تم تنزيل الكتب والبرامج و الفيديو من مكتبة فرحات

مكتبة فرحات الهندسية

www.farahat-library.com/blog

لتحميل أفضل افلام الكارتون الهادفة

مكتبة فرحات المرئية

www.farahat-library.com/cartoon

لتحميل البرامج المفيدة و المنتجه

مكتبة فرحات للبرمجيات

www.farahat-library.com/software

لبيع وشراء المنتجات الغير متوفرة في الاسواق

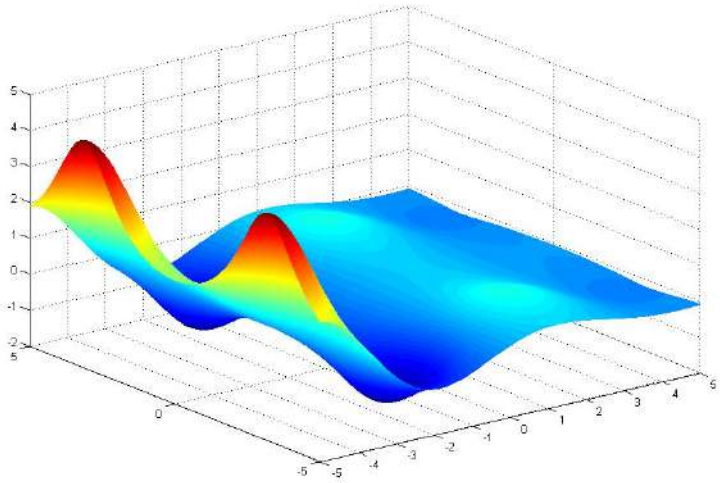
مول فرحات

www.farahat-library.com/mall

دورة الماتلاب خطوة بخطوة

المقدمة في
ملتقى المهندسين العرب

يقوم بشرح الدورة
المهندس أحـــمـــد عـــفـــيـــفـــي ســـلامـــة



جميع الحقوق محفوظة لدى ملتقى المهندسين العرب©
٢٠٠٦

أخواني الكرام توكلنا على الله في بدء شرح برنامج الماتلاب

سنتناول بإذن الله في مقدمة في الماتلاب التالي:

-1تعريف برنامج الماتلاب ومؤسسه
-2واجهة البرنامج
-3بعض الأوامر الأساسية
-4تعريف العمليات الأساسية
-5تعريف بعض المتغيرات
-6الأعداد المركبة
-7المصفوفات
-8الرسم في نظام الإحداثي الثنائي ٢D Plotting
-9الرسم في نظام الإحداثي الثلاثي الأبعاد ٣D Plotting

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

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

أولا: 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 Integrationو كذلك يقوم بحل المعادلات الجبرية Algebraic Equations وكذلك المعادلات التفاضلية Differential Equationsذات الرتب العليا والتي قد تصل من الصعوبة ما تصل، ليس فقط ذلك بل يستطيع البرنامج عمل التفاضل الجزئي، ويقوم بعمل عليات 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 محاكة 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 بإستبدال 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 إستاذ الرياضيات وعلوم الحاسب Computer Science لأكثر من عشرين عاماً في جامعة متشيجين و جامعة ستانفورد وجامعة نيو مكسيكو.
أمضى خمس سنوات عند إثنين من مصنعى الـ Hardware وهما Intel Hypercube organization و Ardent Computerقبل أن يقوم بالإنتقال 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 بكلوريوس الهندسة الكهربية وعلوم الحاسب من جامعة MIT عام ١٩٧٨
كما أنه حاصل على شهادة M.S.E.E من جامعة ستانفورد عام ١٩٨٠

جاك ليتل

يستكلم...

الحمد لله رب العالمين، والصلاة والسلام على خير المرسلين، سيدنا محمد عليه أفضل الصلاة والتسليم
أخواني نبدأ معاً تعليم الماتلاب خطوة بخطوة، والذي اسأل 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 بإستخدام الماتلاب Control System Using the Matlab
4- تطبيقات الأشارة بإستخدام الماتلاب Signal Application Using Matlab
5- عمليات الإشارة الرقمية بإستخدام الماتلاب Digital Signal Processing Using Matlab
6- النظريات الرياضية التقريبية بإستخدام الماتلاب Numerical Application Using Matlab
7- تطبيقات معالجة الصور بإستخدام الماتلاب Image Processing Applications Using Matlab
8- تطبيقات موجات الراديو بإستخدام الماتلاب Radio Frequency Applications Using Matlab
9- التطبيقات الميكانيكية بإستخدام الماتلاب Mechanical Applications Using Matlab
10- تطبيقات الرادار بإستخدام الماتلاب Radar Applications Using Matlab
11- تطبيقات الروبوت بإستخدام الماتلاب Robots Applications Using Matlab
12- التطبيقات الإلكترونية بإستخدام الماتلاب Electronics Applications Using Matlab
13- تطبيقات أشباه الموصلات بإستخدام الماتلاب Semiconductors Applications Using Matlab
14- التطبيقات المستخدمة في صناعة السيارات بإستخدام الماتلاب Automotive Applications Using Matlab
15- التطبيقات المستخدمة في علوم الفضاء والدفاع الجوي بإستخدام الماتلاب Aerospace and Defense Applications Using Matlab
16- تطبيقات الإتصالات بإستخدام 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 مواضيع لابد من أخذه قبل الشروع في التطبيق وهما

1- مقدمة في الماتلاب
2- مقدمة في المحاكاة
3- أنظمة التحكم بإستخدام 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

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

-1يتم شرح جزء معين يتم تحديده في الأسبوع

-2يتم تقديم الشرح في أول كل أسبوع حتى يتسنى للأعضاء الإستفسار بقدر المستطاع خلال فترة الأسبوع وكذلك الرد على إستفساراتهم

-3يجب التقيد بما سيتم شرحه من خلال ترتيب معين، حتى نصل إلى درجة عالية من الفهم والتطبيق للبرنامج.

-4ضرورة تحميل كتاب تعليم الماتلاب وكذلك كتاب المحاكاة بإستخدام الماتلاب **simulink** من خلال موضوع **مكتبة الميكاترونكس!**

-5تحميل برنامج الماتلاب، ويمكن للأعضاء تحميل البرنامج من خلال موضوع **برامج هندسة ميكاترونكس!**

-6يمكن للأعضاء المشاركة في شرح البرنامج، مع الإلتزام بالترتيب المقرر للشرح.

-7في حالة أن كان للعضو إستفسار حول موضوع قد سبق شرحه، فما عليه إلا أن يراسل المشرف حول الإستفسار، حتى يقوم المشرف بوضع إستفساره مع إستفسارات ذلك الجزء المشروح مسبقاً.

-8بالنسبة للأعضاء الجدد الذين تقل مشاركاتهم عن ٢٥ مشاركة لبدء الرسائل الخاصة، يمكنهم بعث إستفساراتهم على بريدي الإلكتروني، ولقد وضعت بريدي الإلكتروني في الملف الشخصي الخاص بي.

-9ضرورة توضيح مايتم شرحه بالصور إذا إقتدى الأمر ذلك.

-10يقم 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!** .

-11تقدم 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**.

-12تقدم 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

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/

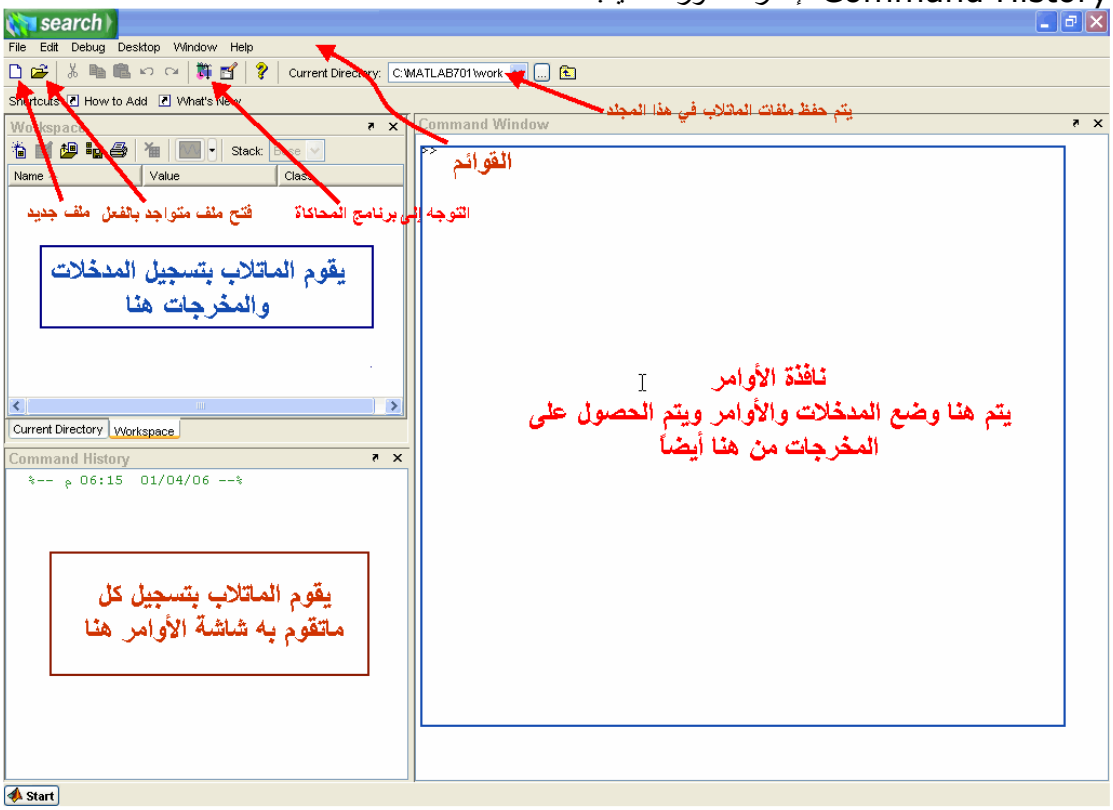
يقدم أيضاً موقع الجامعة لمحة سريعة حول برنامج الماتلاب

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

نسأل الله التوفيق

ثانياً: واجهة البرنامج

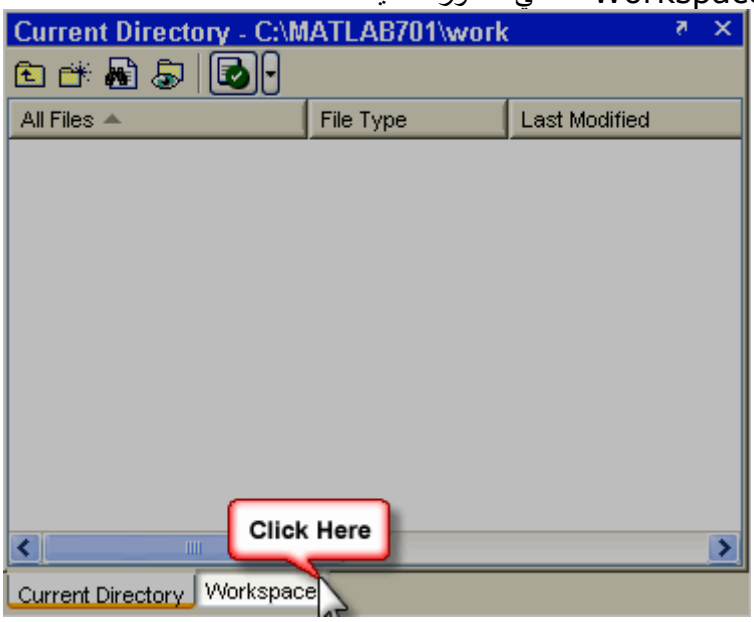
تتسم واجهة البرنامج بالسهولة في التعامل معها، حيث يتم تقسيم مناطق العمل بها إلى ثلاث مناطق رئيسية، وهي كالتالي نافذة الأوامر Command Window و منطقة العمل Workspace و تاريخ الأوامر Command History، إنظر الصورة التالية.



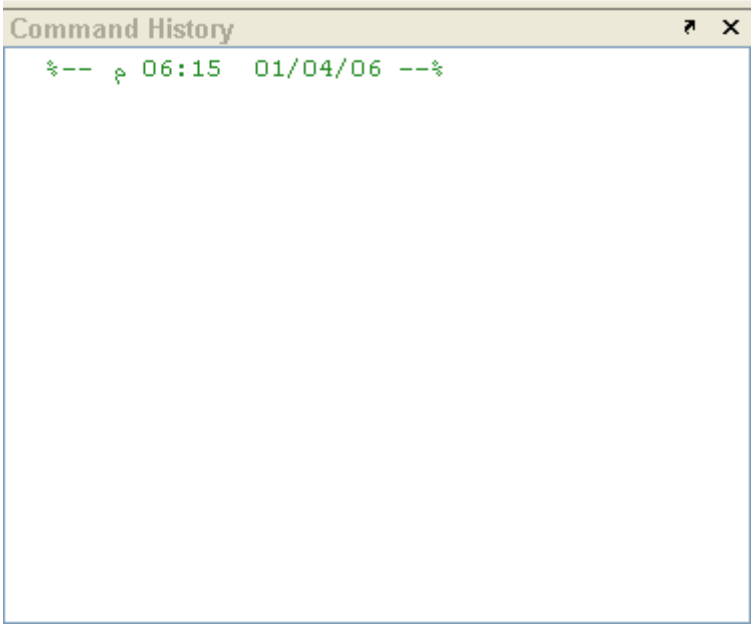
نافذة الأوامر Command Window: حيث يتم إدخال المدخلات Inputs والأوامر Commands، ويعمل الماتلاب على تحليل تلك البيانات ومدى مطابقة المدخلات للوظيفة المطلوبة منه، حتى تحصل على النتائج في نفس الشاشة.

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

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



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



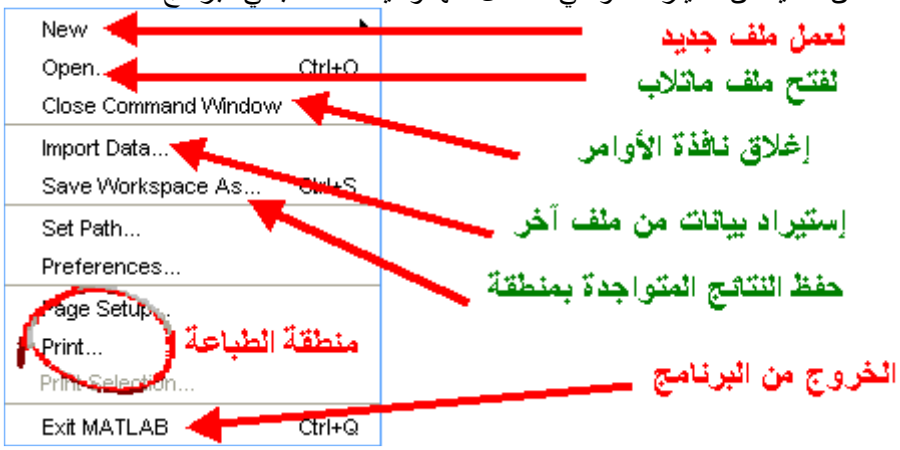
قائمة إبدأ Start: تستخدم هذه القائمة للوصول إلى التطبيق المراد تنفيذه، تستخدم هذه القائمة في المراحل المتقدمة في برنامج الماتلاب

صورة 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

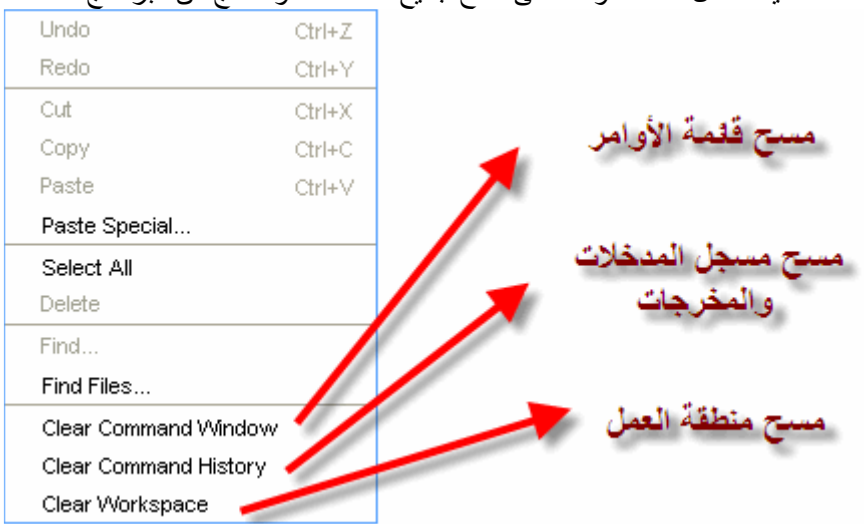


قائمة التعديل Edit

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

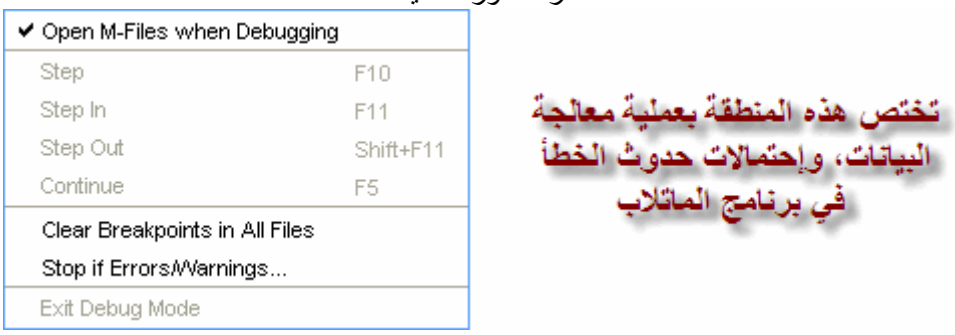
حيث تعمل تلك الأدوات على مسح جميع المدخلات والنتائج من البرنامج



قائمة Debug

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

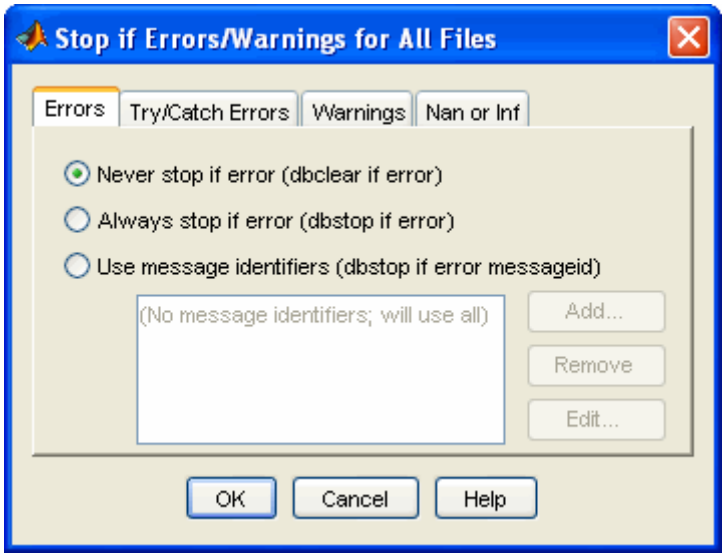
أنظر الصورة التالية



فمثلاً قم بإختيار Stop If Errors/Warnings...

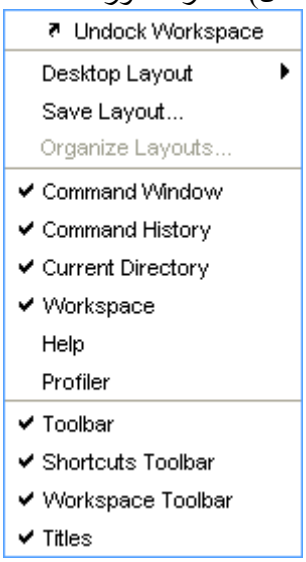
ستلاحظ ظهور نافذة، تعطيك حرية الإختيار في تصرف برنامج الماتلاب عند حدوث أخطاء أو تحذيرات

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



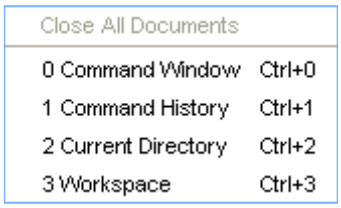
قائمة Desktop:

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

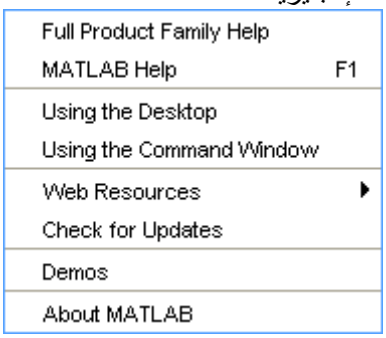


معلومة هامة:
تكون النوافذ في أحد الوضعين
١- Docked: حيث تكون النافذة غير قابلة للتحريك من مكانها.
٢- Undocked: حيث تكون النافذة قابلة للتحريك وتعديل مقاسها أيضاً

يتبقى لدينا قائمتان هما
قائمة Window:
حيث يمكنك التنقل بين ملفات الماتلاب المختلفة، وكذلك النوافذ مثل نافذة الأوامر Command Window وغيرها الكثير.



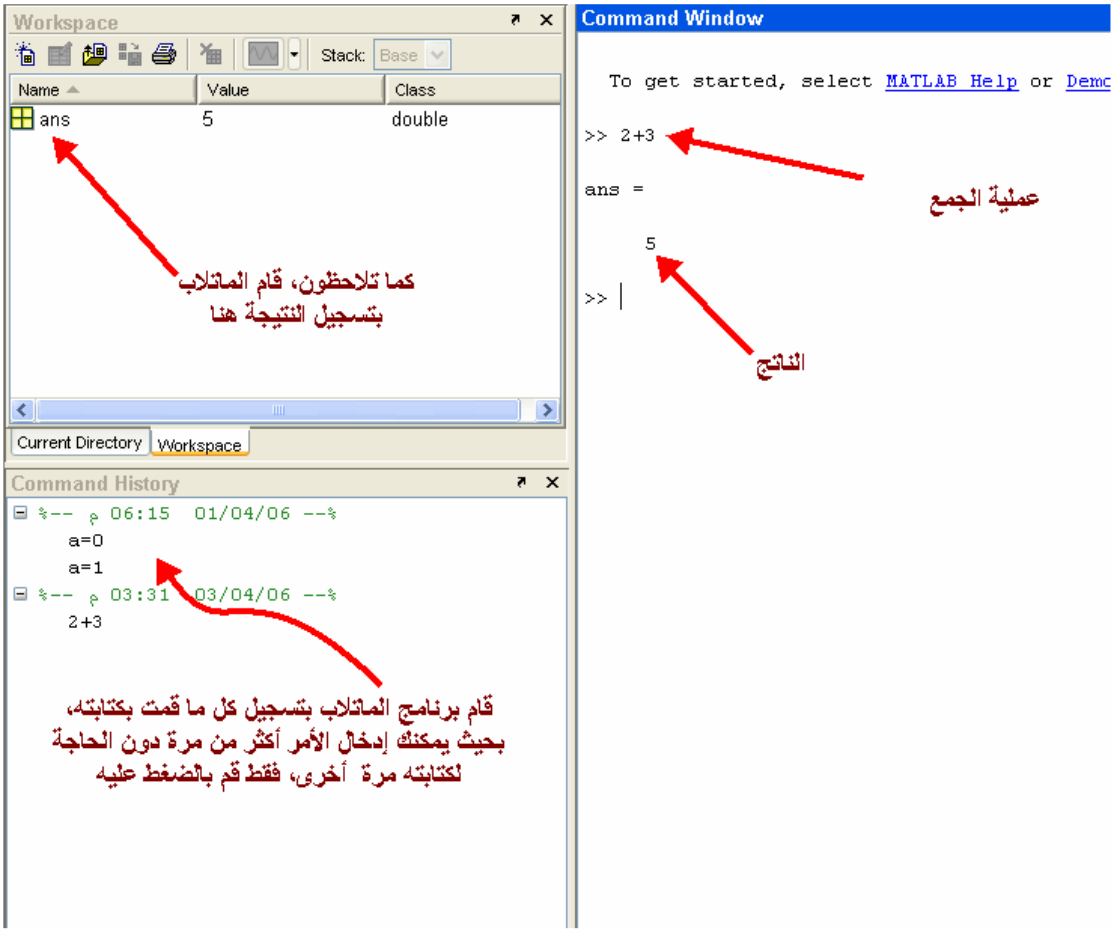
قائمة Help:
حيث تقوم تلك القائمة، بتوفير المساعدات الضرورية في البرنامج، ووسائل الإتصال بالشركة المصنعة، وآخر التحديثات، وكذلك تعلم الماتلاب باللغة الإنجليزية



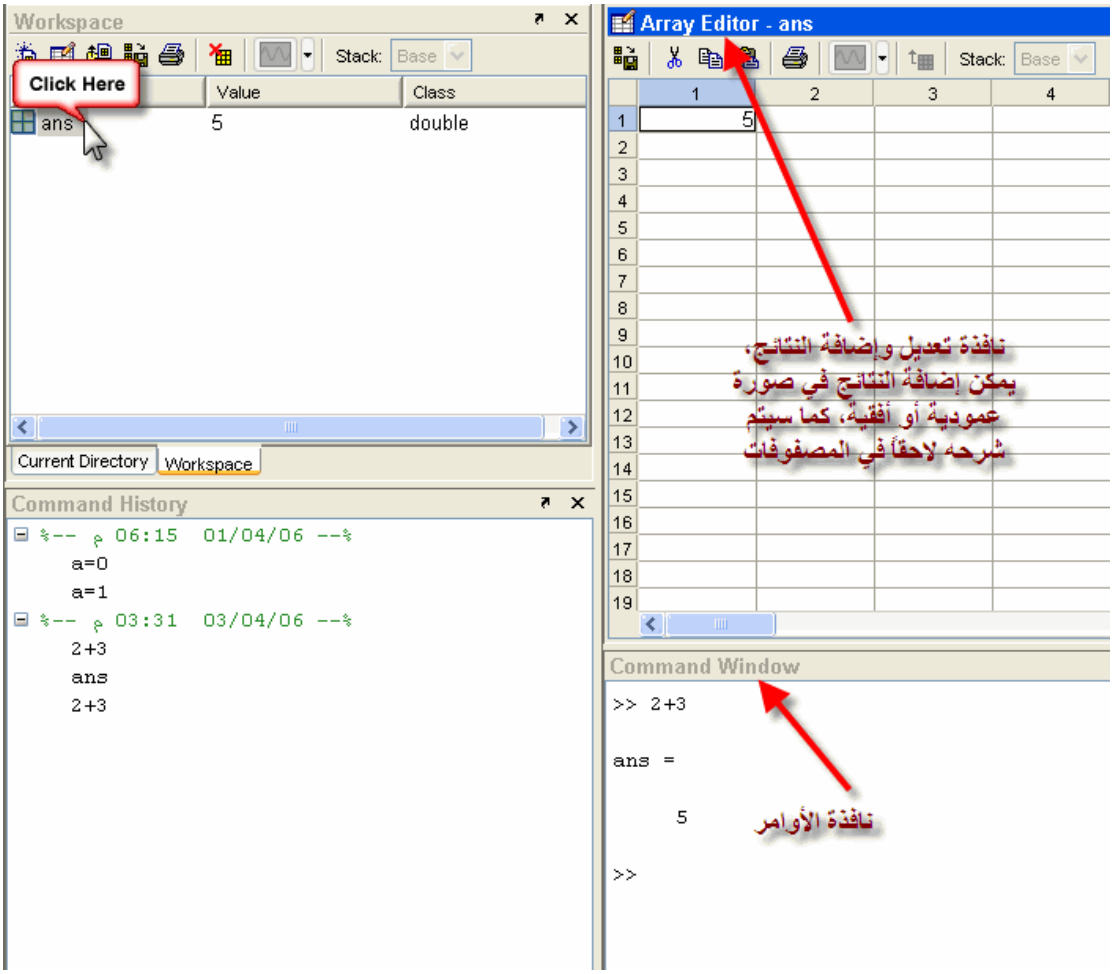
أخواني الكرام، نستكمل برنامج الماتلاب ونتناول اليوم بإذن الله العمليات الأساسية (الجمع و الطرح والضرب والقسمة) وبعض العمليات الهامة مثل وضع الأس لعدد، كما سنتعرف على بعض الأوامر الهامة.

عملية الجمع

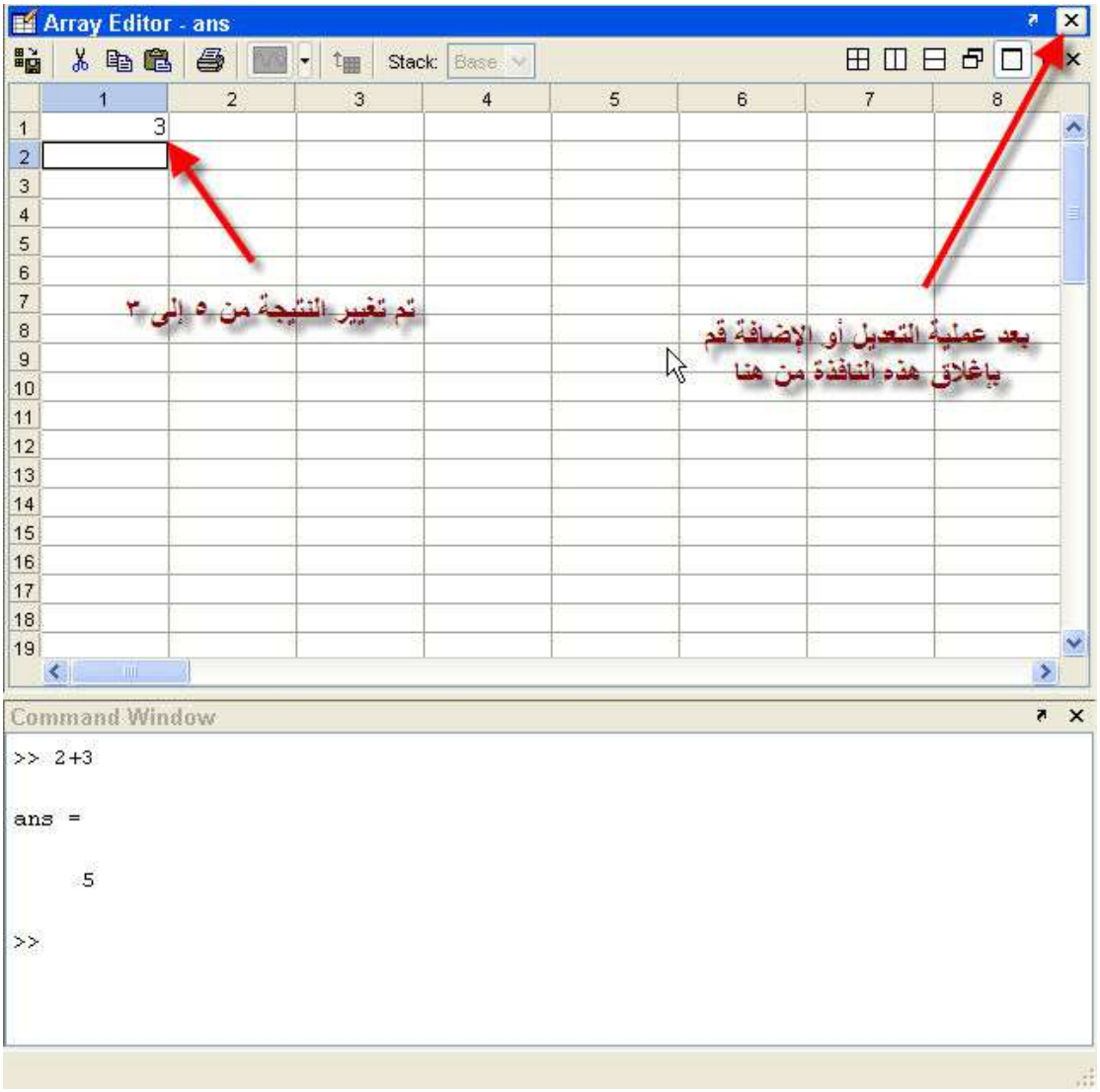
تأخذ علامة الجمع في الماتلاب الرمز المعروف للجمع وهو "+"
فمثلاً إذا قمنا بجمع ٣+٢ سيقوم الماتلاب بوضع الإجابة في صورة أرقام وهو ٥، أنظر الصورة التالية



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



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



ستلاحظ عودة نافذة الأوامر لوضعها الأساسي، قم بكتابة ans في نافذة الأوامر، ستلاحظ ظهور الناتج بالقيمة الجديدة وهي ٣، أنظر الصورة

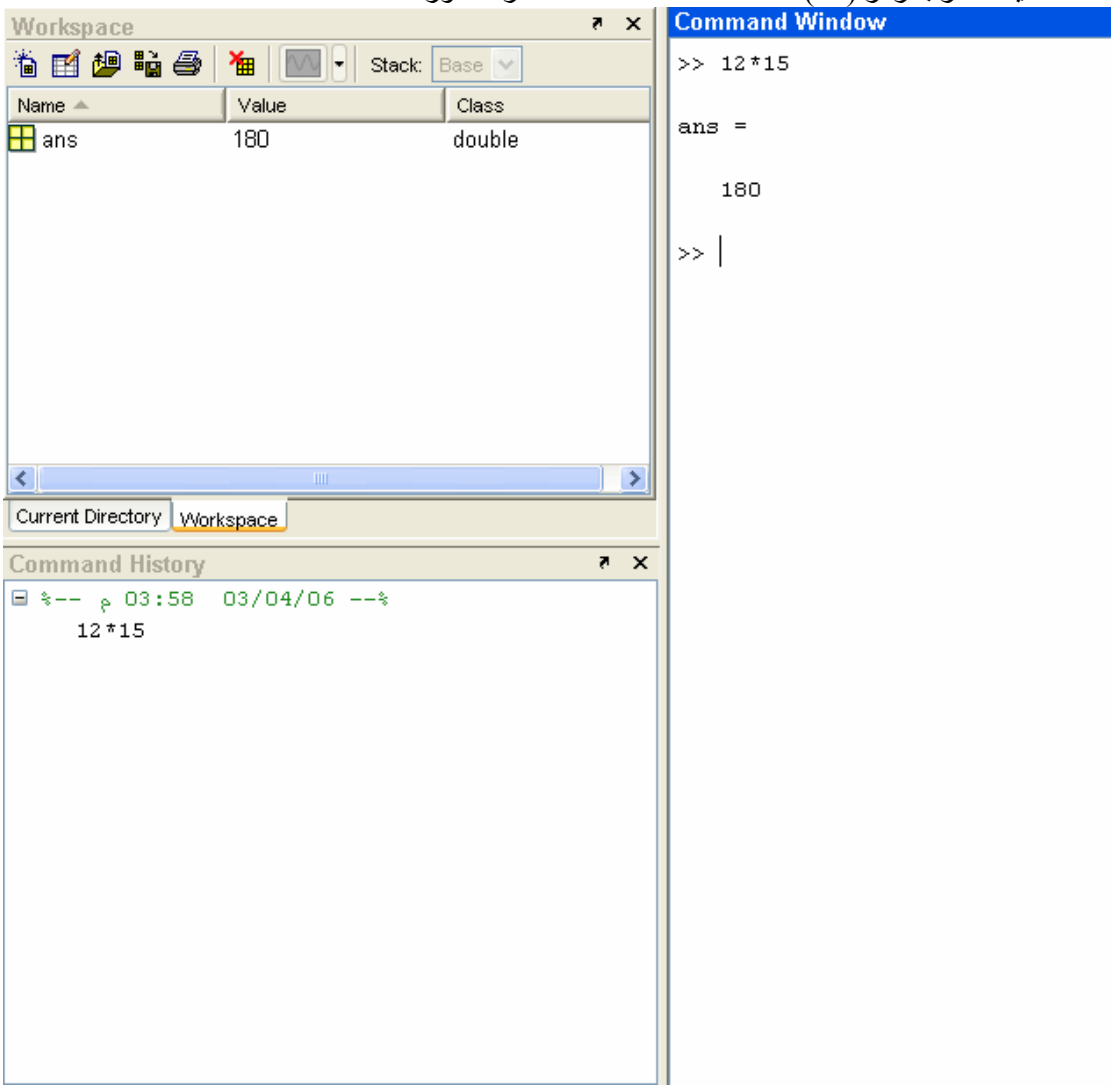
عملية الطرح:

تأخذ عملية الطرح رمز (-) في الماتلاب، فمثلاً ٣-٢=١، أنظر الصورة

```
>> 3-2

ans =

     1
```

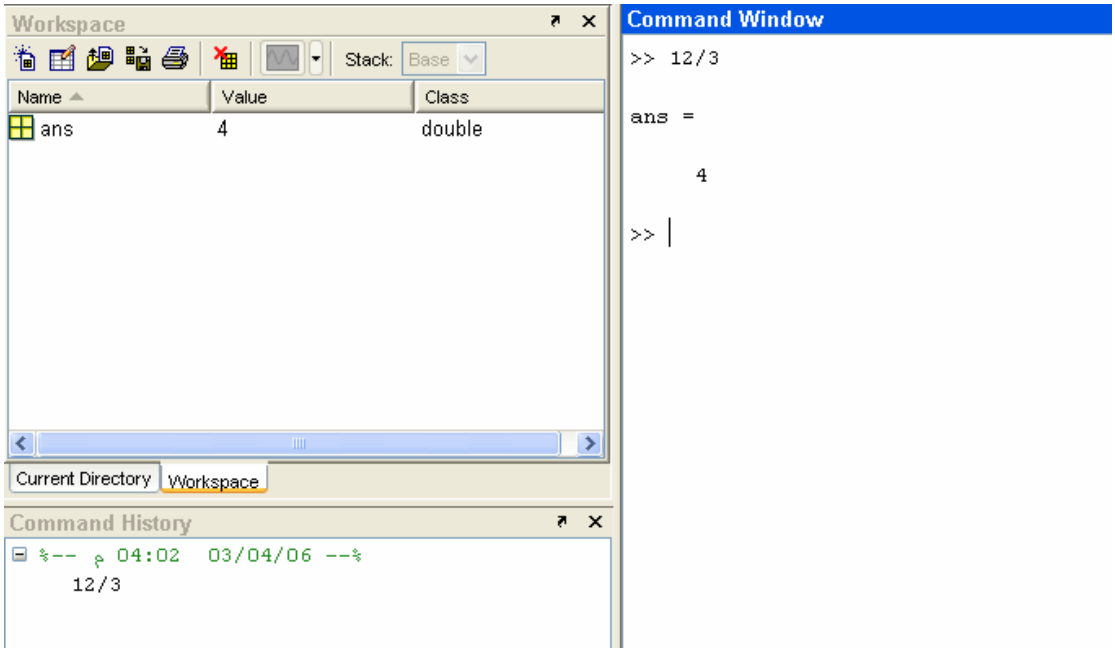
عملية الضرب

تأخذ عملية الضرب رمز (*)، فمثلاً ١٥*١٢= ١٨٠، أنظر الصورة



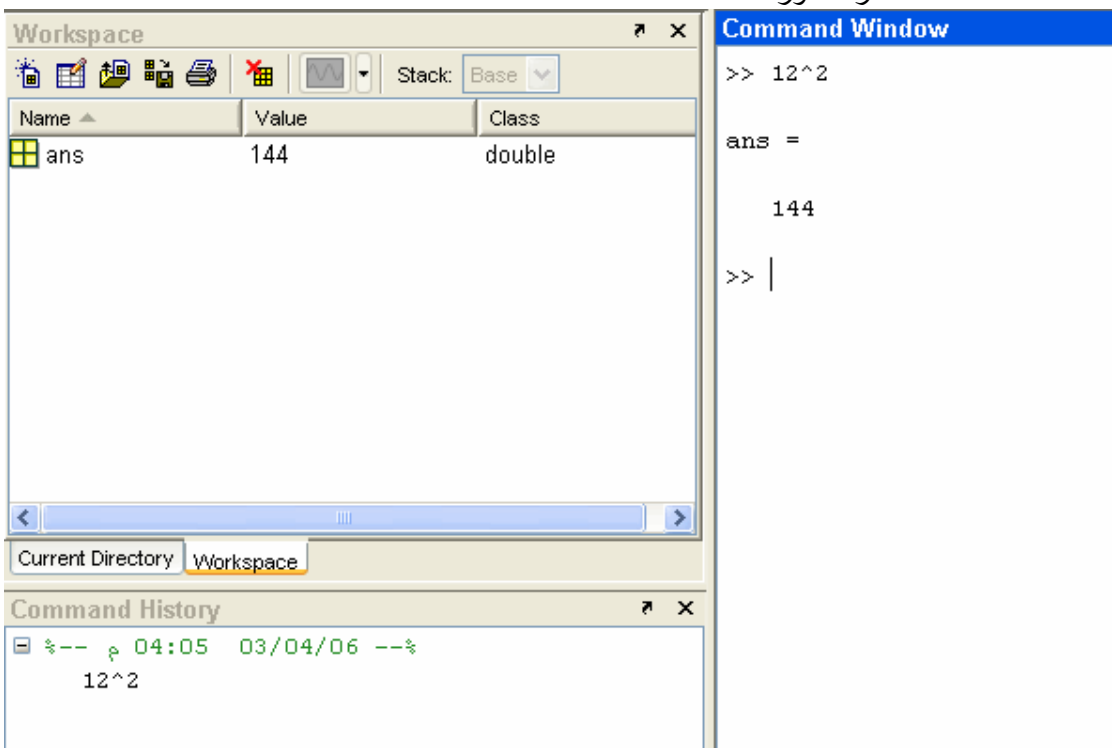
عملية القسمة:

تأخذ عملية القسمة رمز (/)، فمثلاً ١٢ على ٣ تساوى ٤، أنظر الصورة للتتأكد



عملية وضع الأس:

يأخذ رمز الأس (^)، يمكن الحصول على هذا الرمز من خلال الضغط على Shift + 6 في لوحة المفاتيح، فمثلاً ١٢^٢=١٤٤، أنظر الصورة

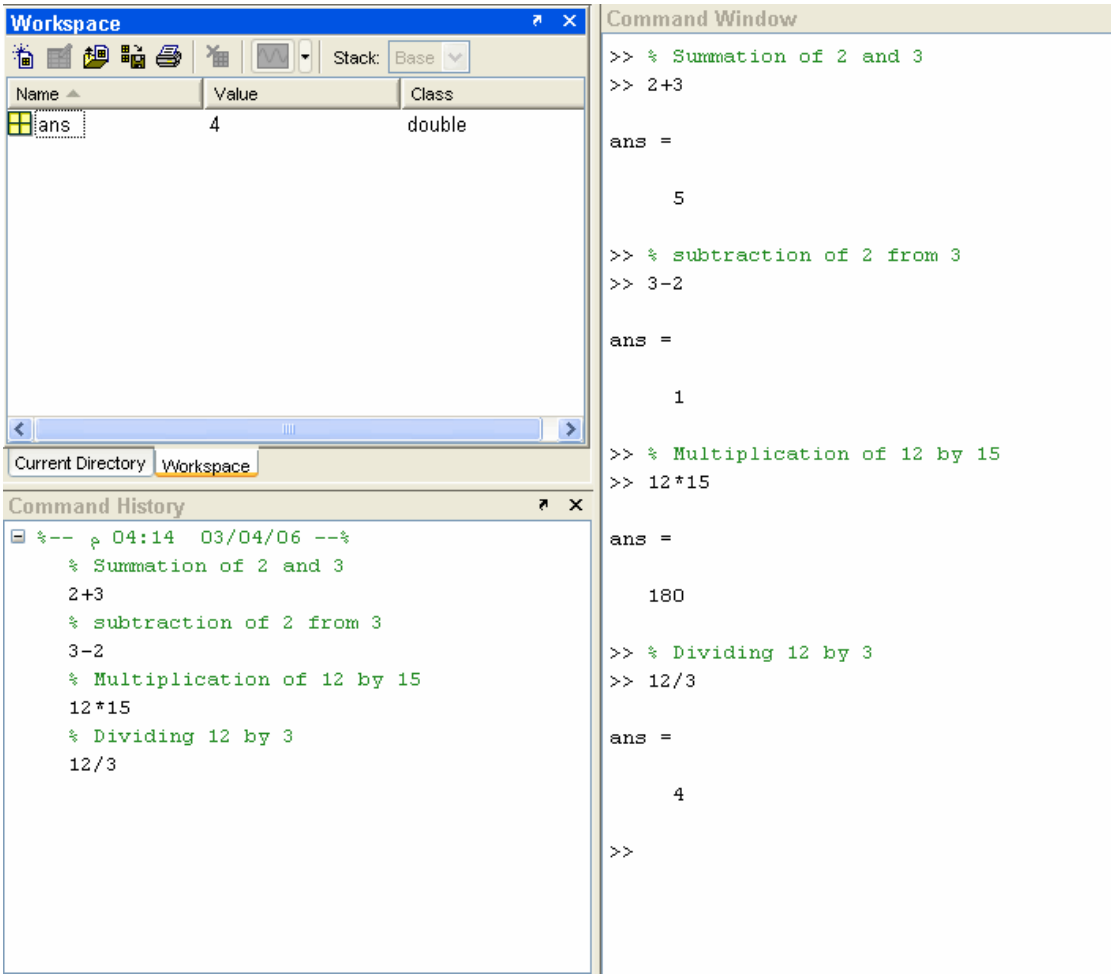


أخذ الجذر التربيعي:
يتم أخذ الجذر التربيعي لأي رقم عن طريق كتابة الأمر sqrt، أنظر الصورة التالية

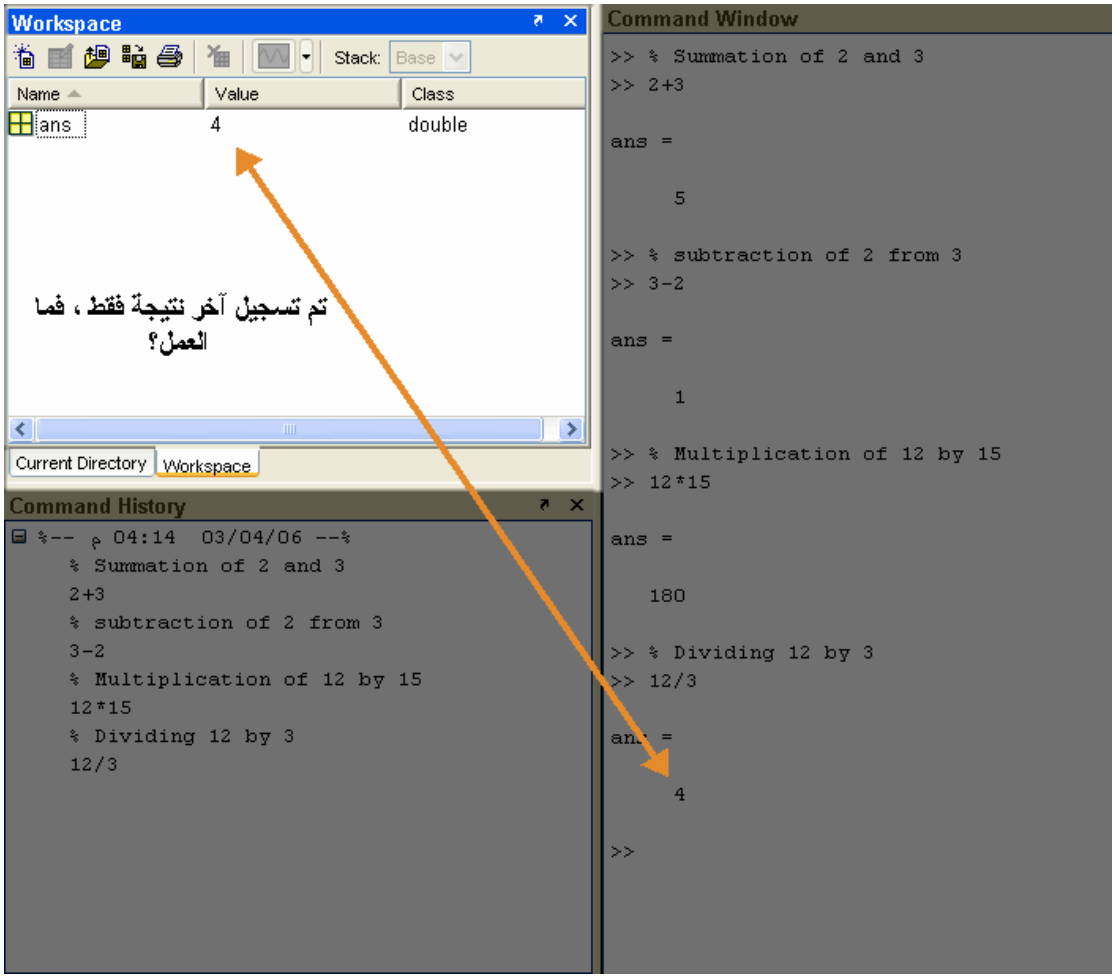
```
>> sqrt(144)

ans =

    12
```

وضع عناوين أثناء البرمجة

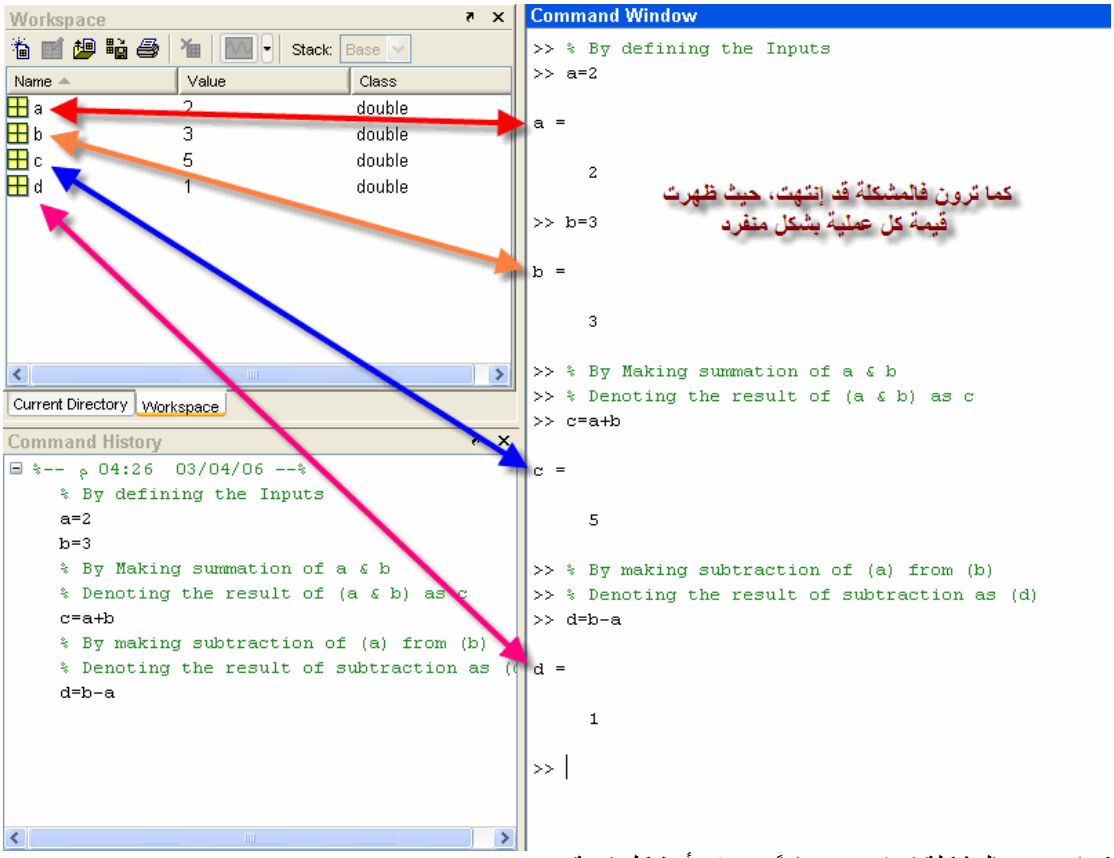
كما تعودنا في برامج Qbasic و C++ وغيرها الكثير من برامج البرمجة، فيتم وضع عناوين لما نقوم به حيث تكون مثل المرجع لنا في معرفة ما نقوم به في جزء ما من البرنامج.
ففي برنامج الماتلاب لوضع عنوان ما، لابد من أن نبدأ بوضع علامة مئوية (%)، ثم نكتب مانريده بعدها، لاحظ الصورة التالية



