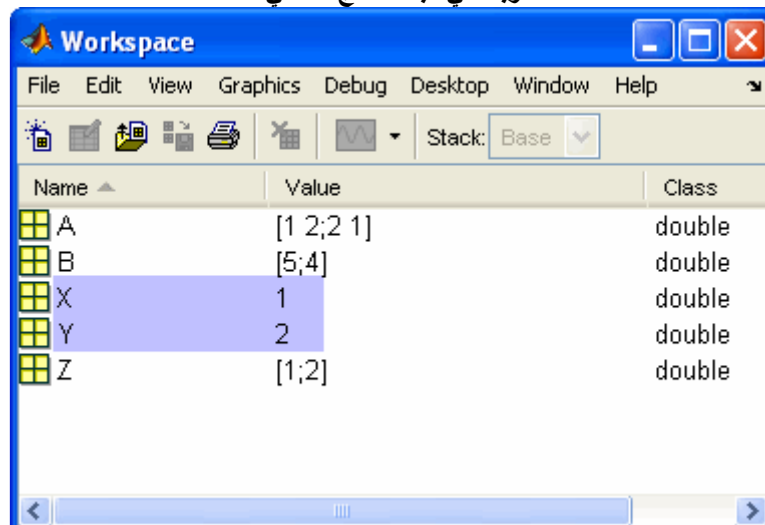
$$Z = A^{-1} \times B$$

ويمكن كتابة ذلك على الماتلاب كالتالي



```
C:\Program Files\MATLAB\R2006a\work\lequa...
File Edit Text Go Cell Tools Debug Desktop Window Help
[Icons] B...
1.0 1.1 x % %
1 - clc
2 - clear
3 - close all
4 - A=[1 2 ; 2 1];
5 - B=[5;4];
6 - Z=inv(A)*B;
7 - X=Z(1);
8 - Y=Z(2);
9
script Ln 9 Col 1 OVR
```

وبالتالي نجد النتائج كالتالي



Name	Value	Class
A	[1 2; 2 1]	double
B	[5; 4]	double
X	1	double
Y	2	double
Z	[1; 2]	double

ولكن هذا يشترط أن يتم إدخال المعاملات **coefficients** للمعادلتين، وهذا بالتالي يتطلب التمحيص والتدقيق في كل معادلة، فإذا كثرت المعادلات إزداد الوقت المستغرق في البحث، فتسهيلاً للمستخدم يجب عمل برنامج لإدخال المعادلات بشكل كام، فمثلاً سنقوم بعمل برنامج لحل ثلاثة معادلات

$$A_1 \times X + B_1 \times Y + C_1 \times Z = D_1$$

$$A_2 \times X + B_2 \times Y + C_2 \times Z = D_2$$

$$A_3 \times X + B_3 \times Y + C_3 \times Z = D_3$$

ولحل هذه المعادلات يجب وضعها في الصورة التالية

$$\begin{pmatrix} A_1 & B_1 & C_1 \\ A_2 & B_2 & C_2 \\ A_3 & B_3 & C_3 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} D_1 \\ D_2 \\ D_3 \end{pmatrix}$$

ولذلك يجب التفكير كالتالي، عندما يقوم المستخدم بإدخال المعادلات الثلاثة، يجب على الماتلاب أن يحدد قيم المعاملات أولاً لإيجاد حل قيم المتغيرات، وحالاً لهذه المشكلة سنجعل الماتلاب يبحث عن علامة "="

لكل معادلة، ثم سنقوم بتعريف الجزء الذي يحتوى على المتغيرات لكل معادلة، ثم سنقوم بفرض أن

$$x=1, y=0, z=0$$

ومنها نحصل على قيم معاملات x في المعادلات الثلاثة

$$x=0, y=1, z=0$$

ومنها نحصل على قيم معاملات y في المعادلات الثلاثة

$$x=0, y=0, z=1$$

ومنها نحصل على قيم معاملات z في المعادلات الثلاثة

وبالتالي نكون قد حصلنا على معاملات المعادلات الثلاثة، لم يتبقى سوى إيجاد قيم معاملات D وسوف يتم توضيح ذلك عند كتابة البرنامج على الماتلاب.

```

C:\Program Files\MATLAB\R2006a\work\equation_solver_1.m*
File Edit Text Go Cell Tools Debug Desktop Window Help
Stack: Base
1 - clc
2 - clear
3 - close all
4 - equ1=input('Please Enter the 1st equation','s'); % The 1st Equation
5 - equ2=input('Please Enter the 2nd equation','s'); % The 2nd Equation
6 - equ3=input('Please Enter the 3rd equation','s'); % The 3rd Equation
7 - ind1=find(equ1=='='); % for finding the position of the equal sign at the 1st Equation
8 - ind2=find(equ2=='='); % for finding the position of the equal sign at the 2nd Equation
9 - ind3=find(equ3=='='); % for finding the position of the equal sign at the 3rd Equation
10 - var1=equ1(1: ind1-1); % for getting the 1st equation without the absolute value
11 - var2=equ2(1: ind2-1); % for getting the 2nd equation without the absolute value
12 - var3=equ3(1: ind3-1); % for getting the 3rd equation without the absolute value
13 - x=1,y=0,z=0; % 1st Condition
14 - a(1,1)=eval(var1); %By substituting in the function (var1) by the 1st Condition
15 - a(2,1)=eval(var2); %By substituting in the function (var2) by the 1st Condition
16 - a(3,1)=eval(var3); %By substituting in the function (var3) by the 1st Condition
17 - x=0,y=1,z=0; % 2nd Condition
18 - a(2,1)=eval(var1); %By substituting in the function (var1) by the 2nd Condition
19 - a(2,2)=eval(var2); %By substituting in the function (var2) by the 2nd Condition
20 - a(2,3)=eval(var3); %By substituting in the function (var3) by the 2nd Condition
21 - x=0,y=0,z=1; % 3rd Condition
22 - a(3,1)=eval(var1); %By substituting in the function (var1) by the 3rd Condition
23 - a(3,2)=eval(var2); %By substituting in the function (var2) by the 3rd Condition
24 - a(3,3)=eval(var3); %By substituting in the function (var3) by the 3rd Condition
25 - % By Getting the Absolute values of the equations
26 - b(1)=str2double(equ1(ind1+1:end)); % By getting the absolute value of the 1st equation
27 - b(2)=str2double(equ2(ind2+1:end)); % By getting the absolute value of the 2nd equation
28 - b(3)=str2double(equ3(ind3+1:end)); % By getting the absolute value of the 3rd equation
29 - b=b';
30 - sol=inv(a)*b;
31 - x=sol(1,1); % The value of X
32 - y=sol(2,1); % The value of Y
33 - z=sol(3,1); % The value of Z
script Ln 33 Col 55 OVR

```

وسنقوم الآن بوضع ثلاثة معادلات كما في الشكل التالي

$$2x+3y+z=5$$

$$3x+4y+5z=9$$

$$3x+3y+7z=3$$

وبالتالي ستكون النتائج كالتالي

Name	Value	Class
a	[2 0 0;3 4 3;1 5 7]	double
b	[5;9;3]	double
abc equ1	'2*x+3*y+z=5'	char
abc equ2	'3*x+4*y+5*z=9'	char
abc equ3	'3*x+3*y+7*z=3'	char
ind1	10	double
ind2	12	double
ind3	12	double
sol	[2.5;0.6923;-0.4231]	double
abc var1	'2*x+3*y+z'	char
abc var2	'3*x+4*y+5*z'	char
abc var3	'3*x+3*y+7*z'	char
x	2.5	double
y	0.6923	double
z	-0.4231	double

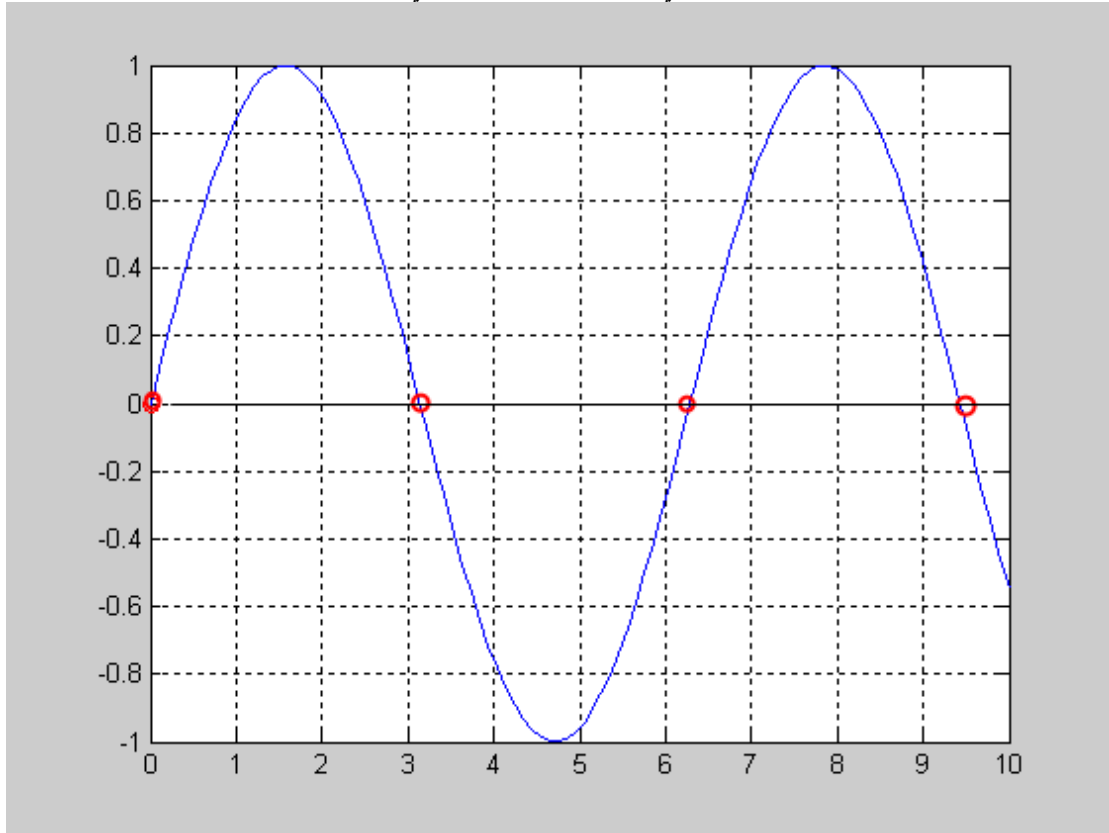
قيم المتغيرات
للمعادلات الثلاثة

وعلى نفس المنوال يمكن حل أي عدد من المعادلات مهما كانت كبيرة، وسنأخذ لاحقاً كيفية عمل نافذة لإدخال تلك المعادلات