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

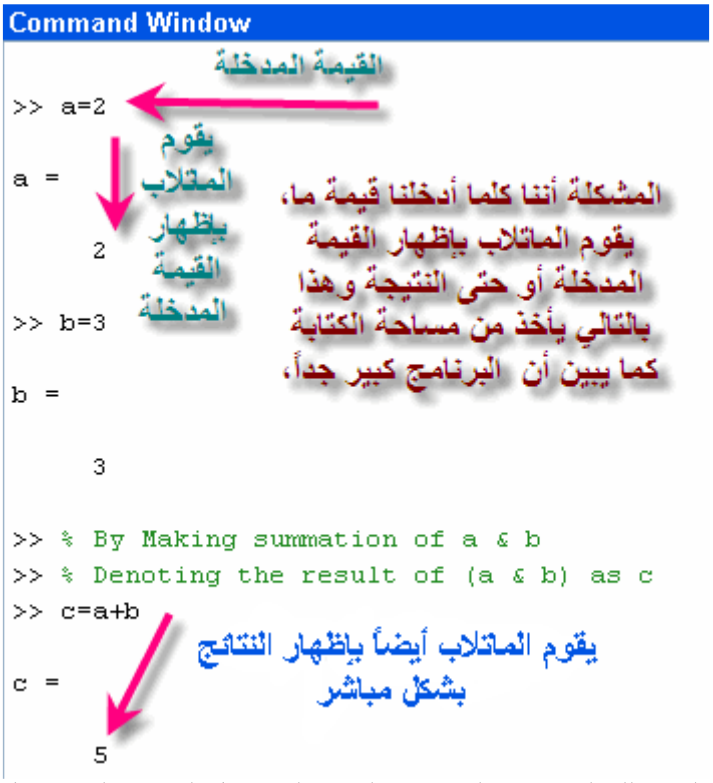


يتم تعريف النتائج بحروف، بحيث يأخذ الحرف القيمة التي يدخلها المستخدم له، أنظر الصورة



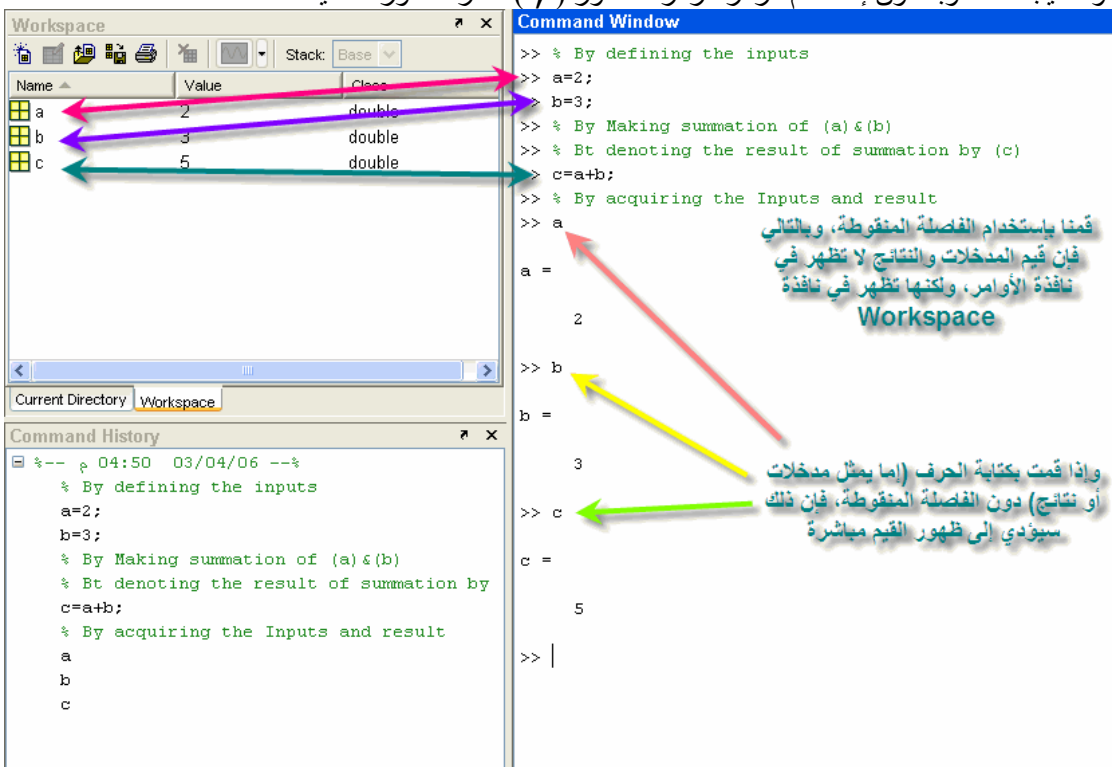
كما ترون فالمشكلة قد إنتهت تماماً، حيث تأخذ كل قيمة حرف معين.

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



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 الرمز المذكور (;) انظر الصورة التالية



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

أنظر 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

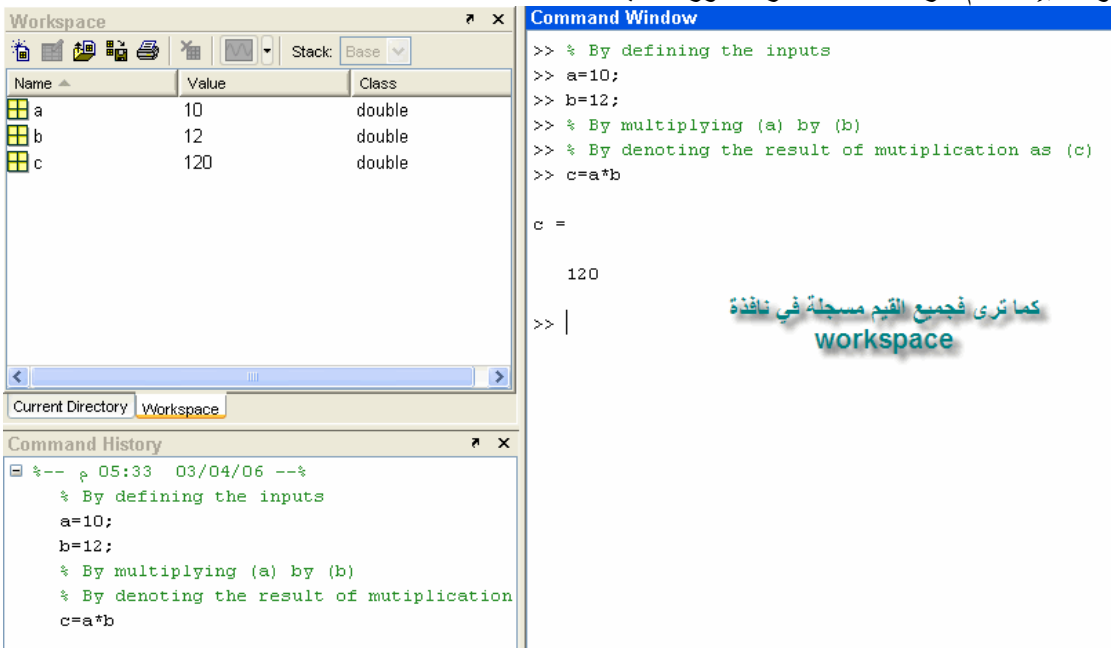
```
>> % the following command will show the complex number
>> i

ans =

        0 + 1.0000i
```

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

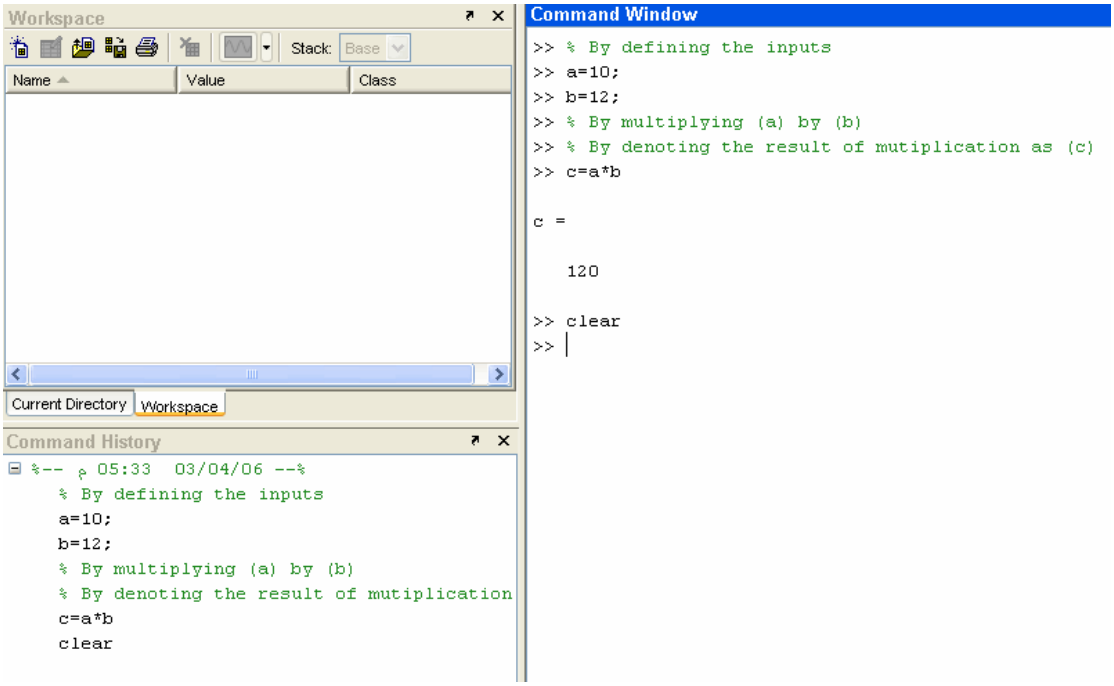
```
Command Window
>> % Overwriting the complex variable i
>> i=3;
>> a=1+3*i

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

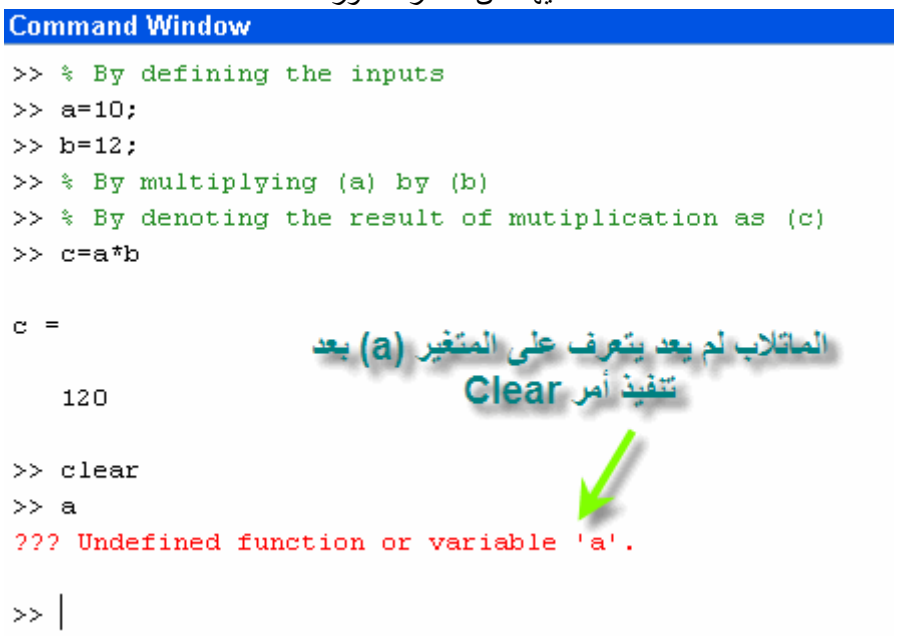
يمكن للماتلاب مسح القيم المدخلة والنتائج (والتي تسجل في نافذة تسجيل النتائج)، دون مسح ما قمت بكتابته، وذلك بإستخدام أمر Clear، أنظر الصورة التالية



ولكن بعد تنفيذ أمر Clear

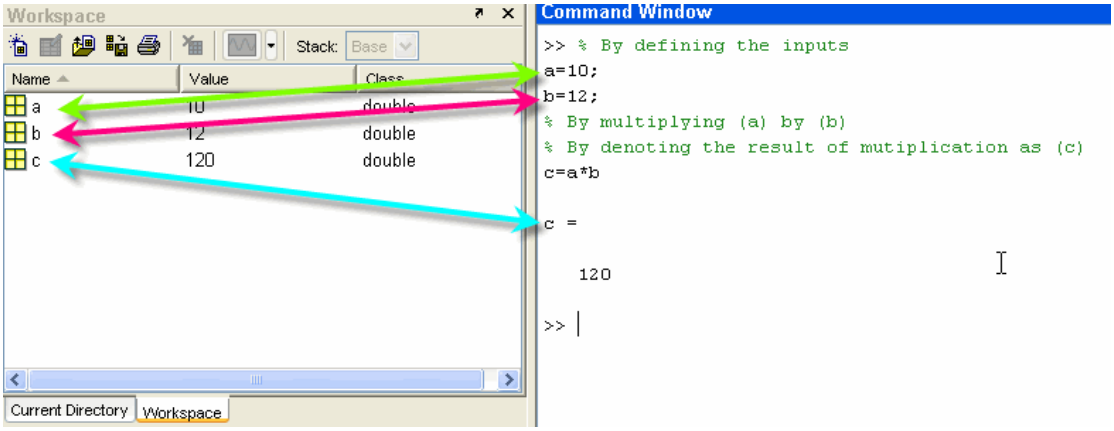


وللتأكد قم بوضع أي حرف من الحروف التي قمت بتعريفها مسبقاً للماتلاب، ستلاحظ ان الماتلاب لا يتعرف عليها الآن، أنظر الصورة

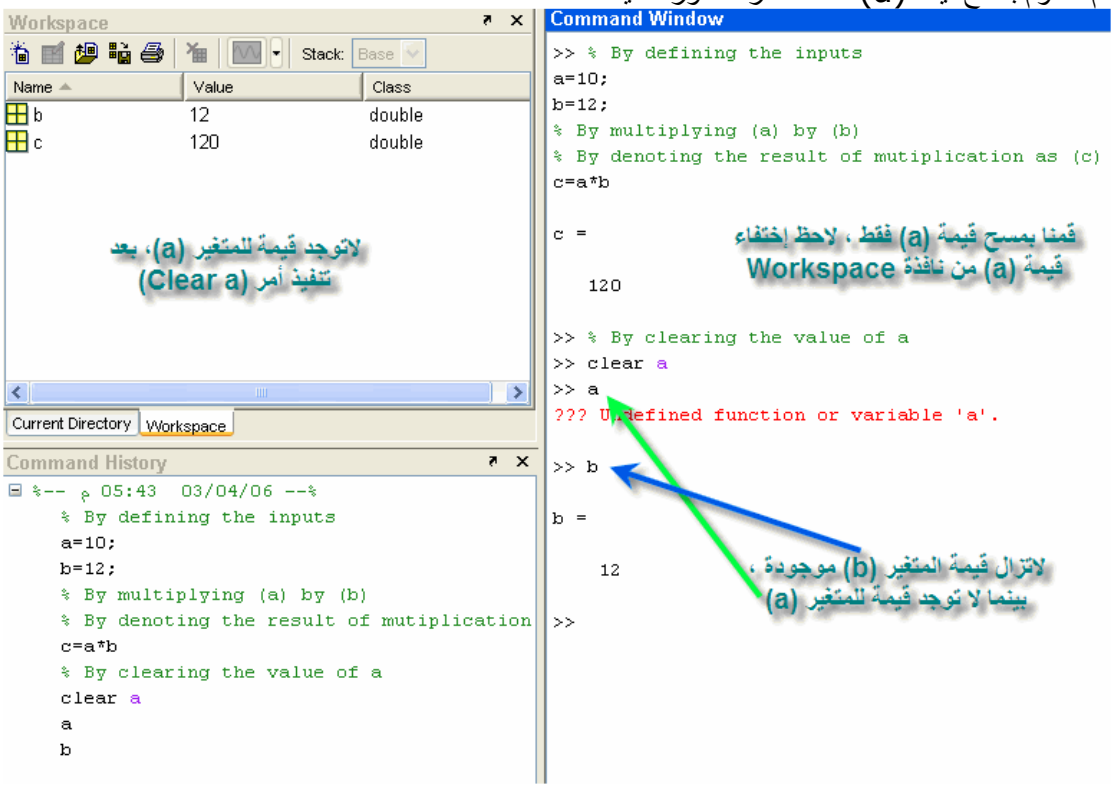


عملية المسح الجزي للمتغيرات:

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



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



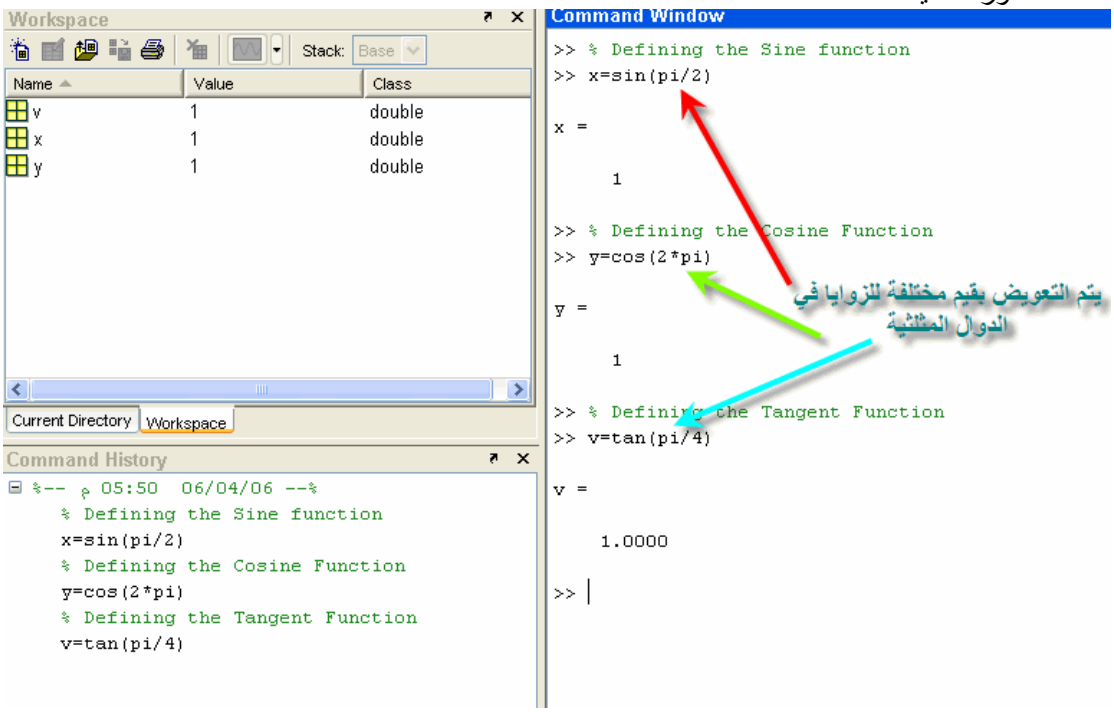
أكتفي هنا بهذا الكم اليوم
ونستكمل بإذن الله غداً
أسألكم الدعاء

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



```
>> % 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 Plotting 2D

الدوال المثلثية العكسية:

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

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، ماهي قيمة الزاوية التي إذا أخذنا لها Sine نحصل على العدد ١ 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 ١

```
>> % 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 ١ 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 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 بإستخدام 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

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

b =

    1.1752
```

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

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

أنظر الصورة التالية للتحقق من النتيجة بإستخدام 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

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

d =

    0.6481
```

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

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

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

a =

    1.1752

>> c=1/a

c =

    0.8509

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

d =

    0.8509
```

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

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

```
>> % 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

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

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

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

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

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

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

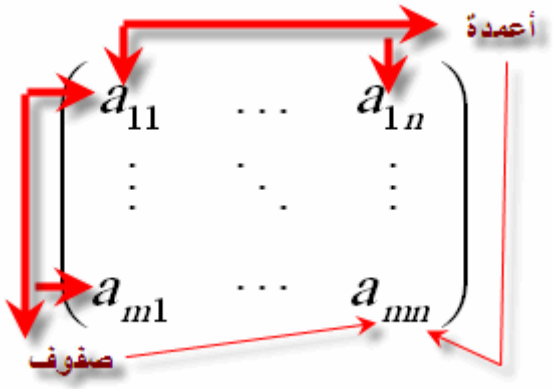
يقوم الماتلاب من خلال التعويض بالمتغير (z) في المعادلات الموضحة الحصول على الدوال الزائدية العكسية.

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

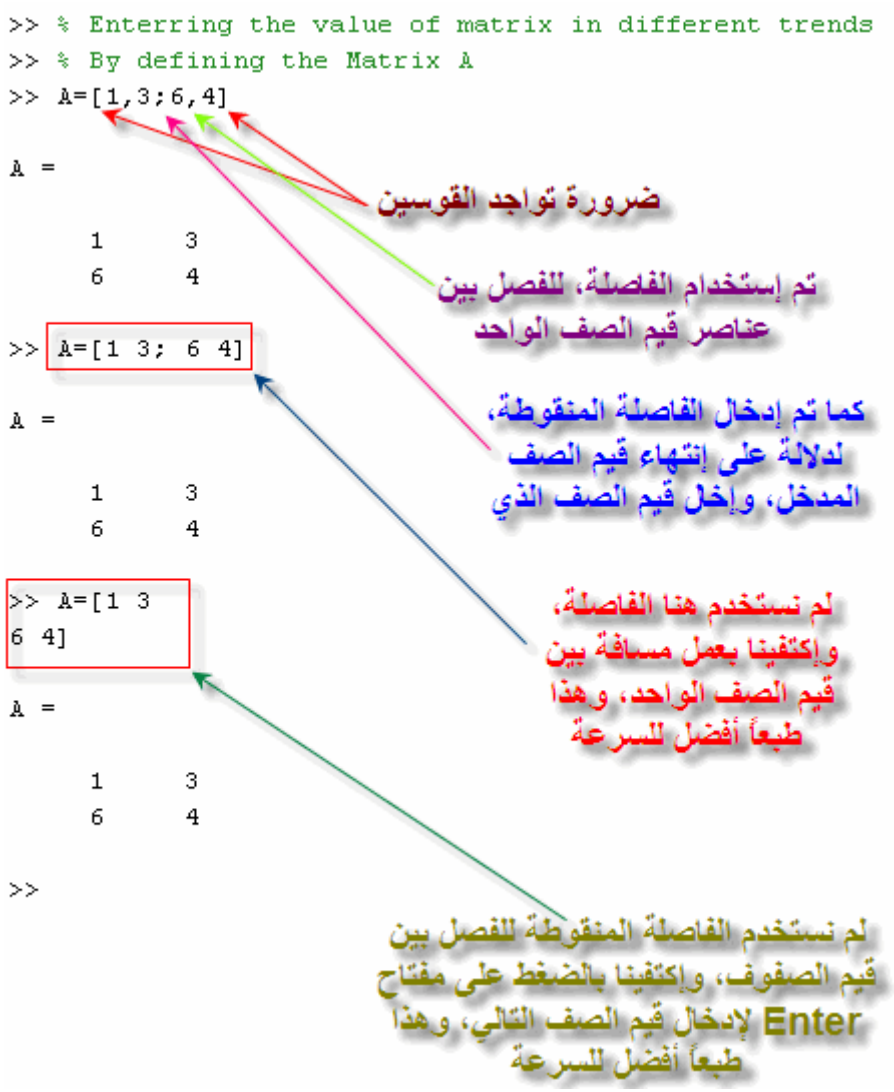


وتستخدم المصفوفات في حل كثيرات الحدود Polynomials، وفي حل مجموعة من المعادلات، كما سيتم شرحه لاحقاً في هذا الإسبوع بإذن الله.

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 (,) Comma أو بعمل مسافة Space بين الأرقام، بعد إدخال قيم الصف الأول يتم فصل عناصر الصف الأول عن عناصر الصف الثاني (الذي سيتم إدخال قيمه) إما بالضغط على مفتاح Enterأو بإستخدام الفاصلة المنقوطة (;) Semicolon، أنظر الصورة التالية



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- القسم
٥- المصفوفة الأسية

الجمع:
قبل البدء في الشروع ببدء إستخدام الماتلاب يجب أولاً أن نذكر شرط جمع مصفوفتين.

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

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}_{3,2} \qquad B = \begin{pmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{pmatrix}_{3,2}$$

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

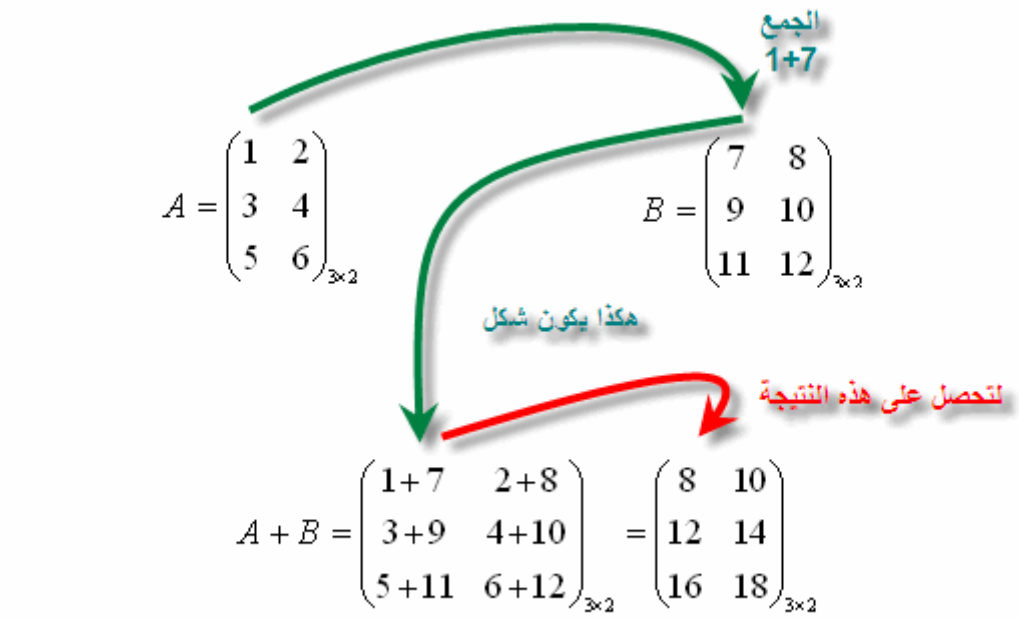
كيف تتم عملية جمع مصفوفتين:
تتم عملية الجمع بجمع العنصر الأول للصف الأول مثلاً في المصفوفة الأولى وما يناظره في المصفوفة الثانية، وبالتالي نكون قد جمعنا العنصر الأول للصف الأول.
وبالتالي نكون قد جمعنا

$$1+7=8$$

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

$$2+8=10$$

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



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

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

<u>الجمع في الماتلاب</u>

يجب أولاً كتابة المصفوفتين B&A، كما تعلمنا سابقاً

ثم إستخدام رمز الجمع (+) للتتم عملية الجمع، أنظر الصورة التالية

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

A =

     1     2
     3     4
     5     6

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

B =

     7     8
     9    10
    11    12

>> % By making addition to both A&B

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

C =

     8    10
    12    14
    16    18
```

يستكمل.....

طرح المصفوفات

فما هو شرط طرح المصفوفات؟

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

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

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

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

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

لنقم الآن بعمل نفس المثال على برنامج الماتلاب
أنظر الصورة التالية

```
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

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

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

ملتقى المهندسين العرب

هذه هي عملية ضرب المصفوفات بالطريقة اليدوية

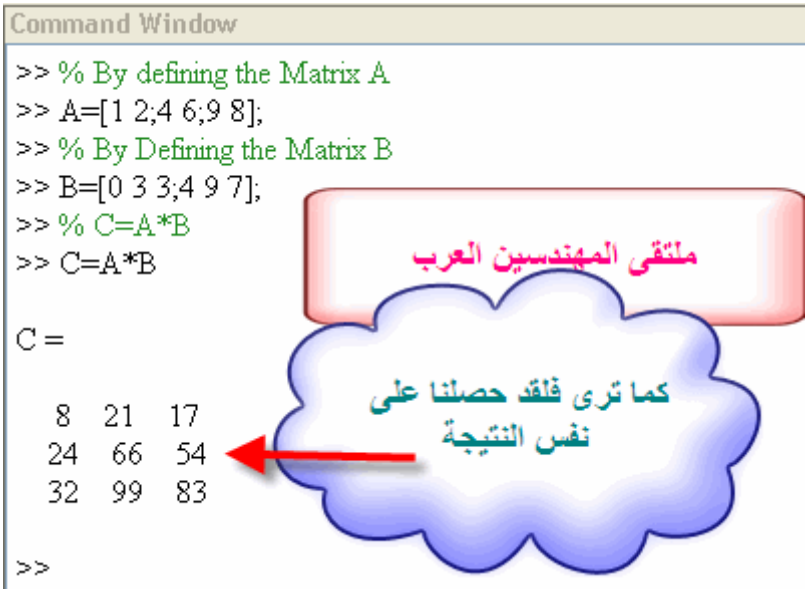
$$C=A\times B=\begin{pmatrix}1 & 2\\ 4 & 6\\ 9 & 8\end{pmatrix}_{3\times2}\times\begin{pmatrix}0 & 3 & 3\\ 4 & 9 & 7\end{pmatrix}_{2\times3}$$

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

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

لنقوم الآن بإدخال نفس المثال على الماتلاب

أنظر الصورة التالية



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 حلهما ليكون الناتج

X=-2

Y=3

فكيف يتم ذلك؟

يمكن وضع المعادلتان في صورة مصفوفة كما في الشكل التالي

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

المعادلتان في صورة المصفوفة

وهنا نذكر أن هنالك طريقتان لحل المعادلتان

1-طريقة الحذف

2-قسمة المصفوفات

وسأذكر سريعاً طريقة الحذف، أنظر الصورة التالية

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

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

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

$\therefore 1.5Y=4.5$

$\therefore Y=3$

$\because 3X+3Y=3$

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

$\therefore X=-2$

طريقة الحذف في حل المصفوفات

أما الطريقة الثانية هي قسمة المصفوفات

لنعود إلى الصورة التالية مرة أخرى

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

المعادلتان في صورة المصفوفة

نجد أنه يمكننا أن نضعها في الصيغة التالية

$$AX = B$$

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

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

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

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

Where inv( ) is the inverse function

وهذا ما يسمى قسمة المصفوفات

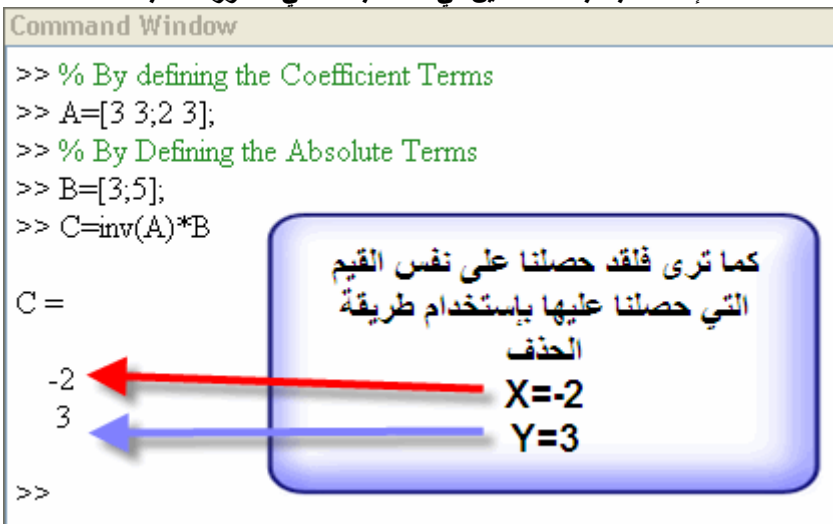
ولكن يشترط عند إيجاد inv أن تكون المصفوفة مربعة (أي عدد الصفوف يساوي عدد الأعمدة(

وبالتالي يمكن إيجاد قيمة X & Y عن طريق وضع المعادلة في الصورة التالية، مع الأخذ في الإعتبار أن تتوفر شرط عملية الضرب بين المصفوفتين

$$\begin{pmatrix}X\\ Y\end{pmatrix} = inv\left(\begin{pmatrix}3 & 3\\ 2 & 3\end{pmatrix}_{2\times 2}\right) \times \begin{pmatrix}3\\ 5\end{pmatrix}_{2\times 1}$$

يجب الإنتباه لشرط عملية ضرب المصفوفة

فإذا قمنا بكتابة المعادلتين في الماتلاب كما في الصورة السابقة



الدرس القادم بإذن الله هو

العمليات على المصفوفات

العمليات على المصفوفات والمتجهات

5

...هات والمصفوفات؟

المتجهات هي مصفوفة ولكن ...عمود و... Column Vec... أو صف واحد Row Vector
...مثلاً الط... التالية لمتجه صفي
!

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

A =

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

Row Vector

وهذه صورة لمتجه عمودي

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

B =

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

Column Vector

أما المصفوفة فهي التي يزيد عدد صفوفها وأعمدتها عن صف واحد أو عمود واحد
وسنتناول العمليات التي تتم على المتجهات أولاً ثم المصفوفات

ماهي العمليات الشائعة على المتجهات؟

1- طول المتجه
2- إضافة عنصر
3- إستبدال عنصر
٤- عملية حذف عنصر
٥- نداء عنصر
٦- نداء عدد عناصر
٧- إيجاد العنصر الأكبر
٨- إيجاد العنصر الأصغر
٩- 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

ة الأولى وهي

طو صفوفة

!

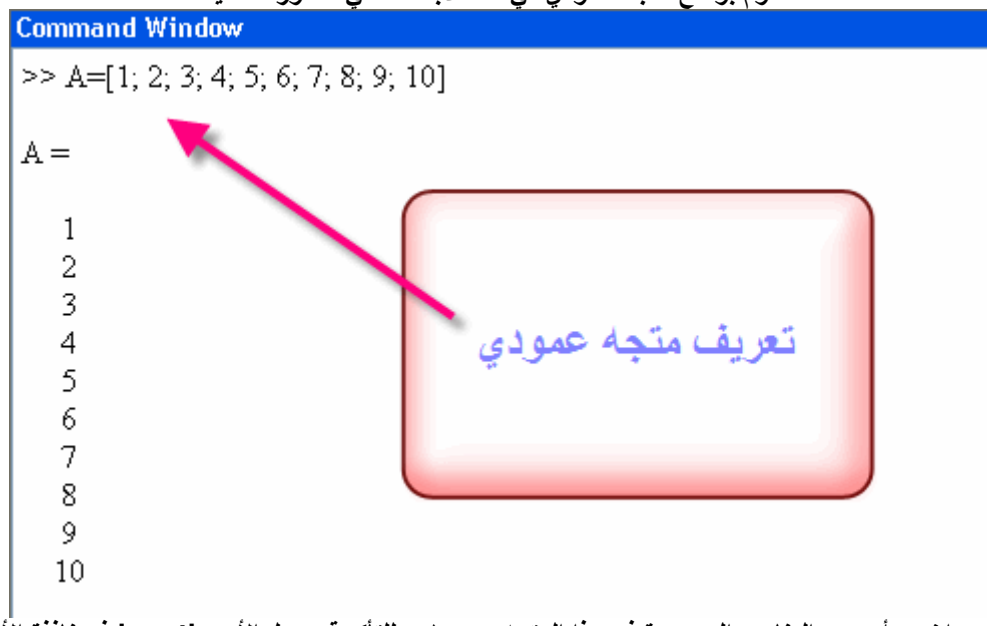
```
Command Window
>> A=[1 2 3 4 5 6 7 8 9 10];
>> % It's required to get the length of A
>> length(A)

ans =

   10
```

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

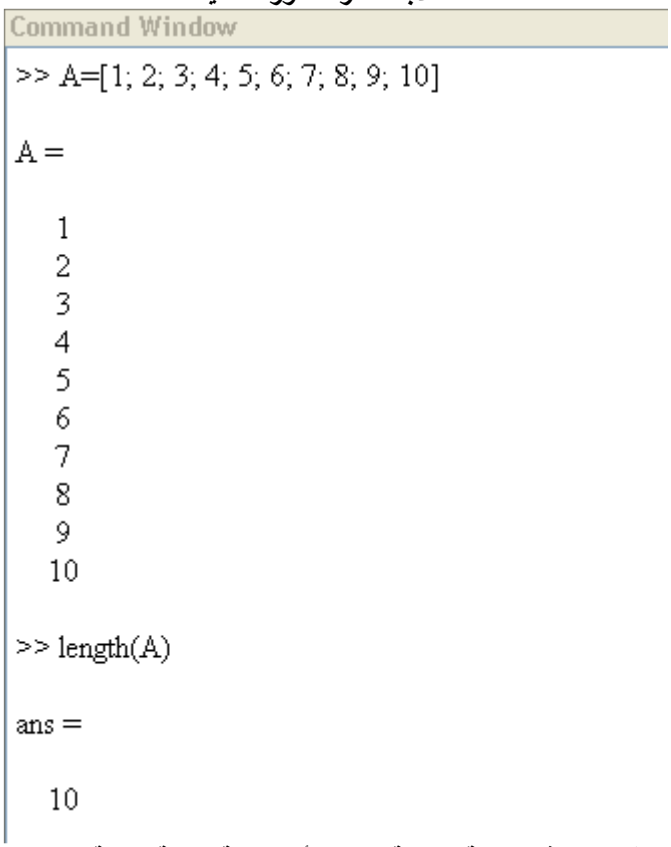
يمكن عمل نفس العملية على متجه عمودي
وننتقل الآن إلى العملية التالية

إضافة عنصر

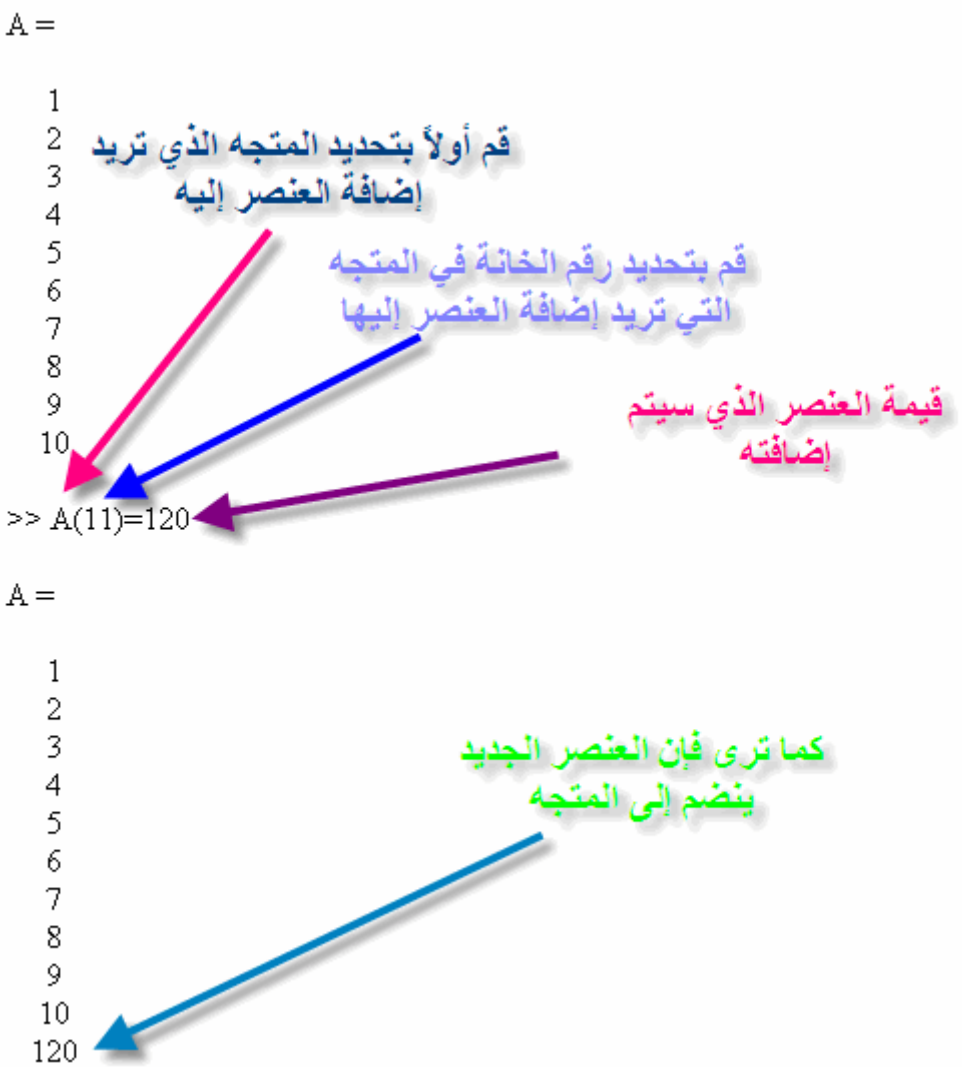
لنقوم بوضع متجه عمودي في الماتلاب، كما في الصورة التالية



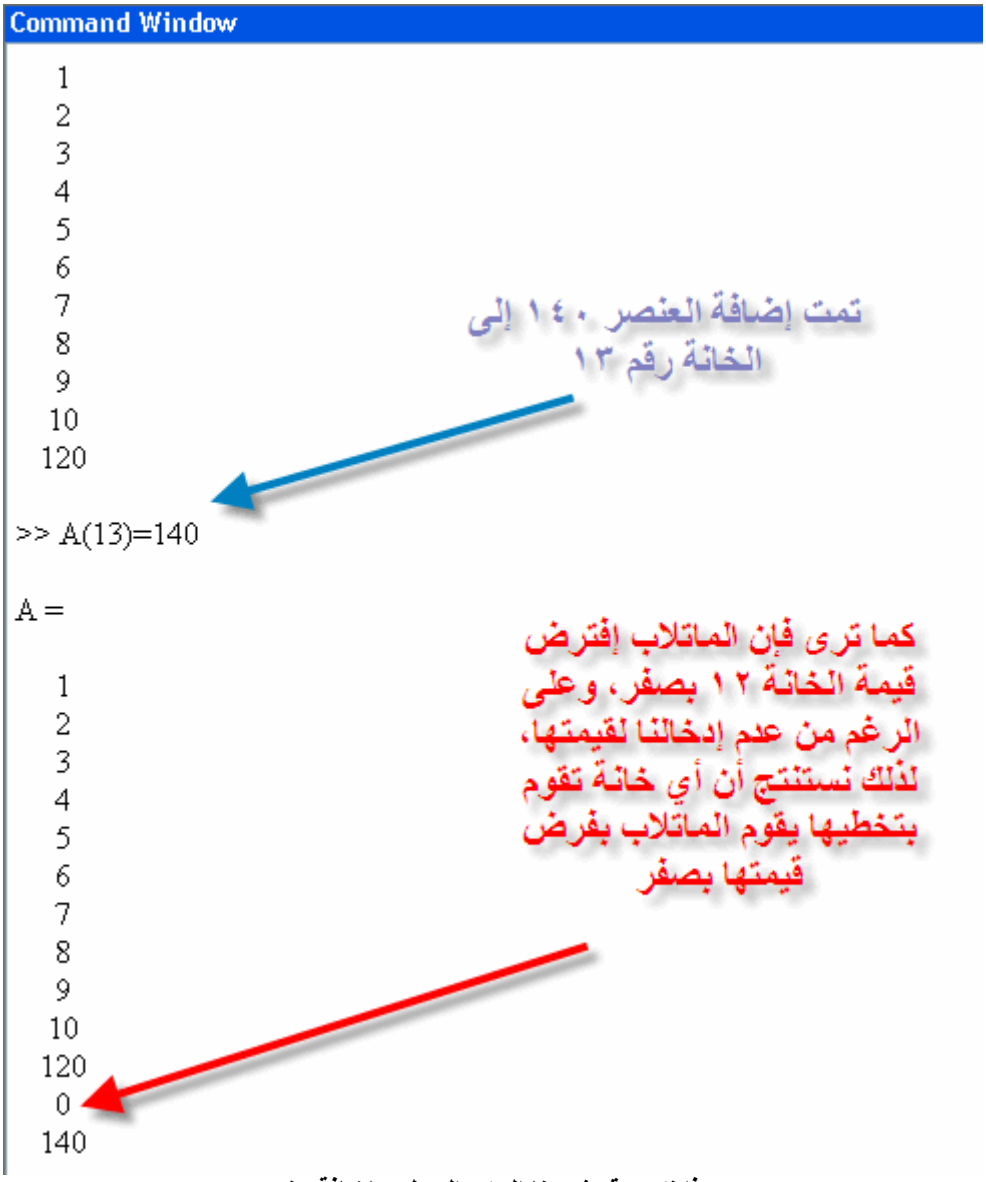
كما هو واضح، أن عدد العناصر الموجودة في هذا المتجه هو ١٠، وللتأكد قم بعمل الأمر **length** في نافذة الأوامر للماتلاب، أنظر الصورة التالية



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



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



وبهذا نكون قد شرحنا الجزء المتعلق بإضافة عنصر

إضافة أكثر من عنصر متتالي

لنفترض أننا نريد إضافة مجموعة من العناصر المتتالية في الخانات ١١و 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 يبدوا 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

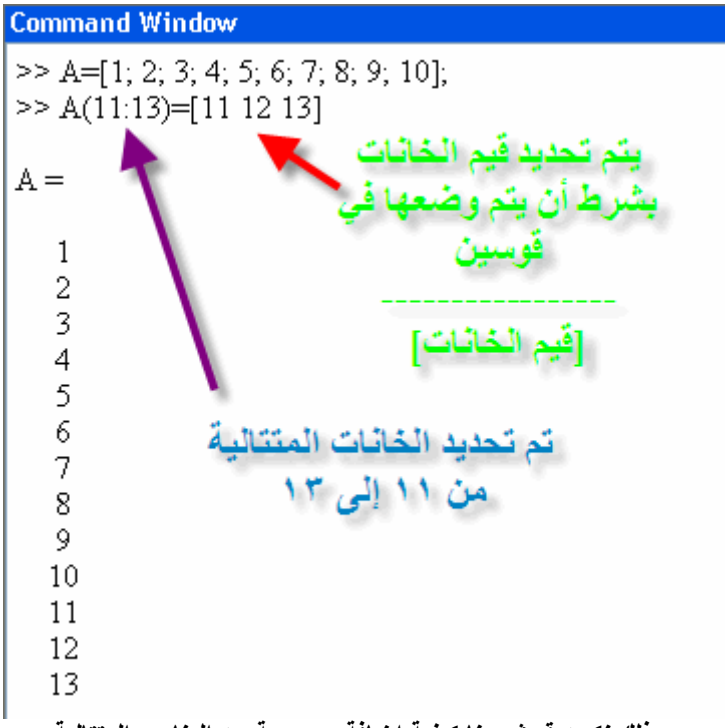
وعندما نريد كتابة مجموعة من الأرقام المتتالية من 10 إلى ١٢٠٠ نكتب

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



وبذلك نكون قد شرحنا كيفية إضافة مجموعة من العناصر المتتالية

إستبدال عنصر

عملية إستبدال عنصر تتطلب عدة شروط

1-أن يكون العنصر موجوداً بالفعل

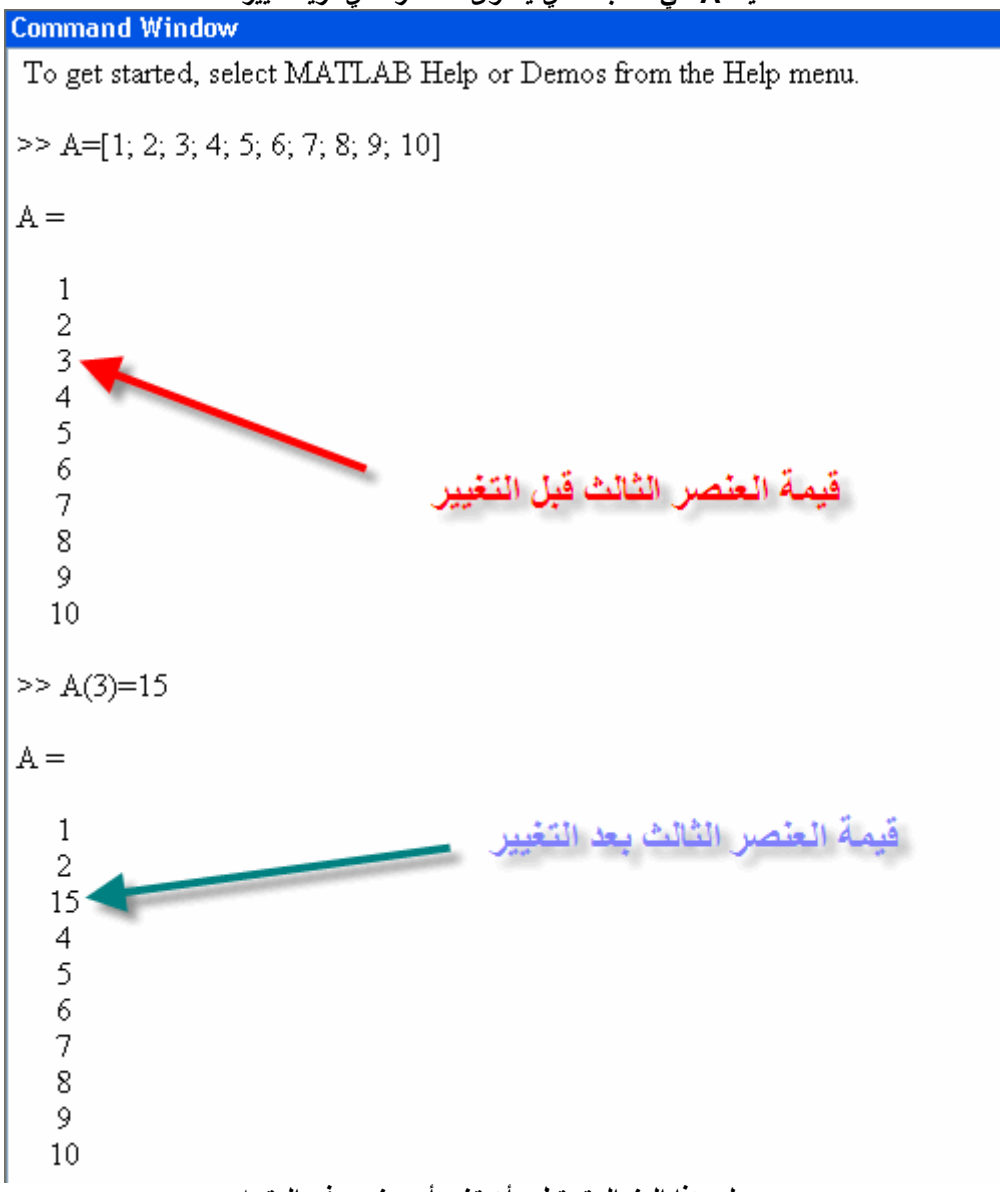
2-أن تحدد مكان هذا العنصر

ففي المثال التالي أردنا أن تستبدل العنصر الثالث بدلاً من الرقم ٣ إلى الرقم ١٥

كل ما علينا فعله هو كتابة التالي

A(3)=15

حيث **A** هي المتجه الذي يحتوى العنصر الذي تريد تغييره

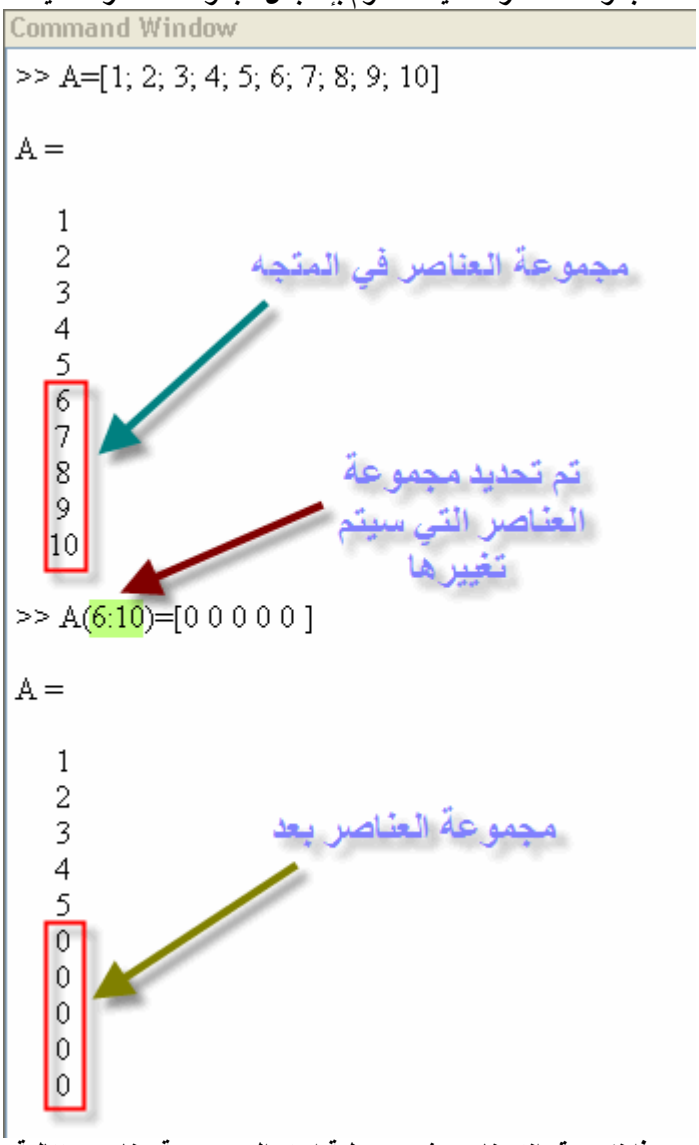


وعلى هذا المنوال تستطيع أن تغير أي عنصر في المتجه

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



وبهذا نكون قد إنتهينا من شرح عملية إستبدال مجموعة عناصر متتالية

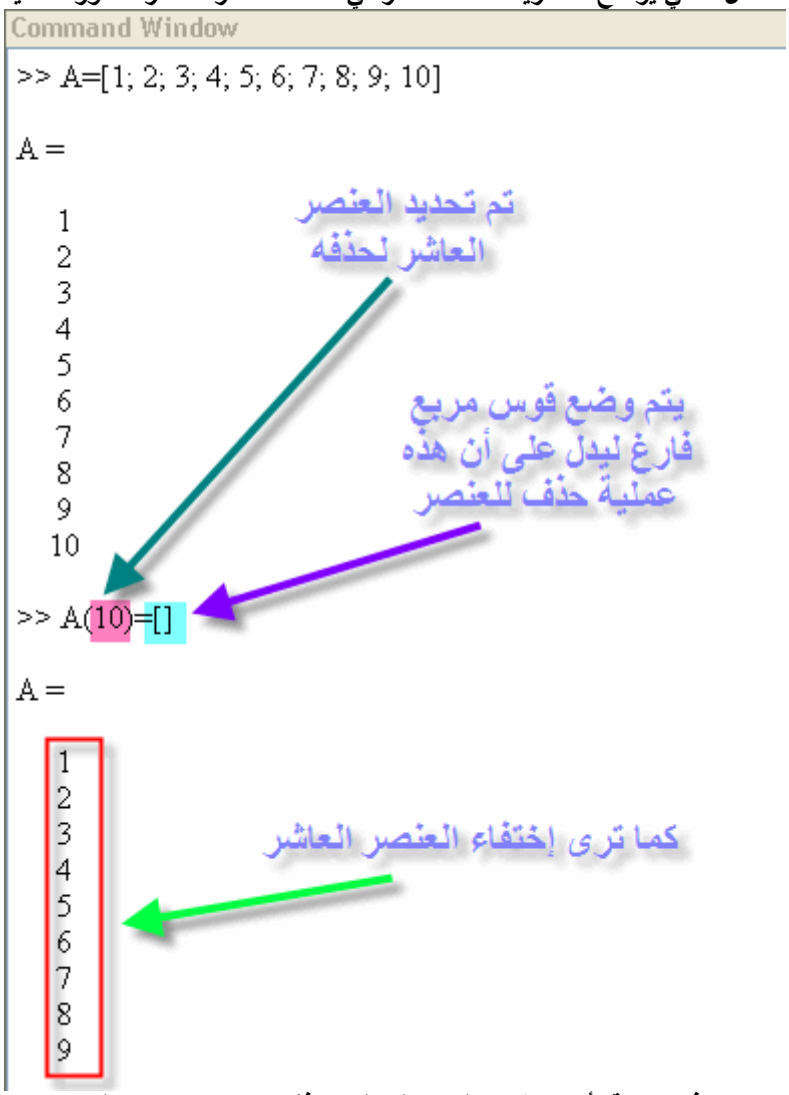
حذف عنصر من المتجه

لتقوم بحذف عنصر من المتجه يجب أن يتوفر الشرطان التاليان

1- تحديد العنصر الذي تريد حذفه

2- وضع أقواص مربعة **Square Brackets** خالية من أي رقم

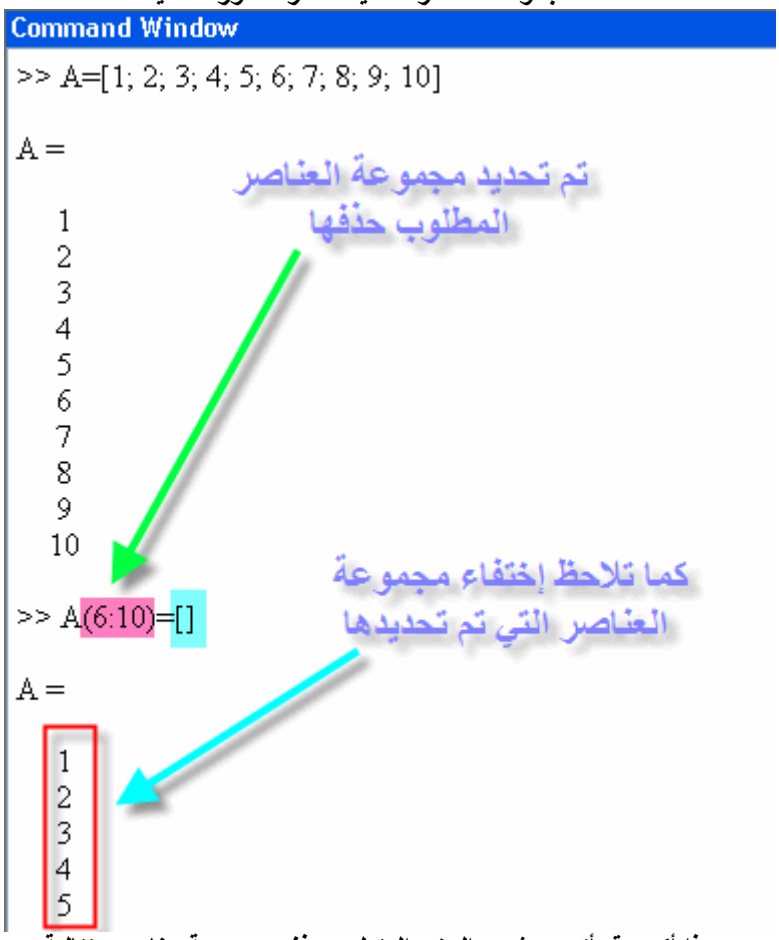
فالمثال التالي يوضح أننا نريد حذف العنصر في الخانة العاشرة، أنظر الصورة التالية



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

حذف مجموعة عناصر متتالية

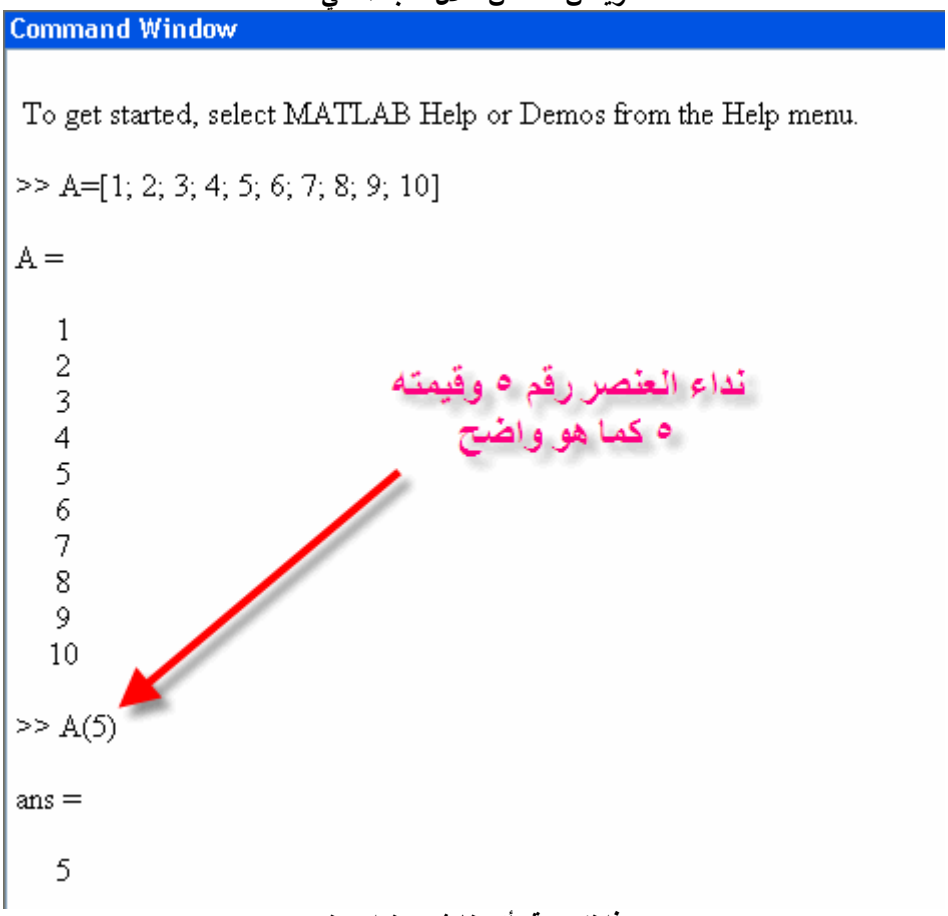
لحذف مجموعة عناصر متتالية، أنظر الصورة التالية



وبهذا أكون قد أتممت شرح الجزء المتعلق بحذف مجموعة عناصر متتالية

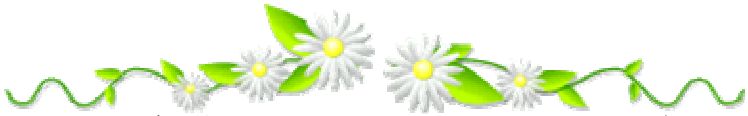
نداء عنصر

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



وبهذا نكون قد أتممنا شرح نداء عنصر

نداء أكثر من عنصر

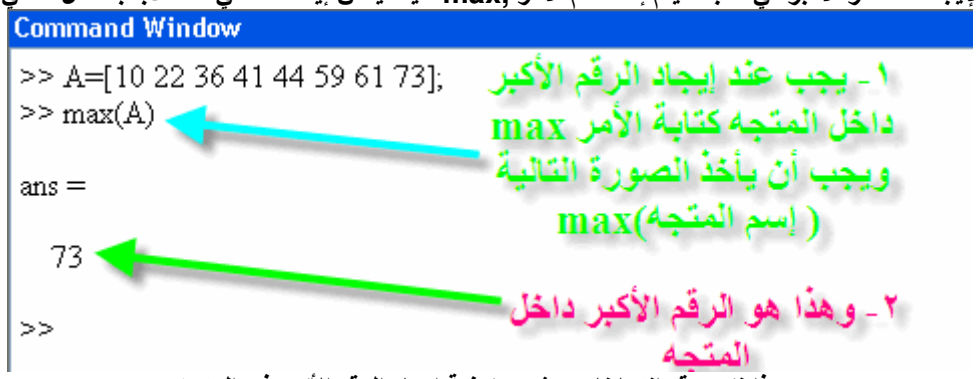


للحصول على قيم مجموعة عناصر محددة من متجه، قم بعمل الآتي على نافذة الأوامر Command Window

وبهذا نكون قد شرحنا الجزء الخاص بنداء أكثر من عنصر

إيجاد العنصر الأكبر في المتجه

لإيجاد العنصر الأكبر في متجه، يتم إستخدام الأمر **max,** حيث يمكن إيتخدامه في الماتلاب بالشكل التالي

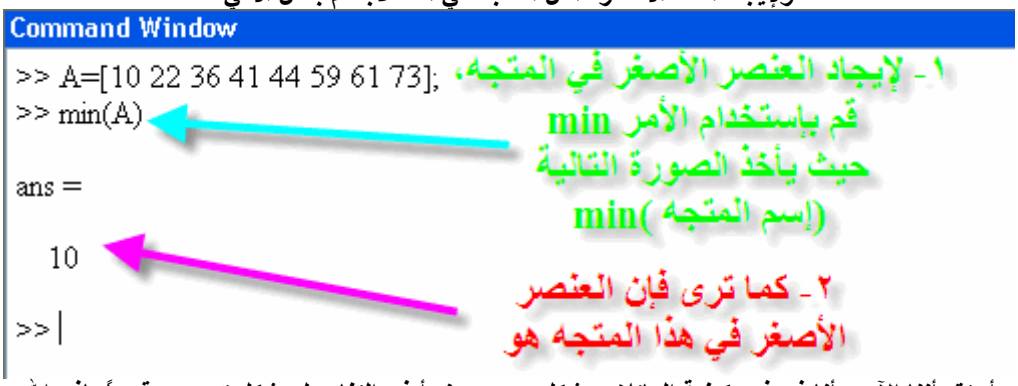


وبهذا نكون قد إنتهاينا من شرح كيفية إيجاد الرقم الأكبر في المتجه

إيجاد العنصر الأصغر في المتجه

لإيجاد العنصر الأصغر في المتجه، يجب إستخدام الأمر **min** وهي إختصار لدى الماتلاب وهي إختصار لكلمة **minimum** أي الأقل

ولإيجاد العدد الأصغر داخل المتجه في الماتلاب قم بعمل الآتي



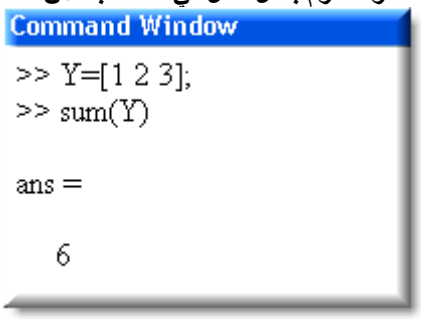
أعتقد أننا الآن بدأنا في فهم كيفية الماتلاب بشكل جيد، وسنبدأ في التفاصيل بشكل تدريجي قريباً بإذن الله

إيجاد مجموع عناصر المتجه

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

Sum (إسم المتجه)

وفالنقوم بعمل مثال في الماتلاب الآن

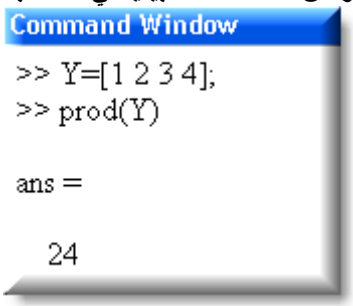


إيجاد حاصل ضرب العناصر في المتجه

يوفر الماتلاب خاصية ضرب عناصر المتجه، وذلك بإستخدام الأمر **prod** وهو إختصار**product**
ويجب أن يأخذ هذا الأمر الصورة التالية في كتابته

prod (إسم المتجه)

والآن لنأخذ مثالاً تطبيقياً في الماتلاب



العمليات على المصفوفات

أولاً يجب تعريف أنواع المصفوفات، فهنالك 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-إستبدال 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 المصفوفةDiagonal

وهذه هي العمليات الشائع إستخدامها في الماتلاب

وسيتم شرحها بالتفصيل بإذن الله

إيجاد حجم المصفوفة

لإيجاد حجم المصفوفة أو دعونا نقول لإيجاد عدد الصفوف والأعمدة لمصفوفة، يجب إستخدام الأمر size, حيث لا يصلح إستخدام 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
```

والآ 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 بإستخدامها 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 ٤٢ في الصف الثاني والعمود الخامس ,نقوم بكتابة التالي في الماتلاب

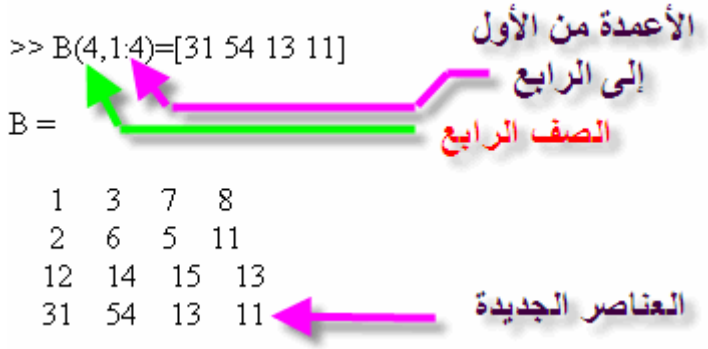
```
>> B(2,5)=42

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 بإفتراضهما صفراً.

فماذا إذا أردنا إضافة عدة عناصر في المصفوفة؟ يمكن إيضاح ذلك بإستخدام المثال التالي

لنقوم أننا نريد إضافة الأعداد٣١ و ٥٤ و ١٣ و 11 في الصف الرابع والعمود الأول الثاني والثالث و الرابع على

التوالي، يمكن ذلك من خلال الماتلاب بالشكل التالي



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، وبقية الصف والعمود بهم قيم، ولكن إذا أردت أن تقوم بحذف صف كامل أو عمود كامل فيمكن ذلك بعمل التالي

نقوم أولاً بعمل مصفوفة للعمل عليها

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

B =

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

لنقل اننا نريد حذف الصف الثالث كله

ولحذف العمود الرابع كله، قم بعمل التالي

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

B =

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

نداء عنصر

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

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

B =

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

لنقل اننا نريد العنصر في الصف الأول والعمود الثالث

```
>> B(1,3)

ans =

    7
```

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

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

ans =

    6    5   11
```

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

```
>> B(2,end)

ans =

   11
```

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، وبعدما يقوم بعمل ذلك ,يقوم بعمل متجه به الرقم الأكبر من كل عمود، أنظر المثال التالي للتوضيح
لدينا الآن مصفوفة تم إنشائها على الماتلاب

```
A =

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

ولنقم بكتابة الأمر max كما ذكرنا مسبقاً

```
>> B=max(A)

B =

    23    15    15    11
```

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

```
>> C=max(B)

C =

    23
```

إيجاد العنصر الأصغر

هذه العملية أيضاً كثيرة الإستخدام في التطبيقات المختلفة، وهي نفس الخطوات السابق ذكرها في إيجاد العنصر الأكبر
ولكن يتم إستخدام الأمر min
وإليكم المثال التالي

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

A =

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

>> B=min(A)

B =

   1    1    2    5

>> C=min(B)

C =

   1
```

إيجاد مجموع العناصر

لإيجاد المجموع كما تعلمنا نقوم بإستخدام الأمر sum ولكن عملية الجمع يقوم الماتلاب بإيجاد جمع كل عمود على

حدى 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 إستخدام أكاديمي، ولكنها هامة جداً، وخصوصاً أن تلك الخاصية تخدم المصفوفة المربعة (عدد الصفوف يساوي عدد الأعمدة)، ويتم إستخدام 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.

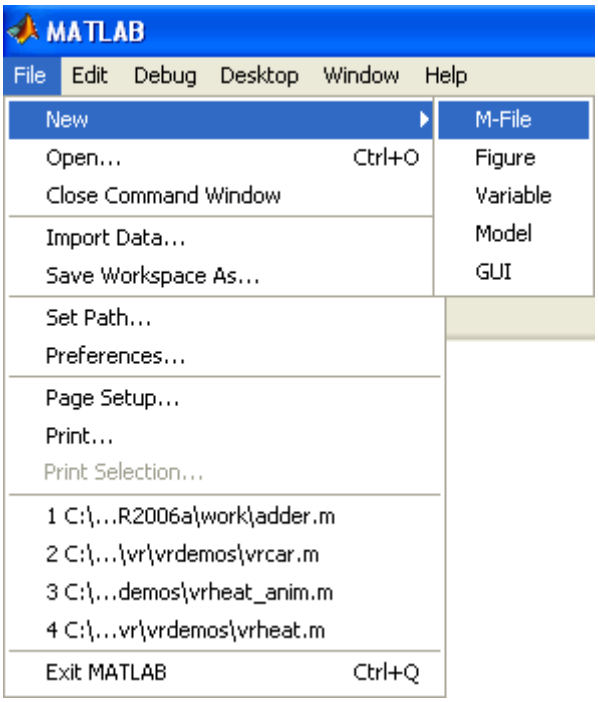
4- إذا حدث خطأ في ترتيب الأوامر لهذا البرنامج الكبير ستقوم بإعادة الإدخال الأوامر من البداية مرة إخرى.

5- يصعب عمل عملية تصحيح للأخطاء Debugging

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

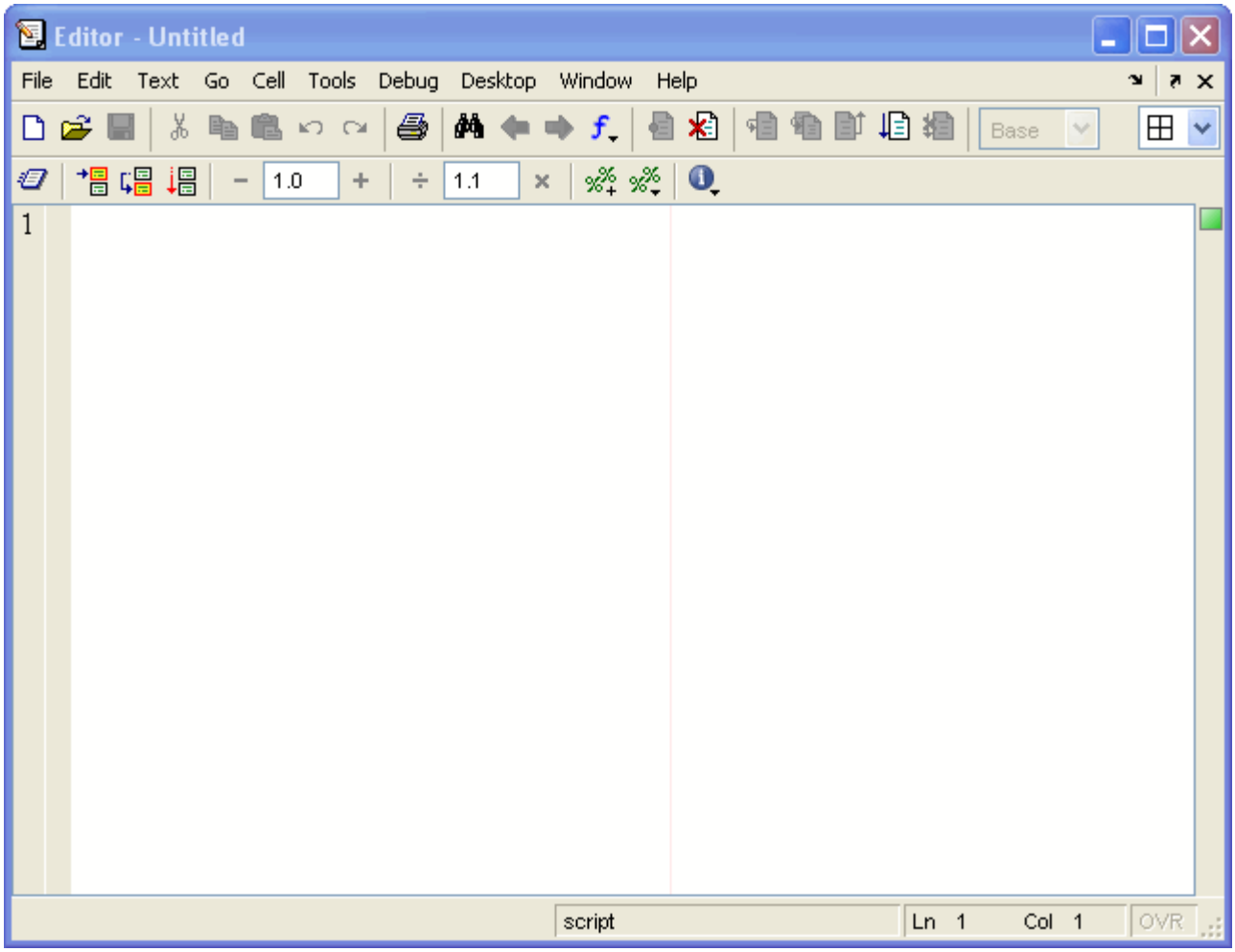
وطبعاً حلاً لهذه المشكلة، تم عمل بما يسمى M-File والتي تعطي القدرة على كتابة البرنامج كاملاً أولاً بدون تشغيل، وبعد الإنتهاء منه يتم تشغيله، هذه الخاصية تعطي القدرة على تعديل القيّم دون الحاجة إلى كتابتها مرة أخرى، أو إعادة إدخال الأوامر التي تعتمد على هذا الأمر.

فكيف يتم تشغيل تلك الخاصية؟ إتبع الصورة التالية



وبالتالي ستظهر نافذه جديدة، تأخذ الشكل التالي

!Error

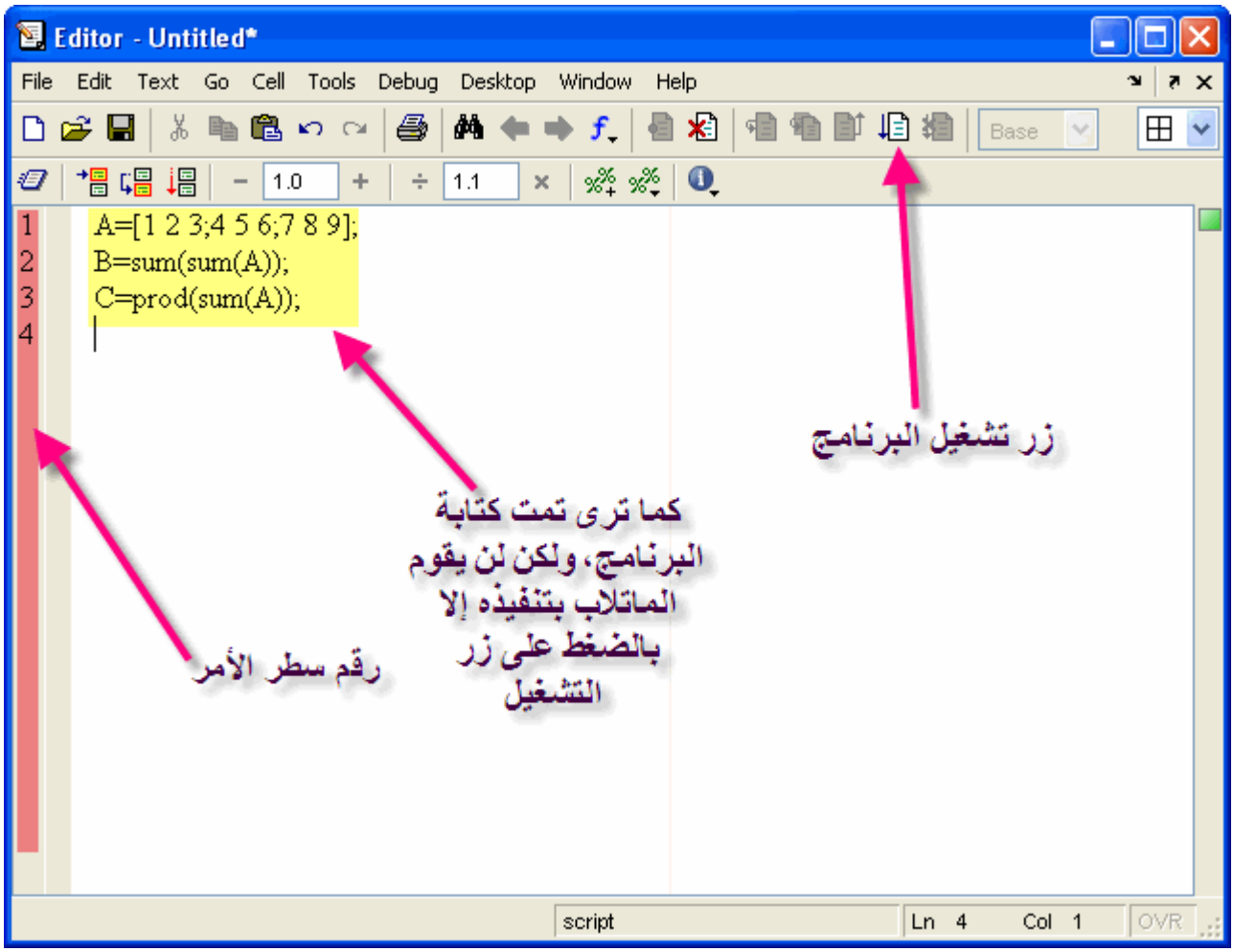


وسنتعرف على نافذة **M-File** بالتفصيل الممل بإذن الله في الدرس القادم

نافذةM-File

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

!Error



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

1- أن لايبدأ بأرقام

2- أن لا يكون أمراً معرفاً في الماتلاب

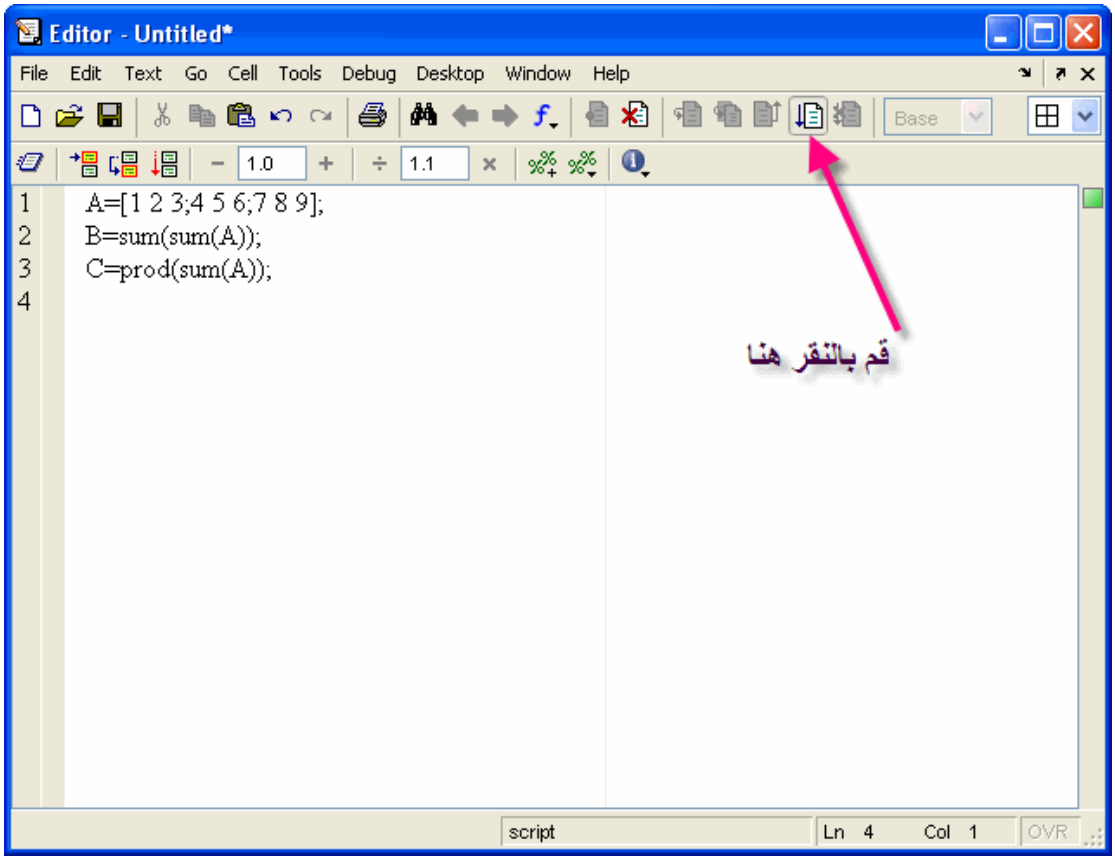
3- أن لا يحتوي الإسم على مسافات فاصلة

٤- أن لاتحتوي على رموز خاصة مثل + ، - ، & ، *

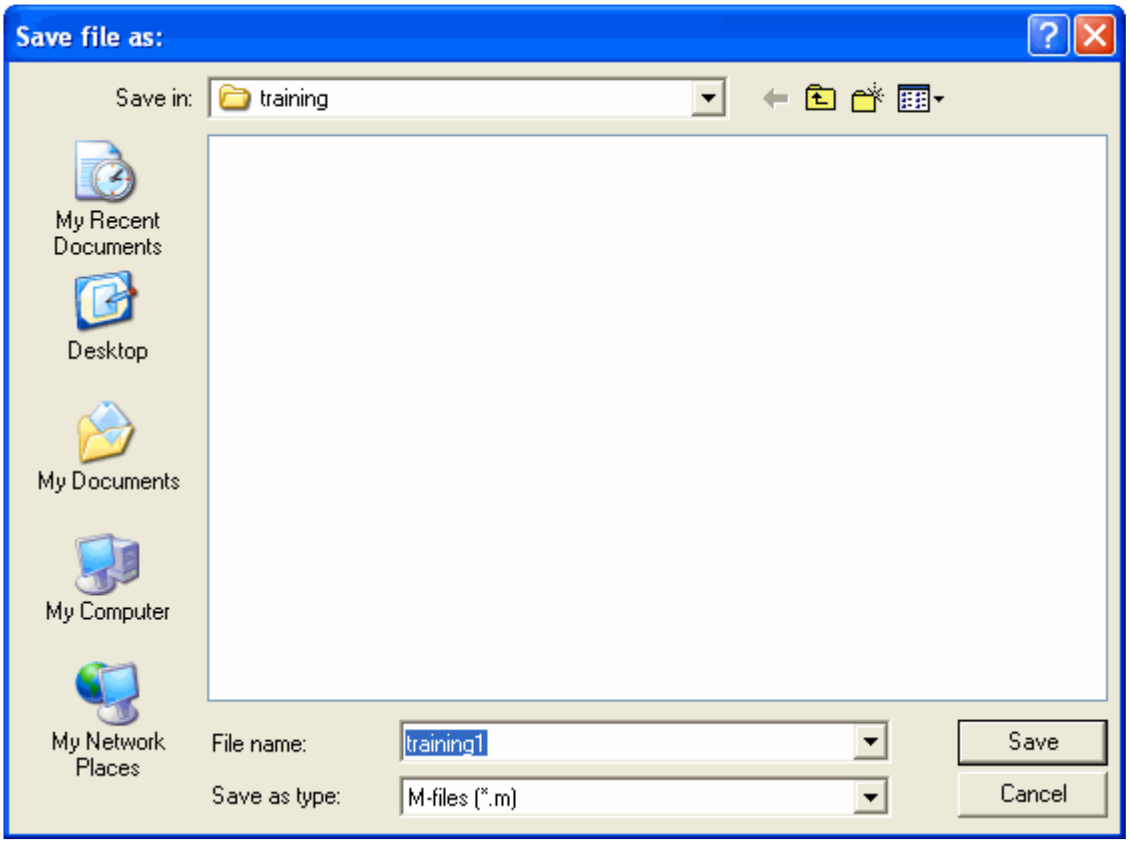
يجب مراعاة تلك الشروط وإلا لن يقوم الماتلاب بتنفيذ البرنامج

فالنقم بتنفيذ المثال المكتوب الآن في النافذة السابقة

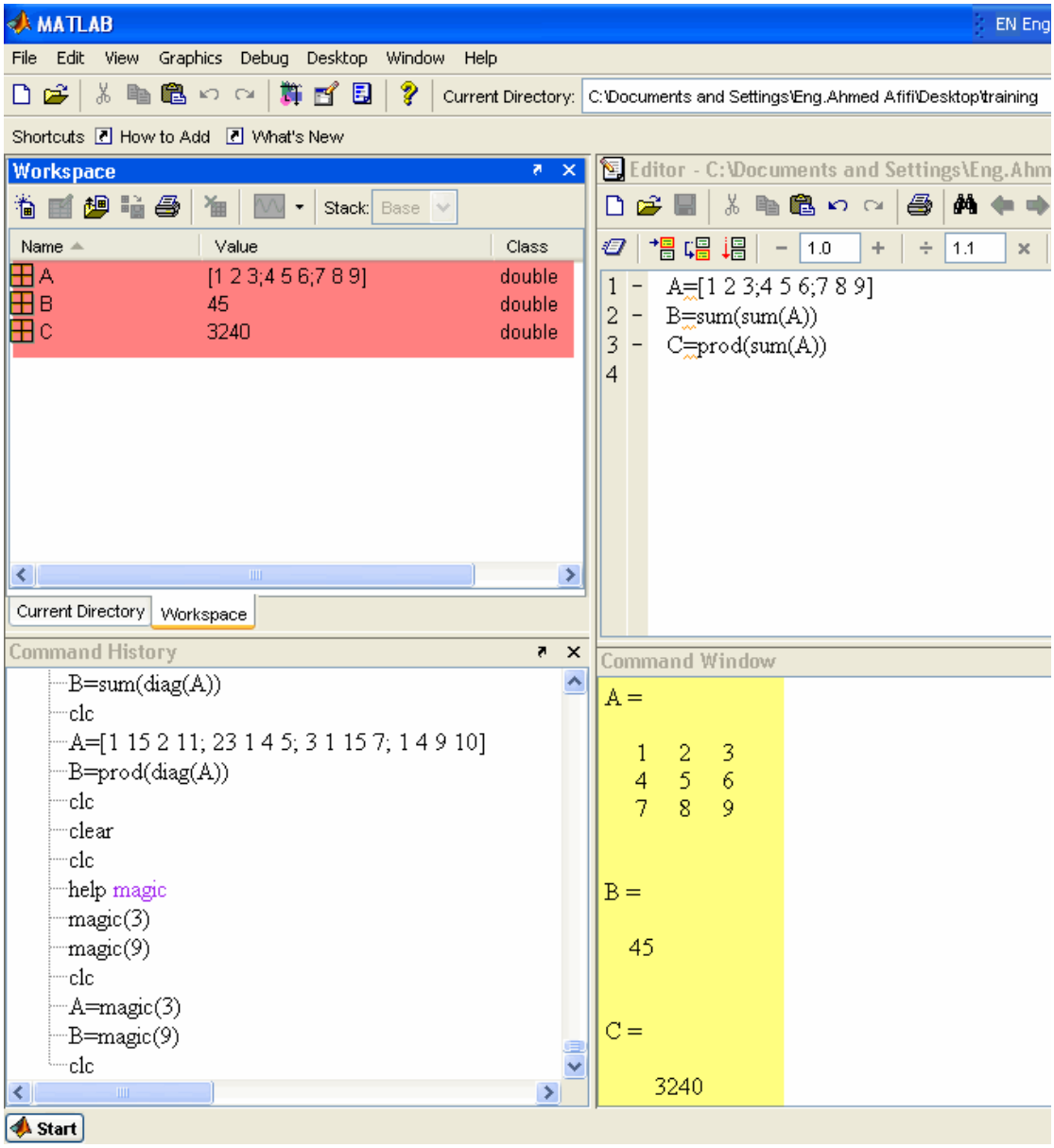
1- يتم الضغط على زر التشغيل كما هو واضح في الصورة التالية



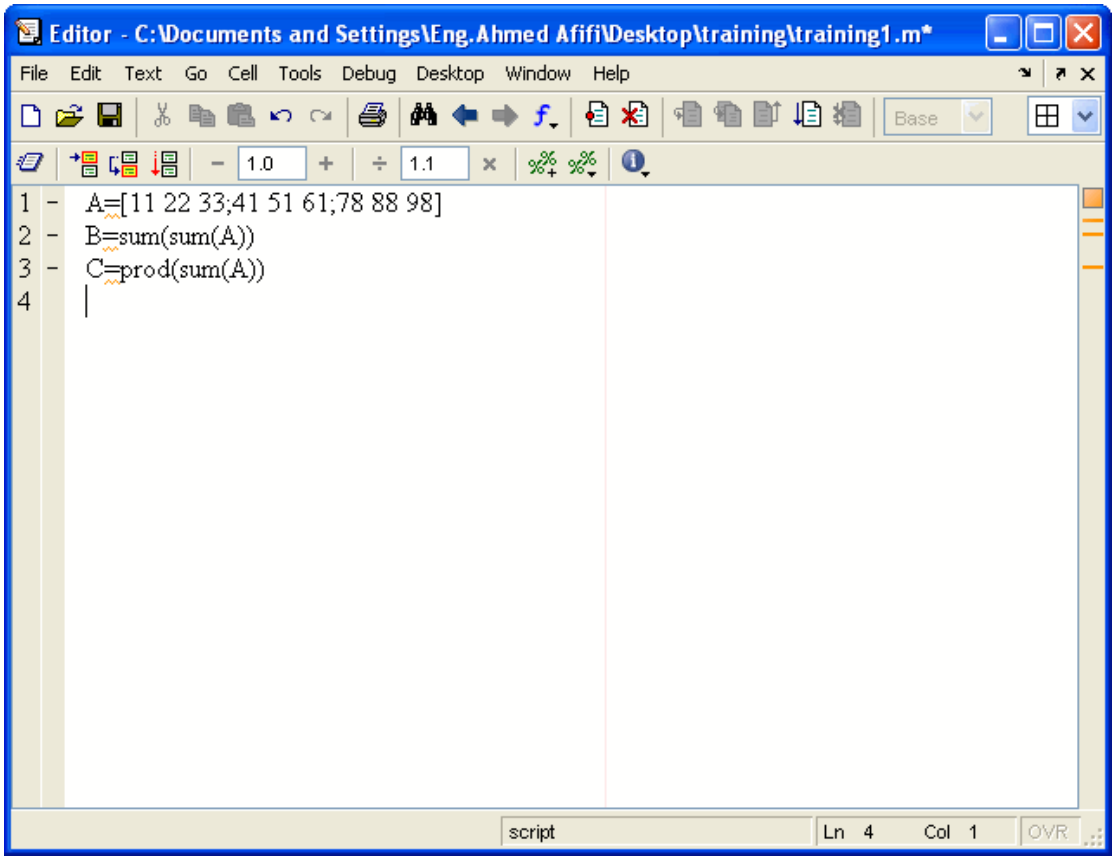
2-سيطالبنا الماتلاب بحفظ البرنامج أولاً، ولنسميه training1



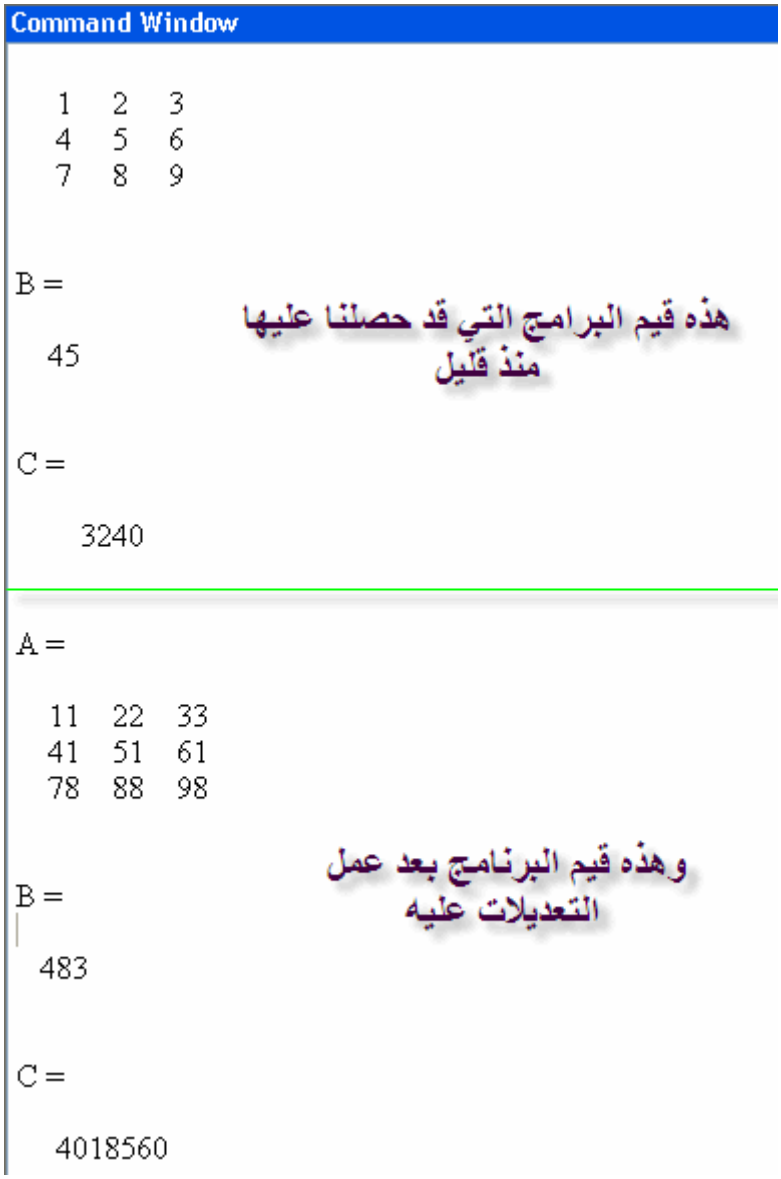
3-ستظهر القيم في كلاً من Command Window and Workspace