وبهذا نكون قد أتمنا التطبيق الأول بنجاح، ولا تترددوا في مراسلتي في حالة عدم وضوح بعض الأجزاء

Zero Crossing

في هذا المثال الهام سنقوم بشرح كيفية رسم معادلة، ووضع دائرة زرقاء على مناطق تقاطعها مع محور السينات، والتي يجب أن تأخذ الشكل التالي



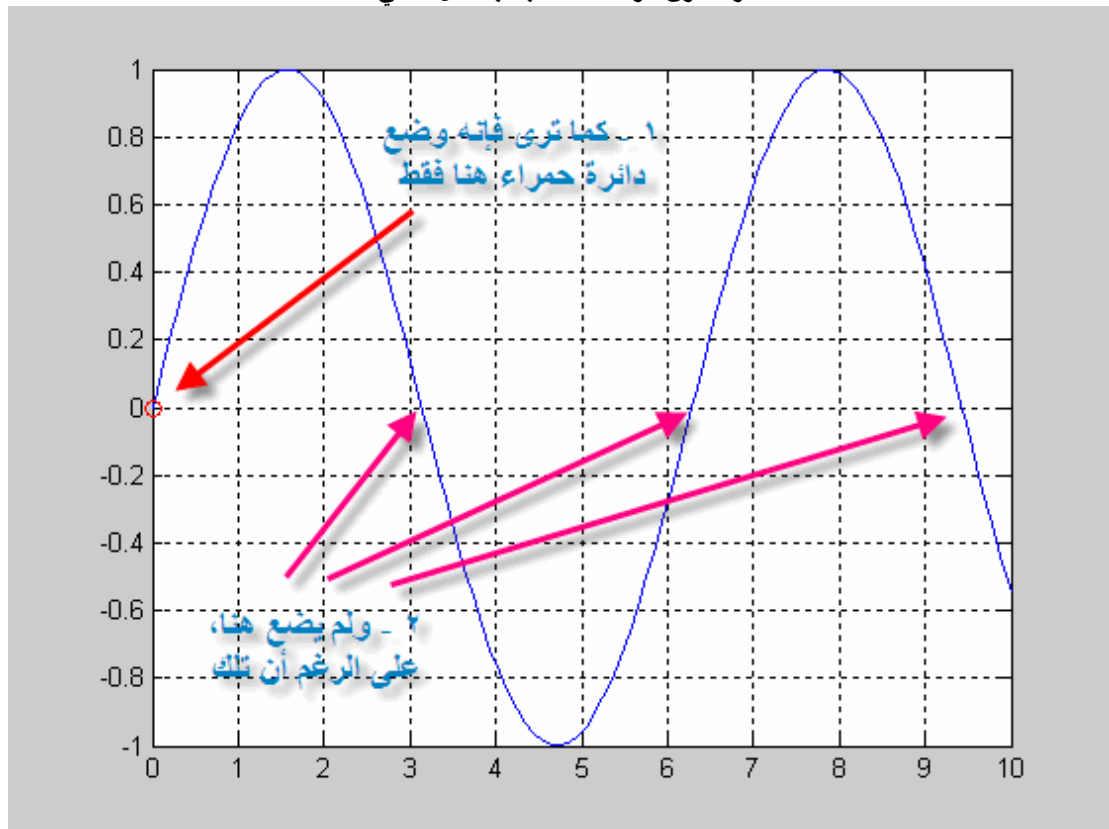
قد يعتقد البعض بأن هذا البرنامج سهلاً تقريباً، ولكن البرنامج يحتاج إلى التفكير قليلاً، فدعونا نفكر في الطريقة التي قد يفكر بها أي شخص الآن، حيث سيقوم بكتابة البرنامج الذي يبحث عن النقاط التي بها $Y=0$ ثم يقوم بإيجاد القيم التي بها تلك النقاط، كما في البرنامج التالي


```

C:\Program Files\WATLAB\R2006a\work\zero_crossing.m*
File Edit Text Go Cell Tools Debug Desktop Window Help
[Icons] Stack: Base
- 1.0 + 1.1 x % % 1
1 - clc
2 - clear
3 - close all
4 - x=linspace(0,10,100);           % Defining the inputs
5 - y=sin(x);                     % By defining the function
6 - ind=find(y==0);               % By searching the position of y=0
7 - x_crossing=x(ind);           % By finding the position of zero crossing at the X-Axis
8 - y_crossing=y(ind);           % By finding the position of zero crossing at the Y=Axis
9 - plot(x,y,x_crossing,y_crossing,'ro');
10 - grid
script Ln 10 Col 5 OVR

```

وستكون الرسمة الناتجة بالشكل التالي



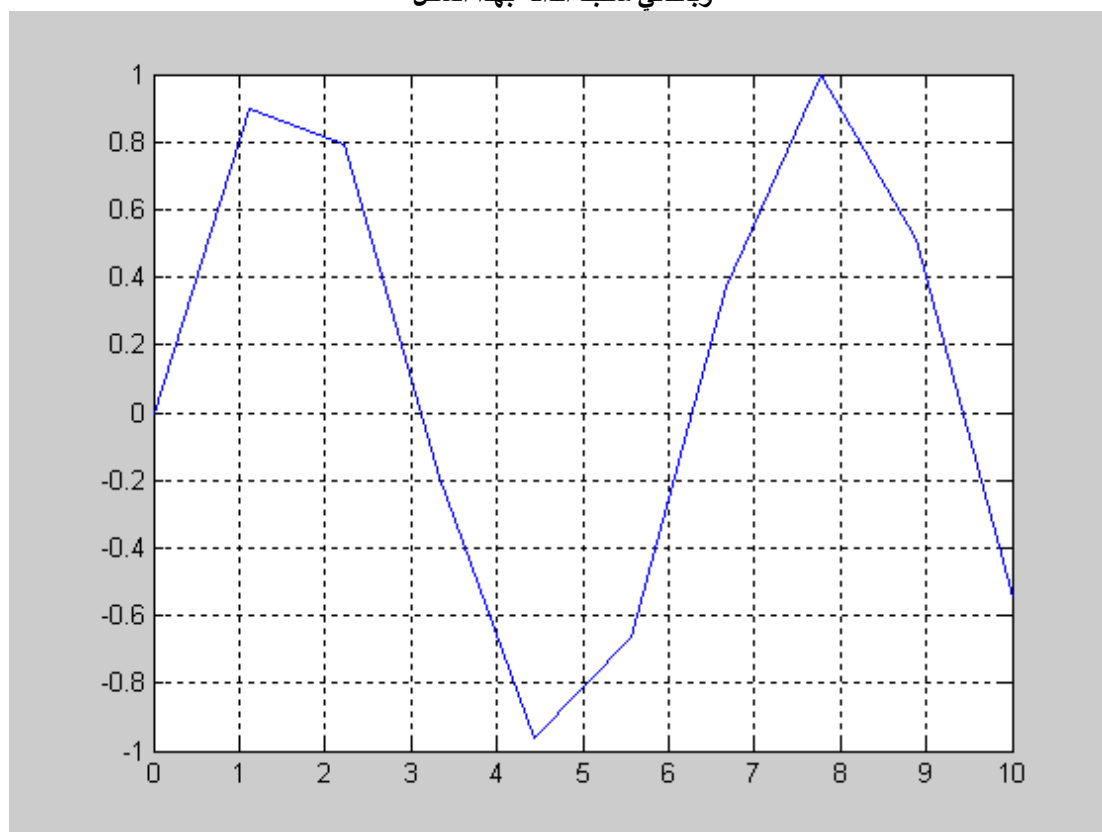
فما سبب هذه المشكلة، حقيقة سببها ما يسمى بـ **Digital Domain**، فما هو **Digital Domain**، معنى ذلك أن أي معادلة يتم رسمها ليست عبارة عن خط متصل وإنما مجموعة من النقاط، تعتمد على عدد النقاط التي تم إختيارها في لرسم الدالة، ثم التوصيل بين تلك النقاط، فمثلاً إذا قمنا بعمل برنامج لرسم دالة الجيب **sine wave** بعدد نقاط قليل، كما في الشكل التالي

```

C:\Program Files\MATLAB\R2006a\work\...
File Edit Text Go Cell Tools Debug Desktop
+ - 1.0 + ÷ 1.1 x % %
1 - clc
2 - clear
3 - close all
4 - x=linspace(0,10,10);
5 - y=sin(x);
6 - plot(x,y)
7 - grid
script Ln 6 Col 10 OVR