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

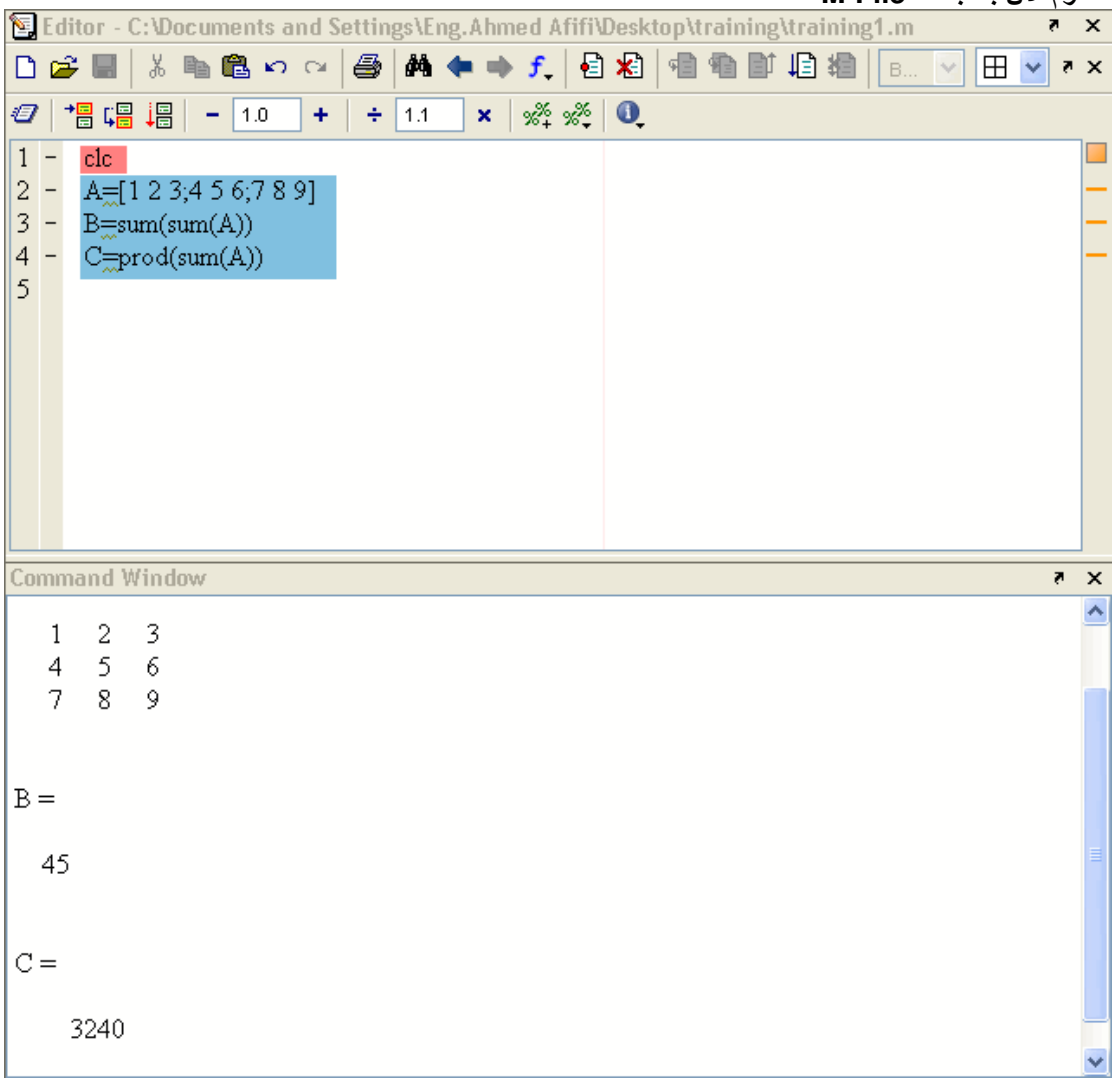


5-سنقوم الآن بتشغيل البرنامج، وسيقوم الماتلاب الآن بالحفظ تلقائياً دون الحاجة لإعادة التسمية، ثم شاهد نافذة الأوامرCommand Window

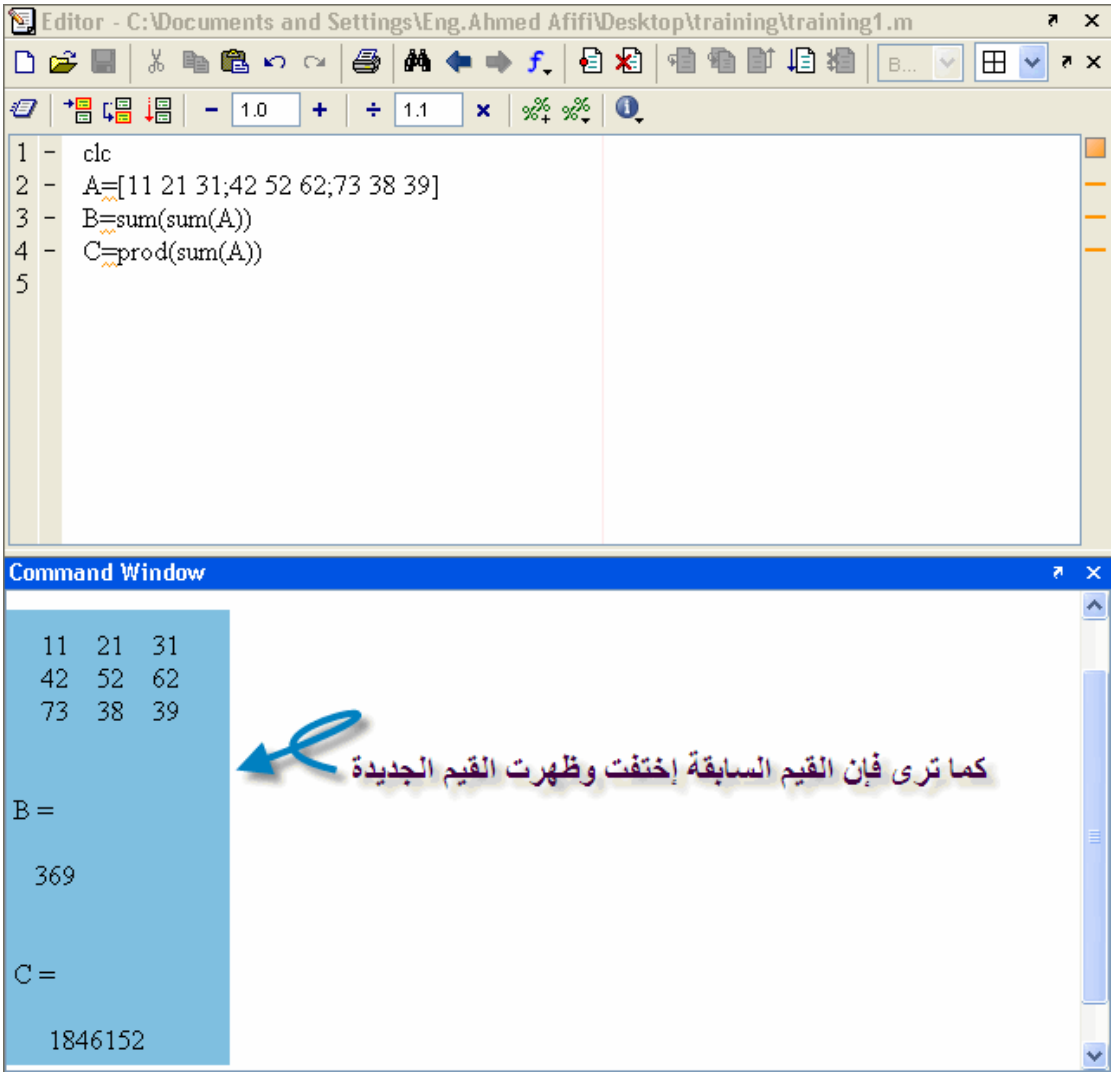


وكما تلاحظ فإنه في كل عملية تحديث للبرنامج ستظل قيم البرنامج القديم موجودة ,فحلاً لهذه المشكلة، يتم وضع الأمر CLCفي أول كل برنامج، وهذا يكون مبدأ في جميع البرامج التي نقوم بعملها لابد من أن تبدأ بهذا الأمر.
ودعونا نقوم بمثال يوضح لنا ذلك

سنقوم الآن بكتابة الـــM-File

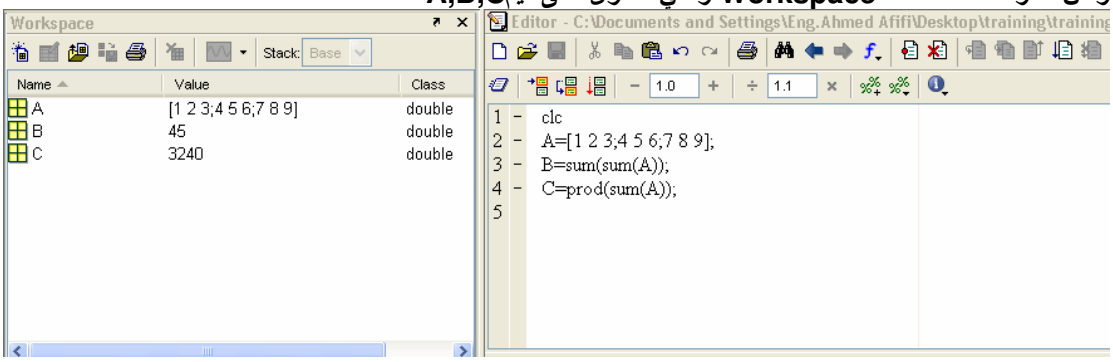


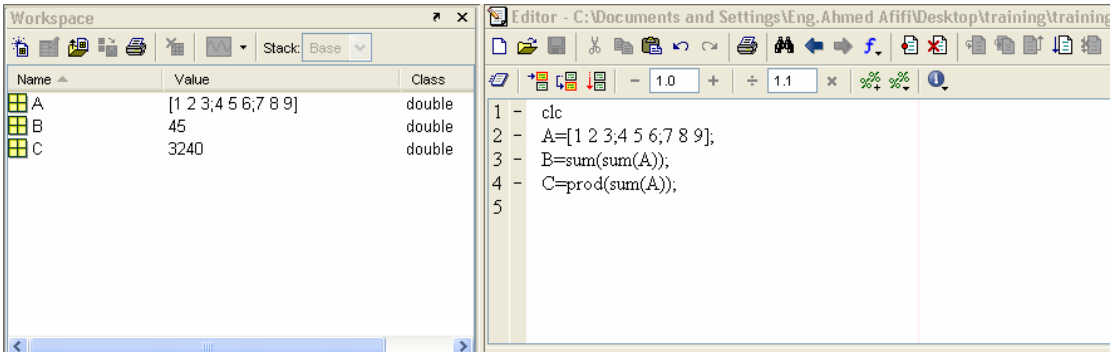
سنقوم الآن بتعديل المثال، وحتى نتأكد أن أمر **CLC** يعمل، ستختفي القيم من **Command Window** وتظهر القيم الجديدة



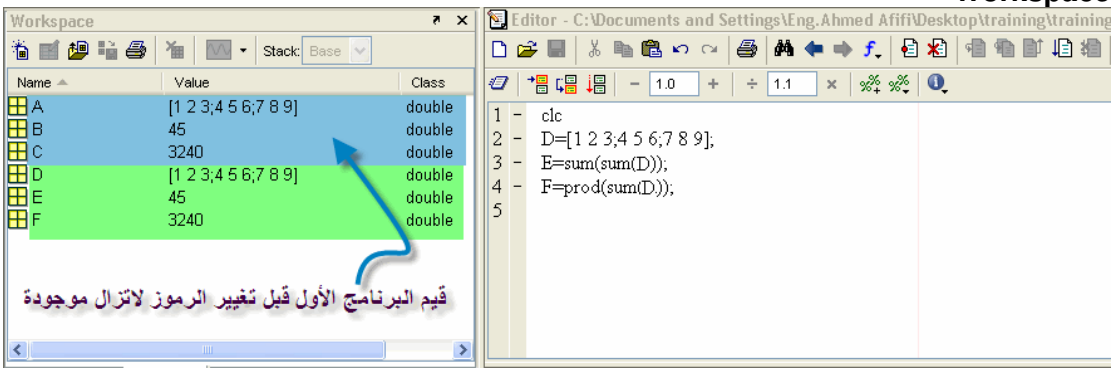
وبهذا نتأكد من أن الأمر CLC يعمل بكفاءة

ولكن دعونا نشاهد نافذة Workspace والتي تحتوى على قيمA,B,C

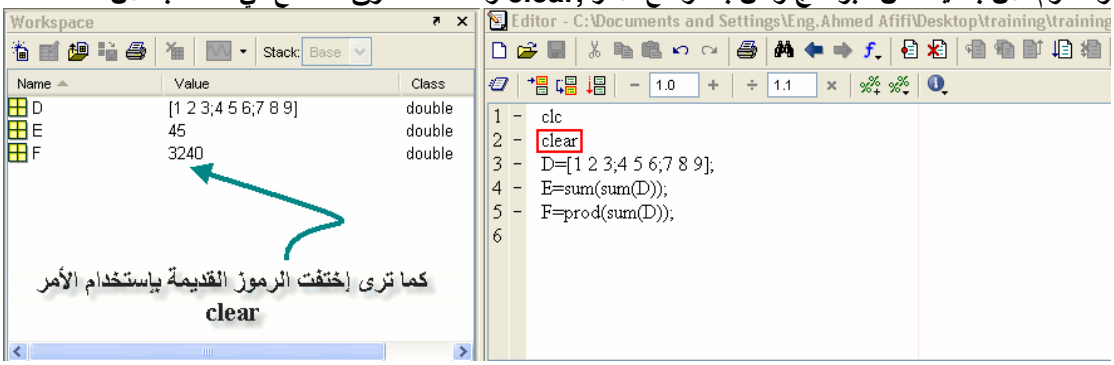




لنقم بتعديل بسيط في البرنامج عن طريق تغيير الرموز فقط من A,B,C إلى D,E,F ومشاهدة النافذة Workspace



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

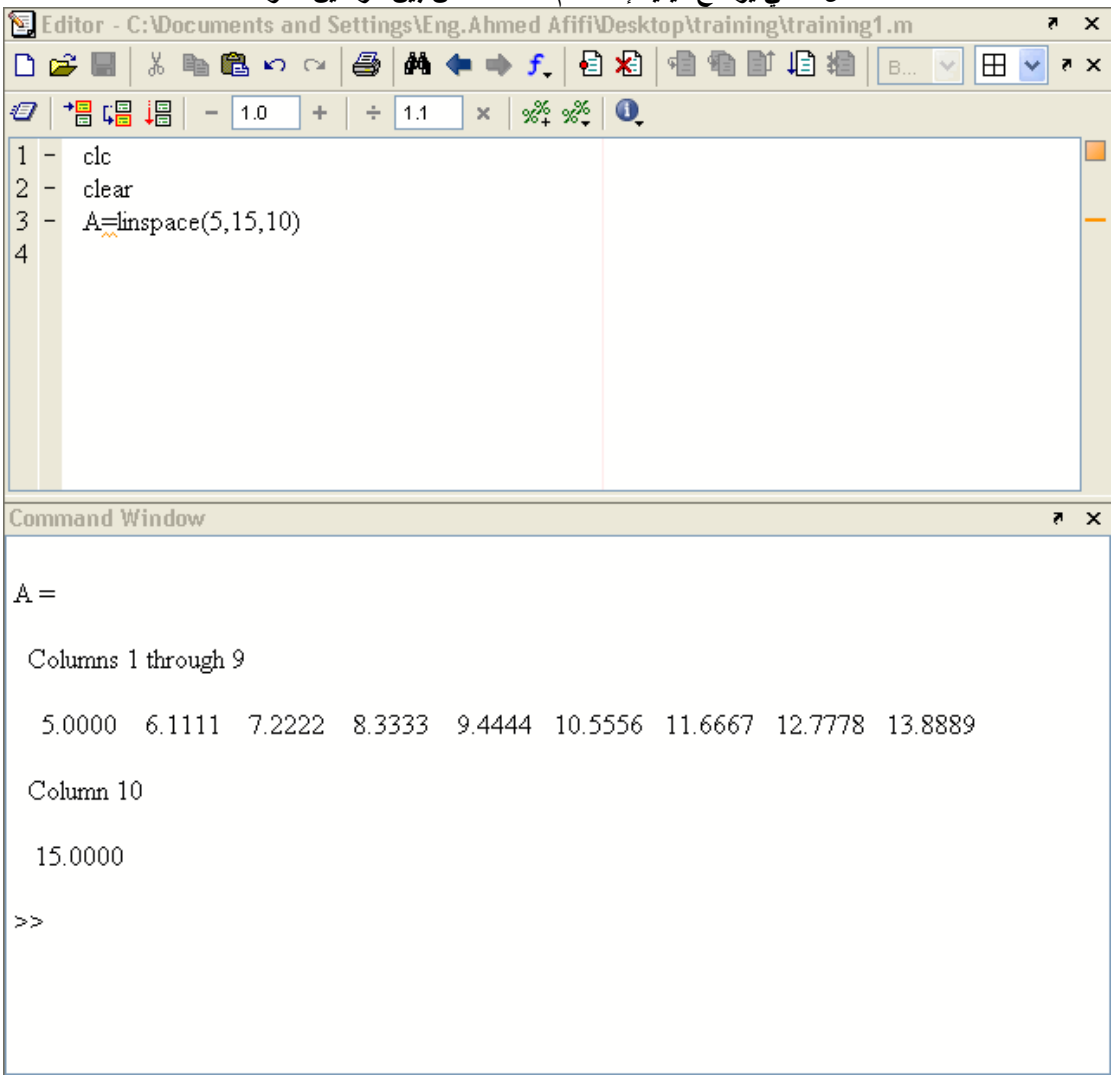
linspace

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



وهذا ما يسمى النظام العشوائي للماتلاب**Randomization System in Matlab**

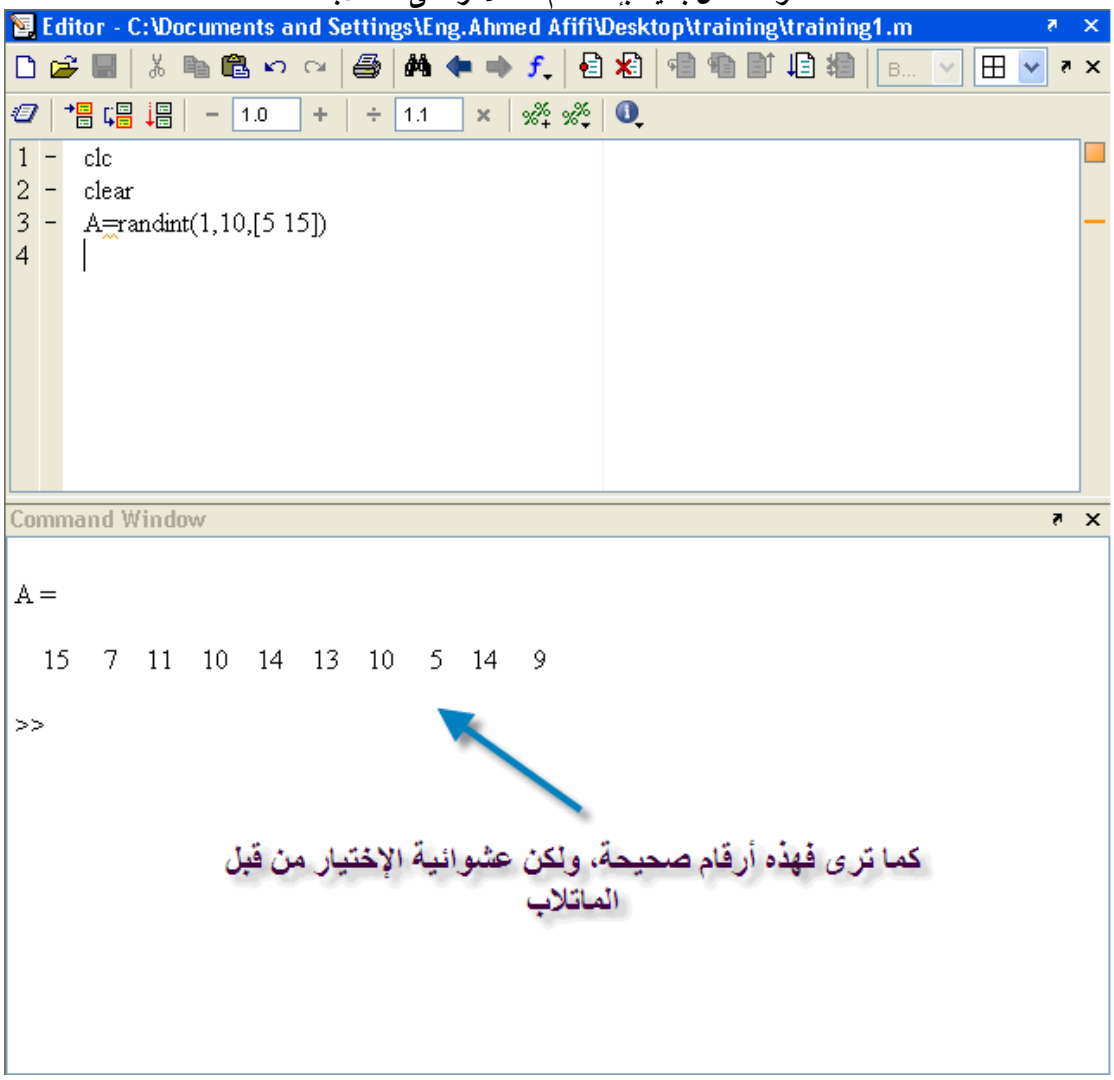
والميزة في هذا النظام هو أن المتجه نظام عشوائي متزايد.

randint

هذا الأمر من ضمن الأوامر والتي تنشأ نظام عشوائي للأرقم، 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])

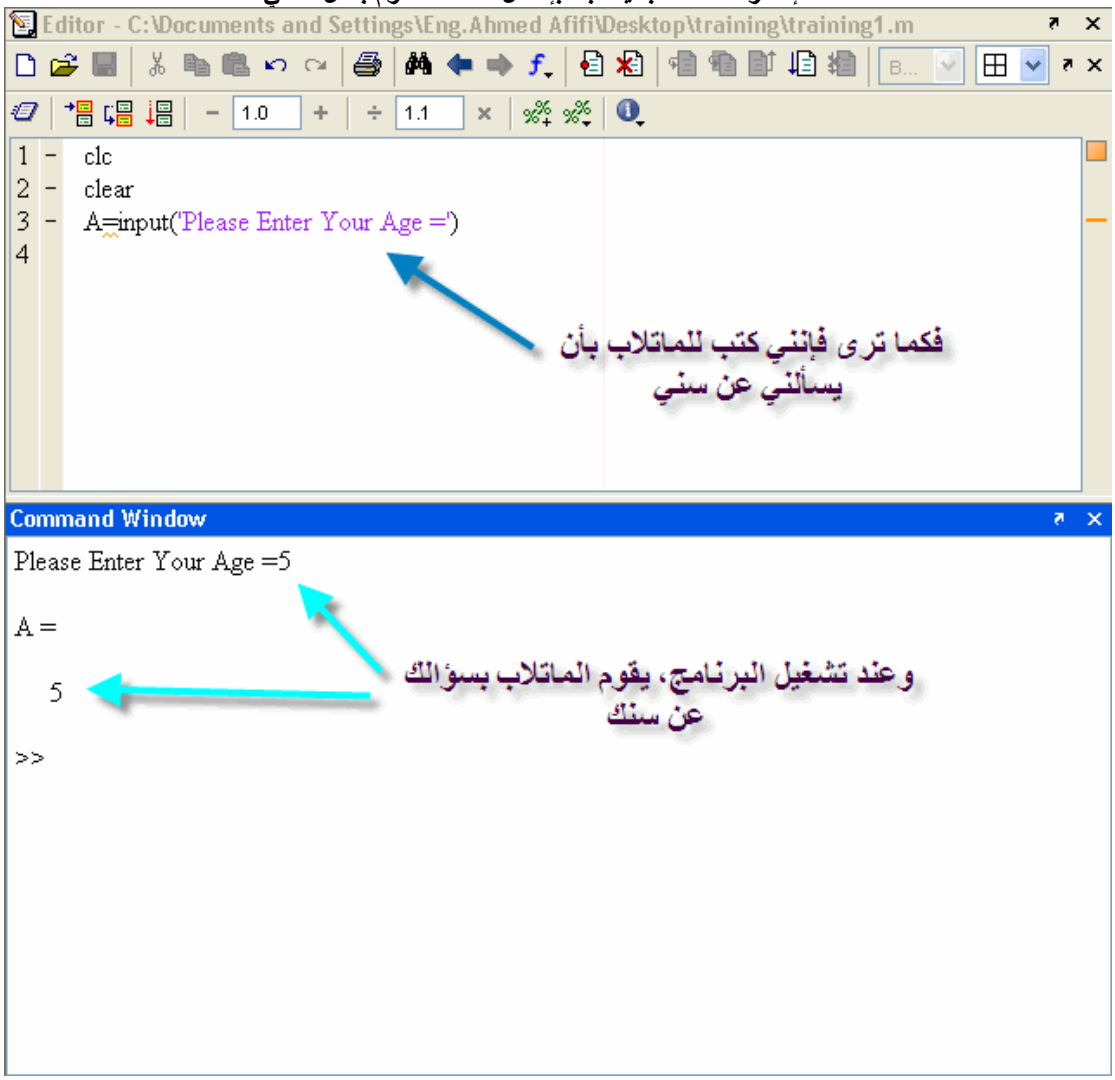
وهذا مثال بسيط بإستخدام هذا الأمر على الماتلاب



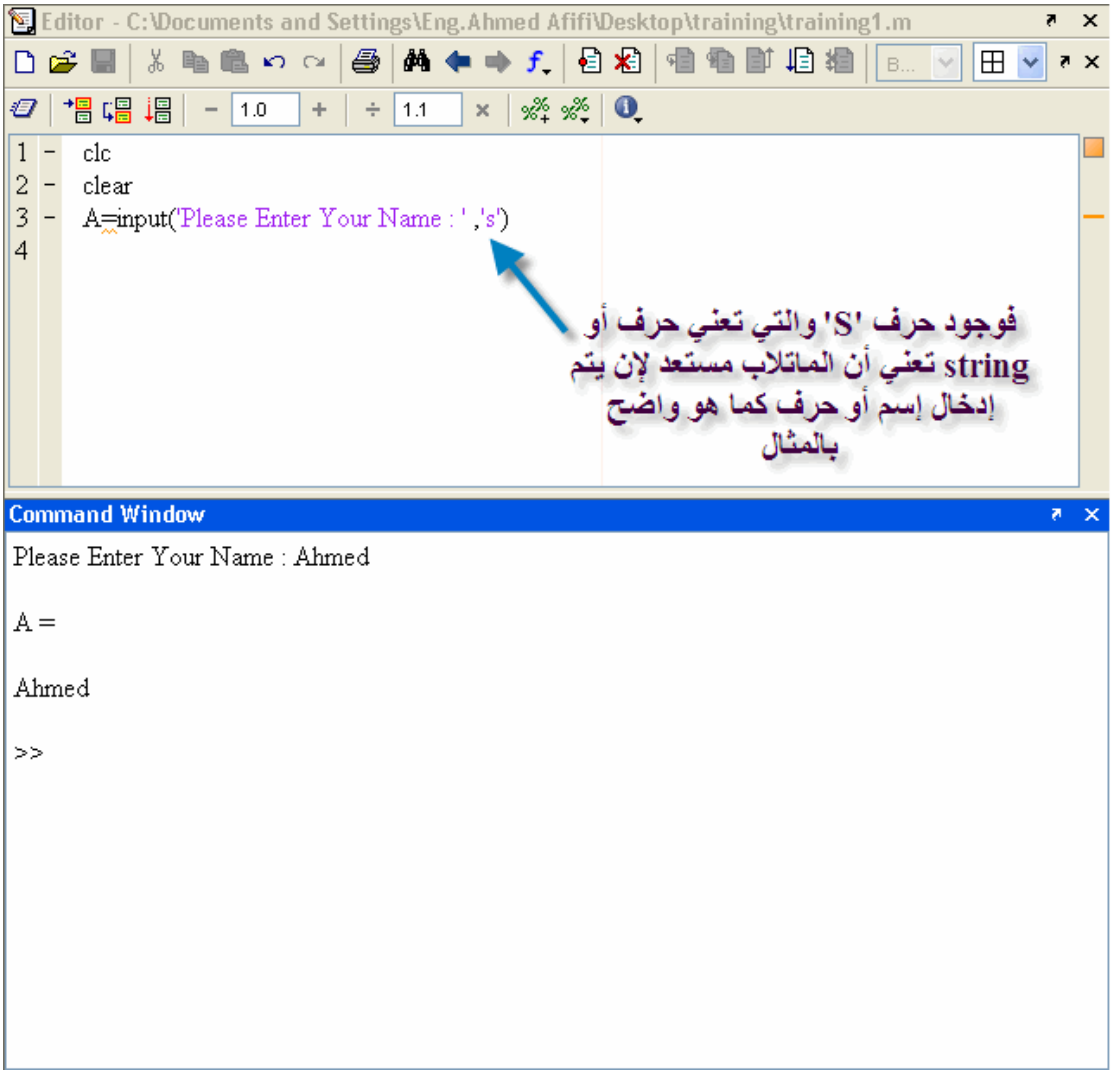
Input

هذا الأمر هام جداً في الماتلاب، حيث يقوم المستخدم من خلال إستخدامه 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



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



Str2num & num2str

أولاً ما الفرق بين String and Character ؟

string هو حرف أو كلمة في الماتلاب

character هو عبارة عن رقم أو مجموعة من الآرقام

وهنالك أمر يقوم بتحويل string to Character والعكس كذلك

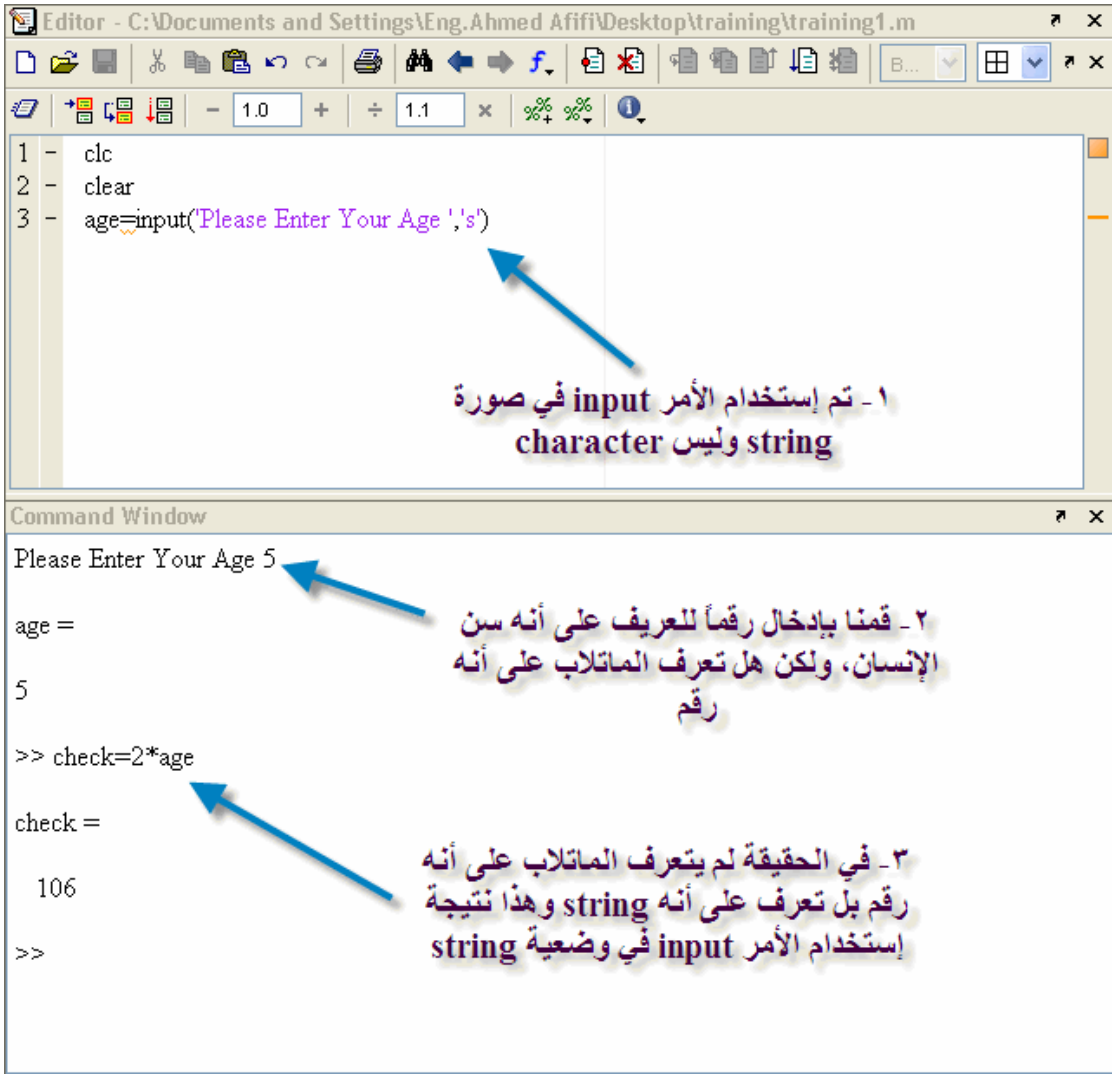
وهما

num2str

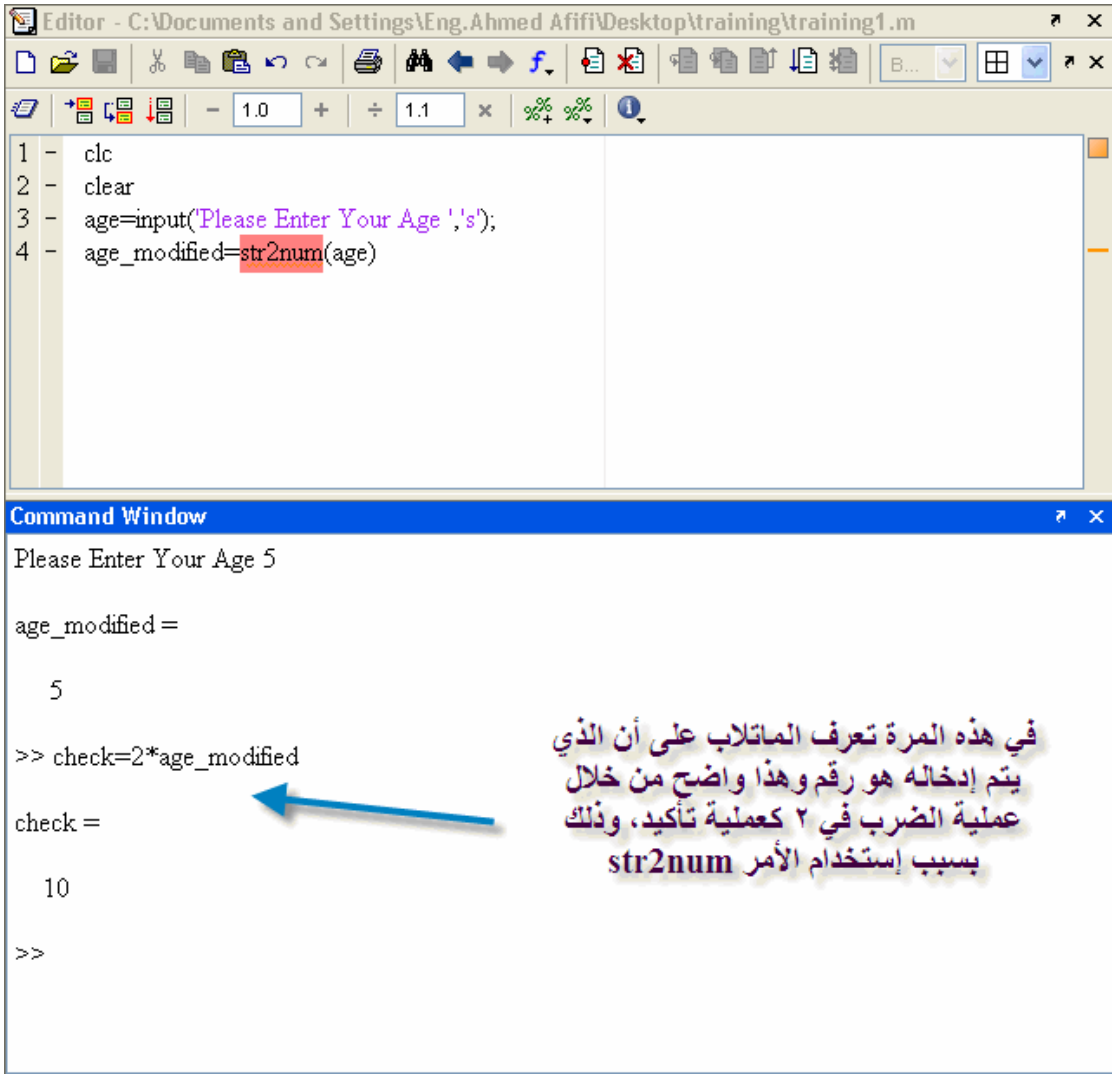
str2num

ولكن فيما يفيدوا هذه الأوامر

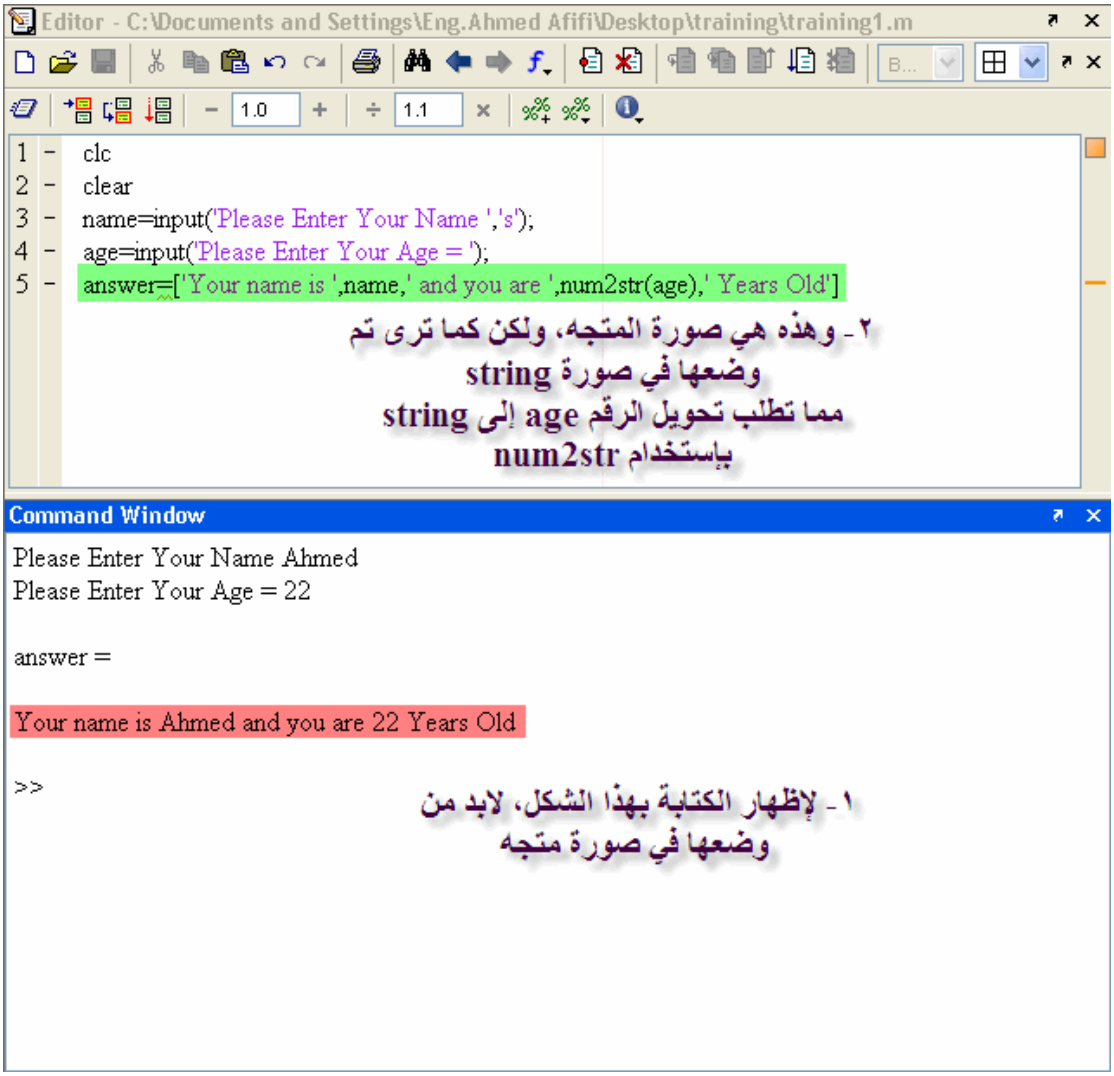
سنقوم بمثال بسيط حتى تفهم المقصود من هذه الأمور، سنقوم بإستخدام الأمر input في وضعية string وسنقوم بإدخال أرقام، ستبدأ تستغرب الآن، فهل الماتلاب حتى بعد إدخالي الرقم سيتعرف عليها كأنها أرقام أم أي شئ غير ذلك، شاهد الصورة التالية


Editor - C:\Documents and Settings\Eng.Ahmed Afifi\Desktop\training\training1.m
1 clc
2 clear
3 age=input('Please Enter Your Age ','s')
١ ـ تم إستخدام الأمر input في صورة character وليس string
Command Window
Please Enter Your Age 5
age =
5
>> check=2*age
check =
106
>>
٢ ـ قمنا بإدخال رقماً للعريف على أنه سن الإنسان، ولكن هل تعرف الماتلاب على أنه رقم
٣ ـ في الحقيقة لم يتعرف الماتلاب على أنه رقم بل تعرف على أنه string وهذا نتيجة إستخدام الأمر input في وضعية string

سنقوم الآن بإستخدام **str2num** لتحويل **string** إلى **character** أو رقم، شاهد الصورة التالية



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



والآن نحن على إتم إستعداد للبدء في الجزء المتعلق بالرسم الثنائي 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 independentوالآخر يسمى dependent

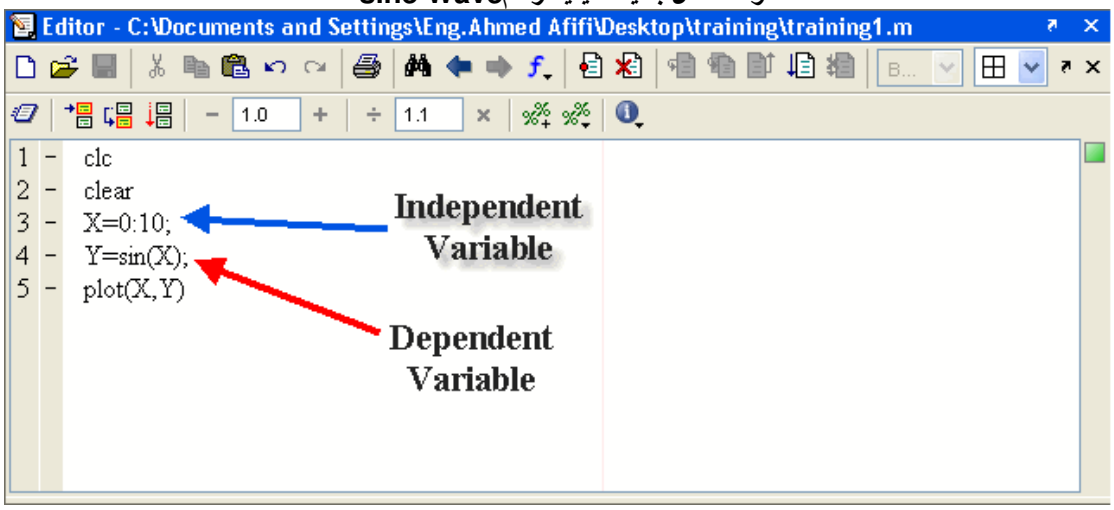
فما المقصود بـــ independent Variable أي المتغير المستقل أي أن قيمه لا تحكمها علاقة

بينما dependent Variable أي المتغير المُعتمد حيث يعتمد قيمه على قيم المتغير المستقل

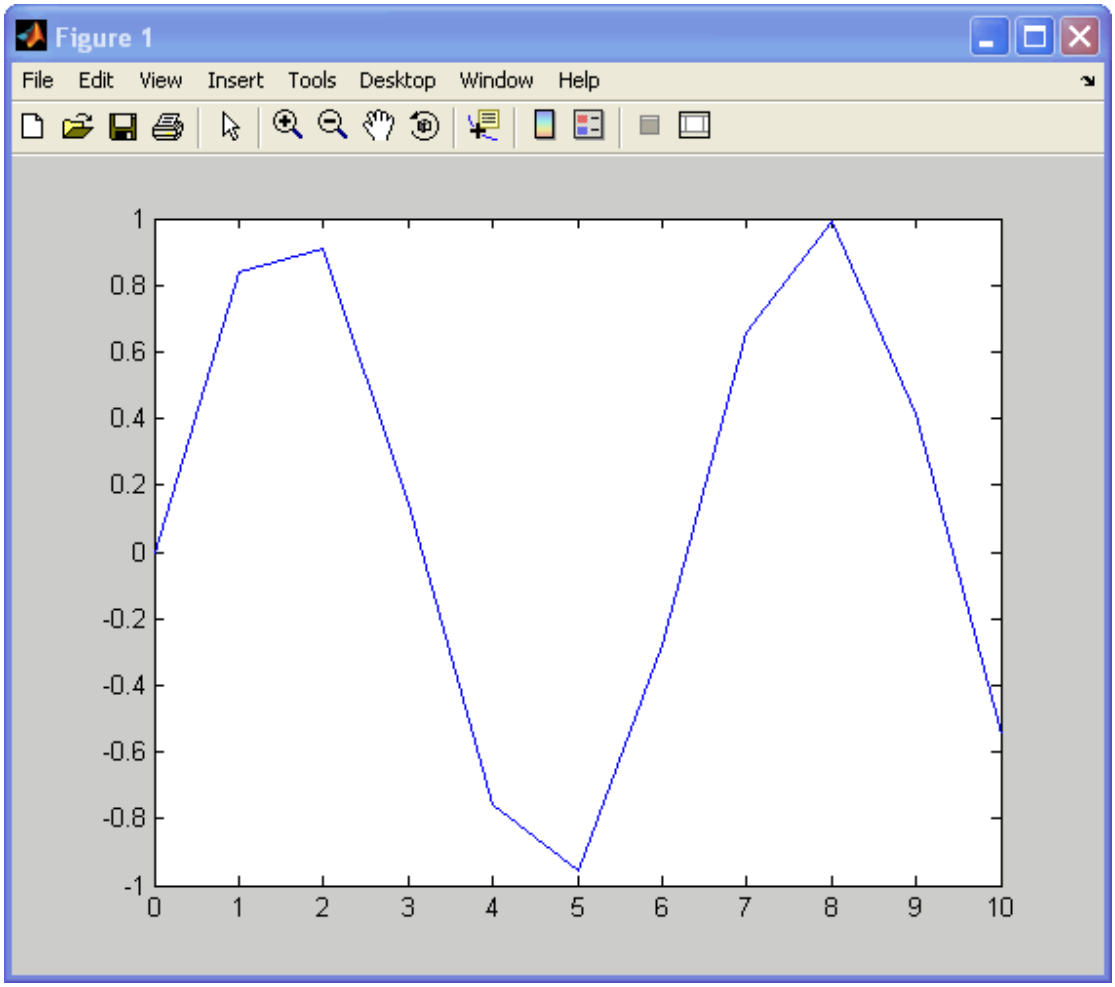
الآن وبعد شرح هذين العنصرين الهامين، سنقوم بشرح الأمر plot حيث يأخذ الصورة التالية

plot(independent variable, dependent variable)

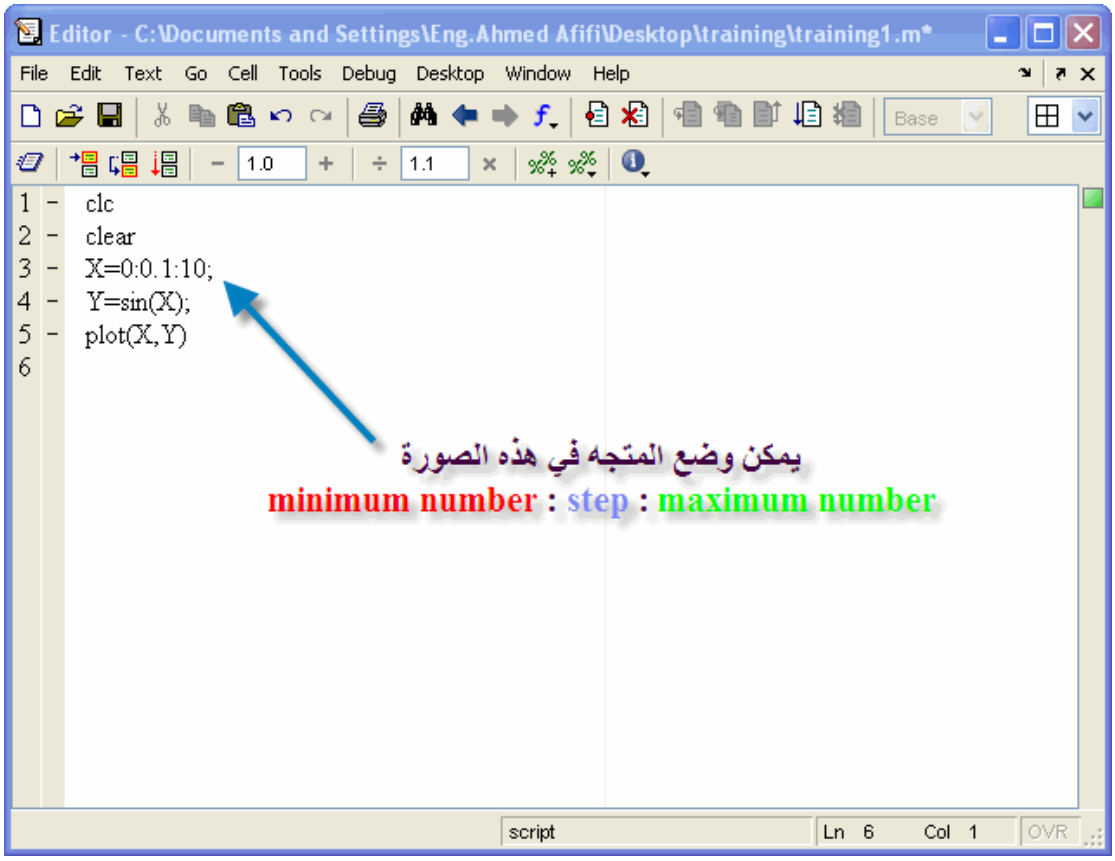
وهذا مثال بسيط لكيفية رسم sine Wave



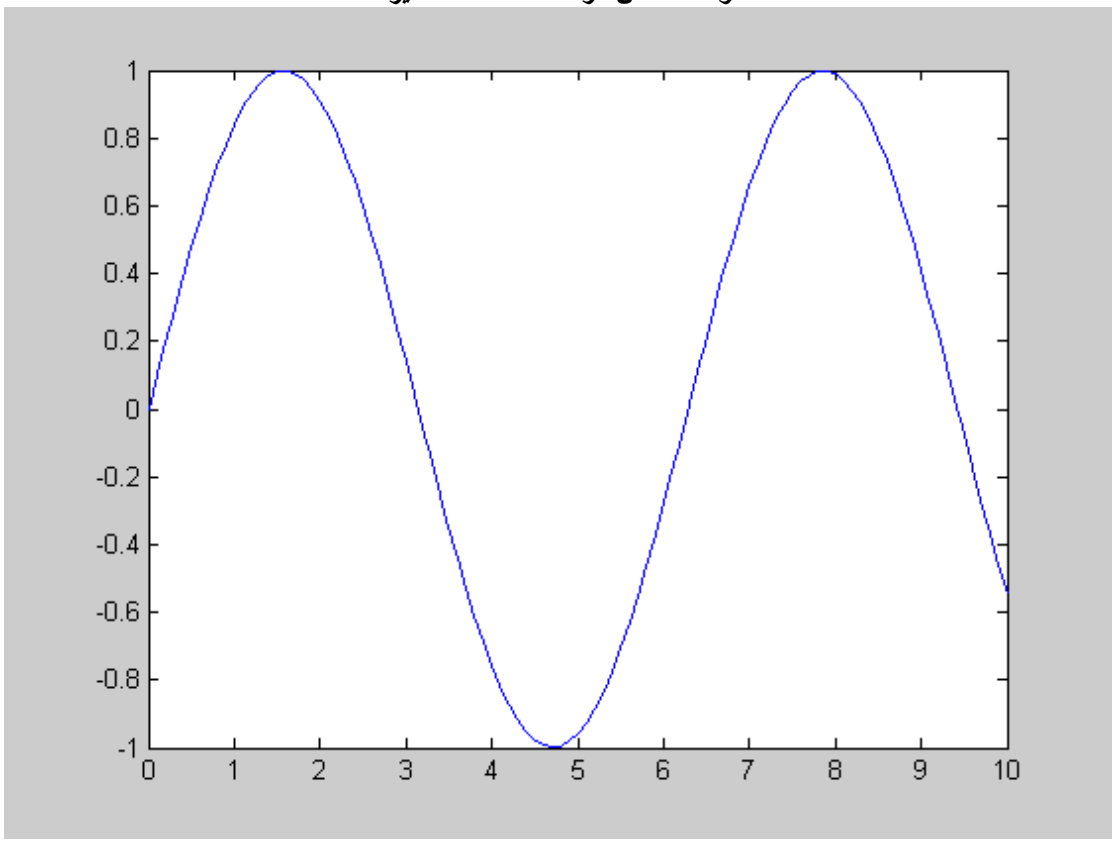
وتلاحظ أننا قد إخترنا ١٠ نقاط فقط لرسم Sine Wave , وهذا عدد قليل لرسم Sine Wave وتلاحظ ظهور الرسمة بالشكل التالي



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



وستلاحظ أن الرسمة قد تحسنت كثيراً



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

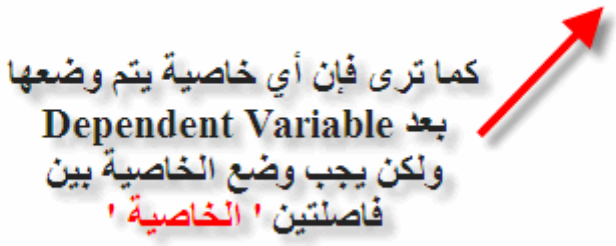
إضافة خصائص إلى الرسومات داخل الماتلاب

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 ونقاك وغيرها، وهذه هي مجموعة الخصائص التي تتم من خلال الماتلاب

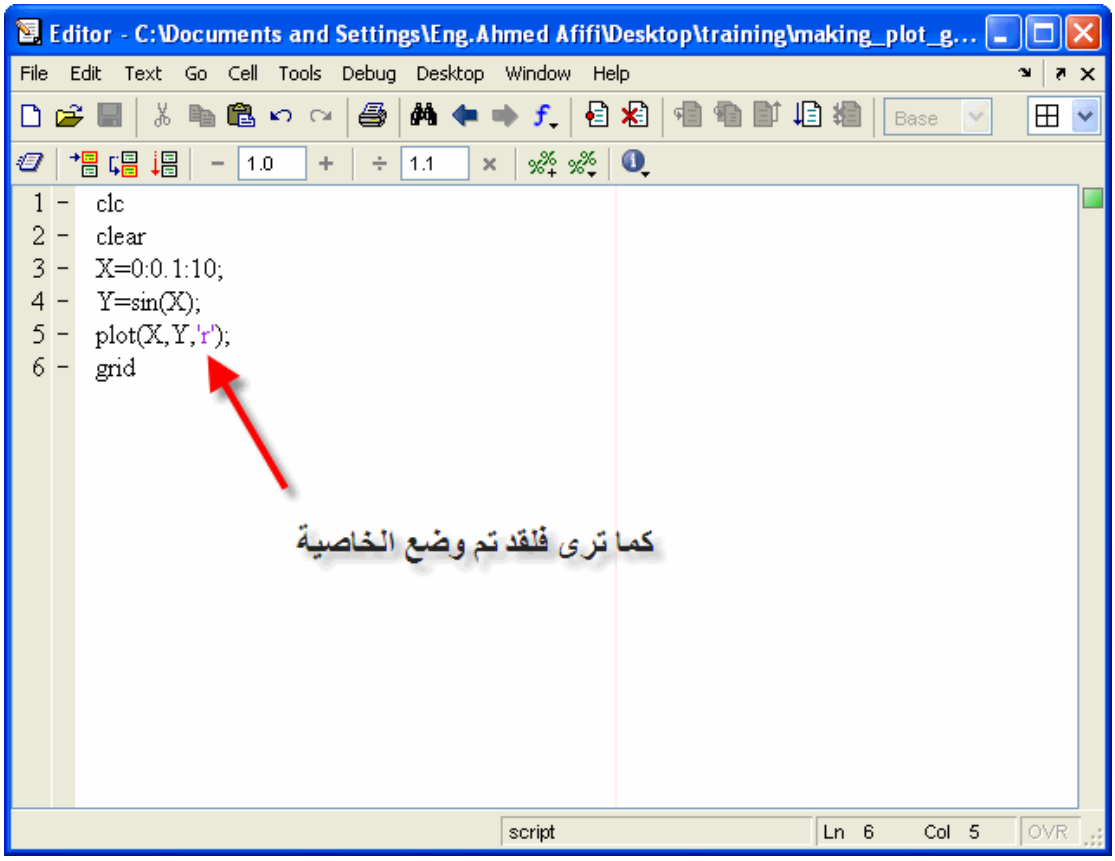
b	blue	.	point	-	solid
g	green	o	circle	:	dotted
r	red	x	x-mark	-.	dashdot
c	cyan	+	plus	--	dashed
m	magenta	*	star	(none)	no line
y	yellow	s	square		
k	black	d	diamond		
		v	triangle (down)		
		^	triangle (up)		
		<	triangle (left)		
		>	triangle (right)		
		p	pentagram		
		h	hexagram		

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

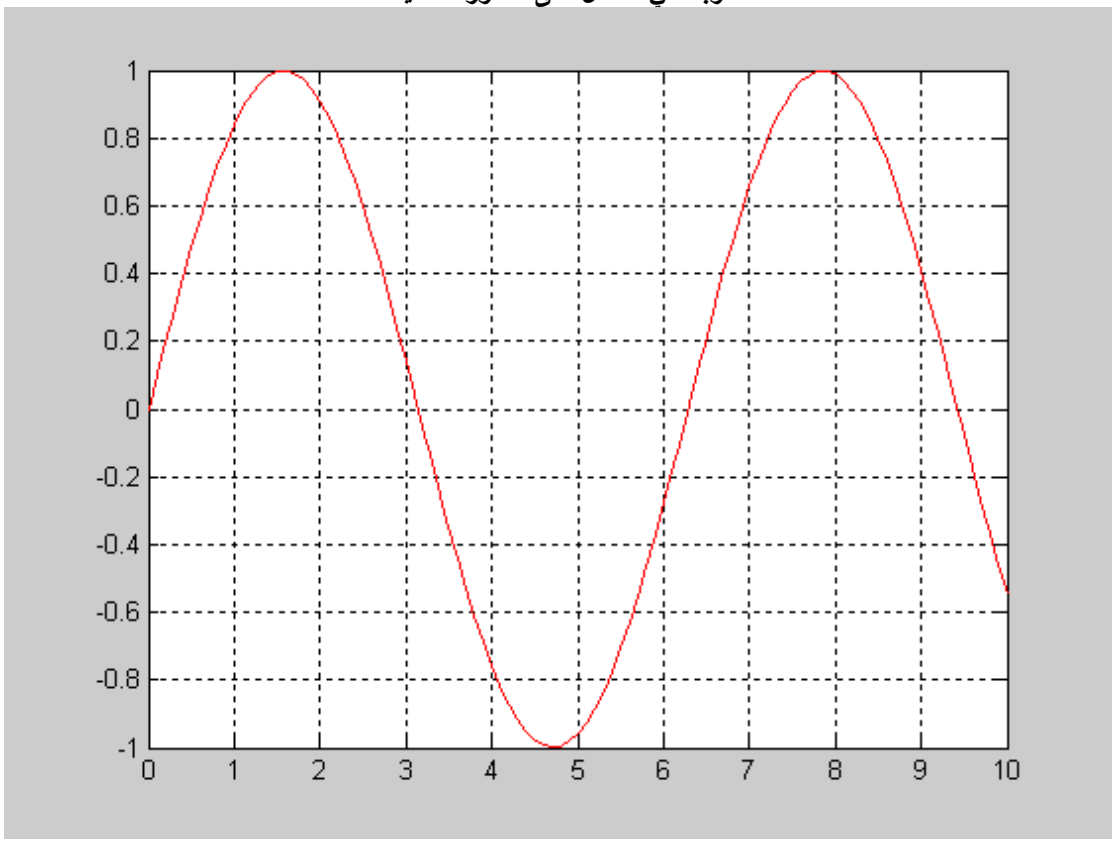
plot(independent Variable, Dependent Variable, ' the property ')



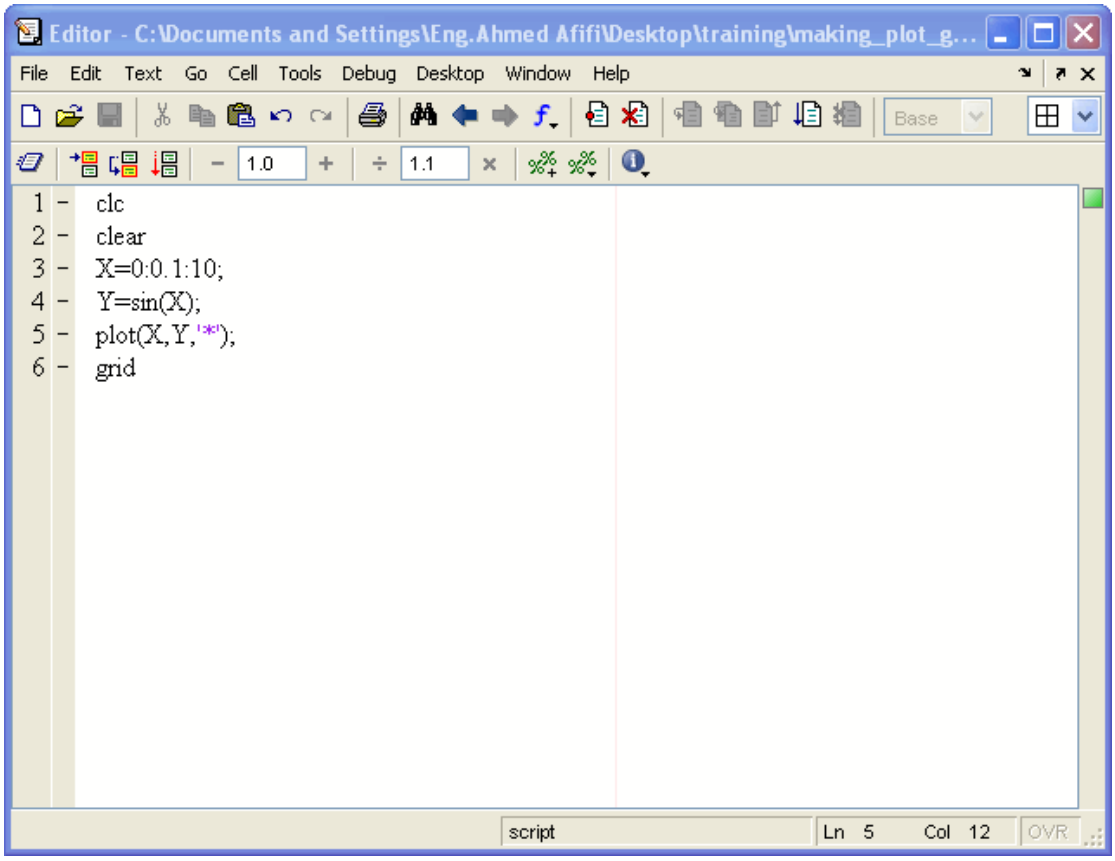
إعتماداً على المثال السابق أخذه سنقوم بتعديل بعض الخصائص
سنقوم مثلاً بتغيير لون الخط إلى الأحمر



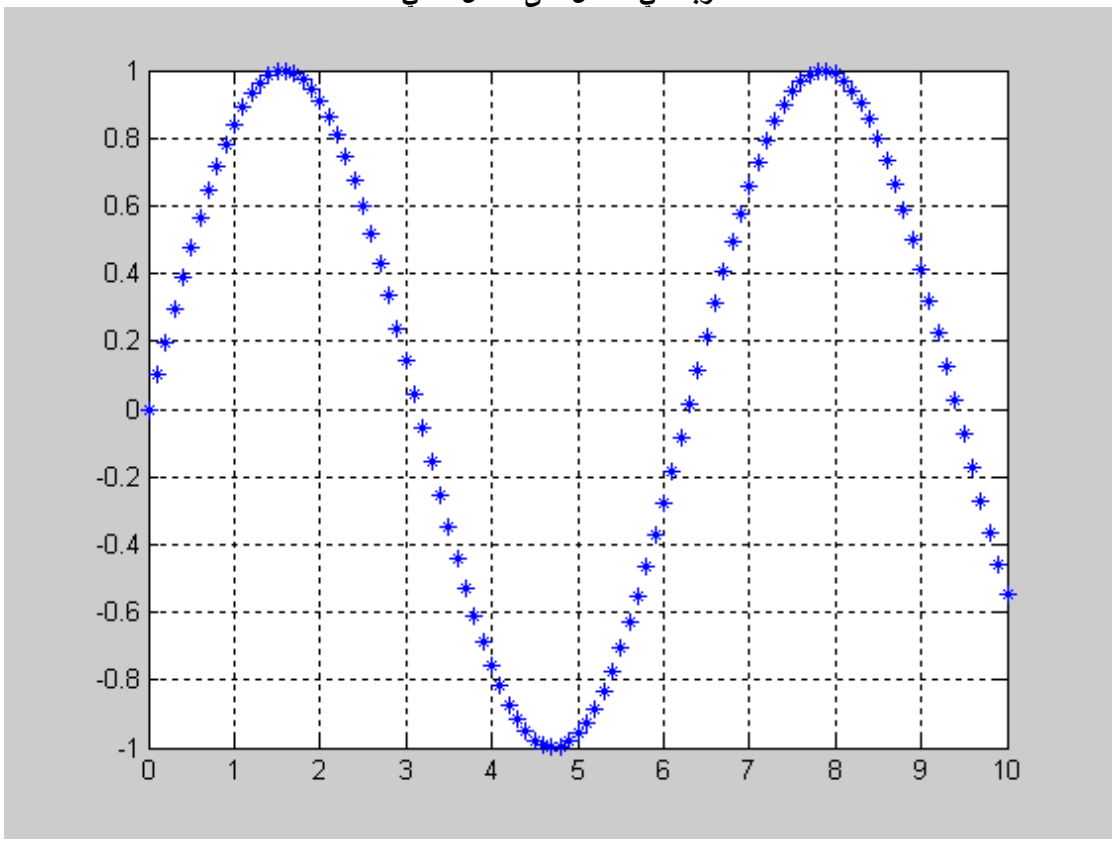
وبالتالي نحصل على الصورة التالية



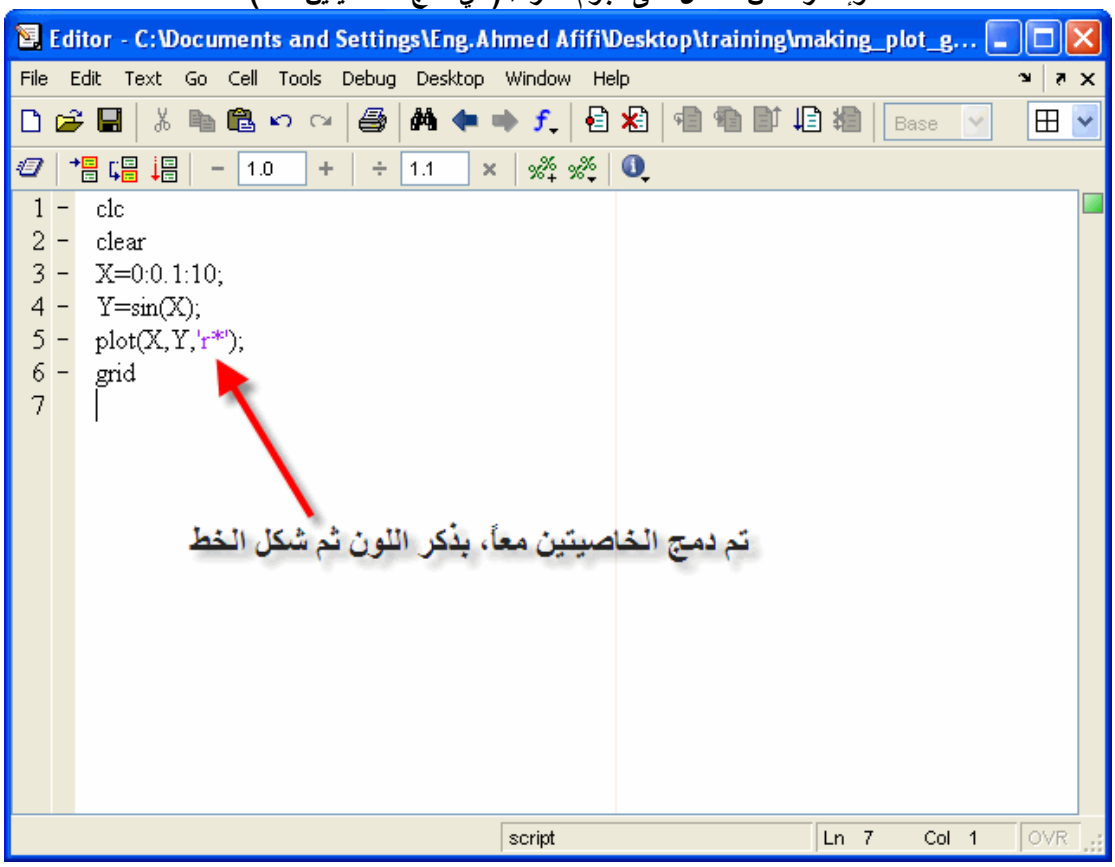
ولنقم الآن بإضافة خاصية جديدة بأن يكون الخط ليس خطاً متصل وإنما عبارة عن نجوم



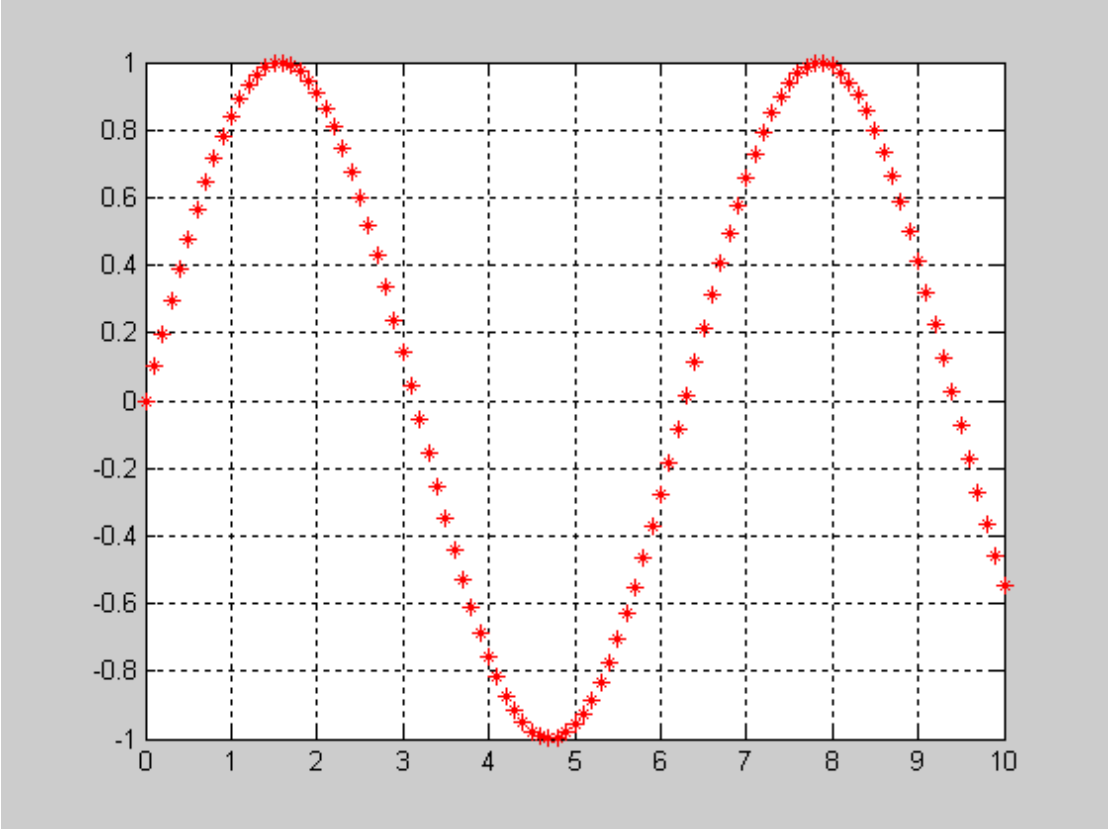
وبالتالي نحصل على الشكل التالي



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



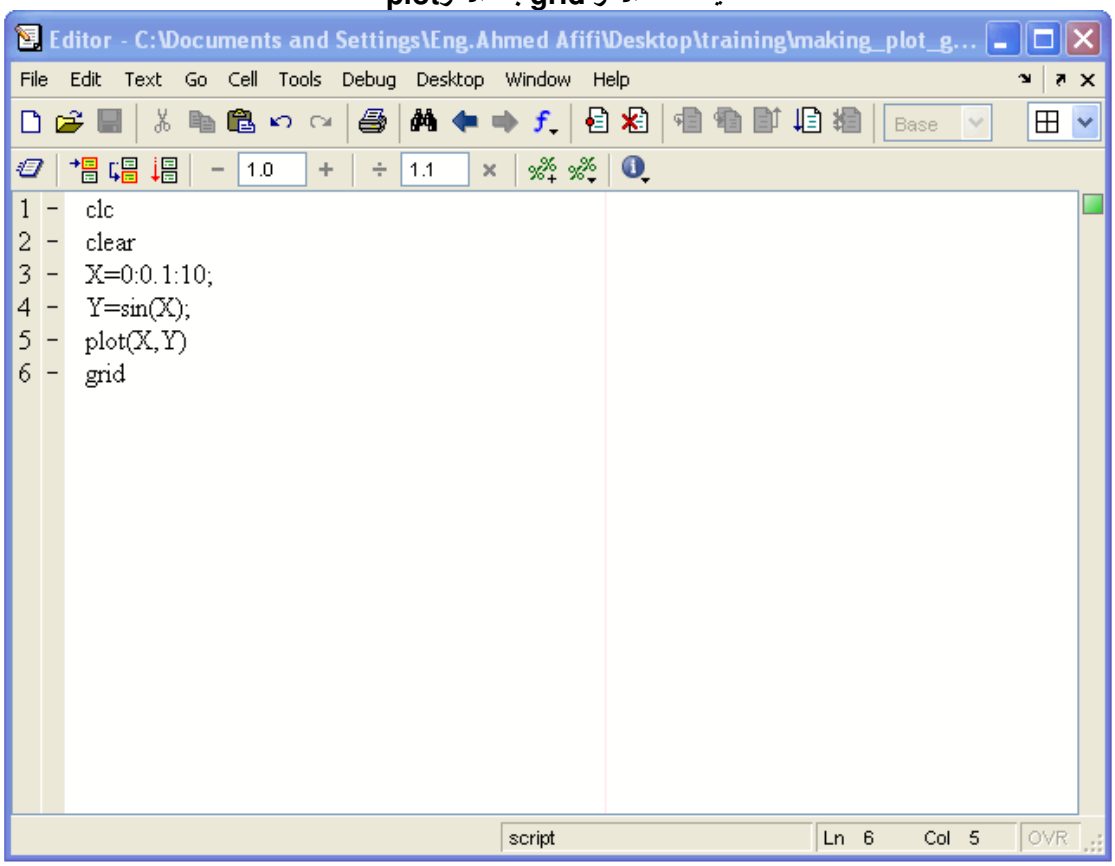
وبالتالي نحصل على الشكل التالي



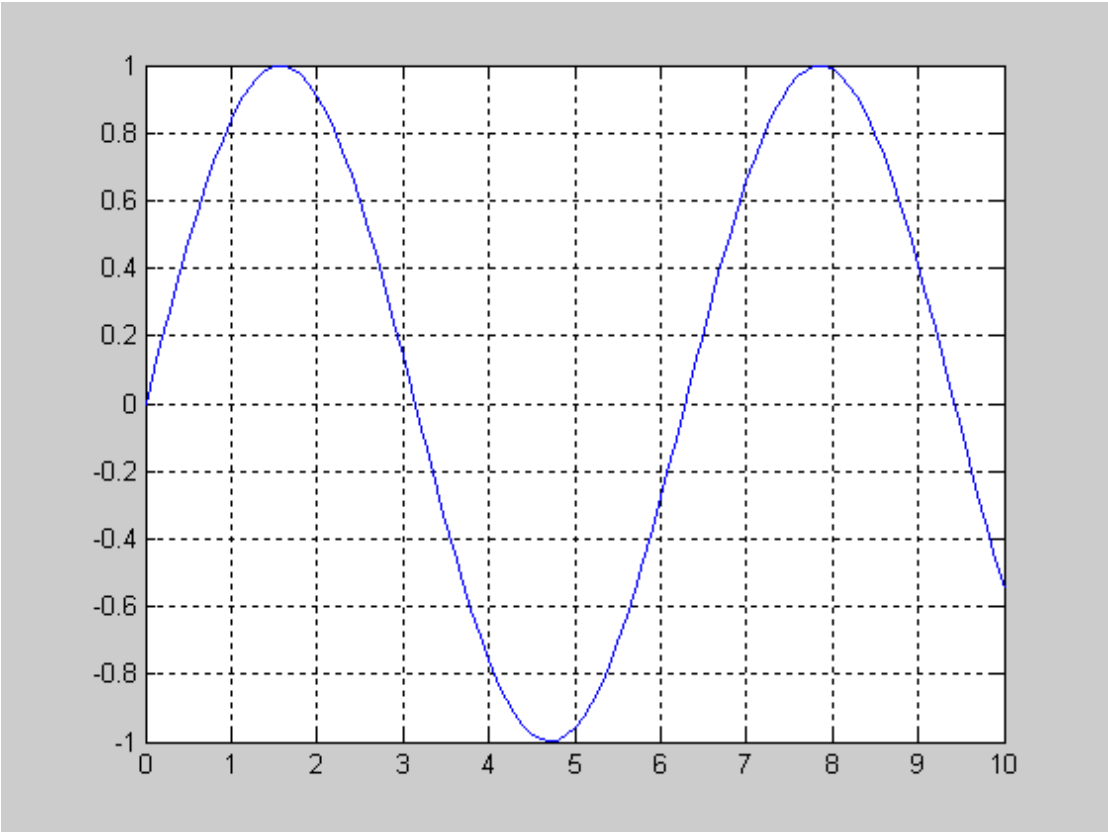
وهنا نكون قد شرحنا خصائص الرسومات داخل الماتلاب

عملية وضع شبكة على الرسم

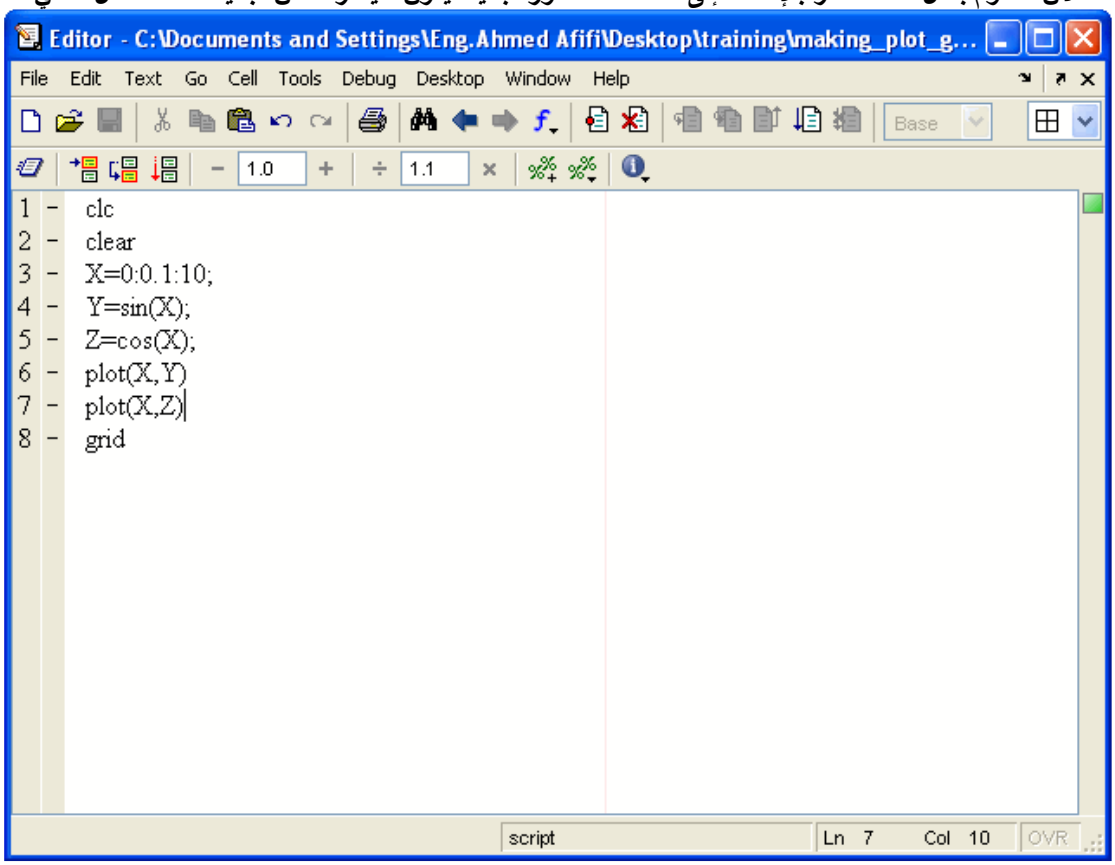
يقوم الماتلاب بوضع شبكة على الرسم، بحيث يكون من السهل تحديد القيم من على الرسم
حيث تأخذ الأمر **grid** بعد الأمر**plot**



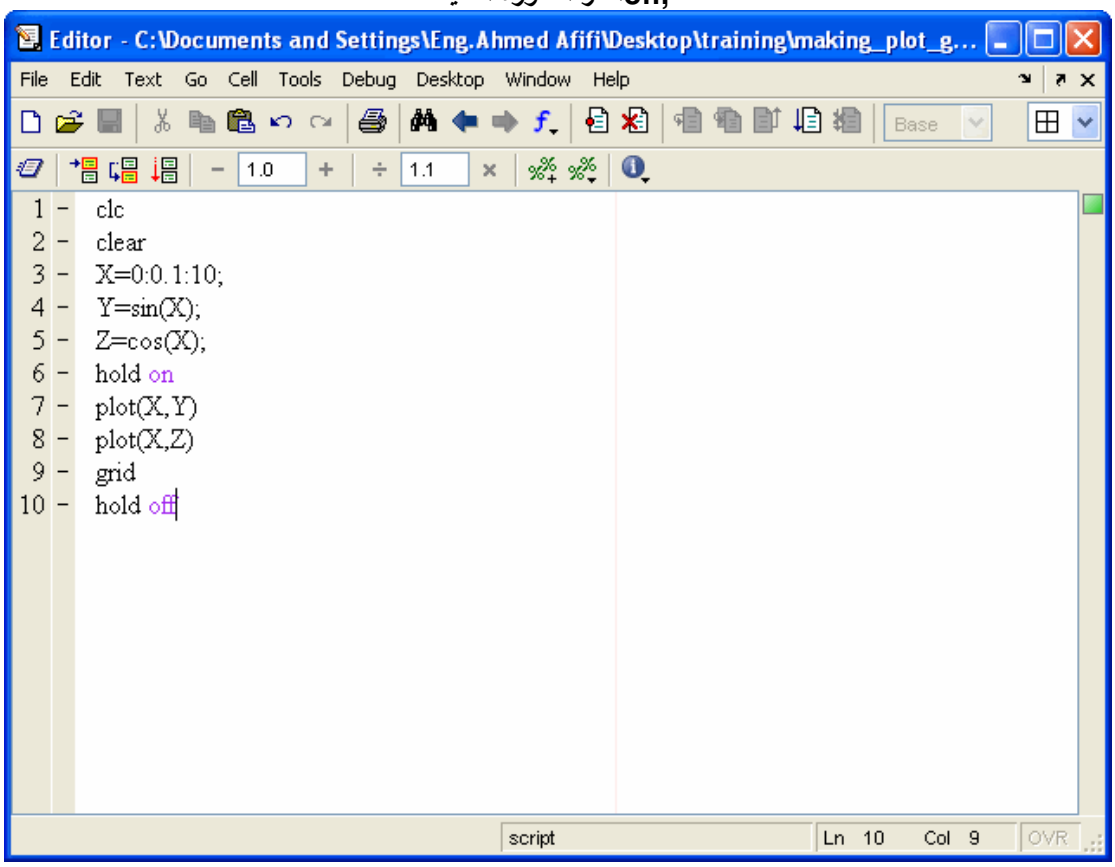
وسيكون شكل الرسم كالتالي



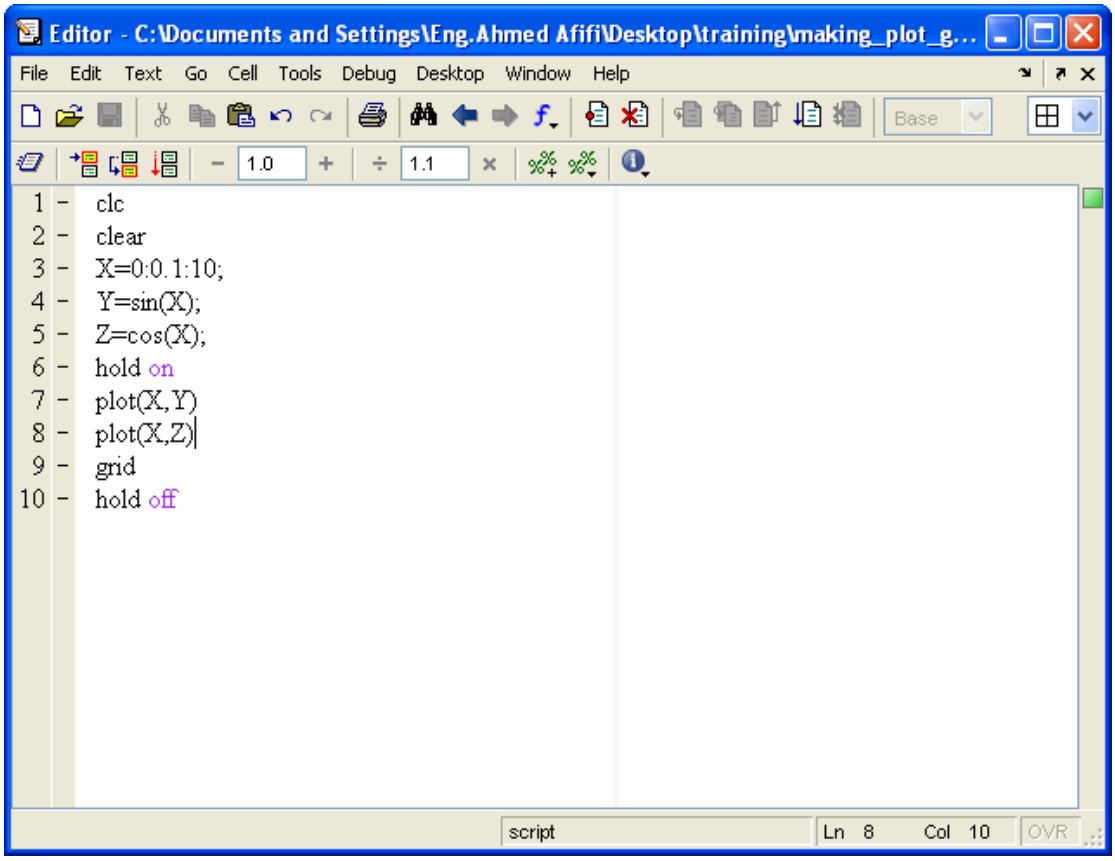
الآن سنقوم بعمل معادلة أخر بالإضافة إلى المعادلة المذكورة بحيث يكون لدينا رسمتان، بحيث تأخذ الشكل التالي



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

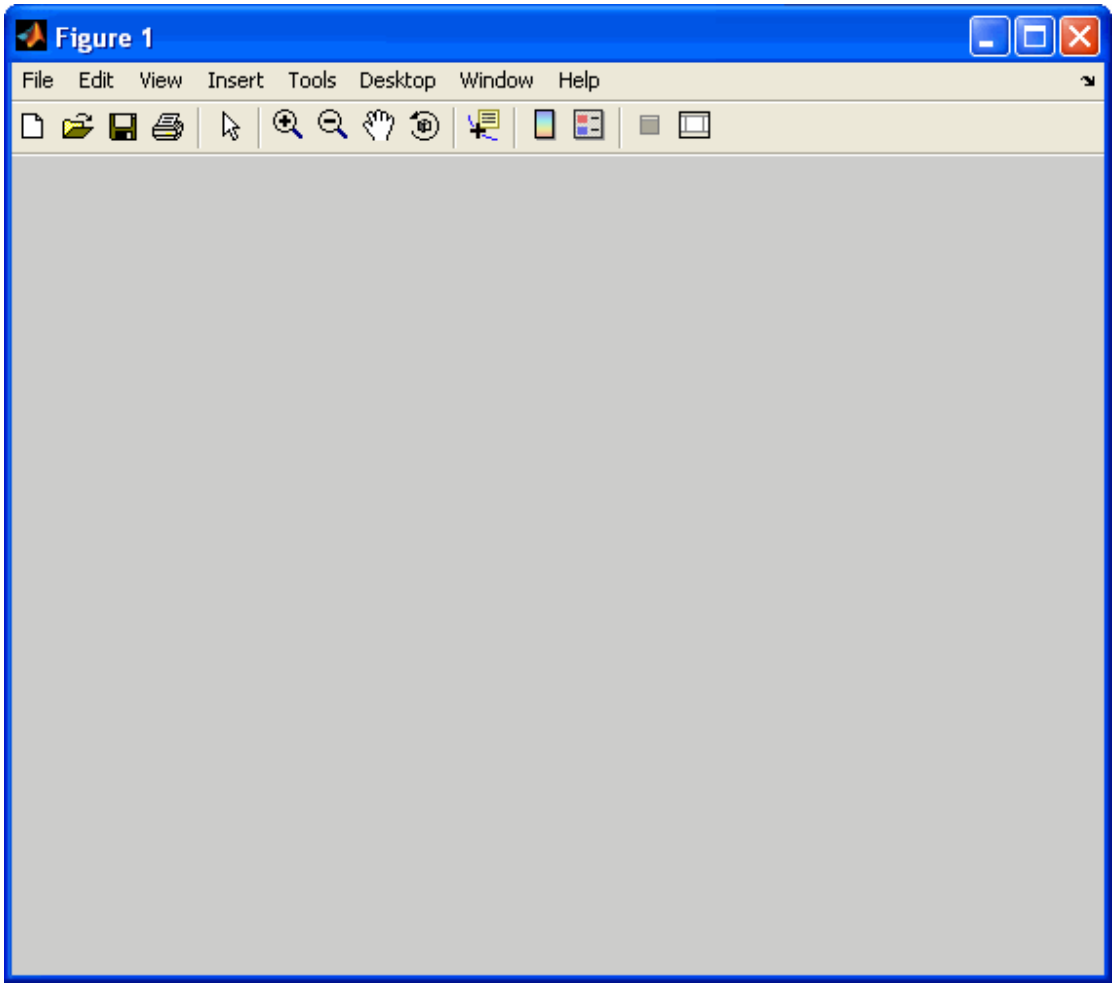


وبالتالي تكون الرسمتان كالتالي



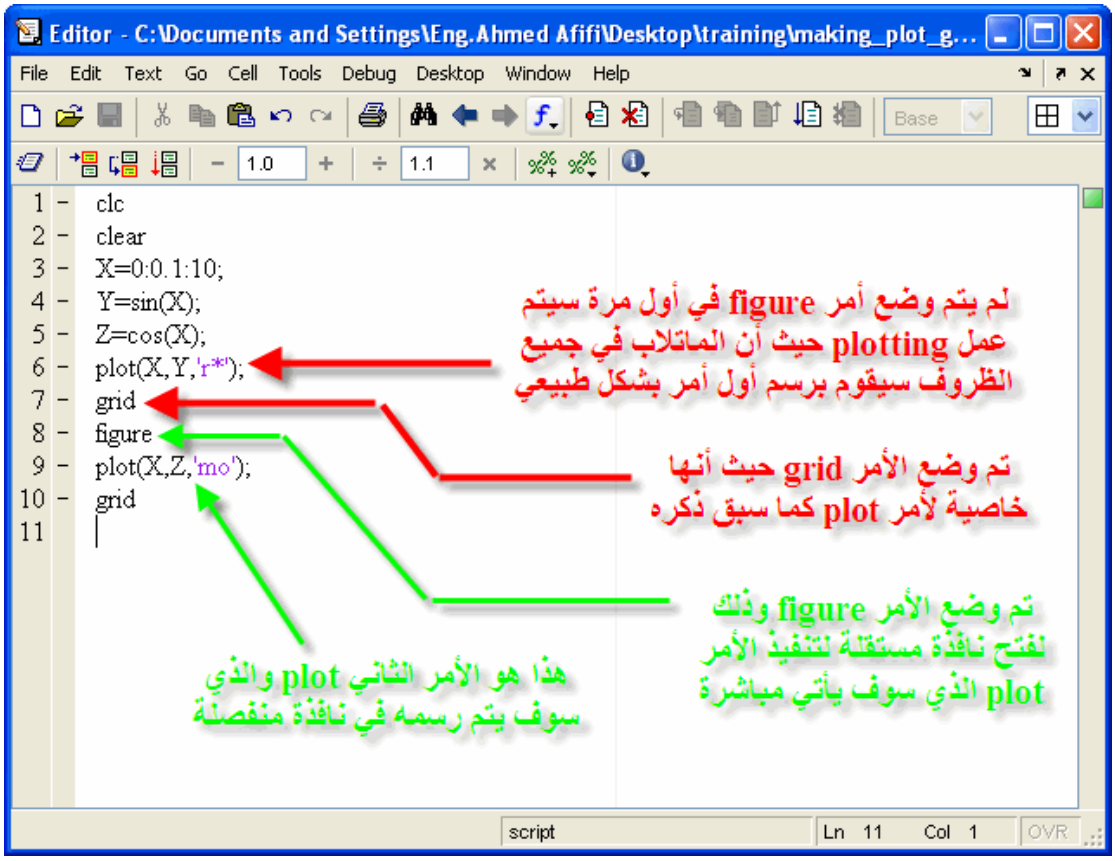
علمية وضع الرسومات في نوافذ منفصلة

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 ان الماتلاب قام بإظهار نافذة رمادية اللون فارغة شاهد الصورة التالية

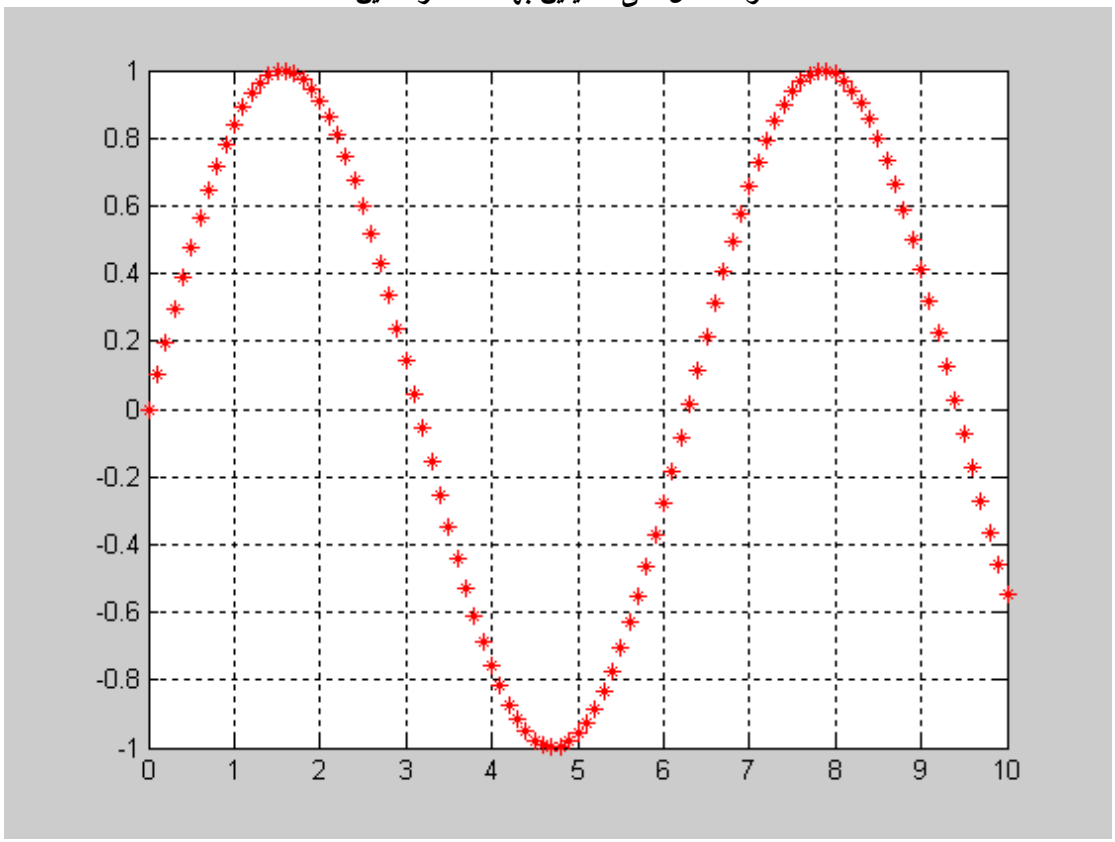


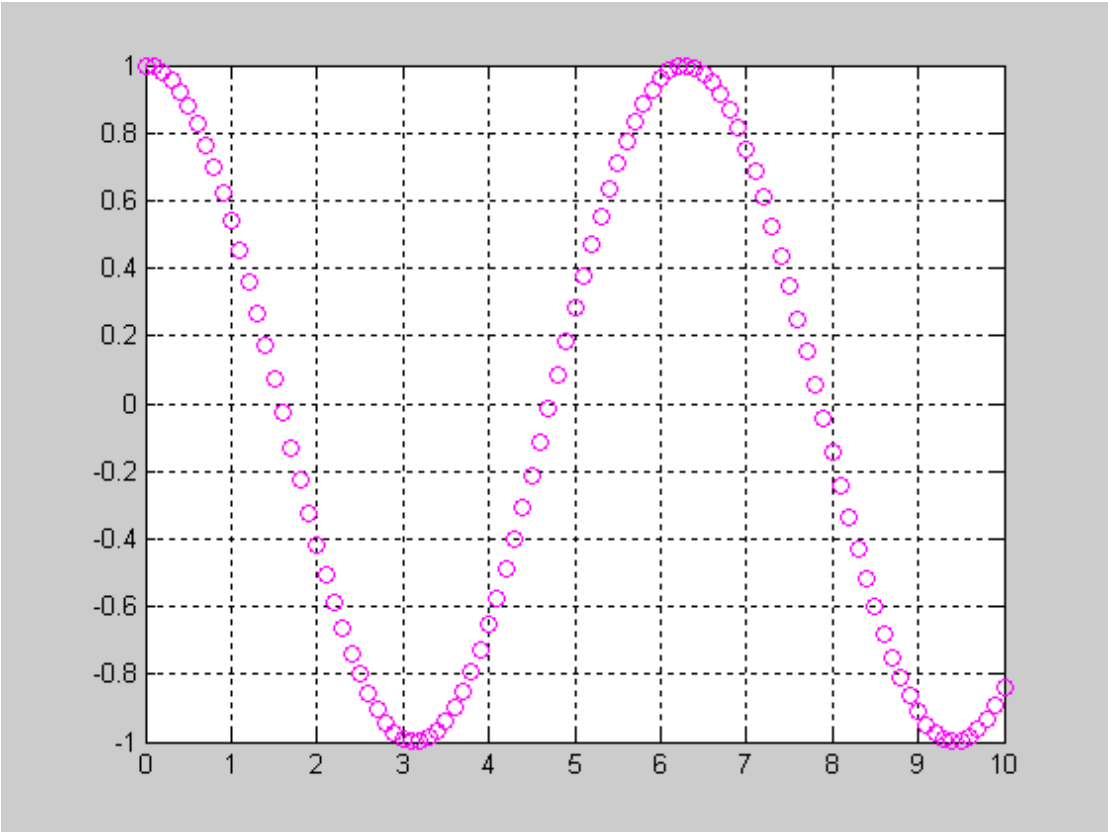
حيث وجود تلك النافذة يعني انه سيتم تنفيذ أمر الرسم **plot** الذي بعد أمر **figure** علماً أنه بعد كل أمر **figure** يتم وضع الخصائص التي تختص بهذه الرسمة مثل أمر **grid** الذي سبق شرحه.