```

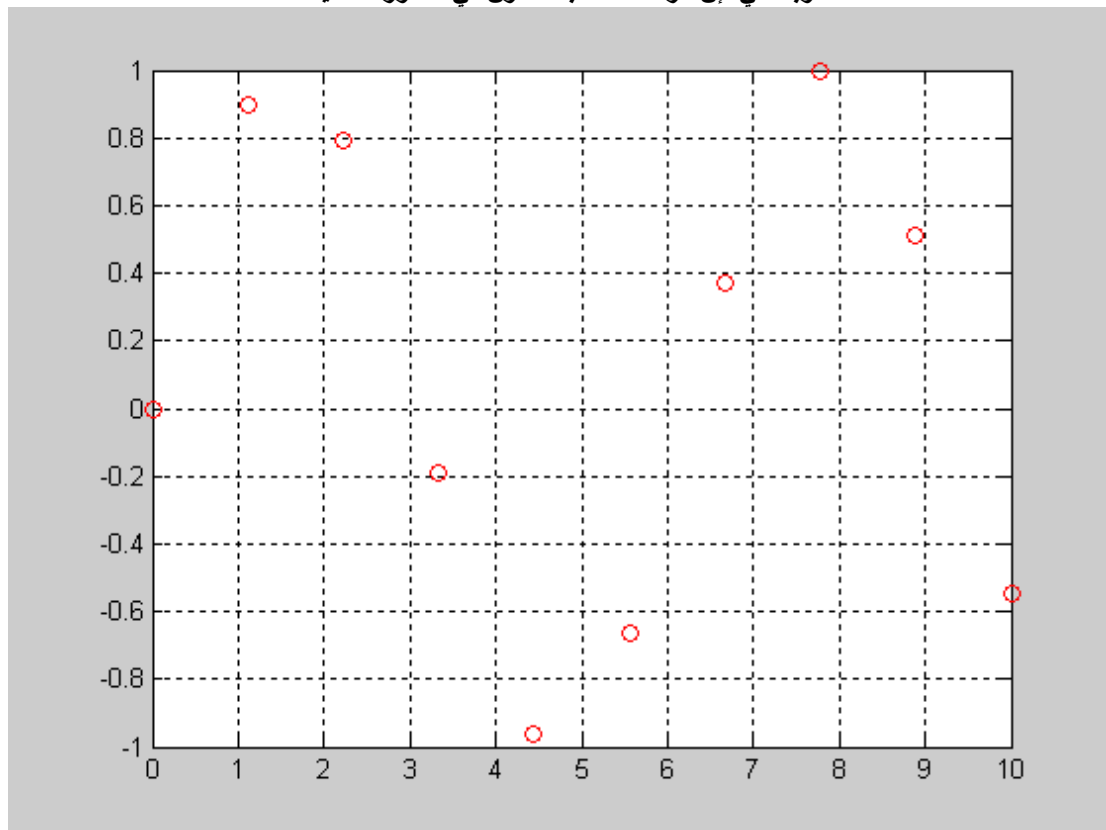
وبالتالي ستجد الدالة بهذا الشكل



كما ترى، فإن الماتلاب قام بتحديد النقاط والتوصيل بينها، وللتأكد من ذلك قم بعمل التالي في البرنامج

```
C:\Program Files\MATLAB\R2006a\work\...
File Edit Text Go Cell Tools Debug Desktop
+ - 1.0 + ÷ 1.1 × % %
1 - clc
2 - clear
3 - close all
4 - x=linspace(0,10,10);
5 - y=sin(x);
6 - plot(x,y,'ro')
7 - grid
script Ln 7 Col 5 OVR
```

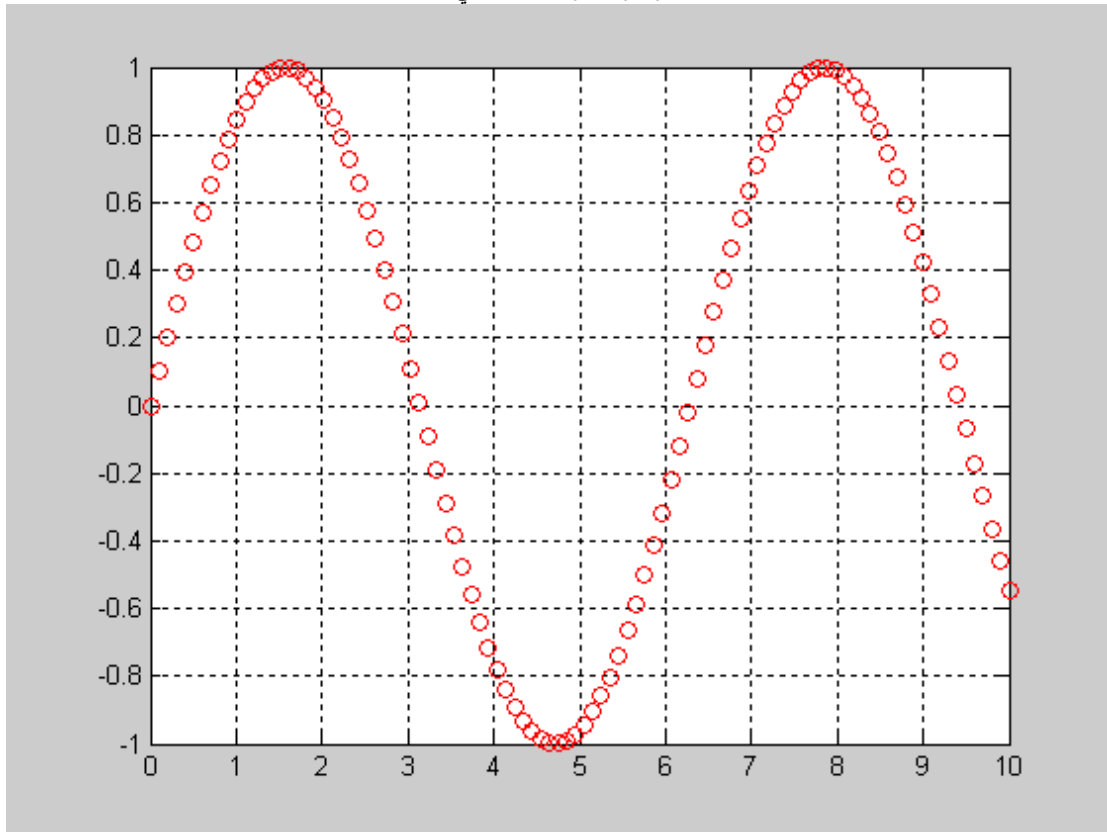
وبالتالي فإن الرسمة الناتجة، تكون في الصورة التالية



وهذا ما يسمى **Digital Domain** والآن لنقم بزيادة عدد النقاط في نفس البرنامج

```
C:\Program Files\MATLAB\R2006a\work\...
File Edit Text Go Cell Tools Debug Desktop
+ - 1.0 + ÷ 1.1 × % % %
1 - clc
2 - clear
3 - close all
4 - x=linspace(0,10,100);
5 - y=sin(x);
6 - plot(x,y,'ro')
7 - grid
script Ln 4 Col 18 OVR
```

وتكون الرسمة كالتالي



كما ترى في الرسمة لا يزال هناك فراغات بين النقاط ، والتي من الممكن أن لا تتقاطع مع محور السينات كما حدث في البرنامج الذي قمنا بعمله، وهذا هو سبب عدم ظهور دائرة حمراء حول منطقة التقاطع، ودعونا نقوم بعمل البرنامج الأول ولكن في صورة **Digital Domain**

```

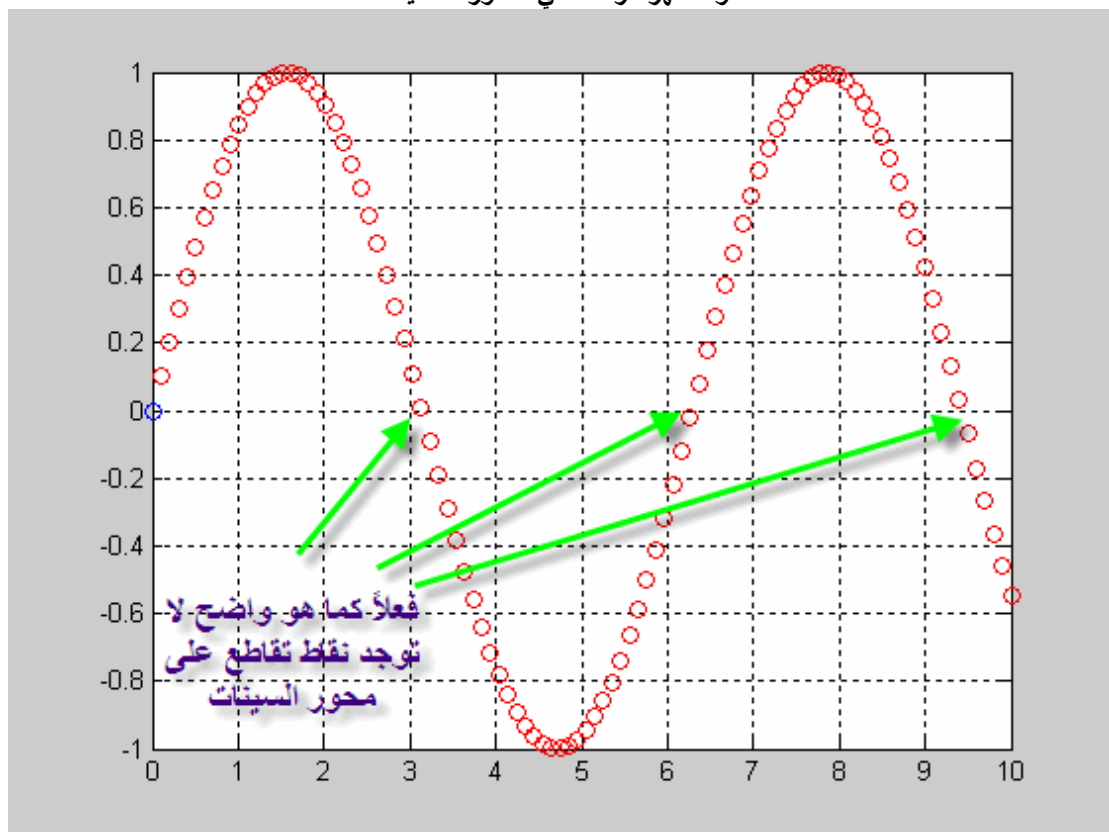
C:\Program Files\WATLAB\R2006a\work\zero_crossing.m*
File Edit Text Go Cell Tools Debug Desktop Window Help
[Icons] Stack: Base
- 1.0 + 1.1 x % % 1
1 - clc
2 - clear
3 - close all
4 - x=linspace(0,10,100);           % Defining the inputs
5 - y=sin(x);                     % By defining the function
6 - ind=find(y==0);               % By searching the position of y=0
7 - x_crossing=x(ind);            % By finding the position of zero crossing at the X-Axis
8 - y_crossing=y(ind);            % By finding the position of zero crossing at the Y=Axis
9 - plot(x,y,'ro',x_crossing,y_crossing,'bo');
10 - grid
11