وهذا مثال بسيط على ذلك



وستحصل على نافذيتين بهما كلتا الرسمتين

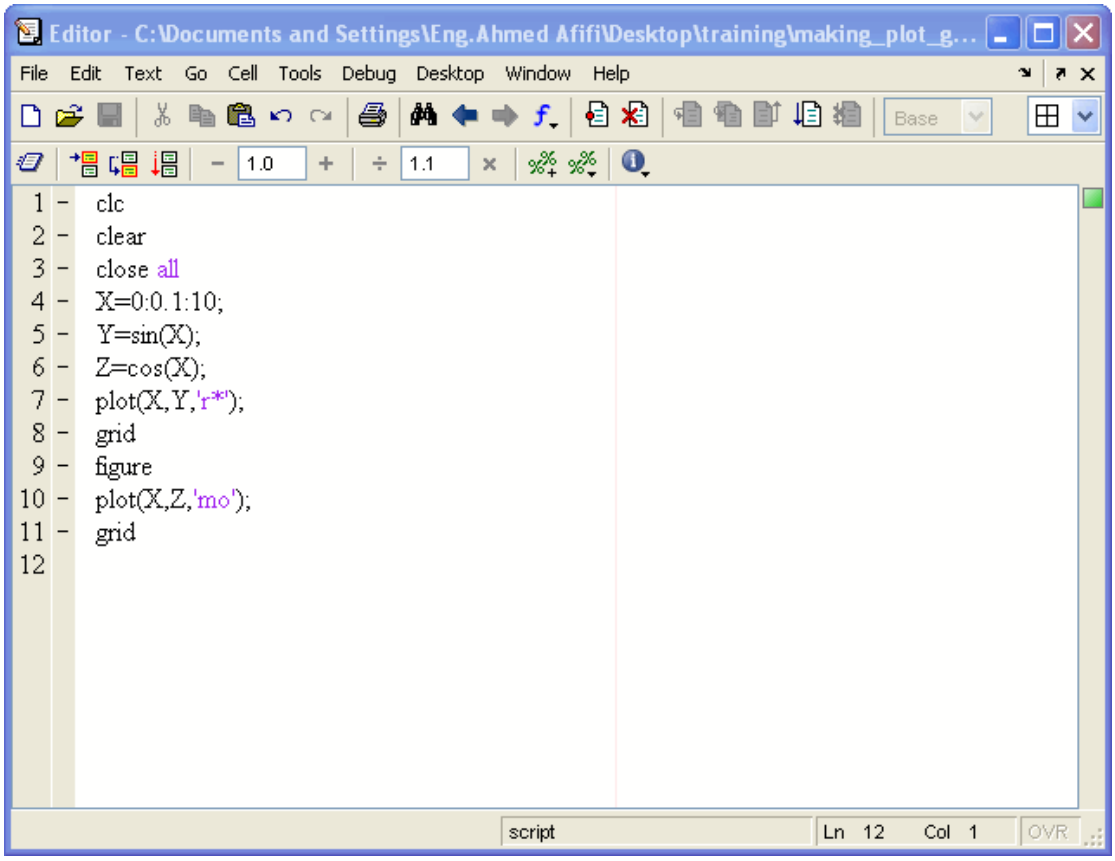




والآن قم بتشغيل البرنامج مرة أخرى، ستلاحظ أن عدد النوافذ قد زاد نافذة واحدة ,فكيف حدث هذا؟
يقوم الماتلاب برسم أول دالة على النافذة الأخيرة التي تم رسم الدالة الثانية بها، ثم يقوم برسم الدالة الثانية في نافذة جديدة بسبب وجود الأمر figure ولحل هذه المشكلة قم بإستخدام الأمر close all بعد الأمر clear بحيث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

clc
clear
close all

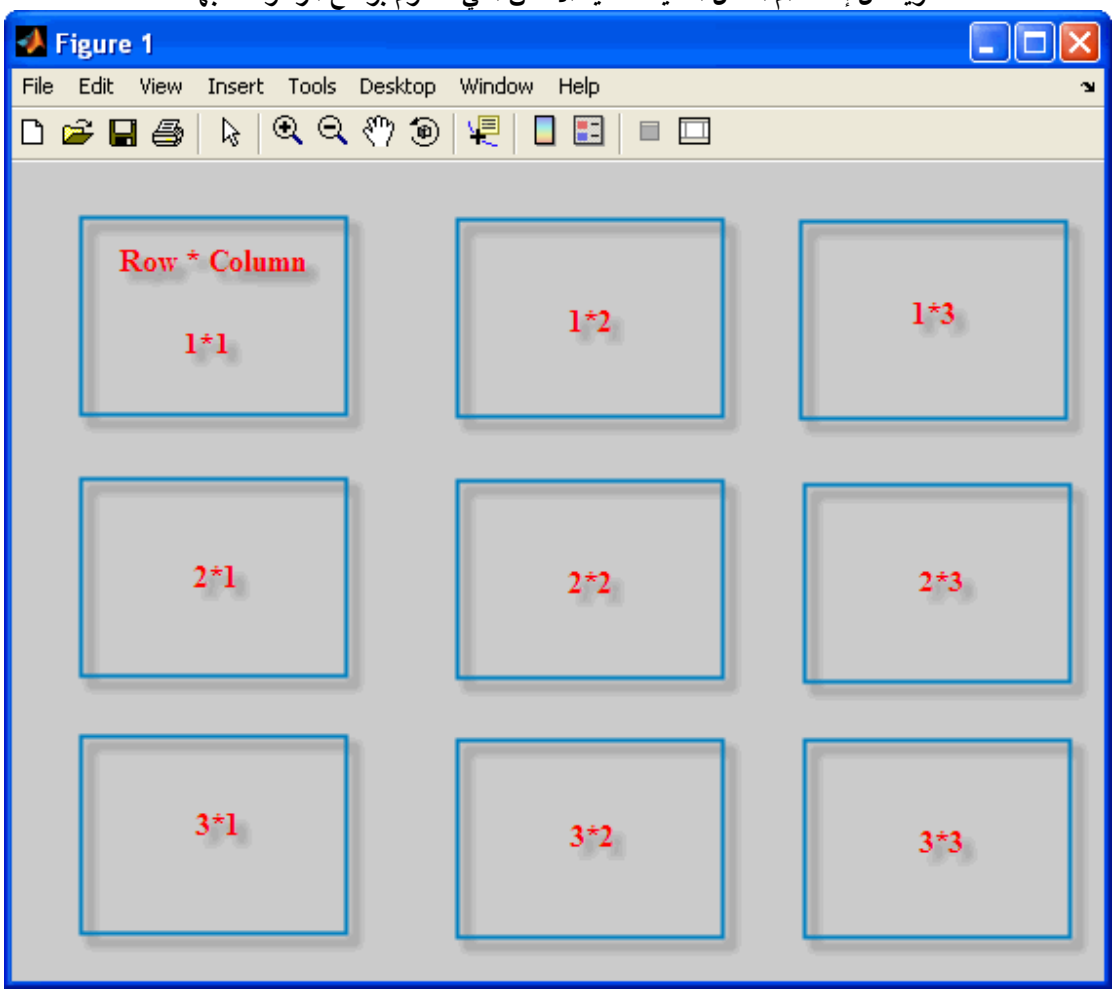
وهذا هو المثال الذي تم عمله منذ قليل بعد التعديل


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
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
script
Ln 12 Col 1
OVR

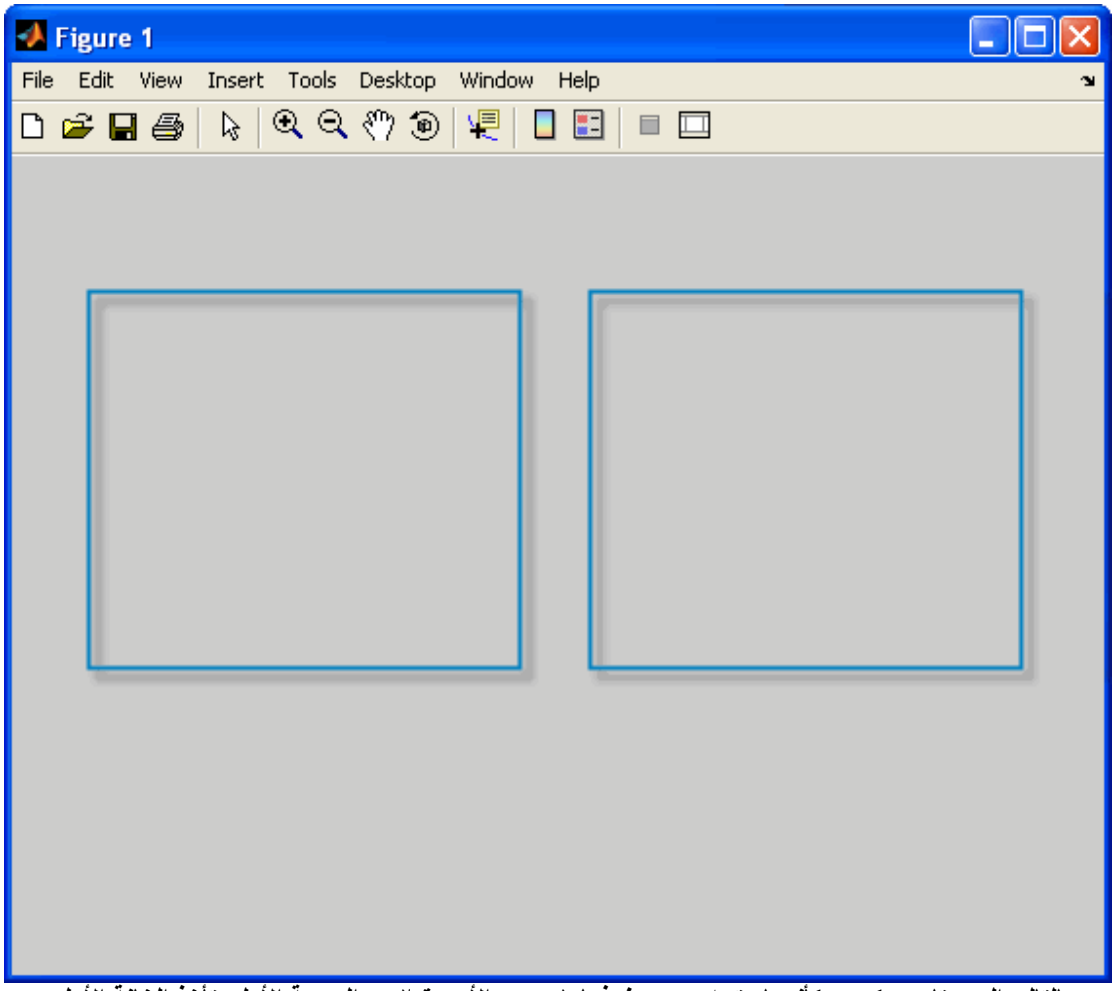
إنشاء رسومات منفصلة في نافذة واحدة

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 بإستخدام 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 إستخدام الأمر معرفة عدد الرسومات التي ستظهرها وكيفية وضعها، ويفضل إستخدام الشكل التالية لتحديد الأماكن التي ستقوم بوضع الرسومات بها



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



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

هذا ما يجب تحديده بالتفصيل عند إستخدام الأمر**subplot**

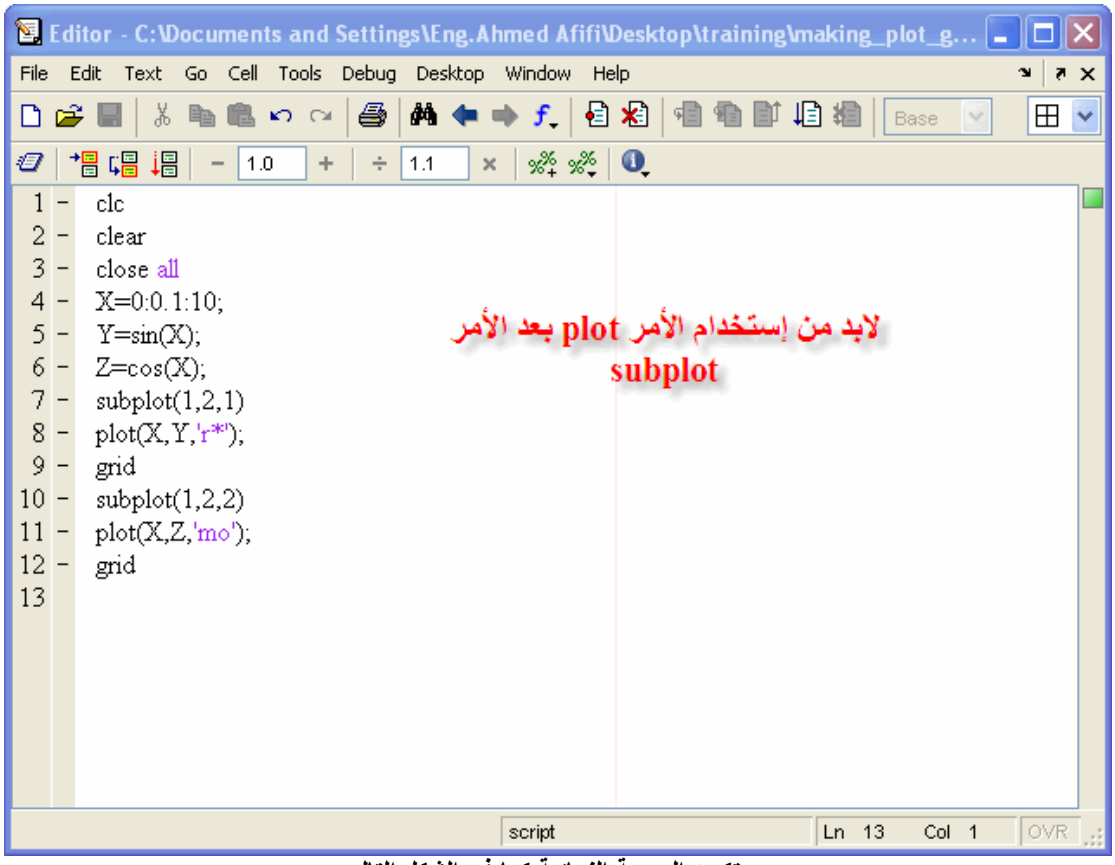
ثم نستخدم الصورة العامة لأمر **subplot** والتي تكون كالتالي

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

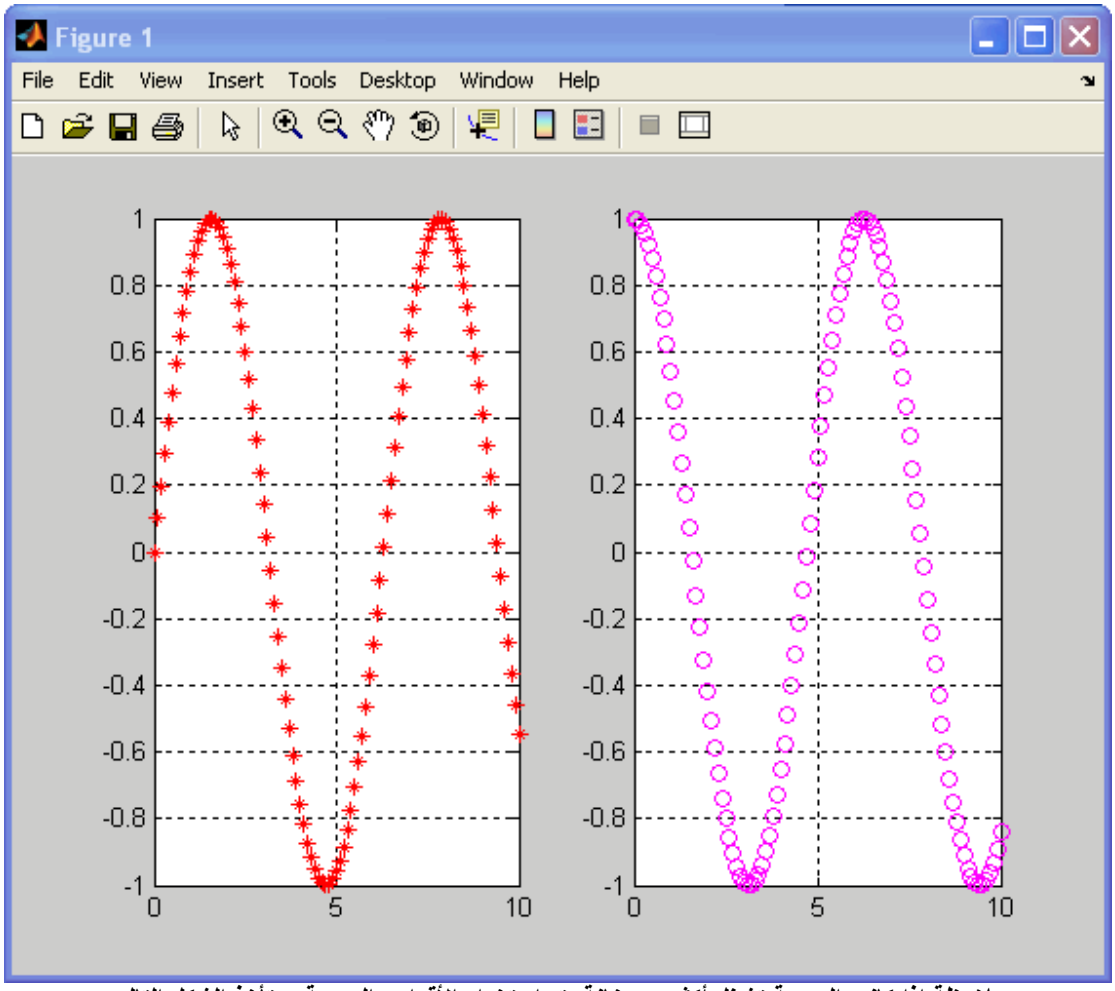
ولرسم الشكل الأول لابد من كتابة الأمر في الصورة التالية

ولرسم الشكل الثاني لابد من كتابة الأمر في الصورة التالية

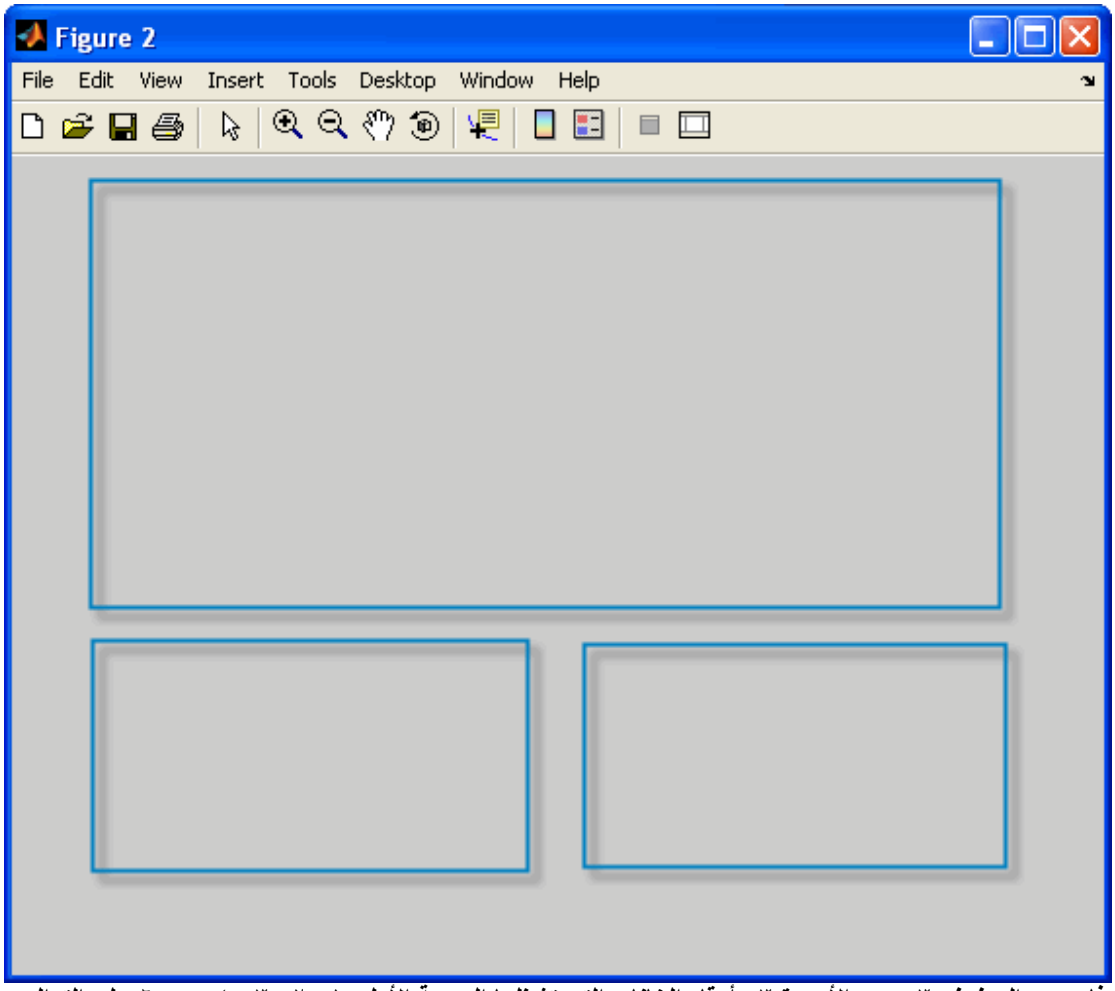
والأن سنقوم بوضع البرنامج كاملاً ليكون المعنى قد وضح تماماً



وستكون الرسمة النهائية كما في الشكل التالي



ملاحظة إذا كانت الرسمة تشغل أكثر من خانة يتم إستخدام 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 و٢ و٣ و٤ و٥ و٦ على التوالي،
وأرقام الخانات التي تشغل الرسمة الثانية ٧ وأرقام الخانات التي تشغل الرسمة الثالثة هي ٩
والبرنامج يكون بالشكل التالي

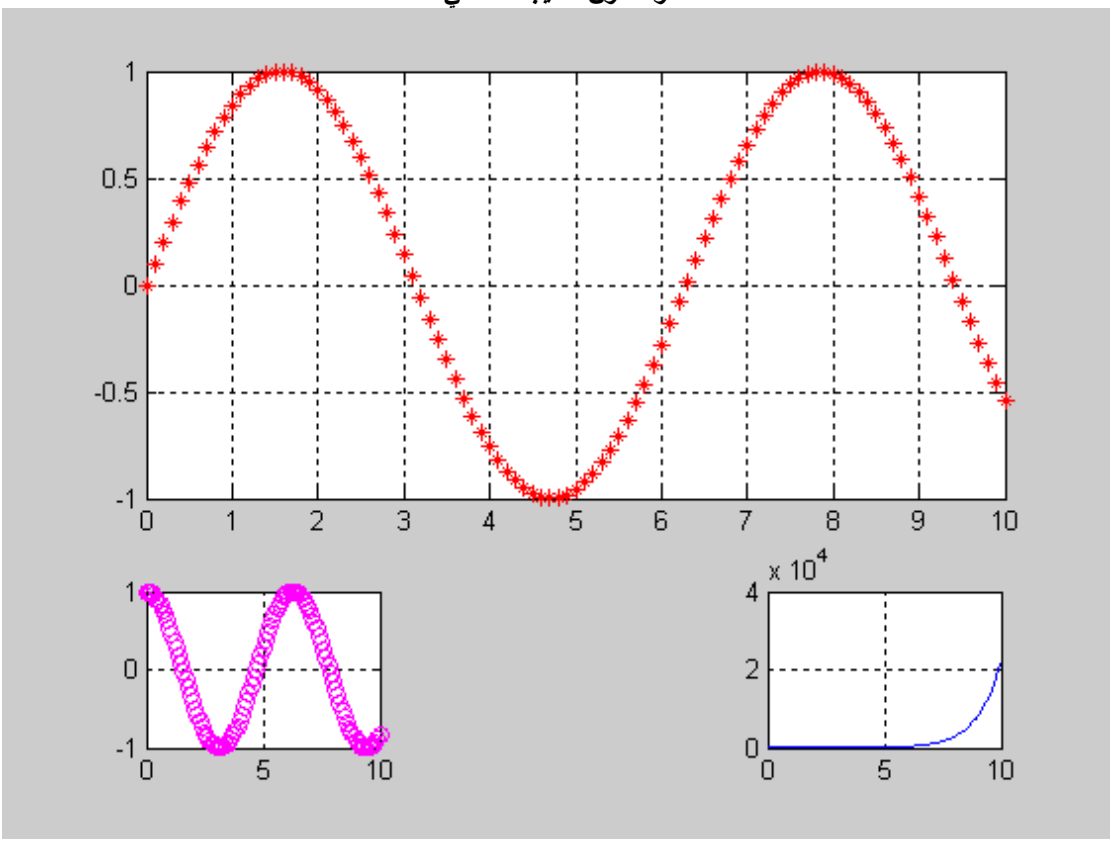
Editor - C:\Documents and Settings\Eng.Ahmed Afifi\Desktop\training\making_plot_g...

File Edit Text Go Cell Tools Debug Desktop Window Help

```
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
```

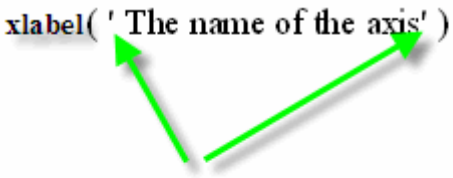
script Ln 17 Col 1 OVR

وستكون النتيجة كالتالي



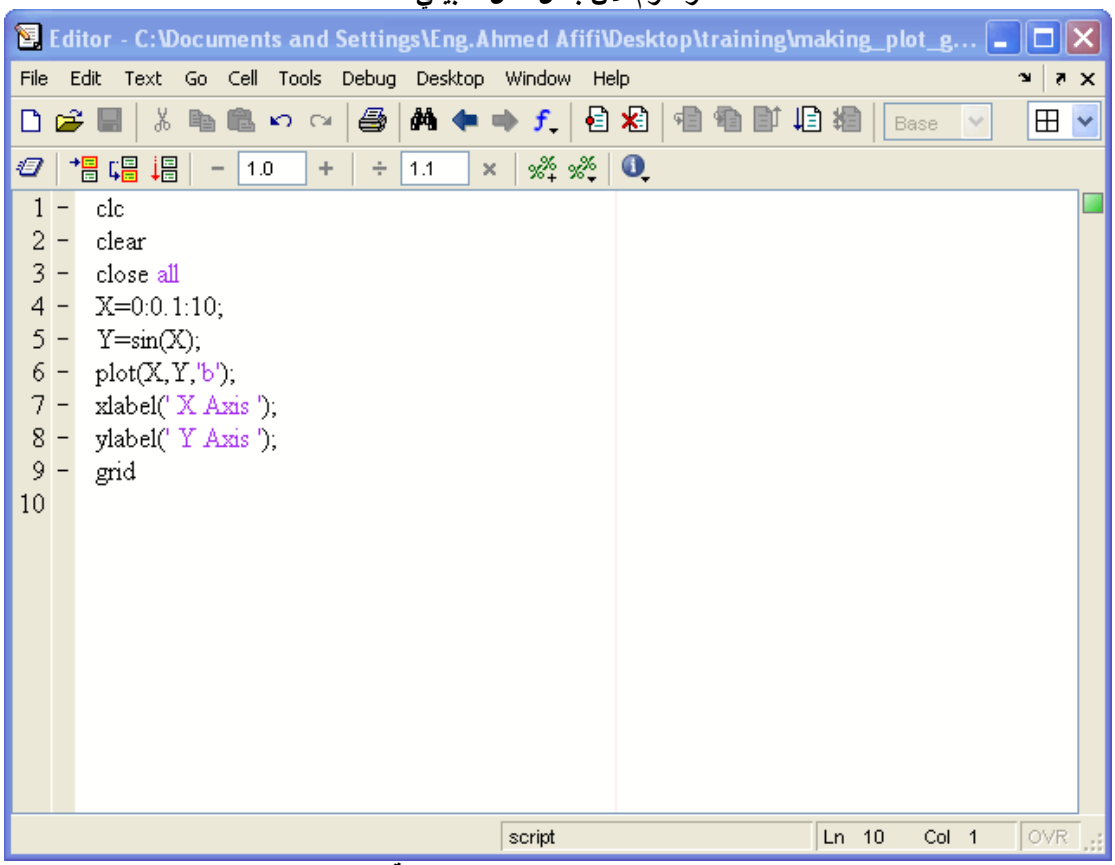
تسمية المحاور

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

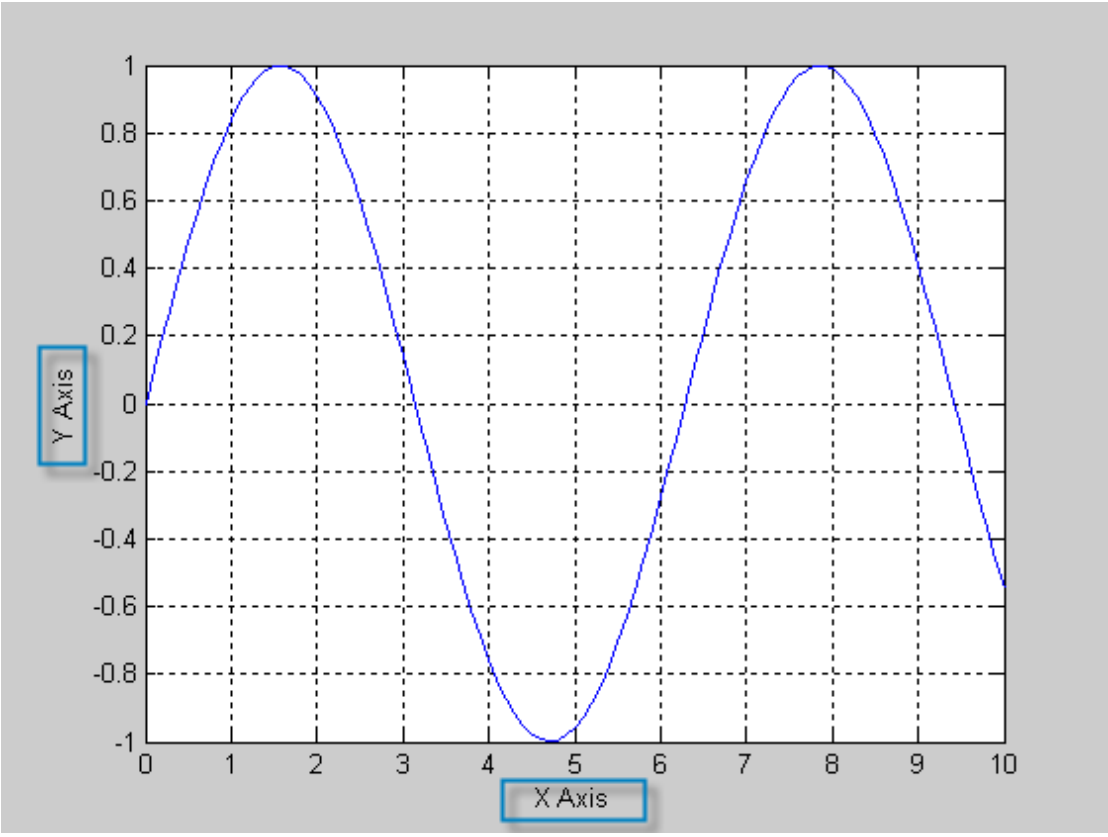


كما ترى لابد من أن يكون إسم المحور
بين فاصلتين كما هو واضح بالشكل

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



وبالتالي نحصل على الصورة التالية



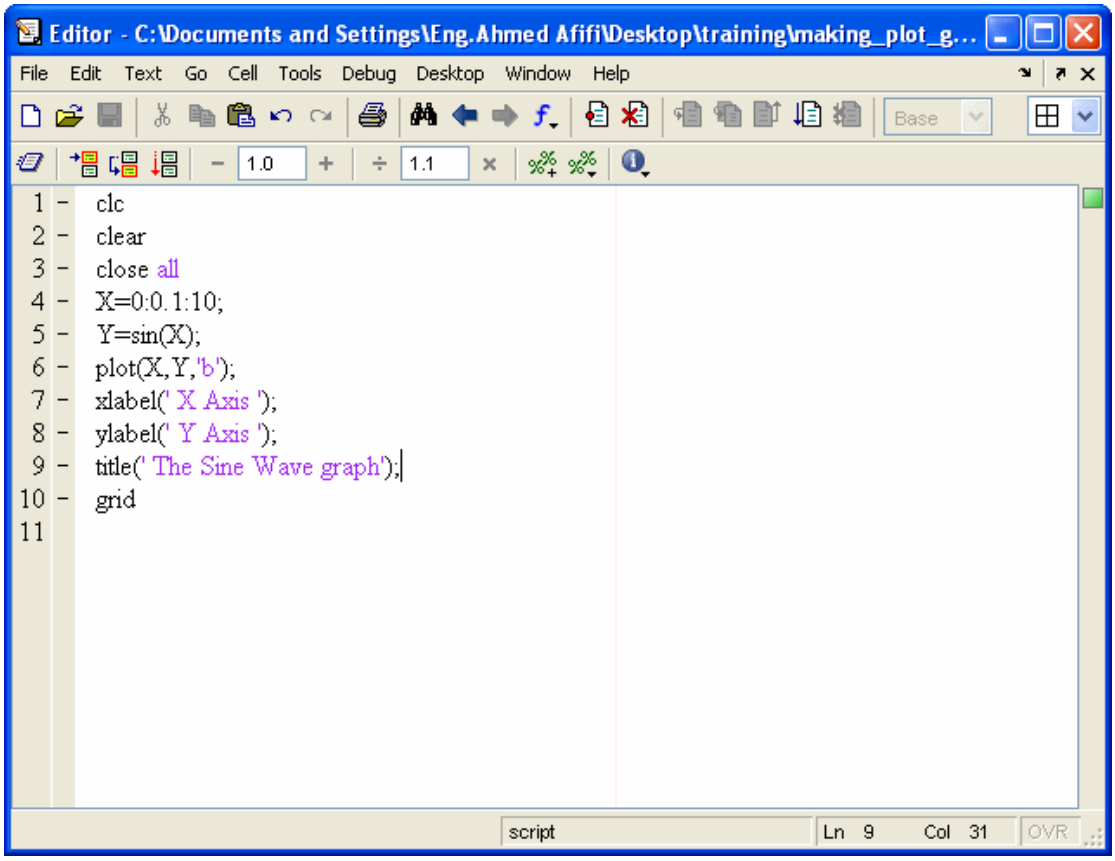
وضع عنوان في أعلى الرسمة

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

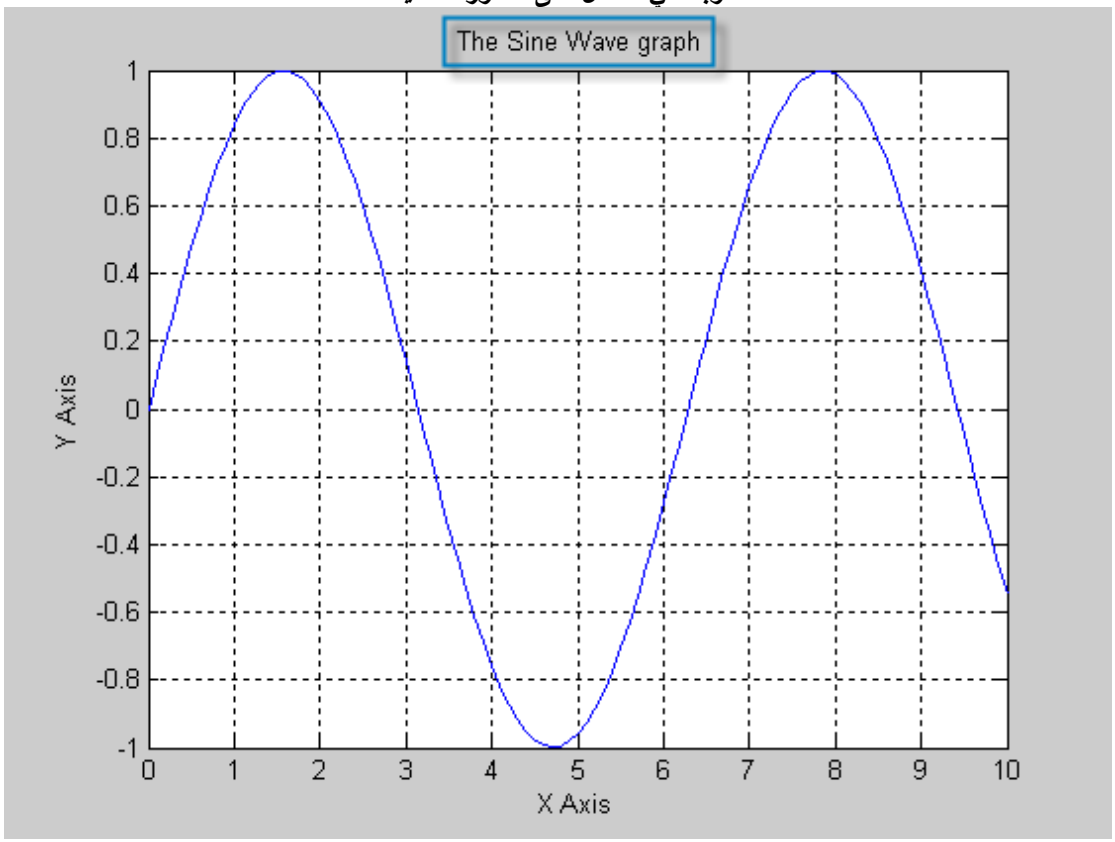
```
Title ('The title of the graph')
```

لابد من وضع العنوان بين فاصلتين كما هو
موضح بالرسم

وبالرجوع إلى المثال السابق ووضع التعديلات عليه كما هو موضح



وبالتالي نحصل على الصورة التالية



وضع نص على نقطة أو أكثر داخل الرسم

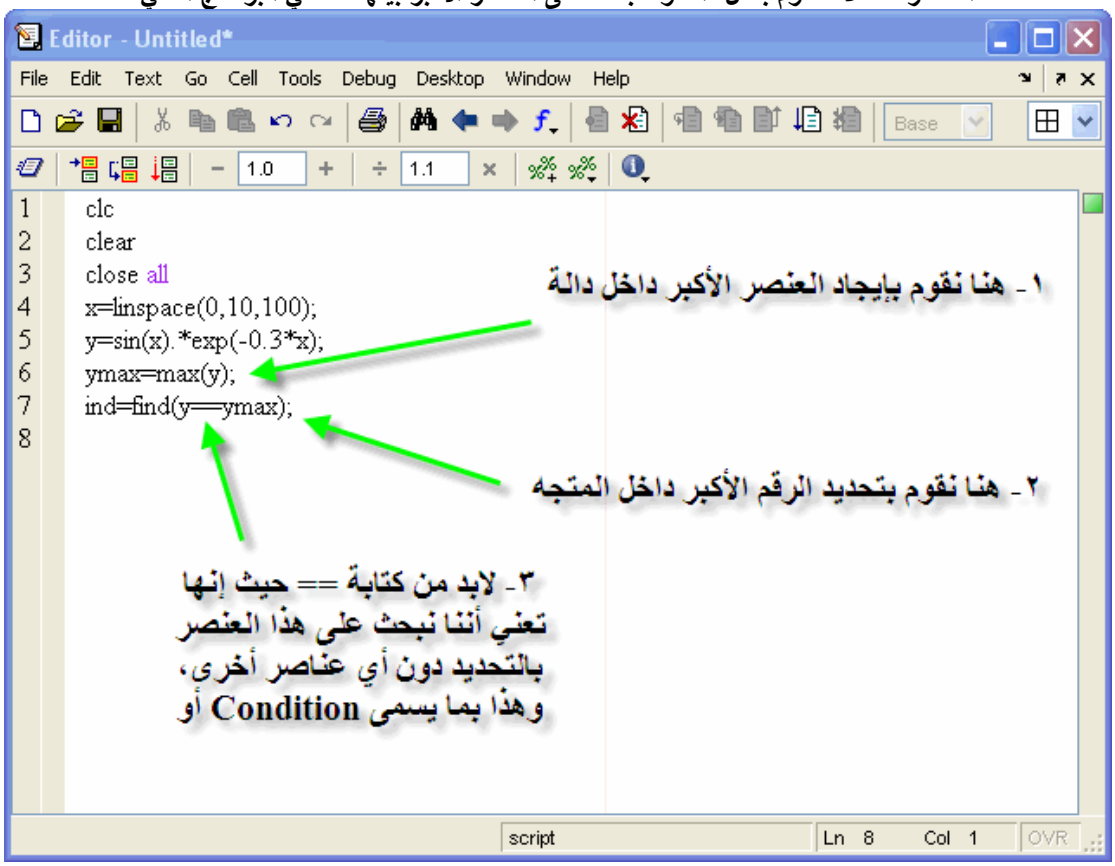
يمكن إضافة نص على نقطة أو أكثر على الرسم، وذلك بإستخدام الأمر **text** ويأخذ الصورة التالية

```
Text (position of the point at X-Axis, position of Y-Axis, ' The text on that point ')
```

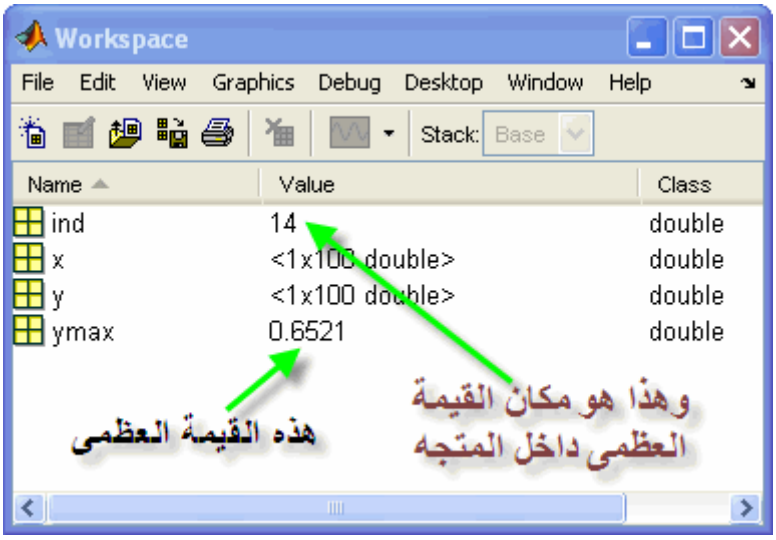
يجب وضع النص بين فاصلتين

وسنأخذ مثالاً بسيطاً في كيفية إيجاد الرقم الأكبر، ثم وضع دائرة حمراء حول النقطة العظمى ووضع كلمة **maximum point**

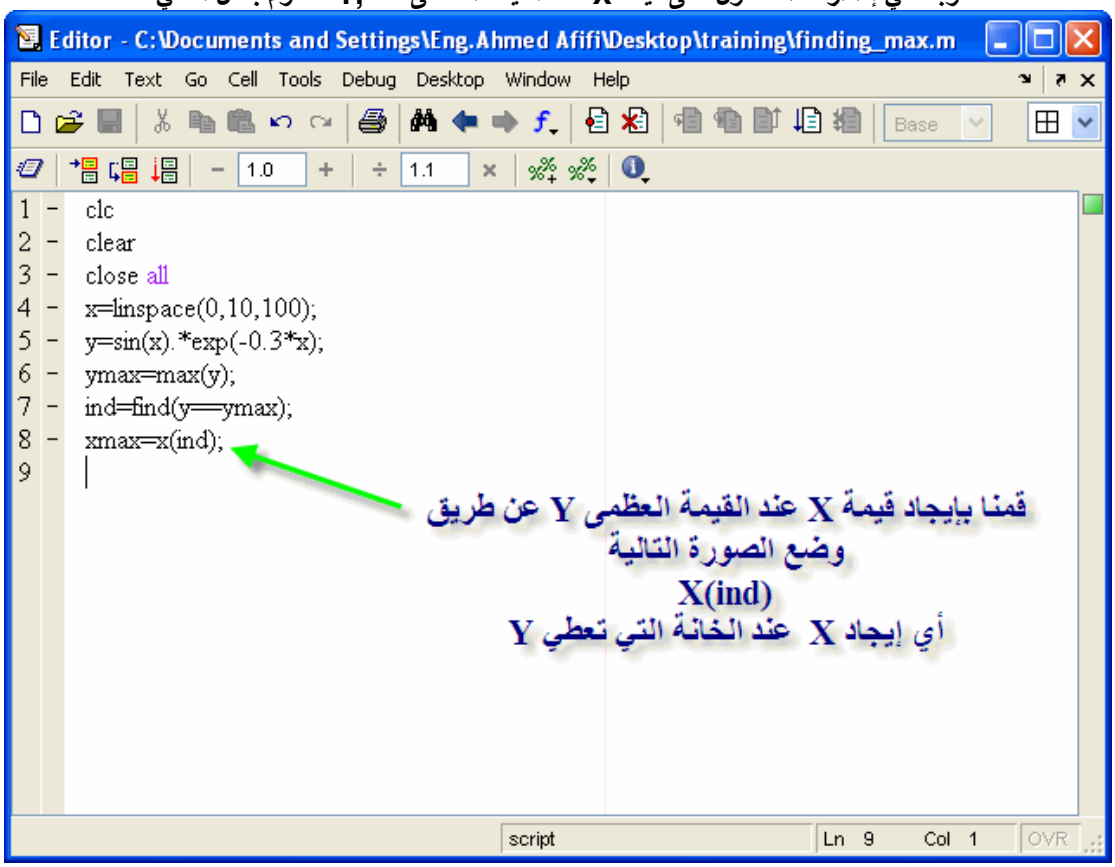
ولكن دعونا نقوم بشرح الأمر الهام **find** هذا الأمر يقوم بإيجاد مكان العنصر داخل المتجه بمجرد تحديد خصائص هذا العنصر، فمثلاً سنقوم بعمل دالة وسنبحث على العنصر الأكبر بينها كما في البرنامج التالي