```

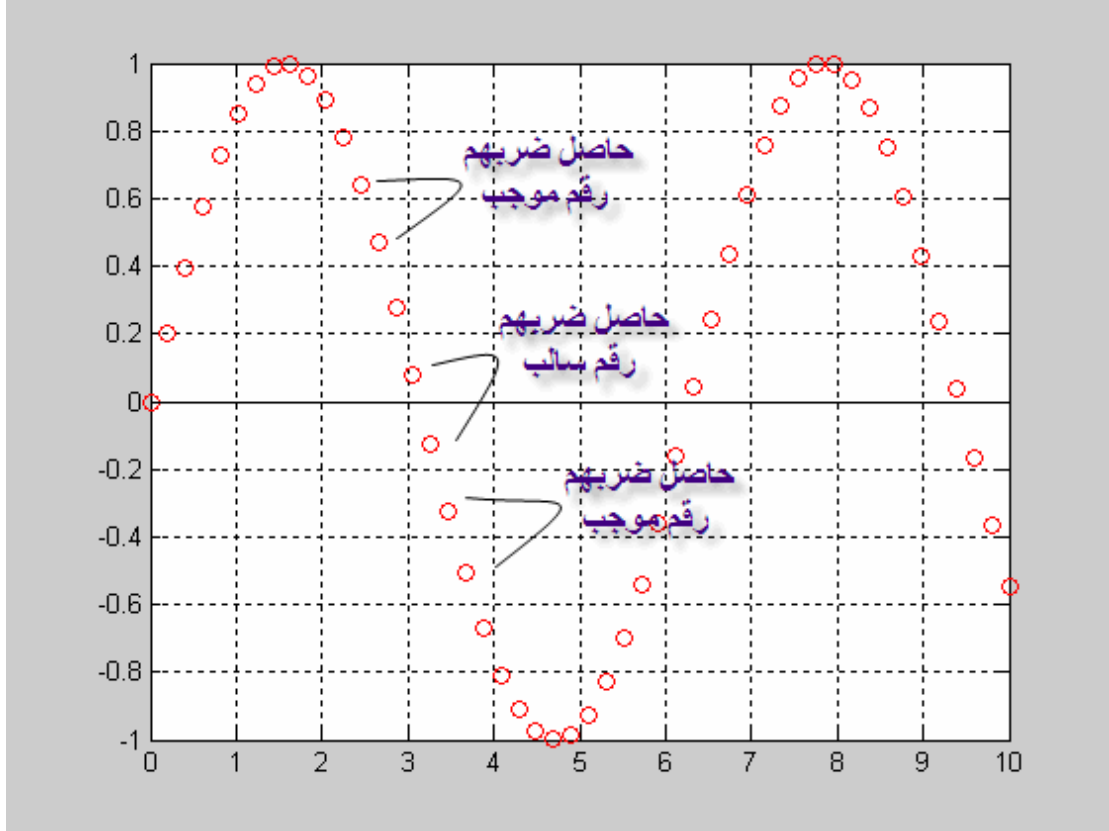
إظهار الدالة في صورة Digital Domain

script Ln 10 Col 5 OVR

وستظهر الرسمة في الصورة التالية



والآن دعونا نناقش حلاً لهذه المشكلة، عند ضرب نقطة في النقطة التي بعدها سنحصل على رقم، ولكن الفكرة أن الرقم قد يكون موجباً في حالة أن النقطتان فوق محور السينات، بينما إذا كانت إحدى النقط فوق محور السينات والأخرى تحت محور السينات فإن الناتج يكون سالباً، أما إذا كان كلا النقطتين أسفل محور السينات فإن ناتج ضرب النقطتين يكون موجباً، أي أن إذا ظهر رقم سالب فهذا يعني نقطة تقاطع، ثم يتم وضع دائرة حمراء على تلك النقطة، شاهد الصورة التالية



كل ما علينا هو ضرب كل نقطة في النقطة التي تليها، وذلك عن طريق تعريف y ثم تعريف نفس الدالة، ولكن متقدمة عنها برقم واحد، ثم ضربهم معاً، فمثلاً إذا كانت قيمة y كالآتي

$$[1 \ 2 \ 3 \ 4]$$

ثم سنقوم بإضافة رقم ليتم تحريك تلك القيم، ويفضل أن يكون الرقم صفراً، كما سيتم حذف الرقم الأخير حيث أنها ستكون أطول بعدد واحد فقط من الدالة الأصلية مما سيترتب عليه خطأ داخل الماتلاب وبالتالي فإن الدالة الجديدة ستكون متأخرة برقمي

$$[0 \ 1 \ 2 \ 3 \ 4]$$

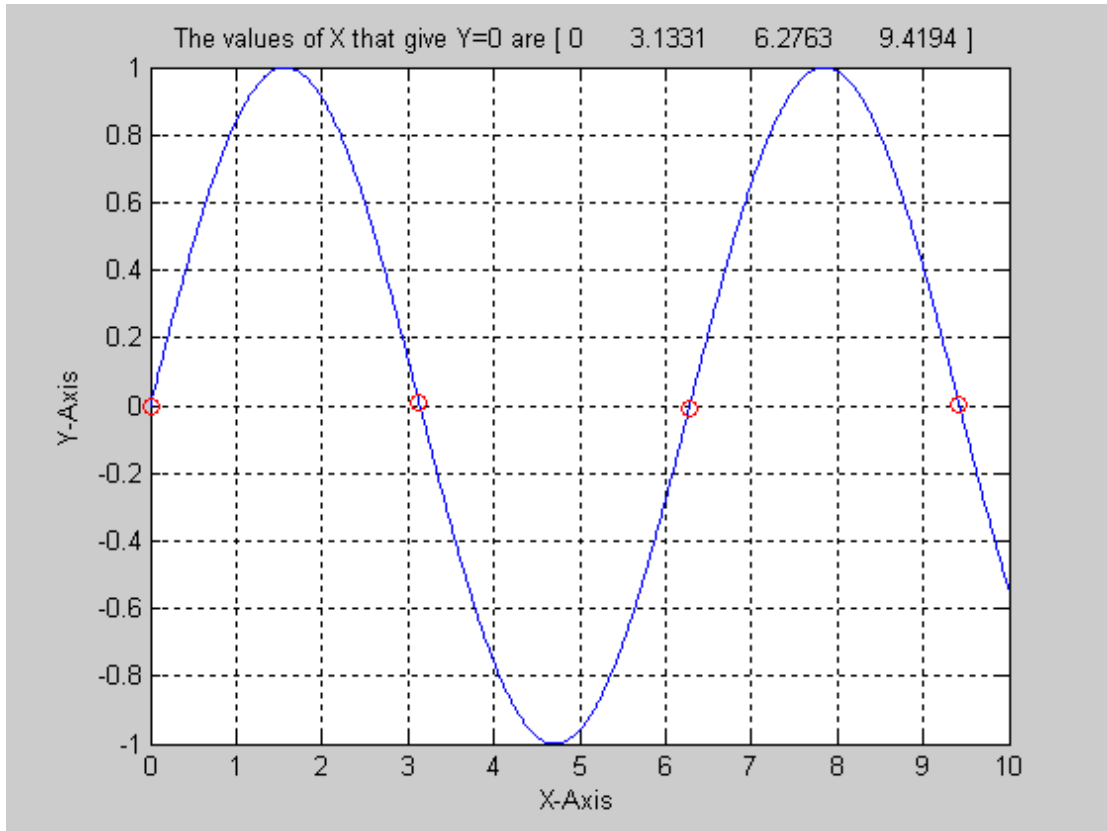
والآن سنقوم بكتابة البرنامج بالتدرج التالي

```

C:\Program Files\MATLAB\R2006a\work\zero_crossing3.m
File Edit Text Go Cell Tools Debug Desktop Window Help
[Icons] B...
- 1.0 + ÷ 1.1 x %>% %>% !
1 - clc
2 - clear
3 - close all
4 - x=linspace(0,10,1000); % By defining the inputs
5 - y=sin(x); % By defining the function
6 - ys=[ 0, y(1:end-1) ]; % By making the shifted backward function
7 - ym=y.*ys; % By multiplying both functions together
8 - ind=find(ym<=0); % بالبحث عن الأرقام السالبة بعد عملية الضرب
9 - x_cros=x(ind); % سنقوم بالبحث عن قيم X المناظرة لتلك الأرقام السالبة
10 - y_cros=y(ind); % إيجاد أماكن قيم Y والتي تساوي صفراً
11 - plot(x,y,x_cros,y_cros,'ro'); % عملية رسم الدالة
12 - str=['The values of X that give Y=0 are', ' ',num2str(x_cros),' '];
13 - title(str);
14 - grid;
15 - xlabel('X-Axis');
16 - ylabel('Y-Axis');
17
script Ln 13 Col 12 OVR

```

وبالتالي ستلاحظ الرسم التالي



وبالتالي تكون الرسمة قد أصبحت صحيحة
وبهذا ينتهي التطبيق الثاني

إيجاد المساحة تحت المنحنى

هذا المثال من التطبيقات الهامة، حيث سنقوم بتعريف المدخلات ورسم الدالة، ثم سنختار نقطتان نقطتان من على الرسم، ثم سنقوم بإيجاد المساحة بين تلك النقطتين، ونقوم بتظليل الجزء المختار، ولكن سنقوم في هذا المثال باستخدام أمرين جديدين وهما

trapez لإيجاد المساحة تحت المنحنى

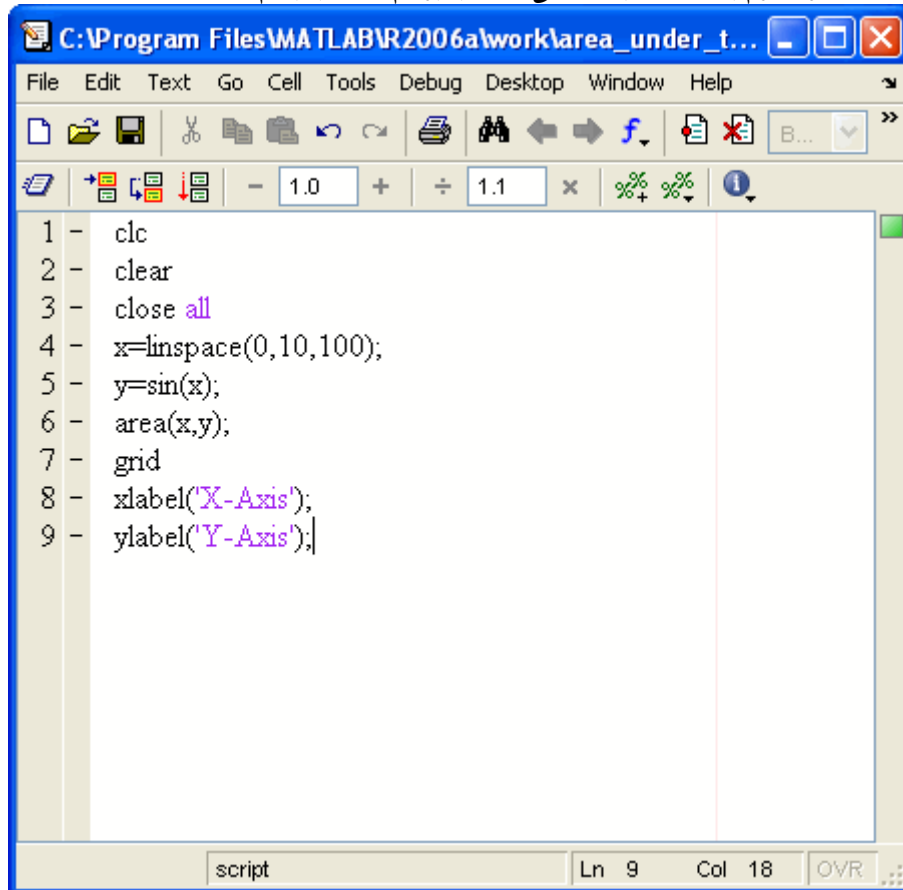
area لتظليل تلك المساحة من الدالة

وسنقوم بشرح الأمر **area**

حيث يأخذ الصورة التالية

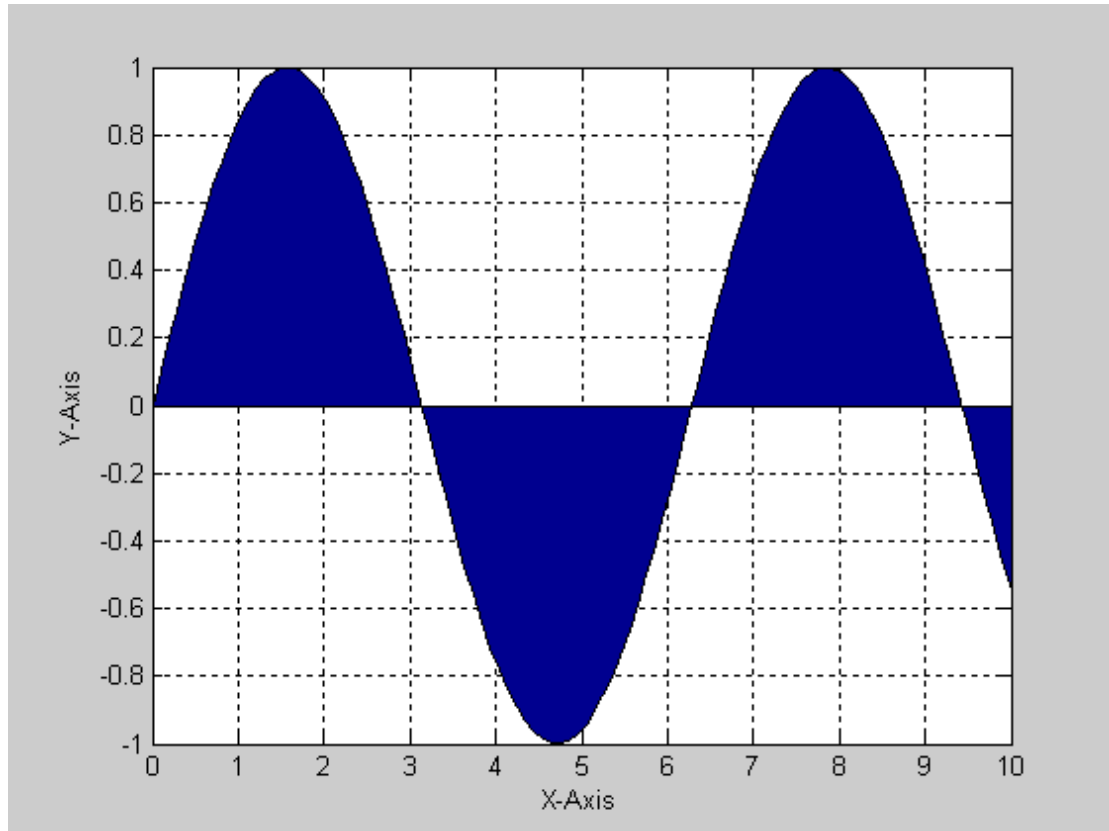
`area(x, y)`

وسنقوم بتنفيذ مثال بسيط على الماتلاب برسم دالة الجيب ثم تظليل تلك الدالة



```
1 - clc
2 - clear
3 - close all
4 - x=linspace(0,10,100);
5 - y=sin(x);
6 - area(x,y);
7 - grid
8 - xlabel('X-Axis');
9 - ylabel('Y-Axis');
```

وستظهر الرسمة كالتالي



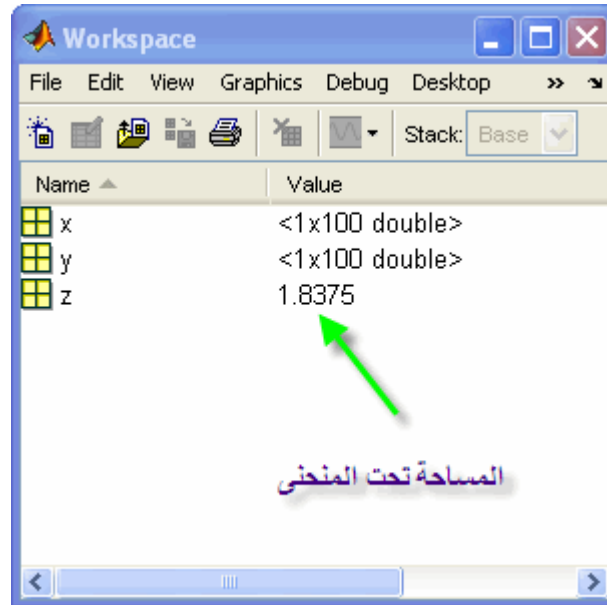
أما بخصوص الأمر **trapz** فيستخدم في إيجاد المساحة تحت المنحنى، حيث يأخذ الصورة التالية
 $\text{trapz}(x, y)$
 وسنقوم بعمل برنامج بسيط في إيجاد المساحة تحت منحنى دالة الجيب

```

C:\Program Files\MATLAB\R2006a\workla...
File Edit Text Go Cell Tools Debug Desktop
[Icons] B...
1.0 1.1 x % %
1 - clc
2 - clear
3 - close all
4 - x=linspace(0,10,100);
5 - y=sin(x);
6 - z=trapz(x,y);
7
script Ln 7 Col 1 OVR

```

وبالتالي ستلاحظ قيمة المساحة من خلال نافذة **Workspace**



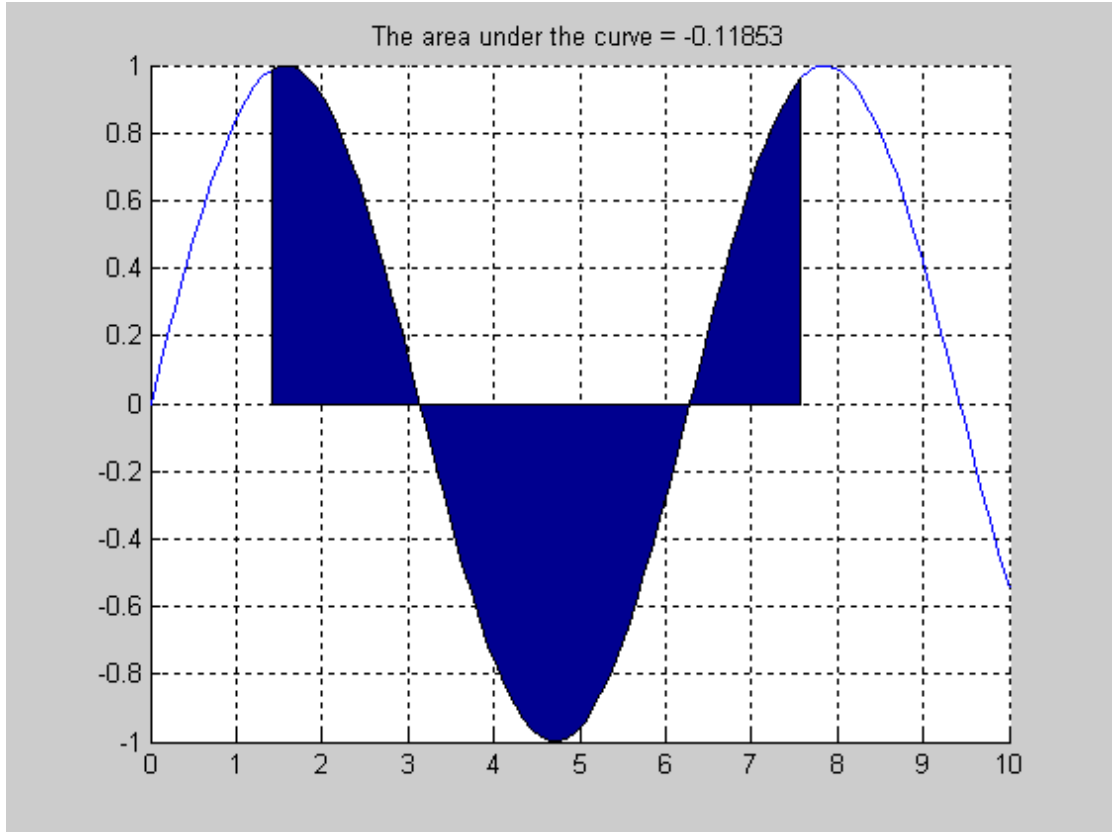
الآن نتوجه إلى البرنامج الذي نريد تنفيذه، سنقوم بإدخال المدخلات **inputs** ثم سنقوم بالتعويض بها في المخرجات **outputs** ثم سنقوم باختيار النقطتان من على الرسم، ثم سنوجد المساحة تحت المنحنى بين تلك النقطتين، ثم سنقوم بتظليل المساحة بين النقطتين.

```

1 - clc
2 - clear
3 - close all
4 - hold on
5 - x=linspace(0,10,100);           % By defining the inputs
6 - y=sin(x);                     % the function to be drawn
7 - plot(x,y)                     % By plotting the function
8 - grid;
9 - [xp yp]=ginput(2);            % By selecting two points from the drawing
10 - ind=find(x>=xp(1) & x<=xp(2)); % By selecting the region in between the two points
11 - x=x(ind);                    % By finding the values of x at that region
12 - y=y(ind);                    % By finding the values of y at that region
13 - area(x,y);                  % By shading that region
14 - z=trapz(x,y);               % By getting the area under the curve
15 - str=['The area under the curve = ',num2str(z)];
16 - title(str);