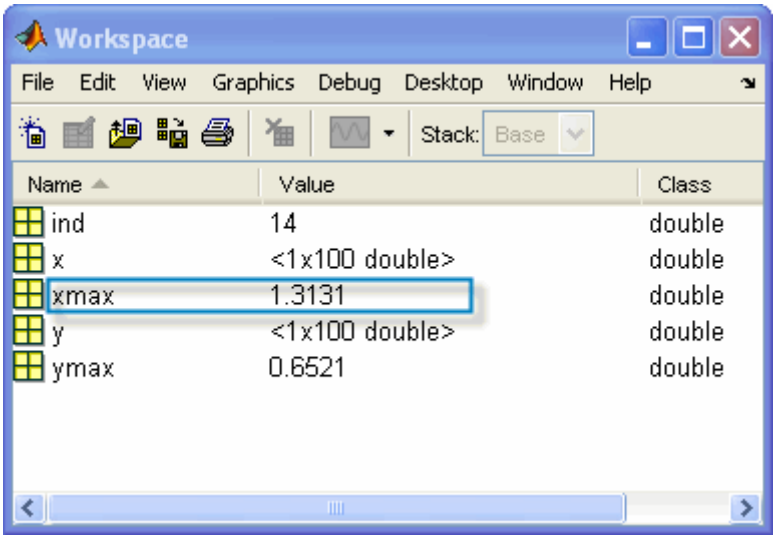
وعند تشغيل البرنامج، نجد القيم كالتالي



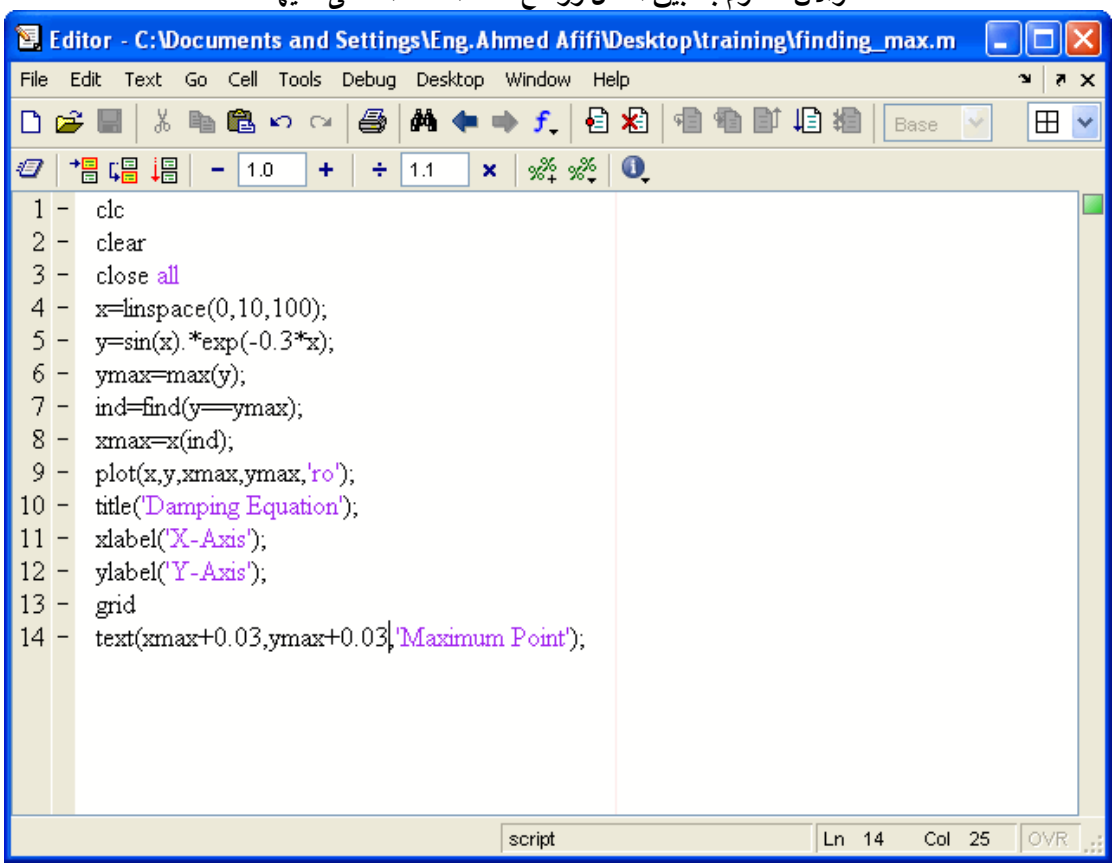
وبالتالي إذا أردنا الحصول على قيمة X عند القيمة العظمى للـــ Y, سنقوم بعمل التالي



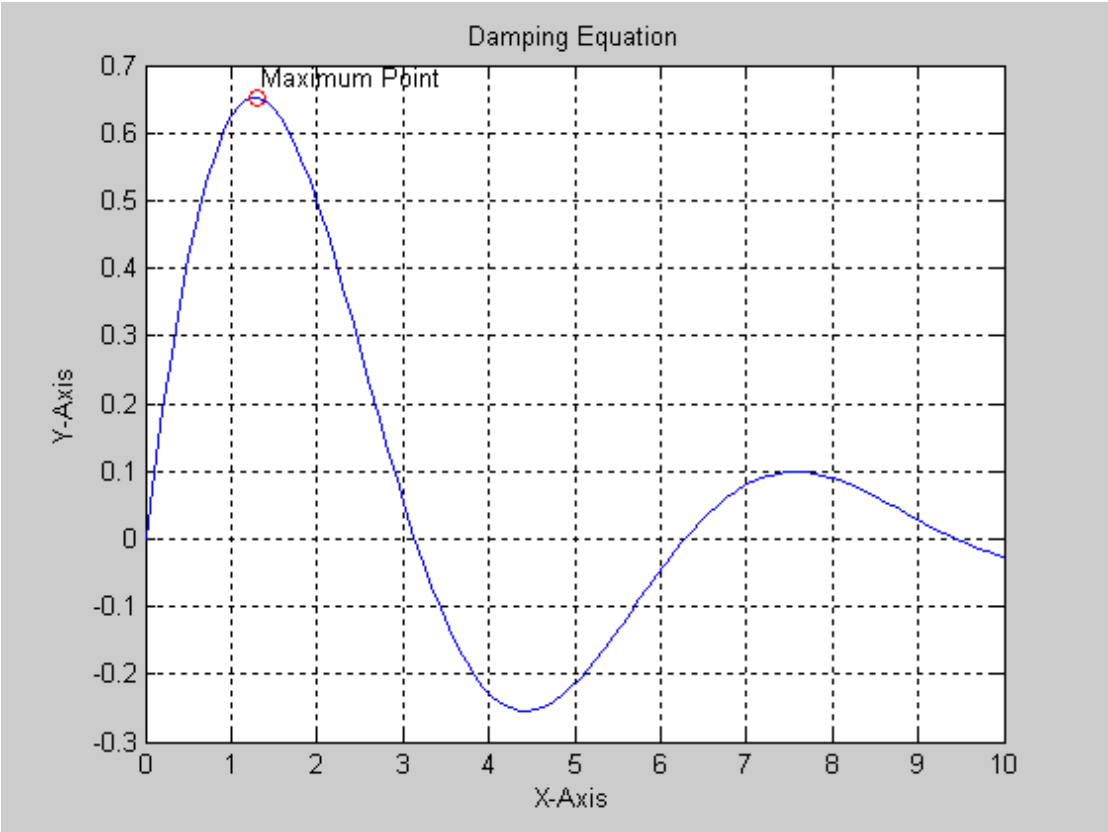
وكما تلاحظ فإن قيمة X والتي تعطي القيمة العظمى Y تظهر في workspace, أنظر الصورة التالية



والآن سنقوم بتطبيق المثال ووضع كلمة النقطة العظمى عليها



وبالتالي ستظهر الرسمة كالتالي

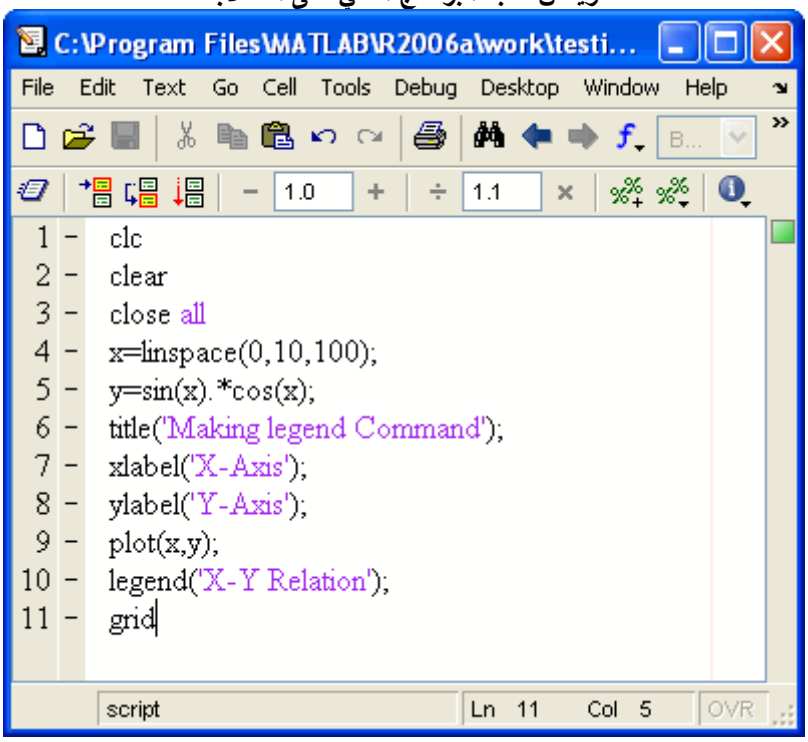

Damping Equation
Maximum Point
Y-Axis
X-Axis
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0
1
2
3
4
5
6
7
8
9
10

الأمر legend

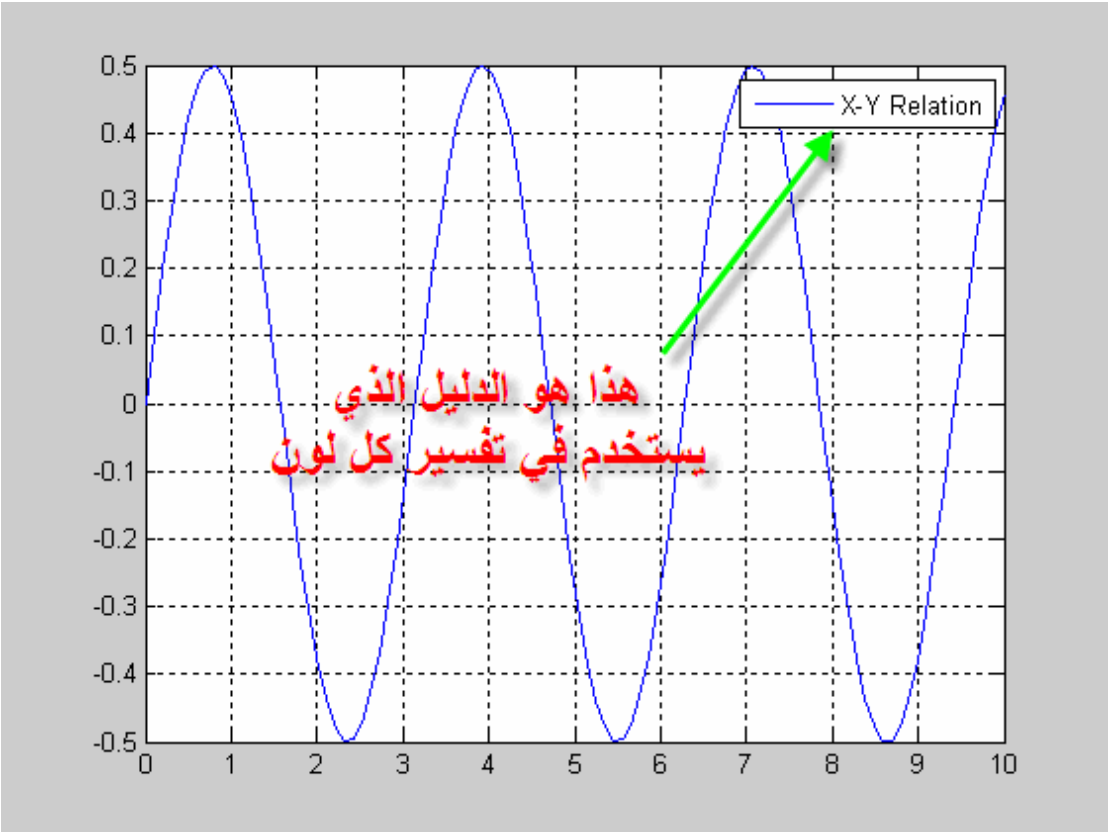
يستخدم هذا الأمر في وضع دليل على صفحة الرسم ليبين ماذا يعني كل لون على الرسم، فمثلاً سنقوم بوضع الأمر legend في المثال التالي علماً أن هذا الأمر لابد من أن يأخذ الصورة التالية

Legend ('the color reference')

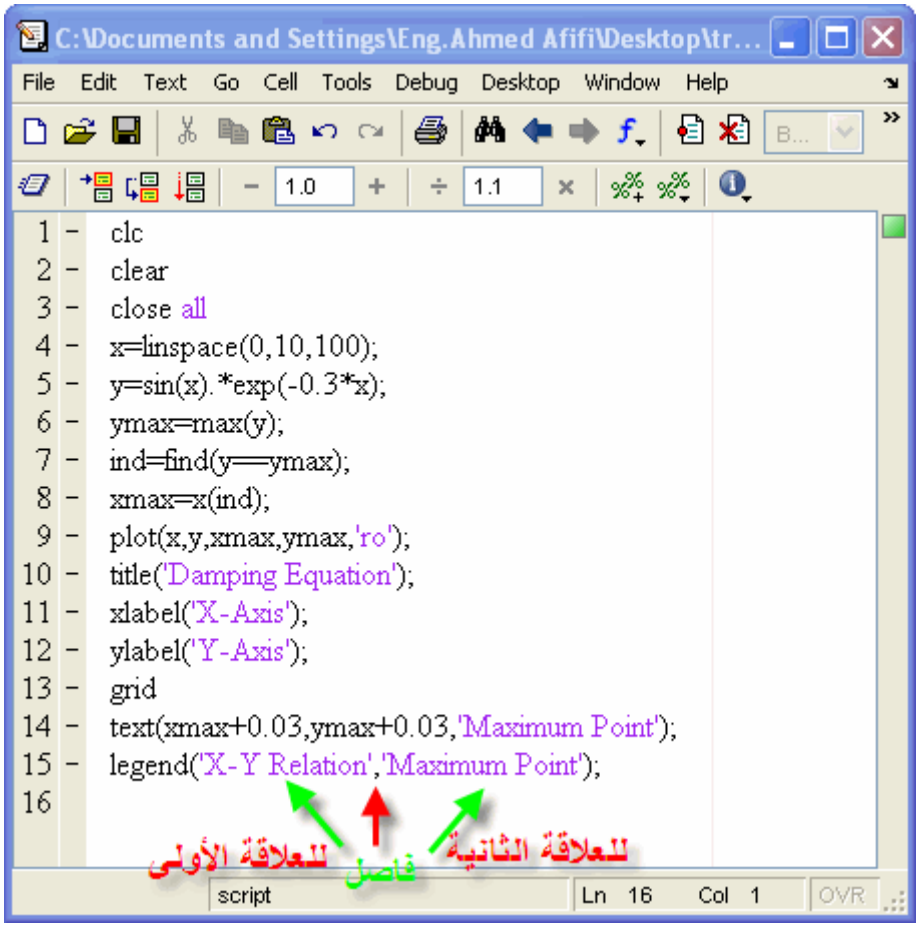
ويمكن كتابة البرنامج التالي على الماتلاب



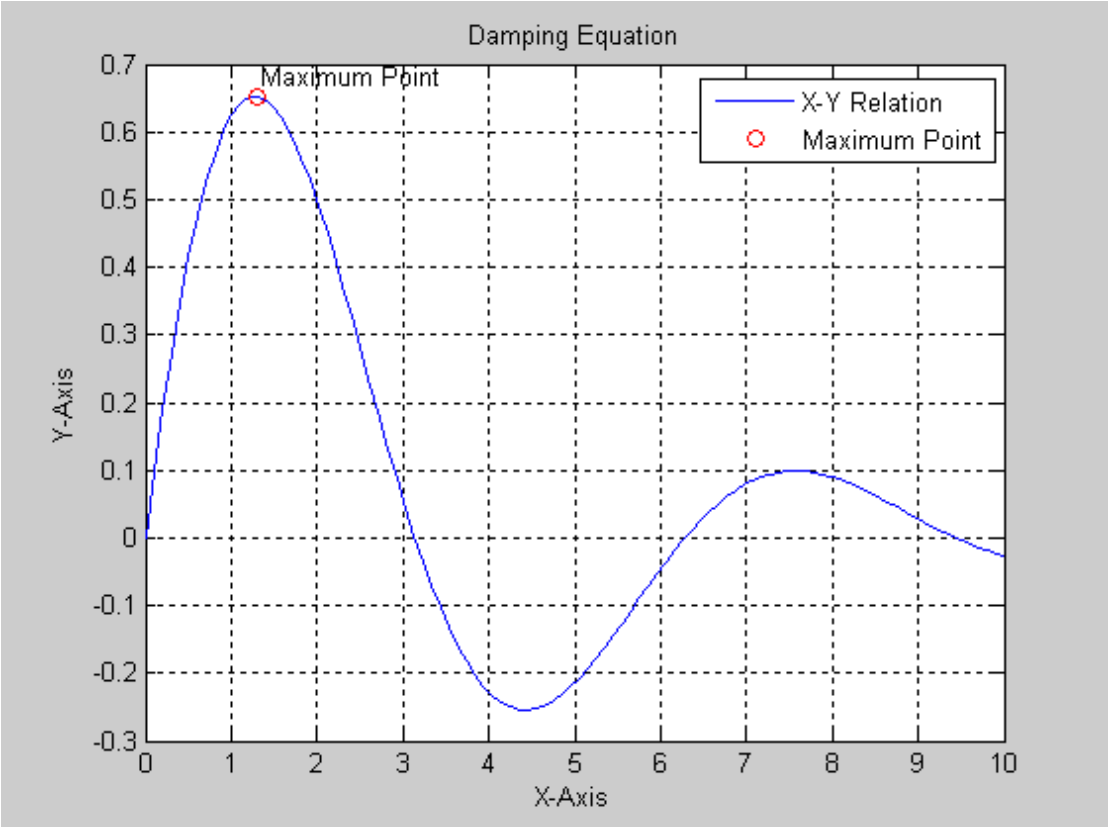
وبالتالي ستجد الناتج كالتالي



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 إثنتان، وبالتالي تتم برمجته بالشكل التالي



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

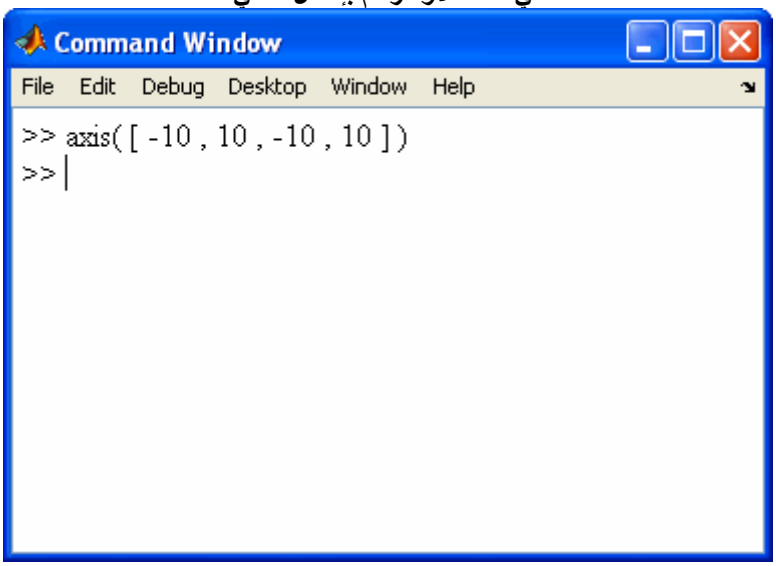
٢- أكبر قيمة لمحور السينات هي ١٠

٣- أقل قيمة لمحور الصادات -١٠

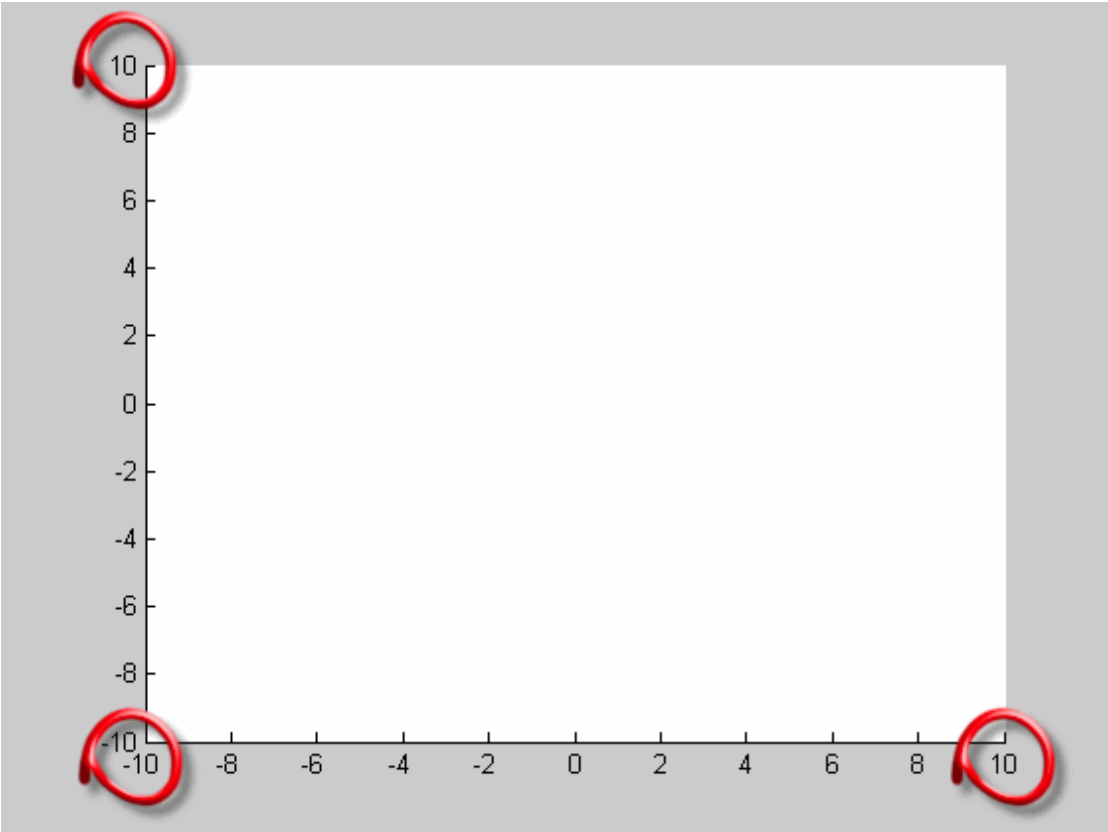
٤- أكبر قيمة لمحور الصادات ١٠

خطوات الحل

في نافذة الأوامر قم بإدخال التالي



وستظهر لك النافذة التالية

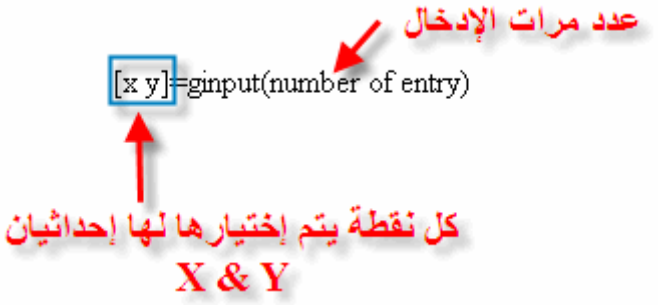


وبالتالي نكون قد أتممنا شرح كيفية فتح نافذة للرسم بنجاح
يمكنك الآن وضع الخصائص التي تريدها على تلك النافذة
أما الدرس القادم فهو مهم جداً وهو كيفية إدخال النقط على الرسم من خلال إستخدام الماوس
ونلقاكم في رعاية الله في الدرس القادم

كيفية إدخال النقاط من خلال الماوس

تعلمنا أنه يمكننا إدخال القيم بإستخدام المتجهات أو المصفوفات، ولكن يوفر الماتلاب قدرة في إدخال النقاط من خلال الرسم بإستخدام الماوس، ونظراً لأننا نقوم بإختيار 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 بإستخدام الماوس، ويتم كتابة ذلك الأمر في الصورة التالية



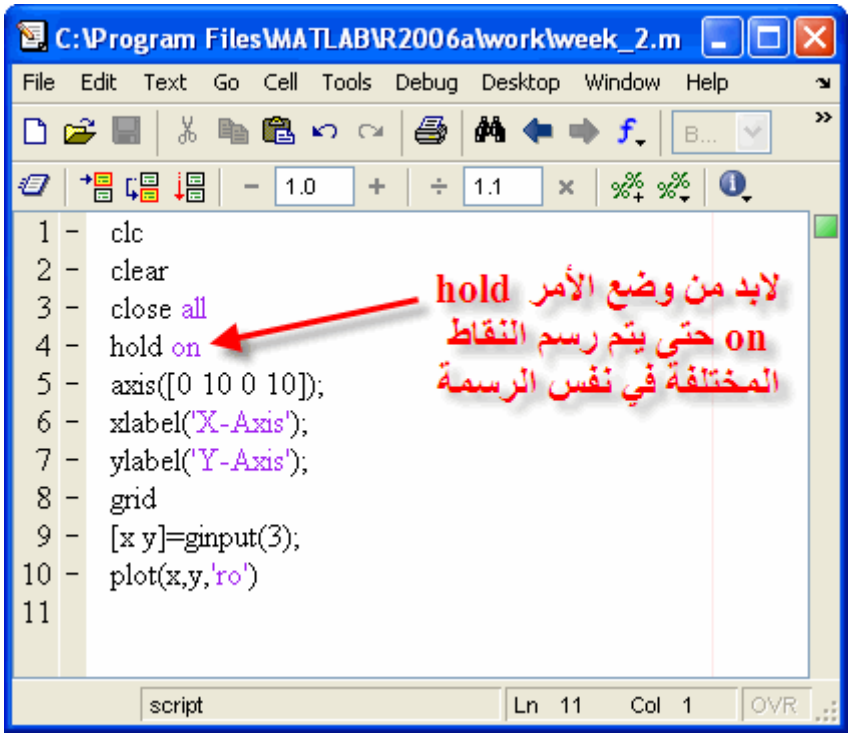
أما إذا أردنا إدخال عدد لا نهائي من النقاط يمكن ذلك بعدم ذكر عدد نقاط الإدخال ,كما في الشكل التالي

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

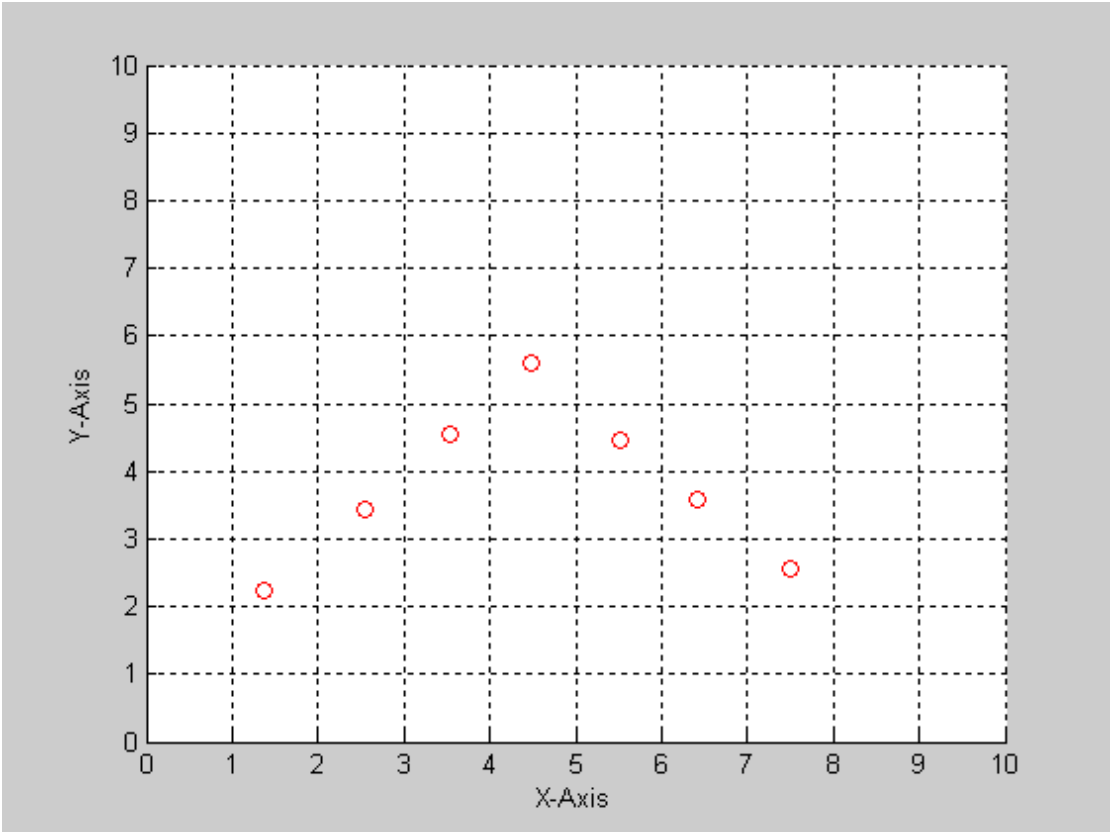
مثال تطبيقي

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 بإستخدام الأمر **ginput**, وهذه النقاط يتم طباعتها على شكل دوائر حمراء.

ويتم كتابة الأوامر بالشكل التالي



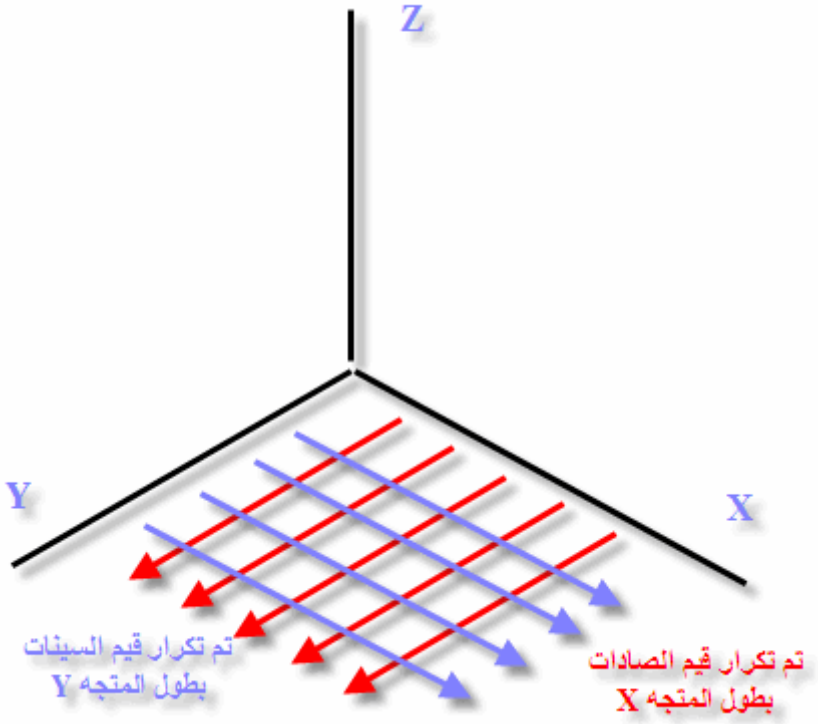
وستظهر لك نافذة لإدخال النقاط ، وبعد إتمام عملية الإدخال إضغط على **Enter** لإتمام الإدخال وستظهر لك النافذة التالية



وبالتالي نكون قد أتممنا شرح هذه الجزئية بنجاح، وسيتم التطرق في دورة البرمجة بإستخدام 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 الإرتفاع، ولكن تلك القيم هي قيم النقاط الموجودة المحاور، ولكن حتى يتم رسم أي نقطة في المستوى الأفقي يجب أن نقوم تعريف ذلك للماتلاب وذلك بإستخدام 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



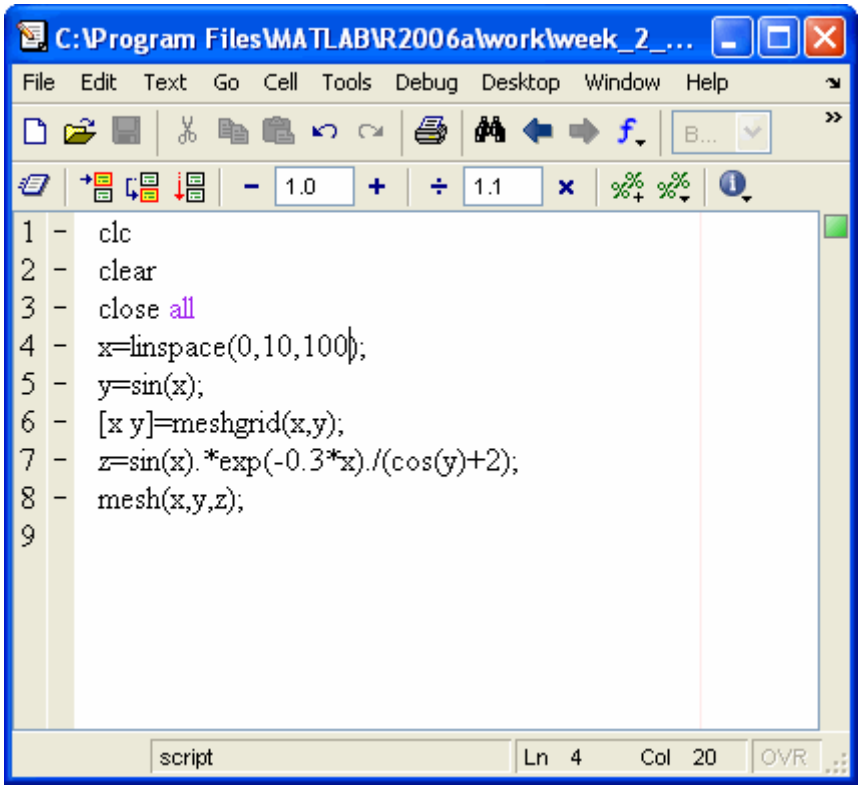
علما أن الأمر meshgrid يأخذ الصورة التالية في كتابته

[x y]=meshgrid(x,y)

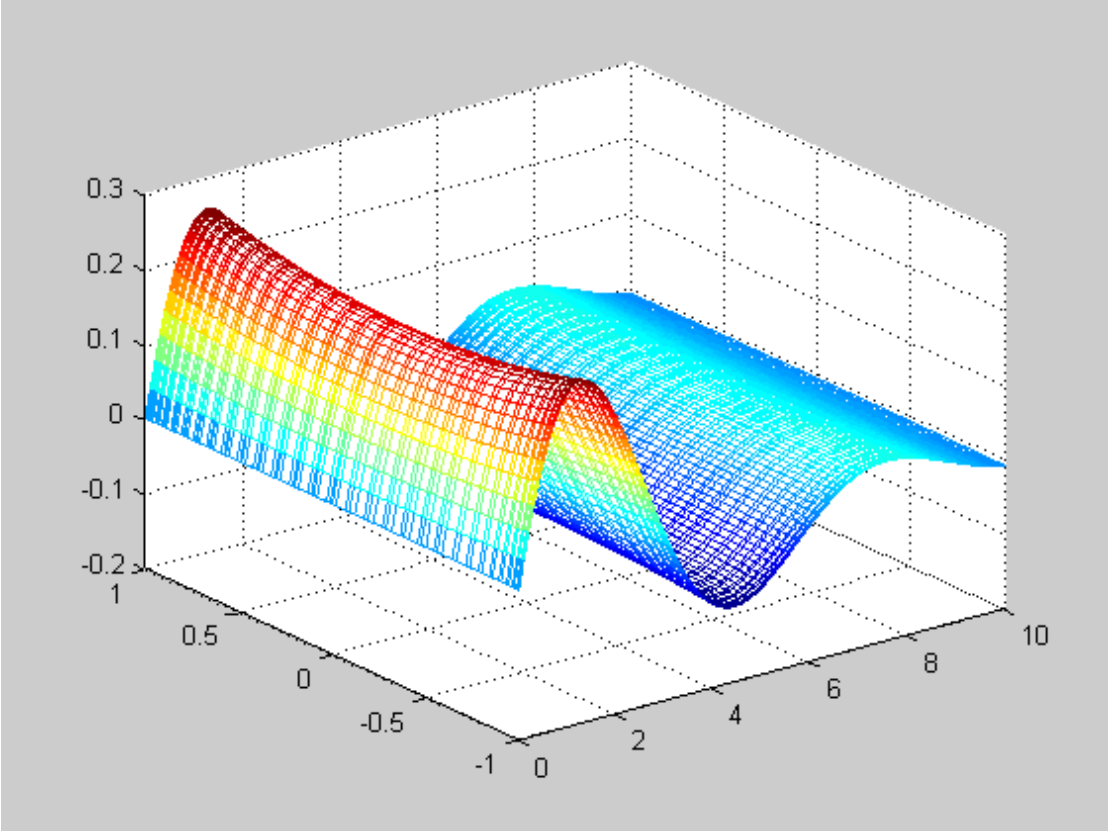
وبعد إستخدام الأمر meshgrid يتم إستخدام الأمر mesh والذي يستخدم كبديل الأمر plot ولكن في الرسم ثلاثي الأبعاد

مثال تطبيقي

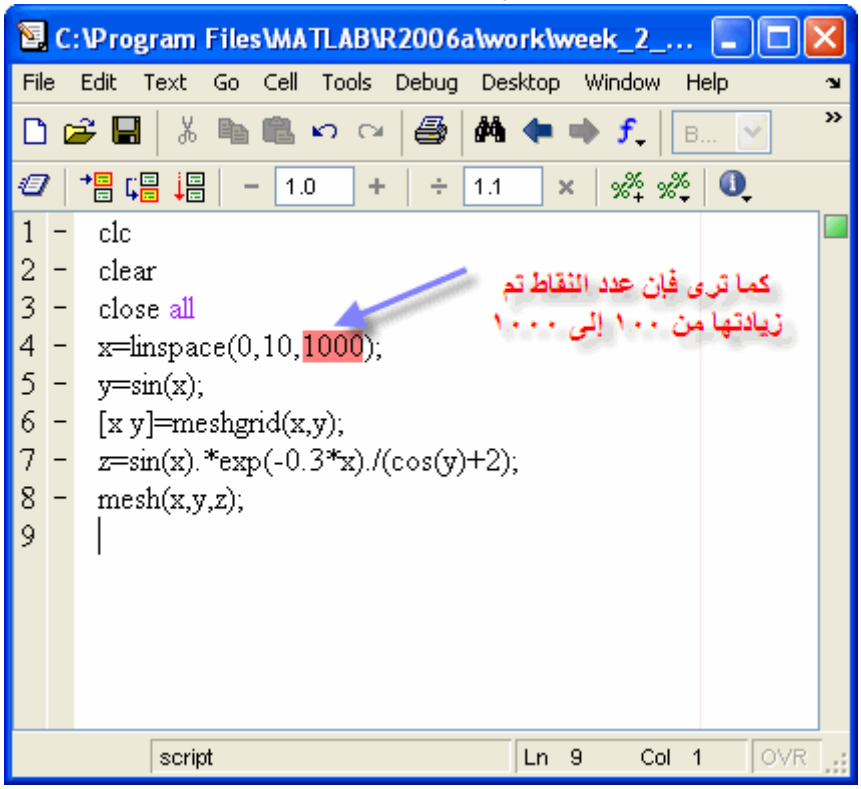
نقوم في هذا المثال بتعريف قيم محور السينات X-Axis وسنقوم بوضع المعادلة التي تصف محور الصادات وعلاقته بمحور السينات، أخيراً وليس آخراً نقوم بوضع العلاقة التي تربط بين محور السينات والصادات.



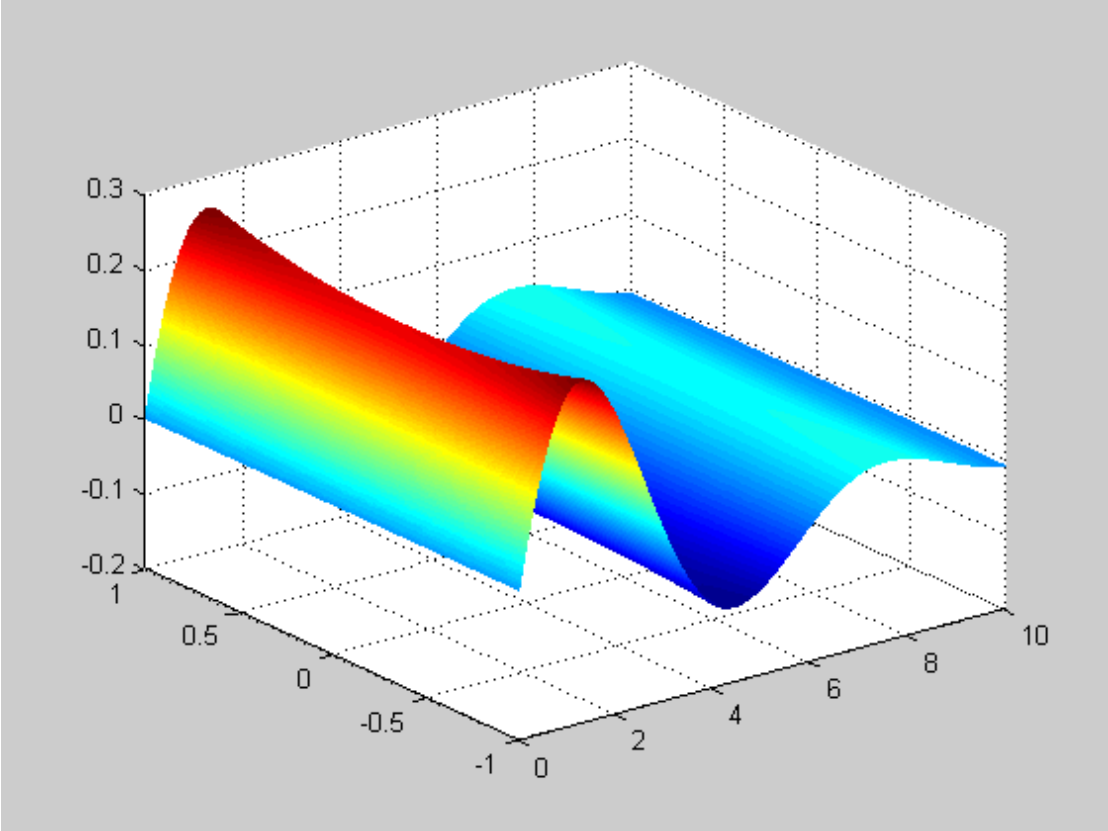
وبالتالي تكون الرسمة الناتجة كالتالي



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



وبالتالي فإن الرسمة 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

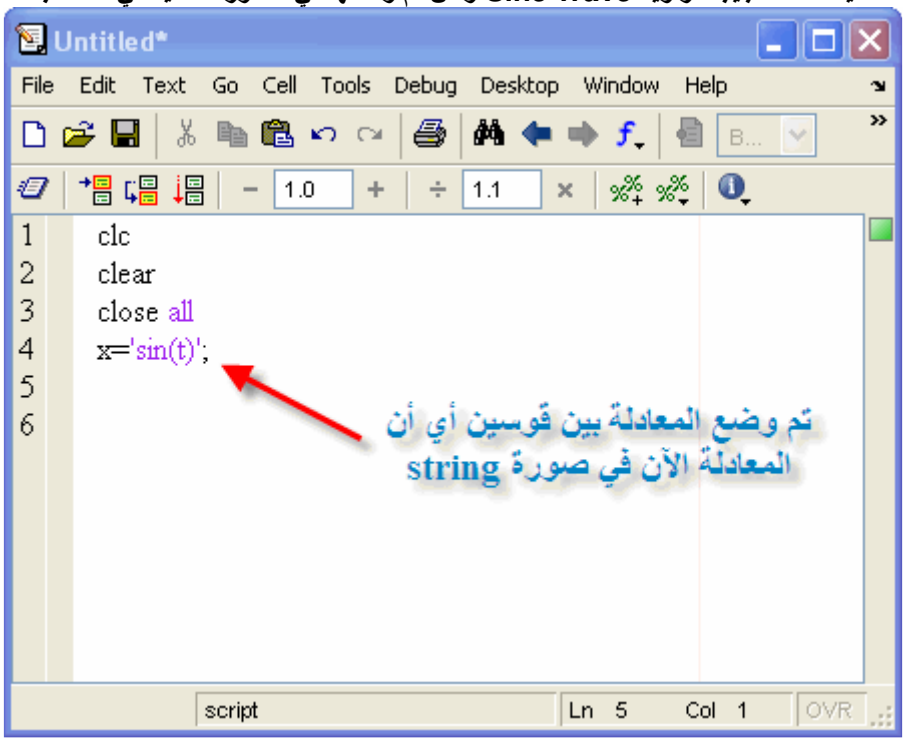
٢- إيجاد نقط تقاطع الرسمة مع محور السينات ووضع علامة عليها

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

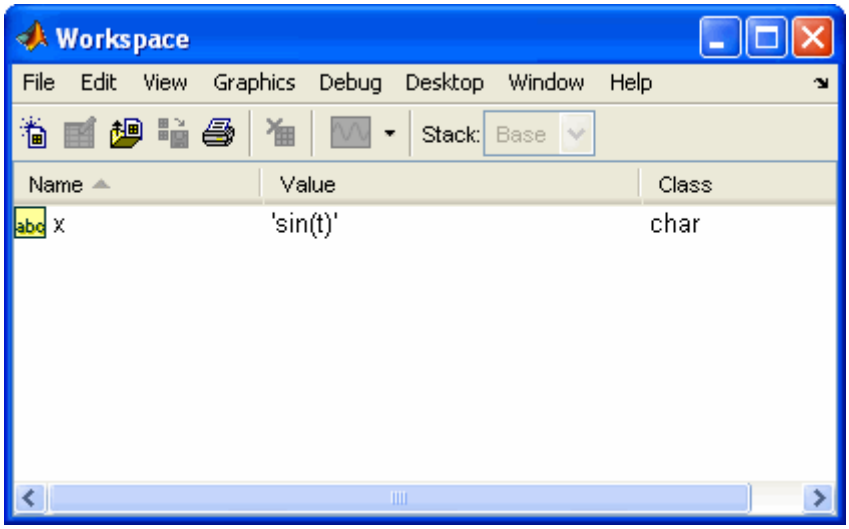
٤- Curve Fitting

EVAL

قبل البدء بالتطبيق الأول، لابد 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



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



ولرسم موجة جيب الزاوية لابد 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 بإستخدام الأمر **eval**,حيث 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

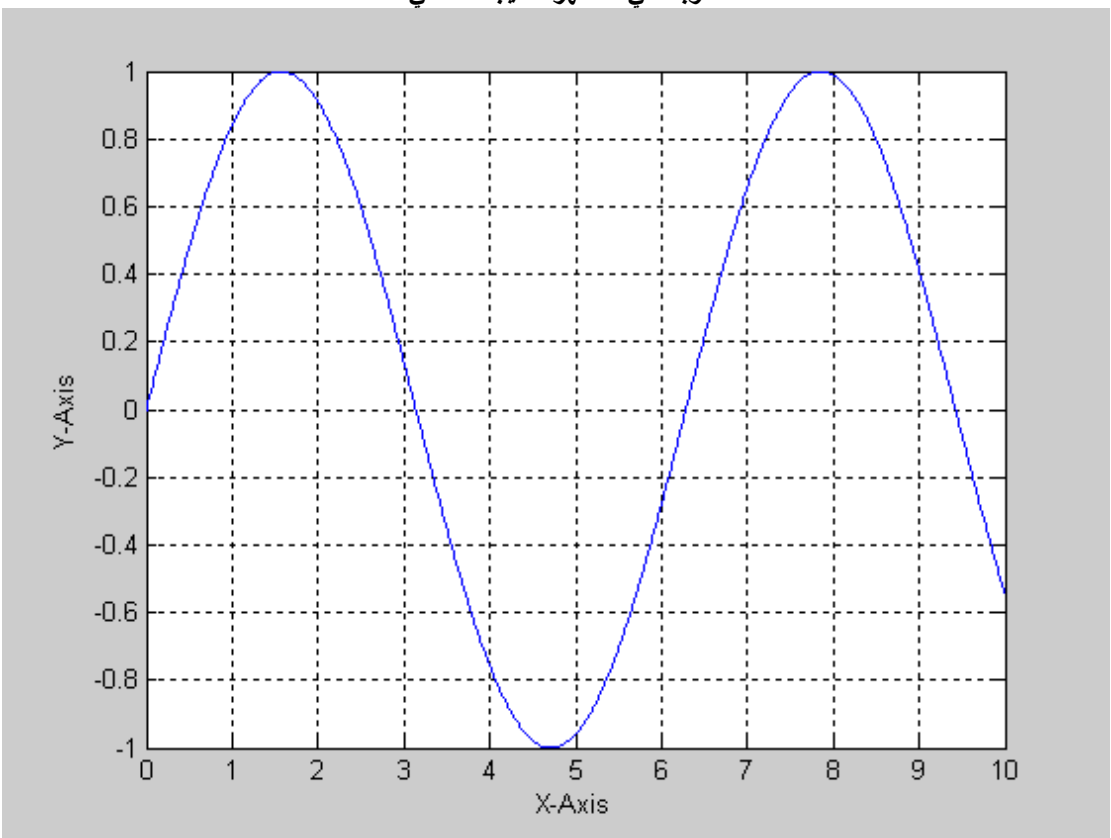
File Edit Text Go Cell Tools Debug Desktop Window Help