```

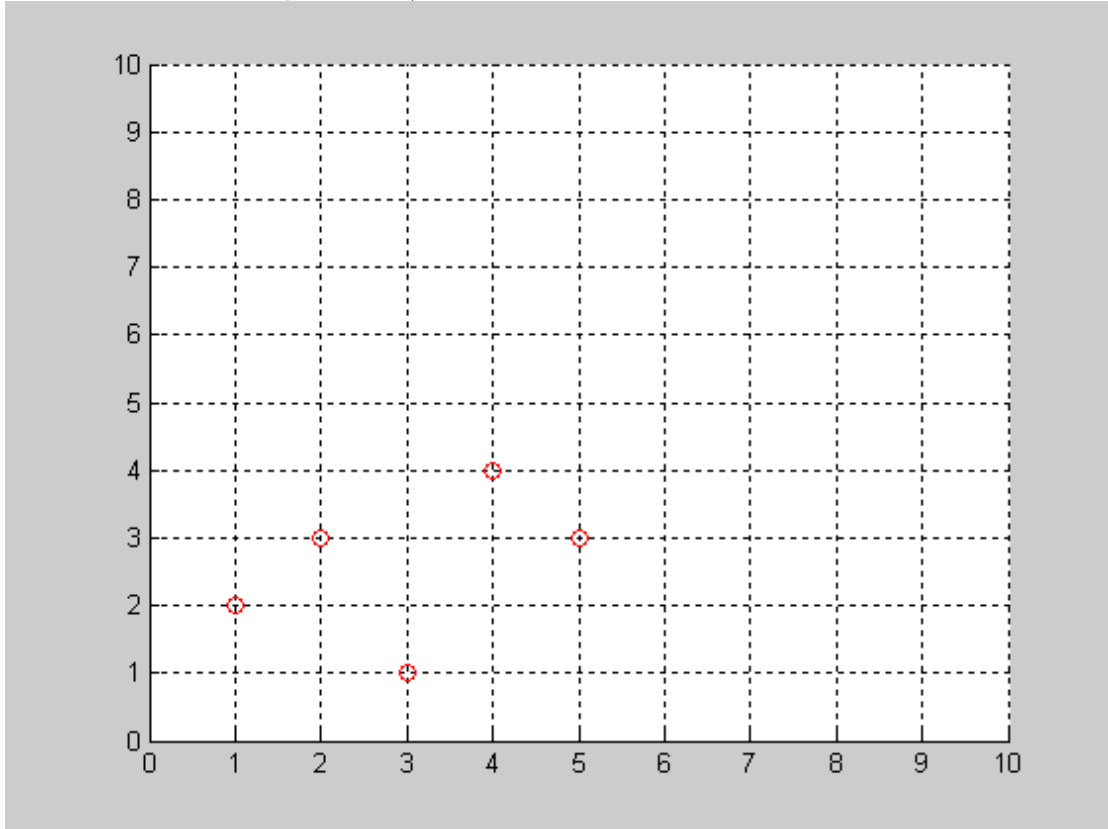
وتم إختيار نقطتان عشوائيتان، وظهرت الرسم كالتالي



وبهذا يكون التطبيق الثالث قد تم بنجاح
التطبيق الرابع والأخير في هذه الدورة سيتم تنفيذه المشاركة القادمة

Curve Fitting

سنقوم الآن بأخذ المثال التطبيقي الثالث والأخير وهو **Curve Fitting** حيث أن هذه العملية هامة جداً في إيجاد علاقة مكافئة لأي نظام، فمثلاً عند إدخال مجموعة من المدخلات **inputs** سنلاحظ أن الخرج **outputs** تأخذ مجموعة من النقاط المشتتة التي لا تجمعها علاقة محددة، أما عند استخدام **curve fitting** سنلاحظ تكون علاقة تقريبية لتوصيف النظام.
وهذه صورة لمجموعة من النقاط الخارجة من النظام لا تجمعها أي علاقة



ولكن سنقوم بشرح أمرين وهما **ones** و **zeros** واللذان لهما القدرة التالية
ones يستطيع أن يكون مصفوفة أو متجه جميع عناصره ١
zeros يستطيع أن يكون مصفوفة أو متجه جميع عناصره صفر
لاحظ الصورة التالية في طريقة كتابة كلا الأمرين

```
C:\Program Files\MATLAB\R2006a\work\ze...
File Edit Text Go Cell Tools Debug Desktop Window >>
[Icons] B... >>
[Icons] 1.0 + ÷ 1.1 x % % 1
1 - clc
2 - clear
3 - close all
4 - x=ones(6,6);
5 - y=zeros(6,6);
6 - |
script Ln 6 Col 1 OVR
```

وستلاحظ ظهور النتائج بالشكل التالي

```
Command Window
File Edit Debug Desktop Window Help

x =

1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1

y =

0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0

>> |
```

أما الآن سنتكلم عن أنواع Curve Fitting
هنالك أنواع عديدة منها

1- linear

2- Quadratic

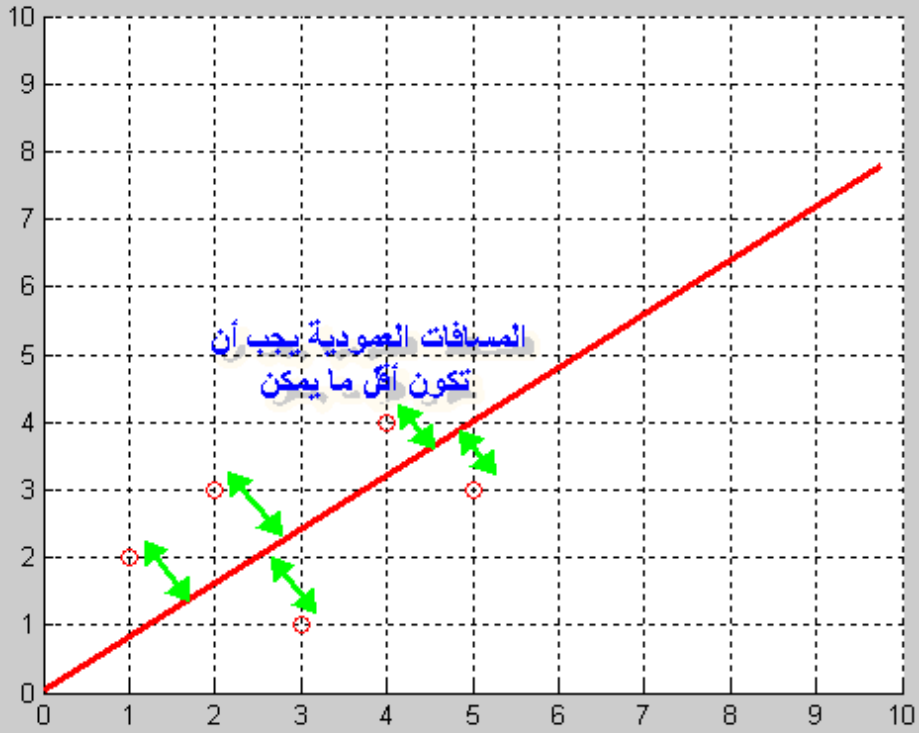
3- Sinusoidal

4- exponential

وسنتناول النوع الأول والرابع، أما الآن سنتناول النوع الأول

Linear Curve Fitting

في هذا النظام يتم إيجاد خط مستقيم بحيث تكون المسافة العمودية بين كل نقطة والخط المستقيم أقل ما يمكن، يمكن مشاهدة الصورة التالية



فكما هو واضح في المثال كل قيمة في محور السينات لها قيمة مناظرة في محور الصادات
 وحيث اننا نستخدم طريقة **Linear Curve Fitting** فإن لكل نقطة على محور الصادات علاقة خطية مع نقطة
 محددة على محور السينات، وهذه العلاقة تكتب في الصورة التالية

$$Y=KX+T$$

Where K & T are constant

فإذا عدنا بالذاكرة للخلف عند حل المعادلات سنجد اننا كنا نقوم بكتابة المعادلات بالشكل التالي

$$AX+BY=C$$

Where

A, B & C are constant

ويمكننا كما تعلمنا كتابة تلك المعادلة في الصورة التالية

$$\begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} = [C]$$

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} A & B \end{bmatrix}^{-1} [C]$$

وبالرجوع إلى المعادلة الخاصة بـ **Linear Curve Fitting** نستطيع كتابتها في الصورة التالية

$$[Y] = [X \quad 1] \begin{bmatrix} K \\ T \end{bmatrix}$$

$$\begin{bmatrix} K \\ T \end{bmatrix} = [X \quad 1]^{-1} [Y]$$

وبهذا نكون قد حصلنا على قيمة كلاً من K & T والتي نستطيع أن نقوم بتعريف مجموعة قيم للمتغير X وبالتالي نقوم

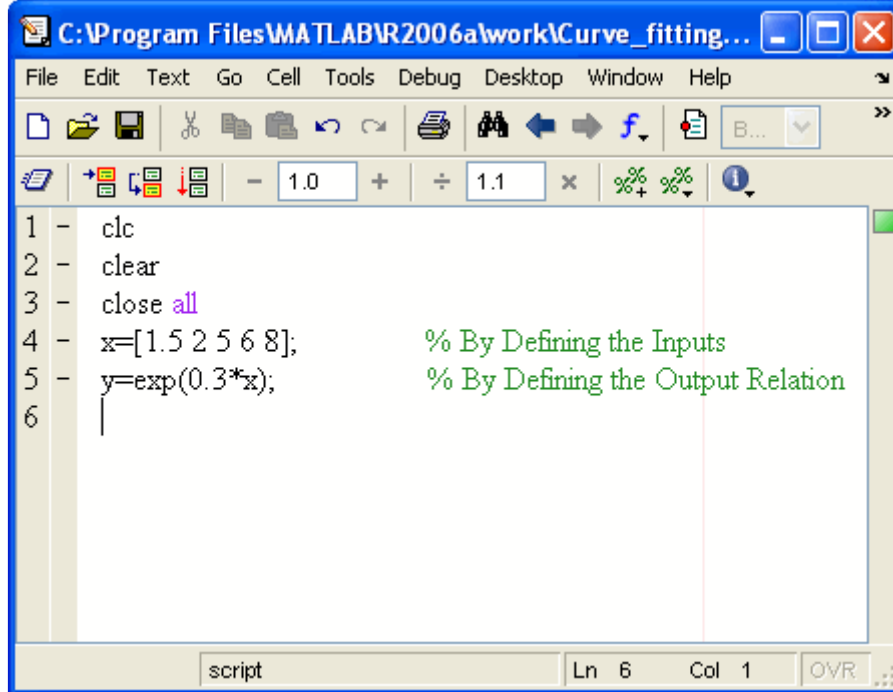
بالحصول على قيمة Y ومنها نقوم برسم العلاقة بين X & Y والتي تمثل خطأ تبعاً للمعادلة التالية

$$Y=KX+T$$

Where K & T are constant

والآن سنقوم بالبدء بكتابة البرنامج في الماتلاب خطوة خطوة

سنقوم الآن بتعريف الماتلاب بمجموعة القيم للمتغير X والعلاقة للنظام التي تعطينا قيمة Y



```
C:\Program Files\MATLAB\R2006a\work\Curve_fitting...
File Edit Text Go Cell Tools Debug Desktop Window Help
[Icons] B... >>
- 1.0 + ÷ 1.1 x % % %
1 - clc
2 - clear
3 - close all
4 - x=[1.5 2 5 6 8];           % By Defining the Inputs
5 - y=exp(0.3*x);           % By Defining the Output Relation
6 |
script Ln 6 Col 1 OVR
```

والآن لنفترض أن لدينا أكثر من قيمة X وبالتالي سنحصل على أكثر من قيمة Y وحيث أن العلاقة بين X & Y خطية كما ذكرنا مسبقاً فإننا بالتالي سيكون لدينا أكثر من معادلة يمكن كتابتها في الصورة التالية

$$Y_1 = K X_1 + T$$

$$Y_2 = K X_2 + T$$

-
-
-

$$Y_n = K X_n + T$$

والتي يمكن وضعها في الشكل التالي

$$\begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ \bullet \\ Y_n \end{bmatrix} = \begin{bmatrix} X_1 & 1 \\ X_2 & 1 \\ X_3 & 1 \\ \bullet & 1 \\ Y_n & 1 \end{bmatrix} \begin{bmatrix} K \\ T \end{bmatrix}$$

$$\begin{bmatrix} K \\ T \end{bmatrix} = \begin{bmatrix} X_1 & 1 \\ X_2 & 1 \\ X_3 & 1 \\ \bullet & 1 \\ Y_n & 1 \end{bmatrix}^{-1} \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ \bullet \\ Y_n \end{bmatrix}$$

وسنقوم في الماتلاب بتحويل متجه الصف **Row Vector** إلى متجه عمودي **Column Vector** ثم إضافة متجه عمودي جميع قيمه واحد باستخدام الأمر **ones** كما تعلمنا مسبقاً

```

1 - clc
2 - clear
3 - close all
4 - x=[1.5 2 5 6 8];           % By Defining the Inputs
5 - y=exp(0.3*x);           % By Defining the Output Relation
6 - x=x';                   % To convert the Row Vector to Column Vector
7 - y=y';                   % To convert The Row Vector To Column Vector
8 - A=[x, ones(length(x),1)];
9 -

```

والآن قد يظن البعض أنه حتى نحصل على قيم **K & T** يجب أن تكون صورة الحل كالآتي

$$AX = B$$

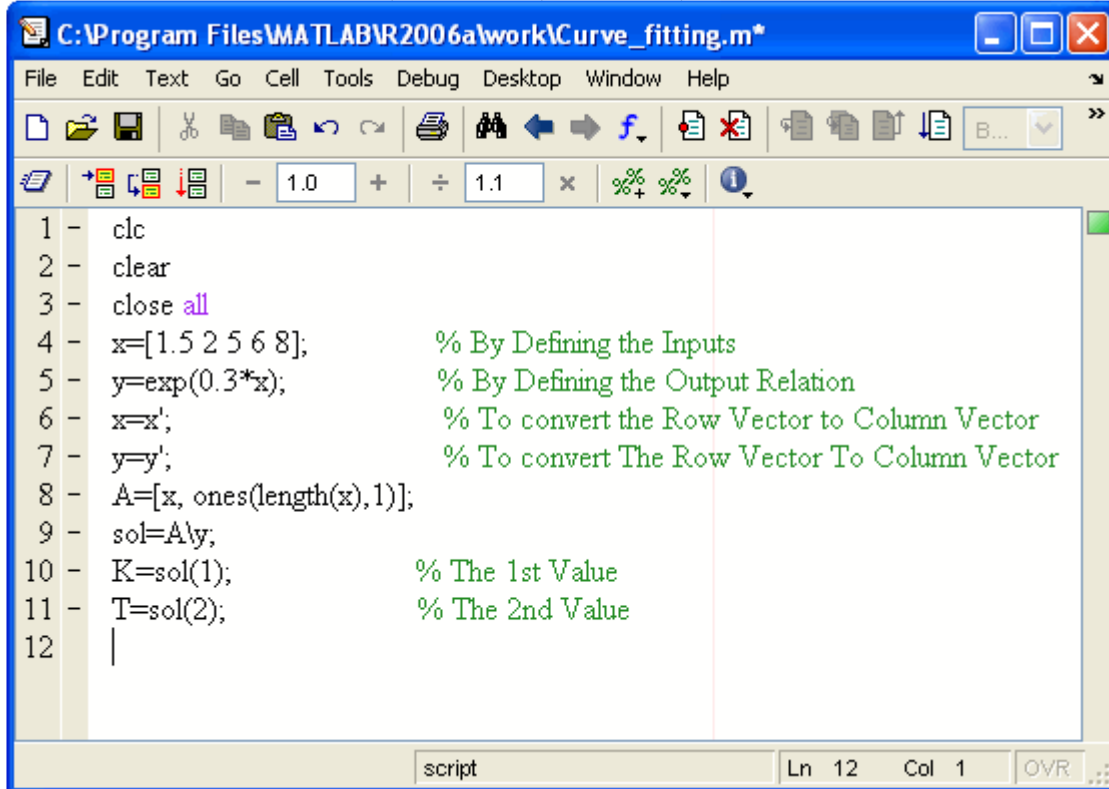
$$X = inv(A) \times B$$

ولكن هذا صحيح إذا كانت قيمة **A** مصفوفة مربعة ، فهل هي كذلك الآن ؟ بالطبع لا ، فما العمل إذا كانت المصفوفة ليست مربعة يتم وضع علامة القسمة مقلوبة (\) ولا يتم استخدام الأمر **inv** أي أن صورة الحل الصحيح تكون

$$AX = B$$

$$X = A \setminus B$$

وبالتالي يكون الحل في الماتلاب كالآتي



```
C:\Program Files\MATLAB\R2006a\work\Curve_fitting.m*
File Edit Text Go Cell Tools Debug Desktop Window Help
+ 1.0 1.1 x % %
1 - clc
2 - clear
3 - close all
4 - x=[1.5 2 5 6 8]; % By Defining the Inputs
5 - y=exp(0.3*x); % By Defining the Output Relation
6 - x=x'; % To convert the Row Vector to Column Vector
7 - y=y'; % To convert The Row Vector To Column Vector
8 - A=[x, ones(length(x),1)];
9 - sol=A\y;
10 - K=sol(1); % The 1st Value
11 - T=sol(2); % The 2nd Value
12 |
script Ln 12 Col 1 OVR
```

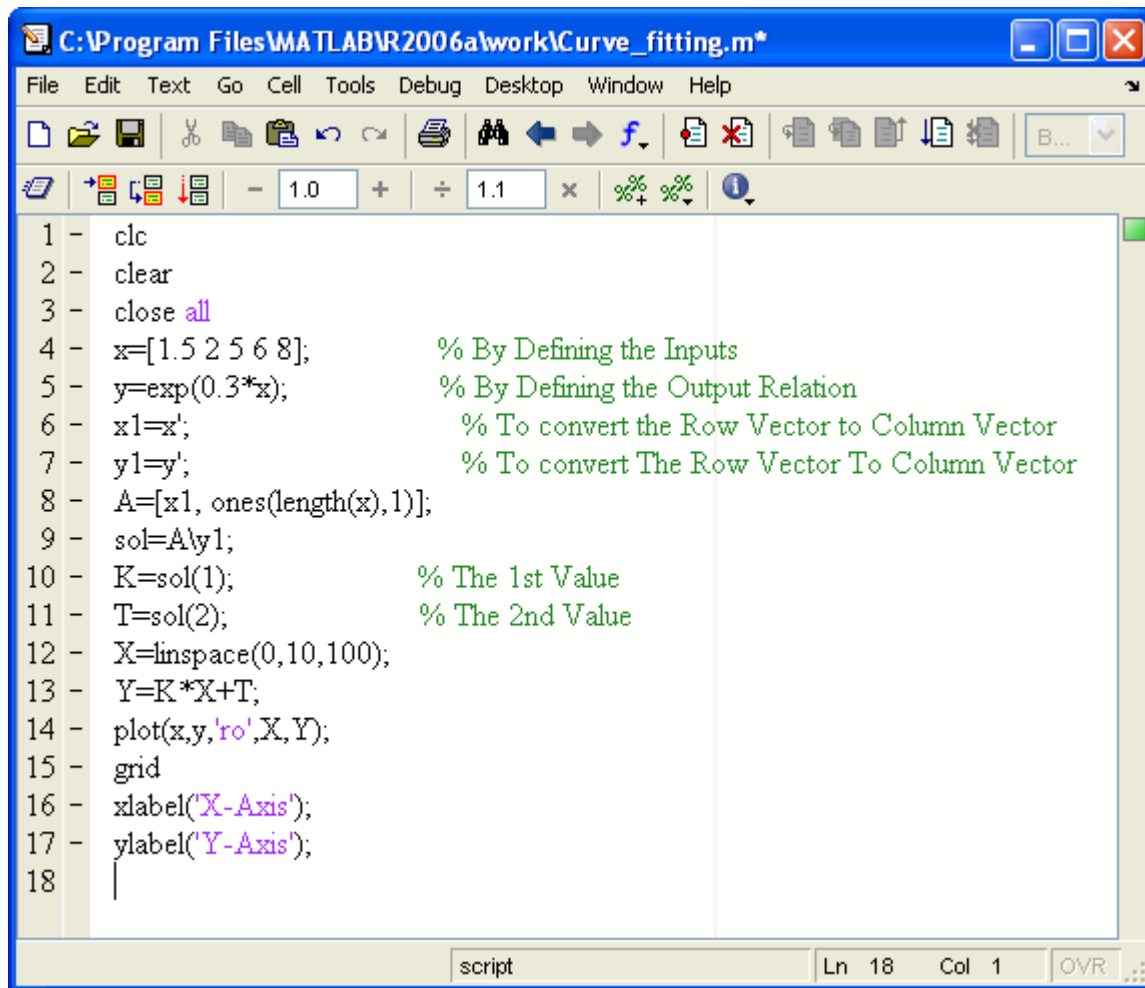
وبالتالي فإن المعادلة الناتجة والتي من خلالها سنرسم خطأ بحيث تكون المسافة العمودية بينه وبين النقاط أقل ما يمكن، تكون في الصورة التالية

$$Y=KX+T$$

Where K & T are constant

والآن سنقوم بتعريف الماتلاب عدة نقاط بحيث نرسم ذلك الخط

!Error

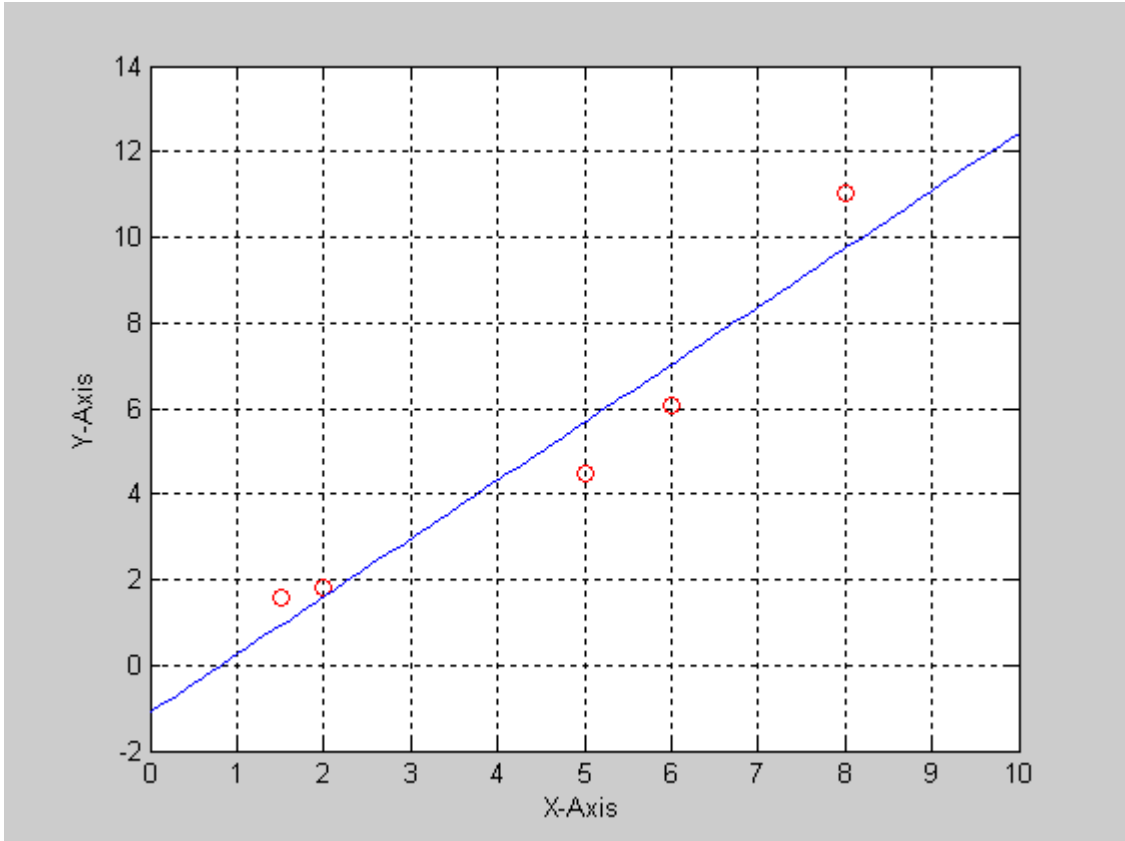


The image shows a screenshot of a MATLAB script editor window. The title bar reads "C:\Program Files\MATLAB\R2006a\work\Curve_fitting.m*". The menu bar includes "File", "Edit", "Text", "Go", "Cell", "Tools", "Debug", "Desktop", "Window", and "Help". The toolbar contains various icons for file operations and editing. The script editor shows the following code:

```
1 - clc
2 - clear
3 - close all
4 - x=[1.5 2 5 6 8];           % By Defining the Inputs
5 - y=exp(0.3*x);           % By Defining the Output Relation
6 - x1=x';                  % To convert the Row Vector to Column Vector
7 - y1=y';                  % To convert The Row Vector To Column Vector
8 - A=[x1, ones(length(x),1)];
9 - sol=A\y1;
10 - K=sol(1);              % The 1st Value
11 - T=sol(2);             % The 2nd Value
12 - X=linspace(0,10,100);
13 - Y=K*X+T;
14 - plot(x,y,'ro',X,Y);
15 - grid
16 - xlabel('X-Axis');
17 - ylabel('Y-Axis');
18 - |
```

The status bar at the bottom indicates "script", "Ln 18", "Col 1", and "OVR".

وبالتالي نحصل على الرسم بالشكل التالي



وننتقل إلى التطبيق الذي يليه وهو
Exponential Curve Fitting

Exponential Curve Fitting

سنأخذ الآن التطبيق الأخير في هذه الدورة وهو , Exponential Curve Fitting ويمكن كتابة العلاقة بين X & Y بالشكل التالي

$$Y = K \times e^X + T$$

وإذا وجدت أكثر من نقطة، فهذا يعني وجود أكثر من معادلة والتي تكتب في الصورة التالية

$$Y_1 = K \times e^{X_1} + T$$

$$Y_2 = K \times e^{X_2} + T$$

•
•

$$Y_n = K \times e^{X_n} + T$$

والتي يمكن كتابتها في صورة المصفوفة

$$\begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ \bullet \\ Y_n \end{bmatrix} = \begin{bmatrix} e^{X_1} & 1 \\ e^{X_2} & 1 \\ e^{X_3} & 1 \\ \bullet & 1 \\ e^{X_n} & 1 \end{bmatrix} \begin{bmatrix} K \\ T \end{bmatrix}$$

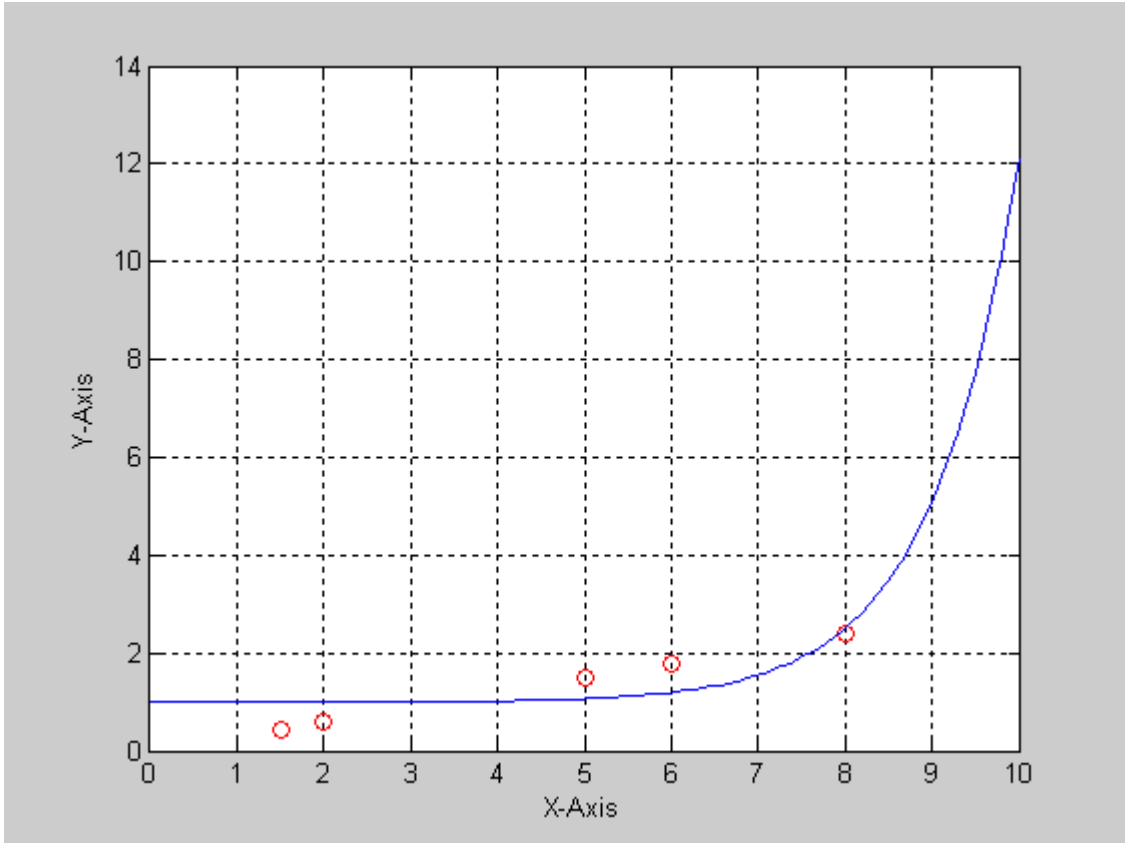
$$\begin{bmatrix} K \\ T \end{bmatrix} = \begin{bmatrix} e^{X_1} & 1 \\ e^{X_2} & 1 \\ e^{X_3} & 1 \\ \bullet & 1 \\ e^{X_n} & 1 \end{bmatrix}^{-1} \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ \bullet \\ Y_n \end{bmatrix}$$

والآن سنقوم بوضع البرنامج على الماتلاب

!Error

```
1 - clc
2 - clear
3 - close all
4 - x=[1.5 2 5 6 8];           % By Defining the Inputs
5 - y=0.3*x;                  % By Defining the Output Relation
6 - x1=x';                   % To convert the Row Vector to Column Vector
7 - y1=y';                   % To convert The Row Vector To Column Vector
8 - A=[exp(x1), ones(length(x),1)];
9 - sol=A\y1;
10 - K=sol(1);                % The 1st Value
11 - T=sol(2);                % The 2nd Value
12 - X=linspace(0,10,100);
13 - Y=K*exp(X)+T;
14 - plot(x,y,'ro',X,Y);
15 - grid
16 - xlabel('X-Axis');
17 - ylabel('Y-Axis');
18
```

وبالتالي سيظهر الناتج كالتالي



هذا وتكون قد أنهينا دورة الماتلاب بفضل الله

الخاتمة

الحمد لله الذي أنعم علي ووفقتني في إنهاء هذه الدورة، وأسأل الله أن يجمعني مع أحبي
في مستقر رحمته، وأن يجعل ثمرة هذا العمل جنة الفردوس، وان يكرمنا بالنظر إلى
وجهة الكريم، إنه ولي ذلك والقادر عليه.
وتقبلوا خالص تحياتي وإحترامي
أخوكم



أحمد عفيفي



شكر خاص من موقع اليوسوليد

www.yousolid.com