Stack: Base

```
1 -  clc
2 -  clear
3 -  close all
4 -  x='sin(t)';                % By Writting 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
```

script Ln 11 Col 1 OVR

وبالتالي ستظهر النتيجة كالتالي



ونحن الآن على أتم إستعداد لتنفيذ التطبيق الأول

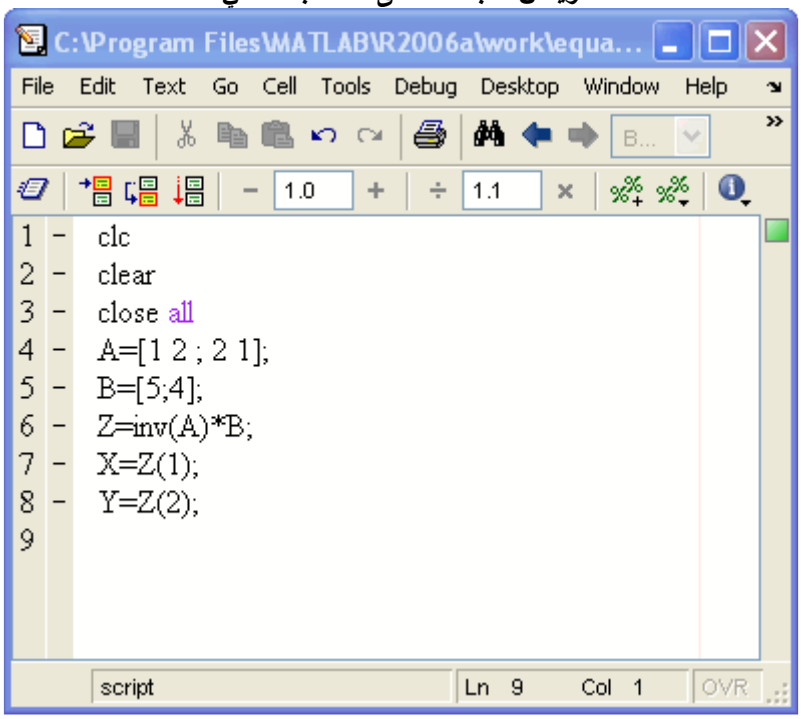
حل ثلاثة معادلات

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

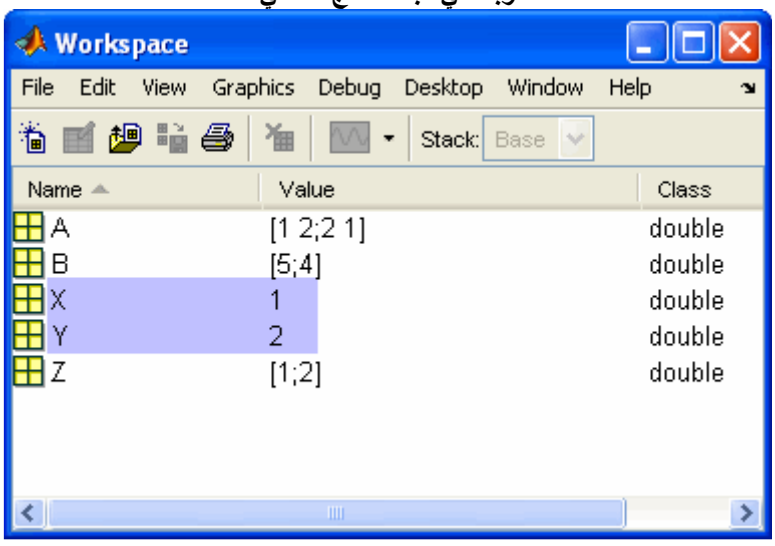
$$X+2Y=5$$
$$2X+Y=4$$

$$\begin{pmatrix}1 & 2\\ 2 & 1\end{pmatrix}\begin{pmatrix}X\\ Y\end{pmatrix}=\begin{pmatrix}5\\ 3\end{pmatrix}$$
$$A\times Z=B$$
$$Z=A^{-1}\times B$$

ويمكن كتابة ذلك على الماتلاب كالتالي



وبالتالي نجد النتائج كالتالي



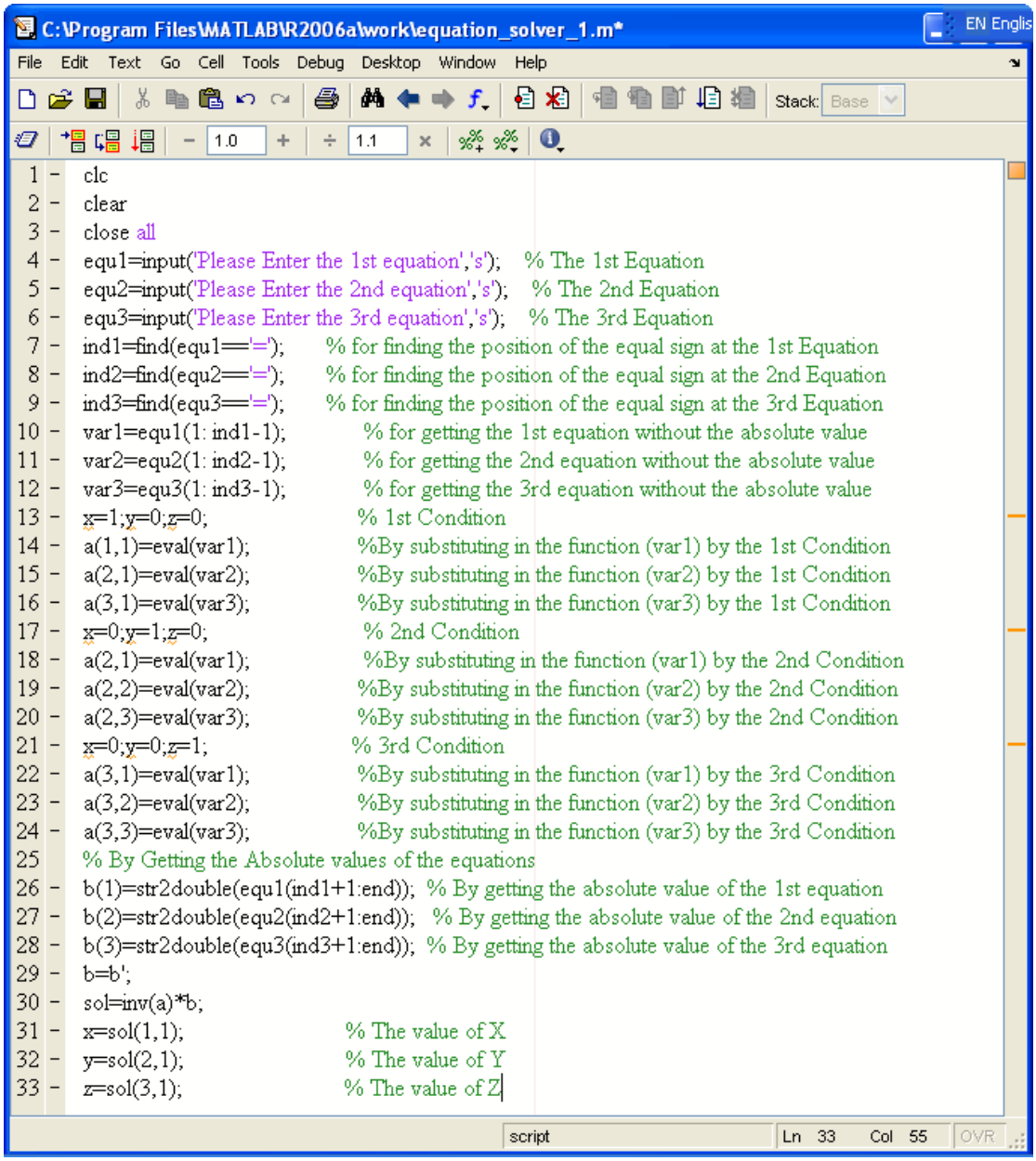
ولكن هذا يشترط أن يتم إدخال المعاملات coefficients للمعادلتين، وهذا بالتالي يتطلب التمحيص والتدقيق في كل معادلة، فإذا كثرت المعادلات إزداد الوقت المستغرق في البحث، فتسهيلاً للمستخدم يجب عمل برنامج لإدخال المعادلات بشكل كام، فمثلاً سنقوم بعمل برنامج لحل ثلاثة معادلات

$$A_1 \times X + B_1 \times Y + C_1 \times Z = D_1$$
$$A_2 \times X + B_2 \times Y + C_2 \times Z = D_2$$
$$A_3 \times X + B_3 \times Y + C_3 \times Z = D_3$$

ولحل هذه المعادلات يجب وضعها في الصورة التالية

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

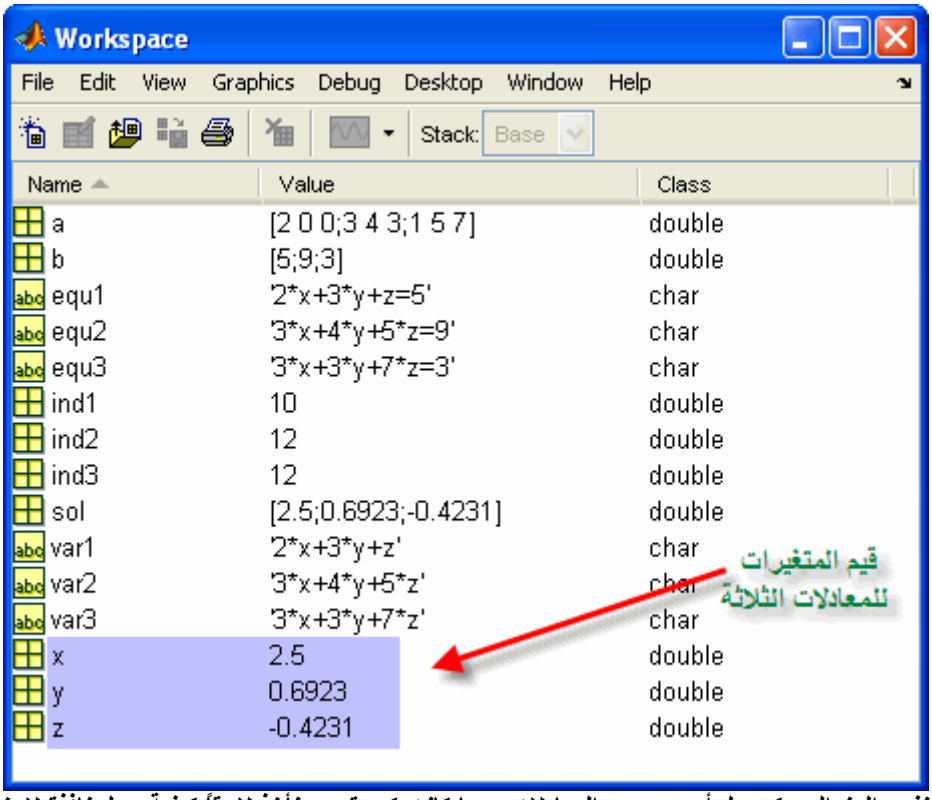
ولذلك يجب التفكير كالتالي، عندما يقوم المستخدم بإدخال المعادلات الثلاثة، يجب على الماتلاب أن يحدد قيم المعاملات أولاً لإيجاد حل قيم المتغيرات، وحلاً لهذه المشكلة سنجعل الماتلاب يبحث عن علامة"="
لكل معادلة، ثم سنقوم بتعريف الجزء الذي يحتوى على المتغيرات لكل معادلة، ثم سنقوم بفرض أن
x=1 , y=0,z=0
ومنها نحصل على قيم معاملات **x** في المعادلات الثلاثة
ثم سنفرض أن**x=0, y=1, z=0**
ومنها نحصل على قيم معاملات **y** في المعادلات الثلاثة
ثم سنفرض أن**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.



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

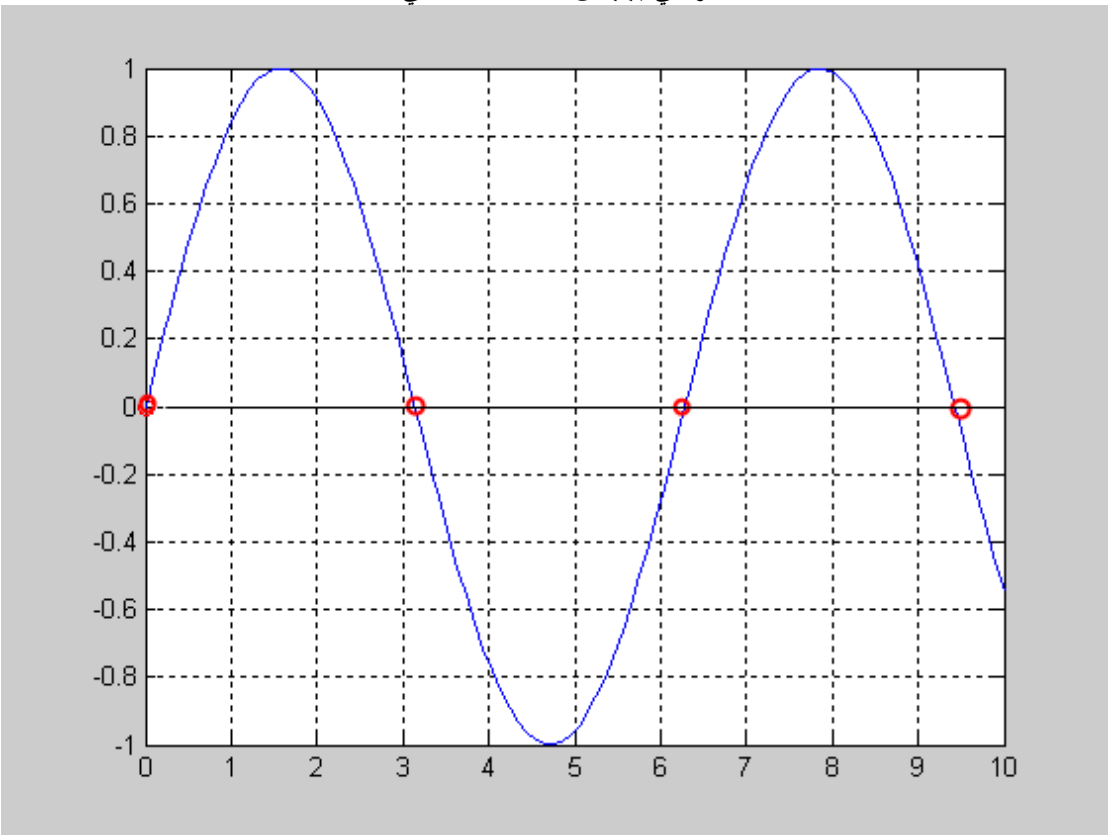


وعلى نفس المنوال يمكن حل أي عدد من المعادلات مهما كانت كبيرة، وسنأخذ لاحقاً كيفية عمل نافذة لإدخال تلك المعادلات

وبهذا نكون قد أتممنا 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\MATLAB\R2006a\work\zero_crossing.m*

File Edit Text Go Cell Tools Debug Desktop Window Help

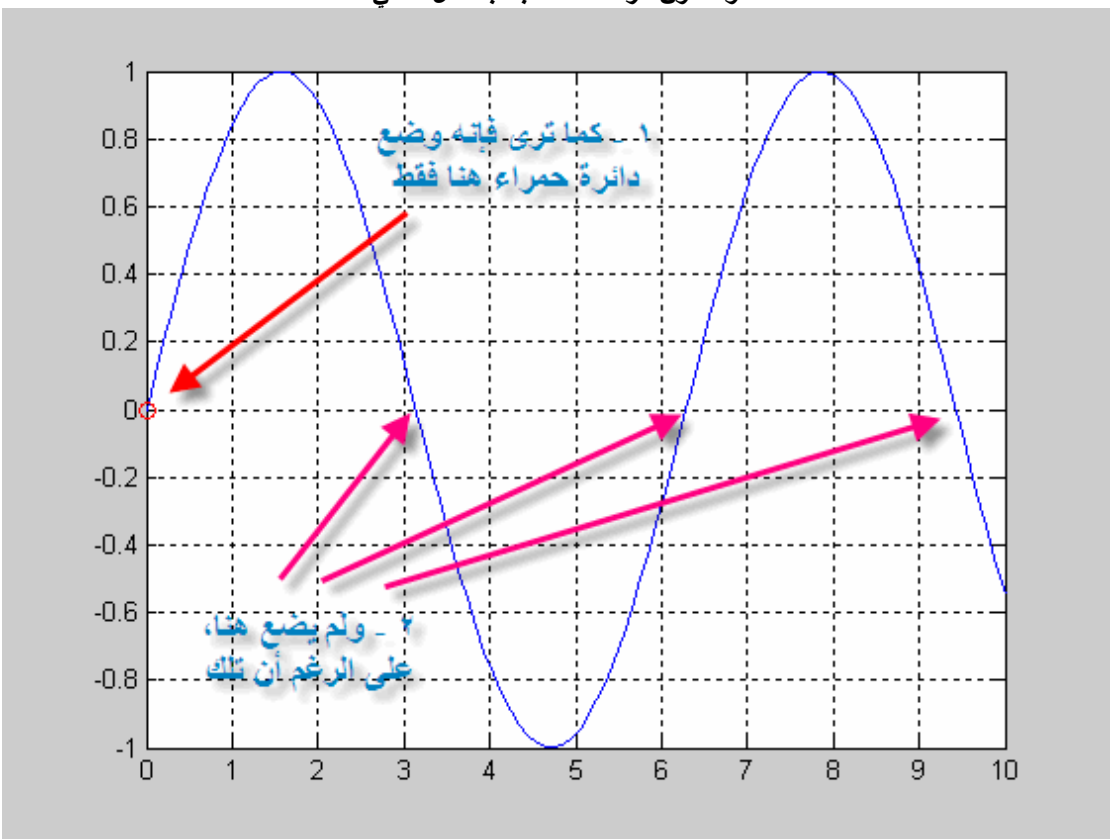
Stack: Base

1.0 1.1

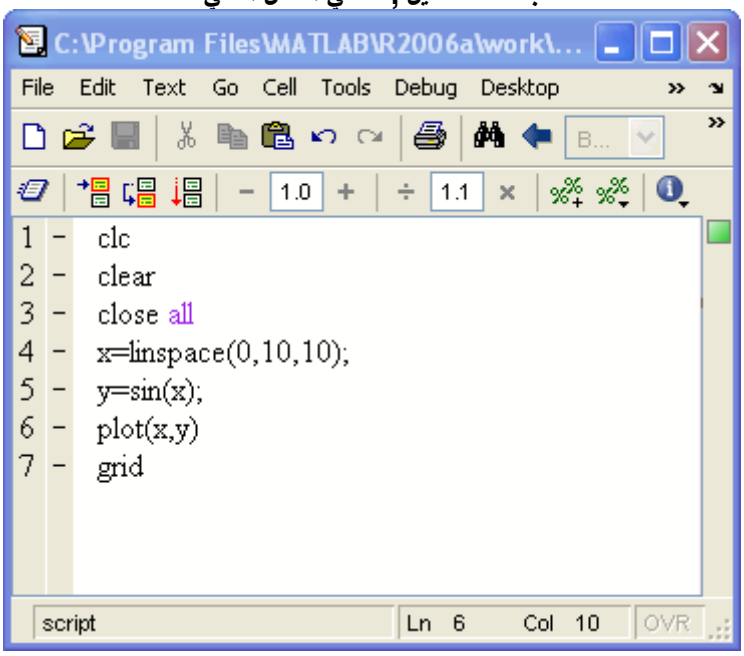
```
1 -  clc
2 -  clear
3 -  close all
4 -  x=linspace(0,10,100);          % Defining the inputs
5 -  y=sin(x);                      % By defining the function
6 -  ind=find(y==0);                % By searching the position of y=0
7 -  x_crossing=x(ind);             % By finding the position of zero 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

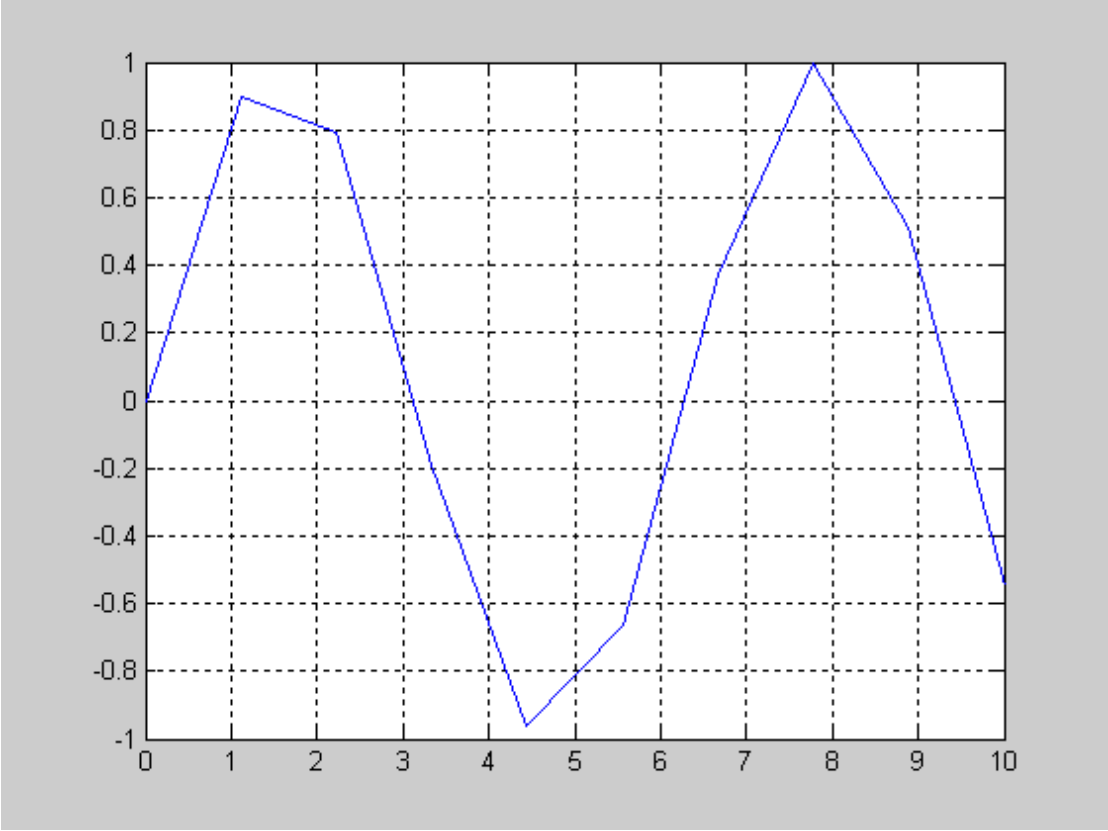
وستكون الرسمة الناتجة بالشكل التالي



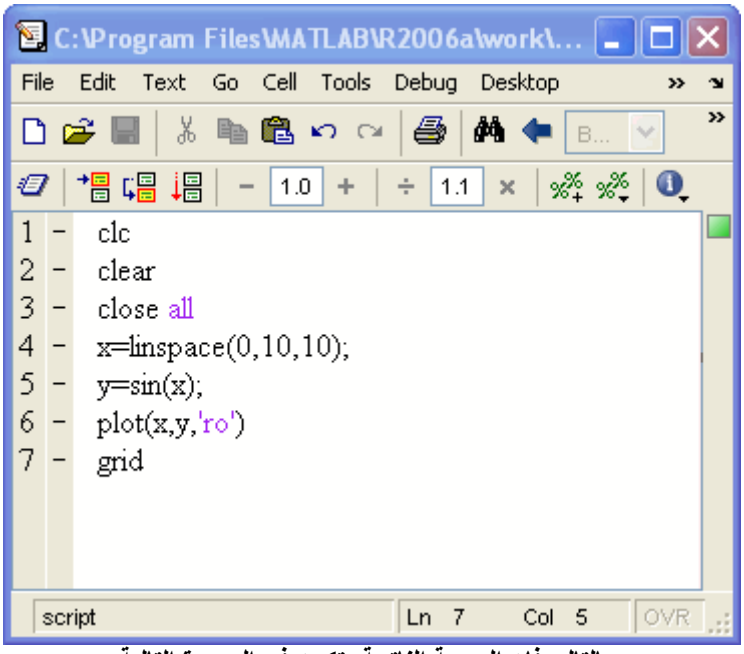
فما سبب هذه المشكلة، حقيقة سببها ما يسمى بـ Digital Domain, فما هو Digital Domain
معنى ذلك أن أي معادلة يتم رسمها ليست عبارة عن خط متصل وإنما مجموعة من النقاط، تعتمد على عدد النقاط التي تم إختيارها في لرسم الدالة، ثم التوصيل بين تلك النقاط، فمثلاً إذا قمنا بعمل برنامج لرسم دالة الجيب sine wave بعدد نقاط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



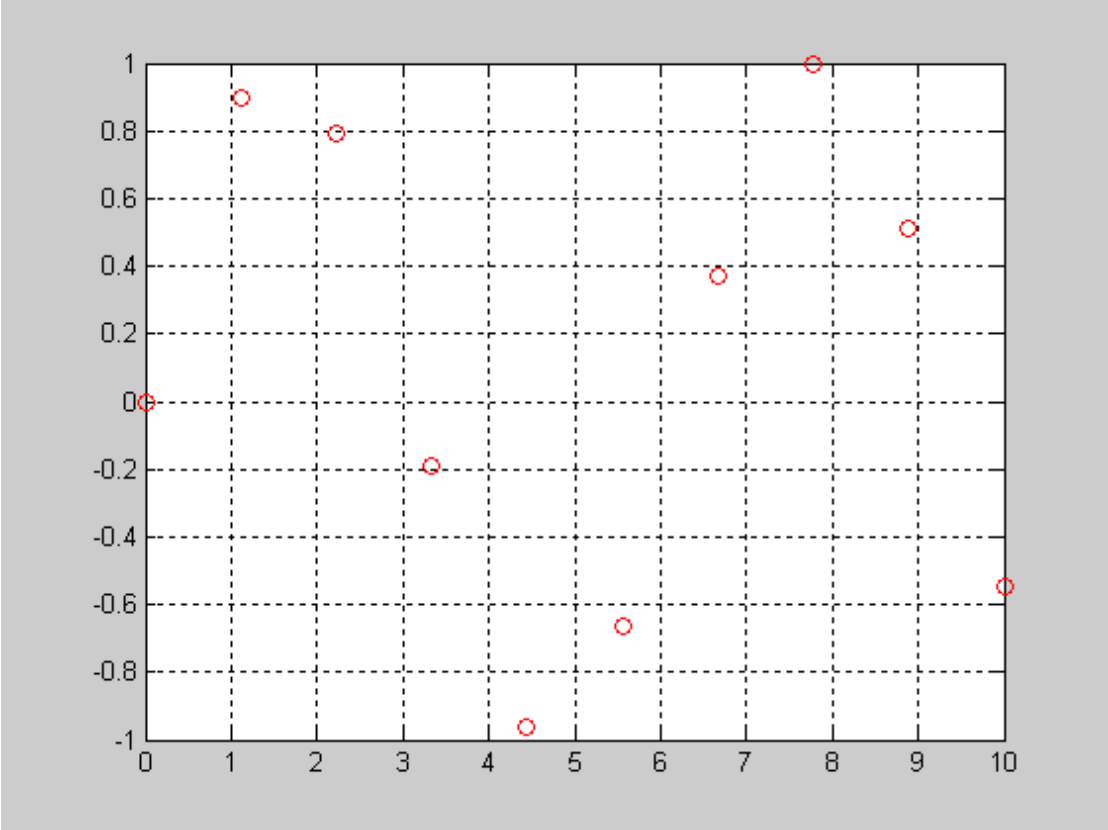
وبالتالي ستجد الدالة بهذا الشكل



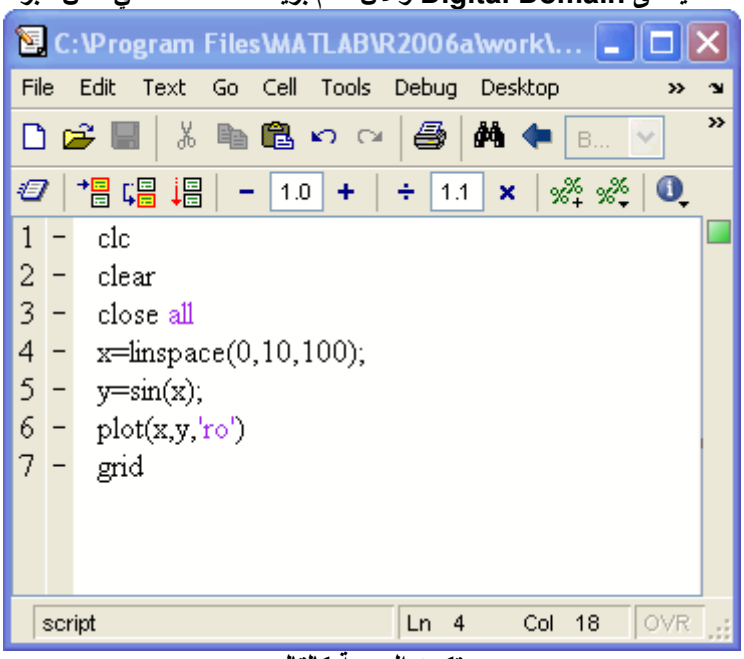
كما ترى، فإن الماتلاب قام بتحديد النقاط والتوصيل بينها، وللتأكد من ذلك قم بعمل التالي في البرنامج



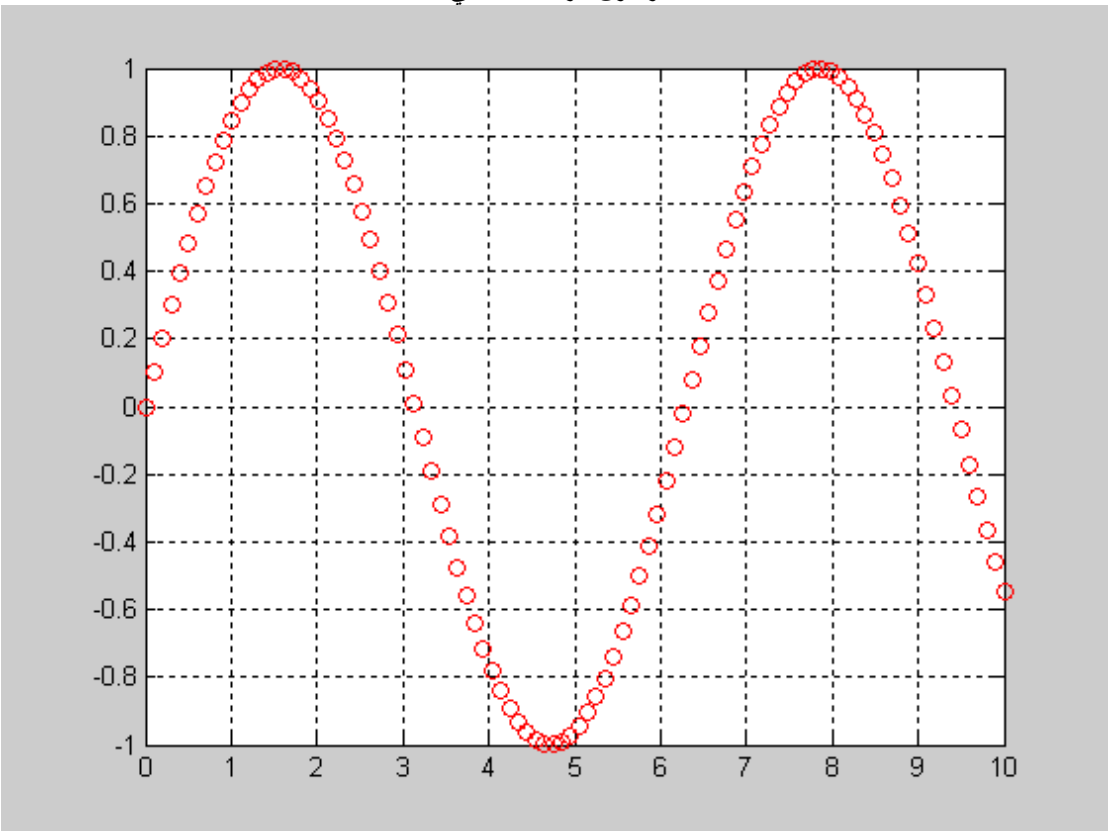
وبالتالي فإن الرسمة الناتجة، تكون في الصورة التالية



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



وتكون الرسمة كالتالي



كما ترى في الرسمة لايزال هنالك 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\MATLAB\R2006a\work\zero_crossing.m*

File Edit Text Go Cell Tools Debug Desktop Window Help

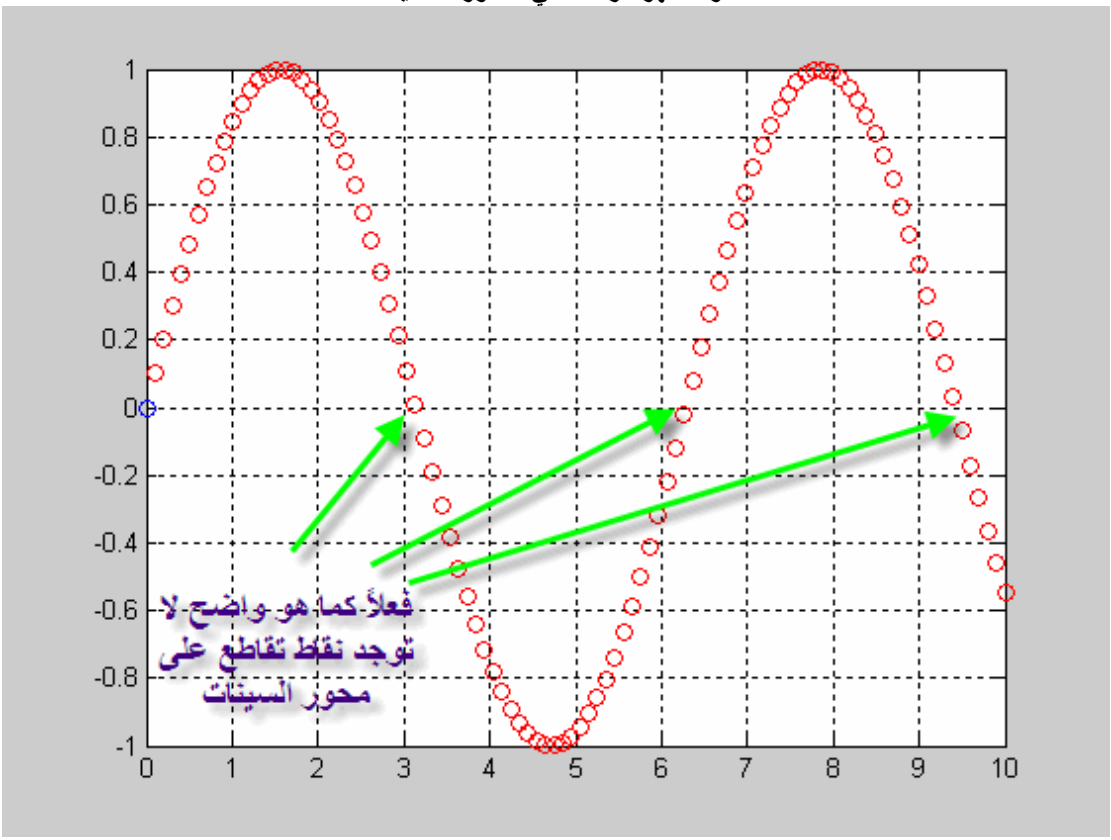
Stack: Base

```
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
```

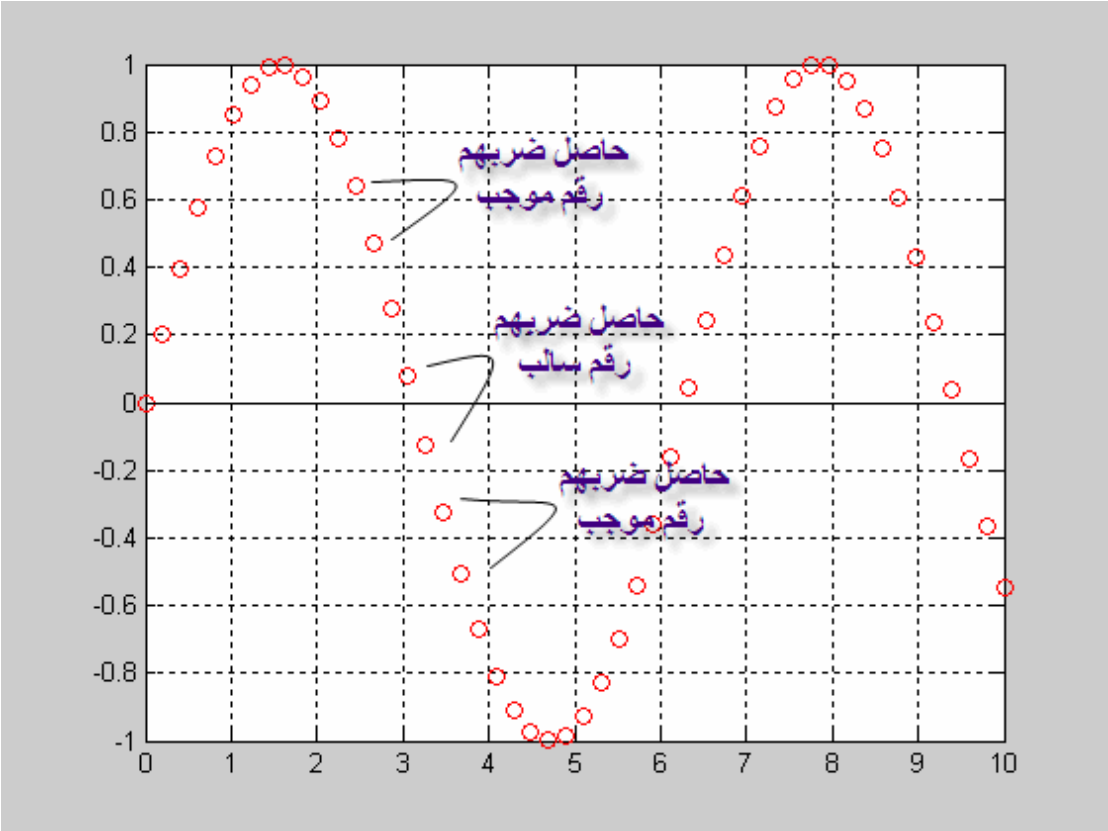
لإظهار الدالة في صورة Digital Domain

script Ln 10 Col 5 OVR

وستظهر الرسمة في الصورة التالية



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



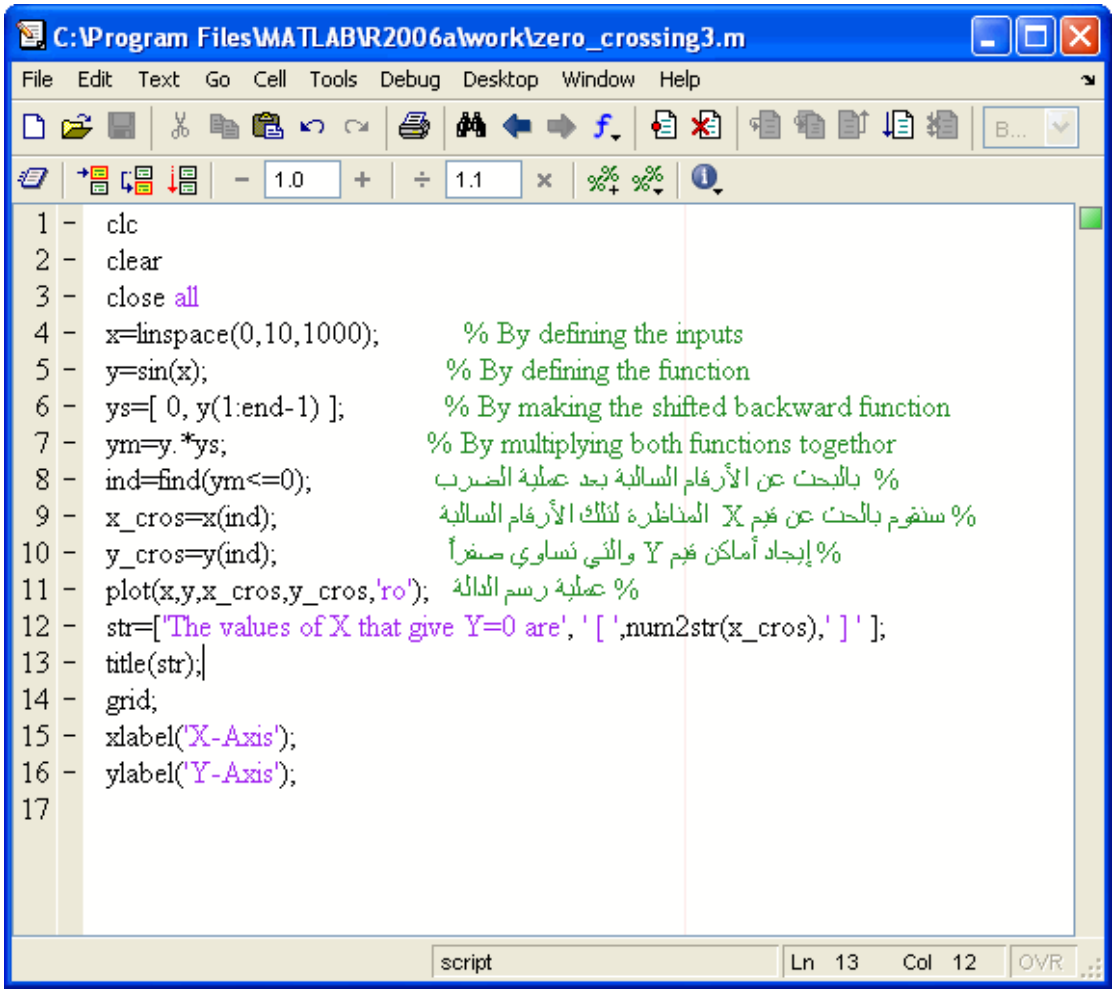
كل ما علينا هو ضرب كل نقطة في النقطة التي تليها، وذلك عن طريق تعريف y ثم تعريف نفس الدالة، ولكن متقدمة عنها برقم واحد، ثم ضربهم معاً، فمثلاً إذا كانت قيمة y كالآتي

[1 2 3 4]

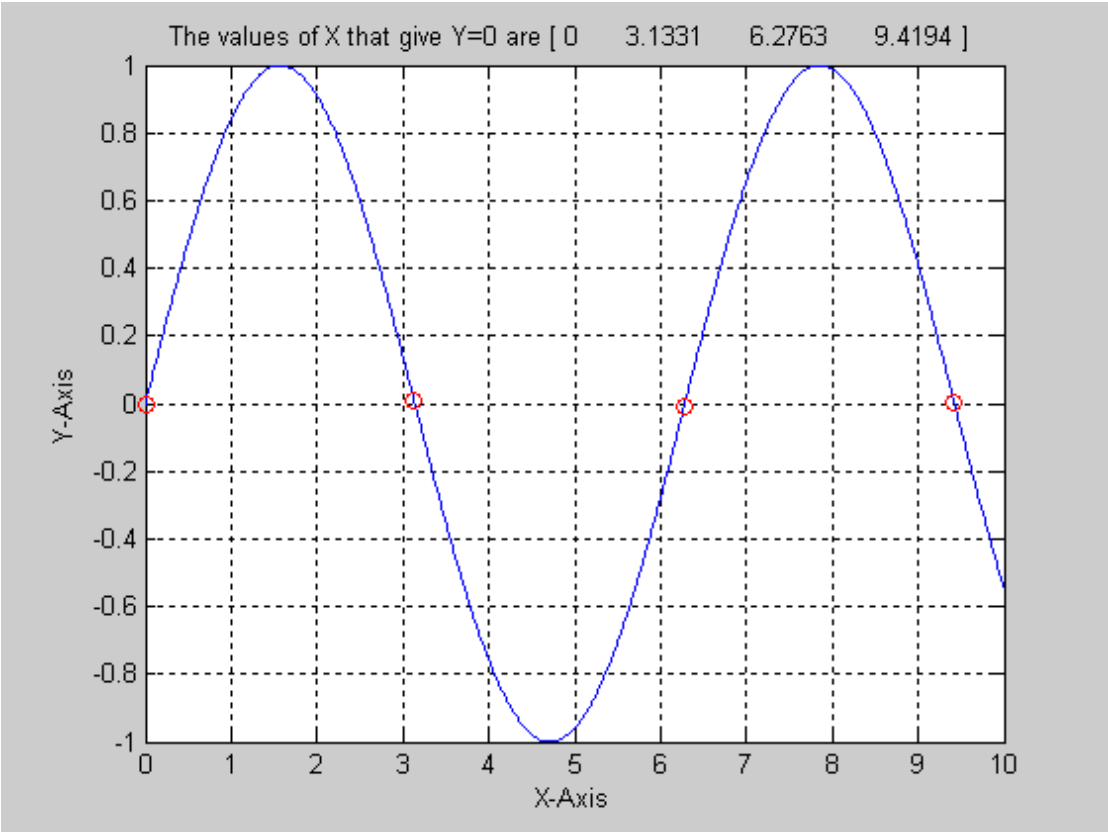
ثم سنقوم بإضافة رقم ليتم تحريك تلك القيم، ويفضل أن يكون الرقم صفراً، كما سيتم حذف الرقم الأخير حيث أنها ستكون أطول بعدد واحد فقط من الدالة الأصلية مما سيترتب عليه خطأ داخل الماتلاب وبالتالي فإن الدالة الجديدة ستكون متأخرة برقمهي

[0 1 2 3 4]

والآن سنقوم بكتابة البرنامج بالتدريج التالي



وبالتالي ستلاحظ الرسمة التالية



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 بإستخدام أمرين جديدين وهما

trapz لإيجاد المساحة تحت المنحنى

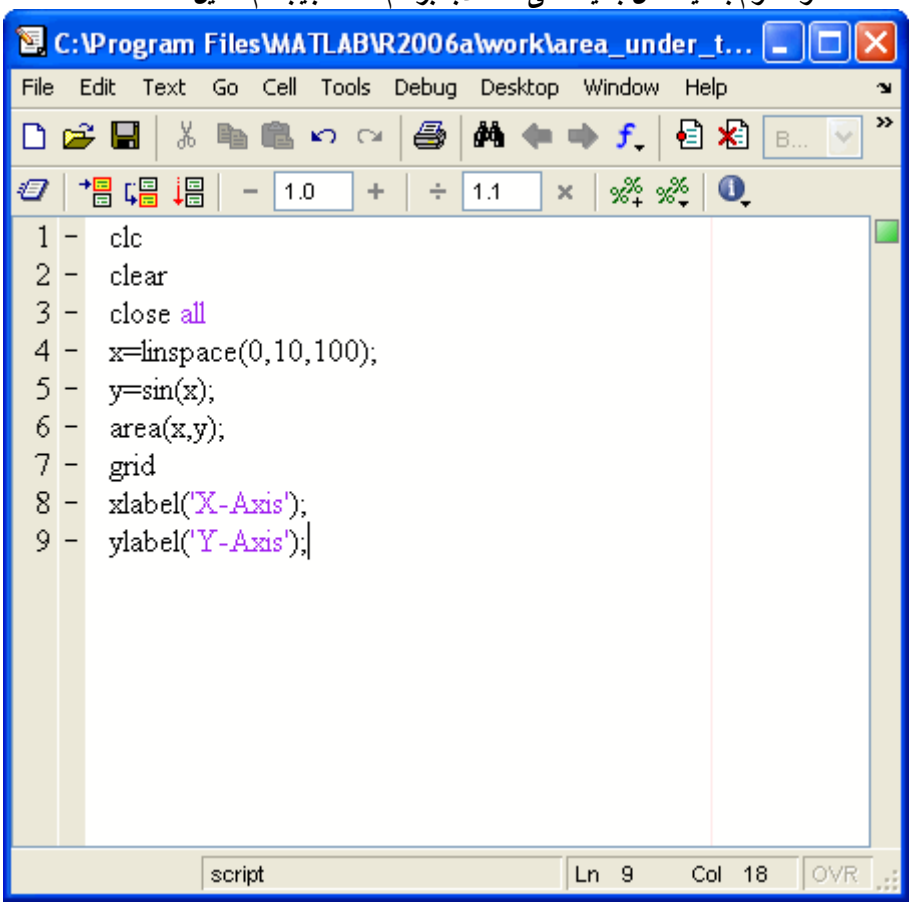
area لتظليل تلك المساحة من الدالة

وسنقوم بشرح الأمر **area**

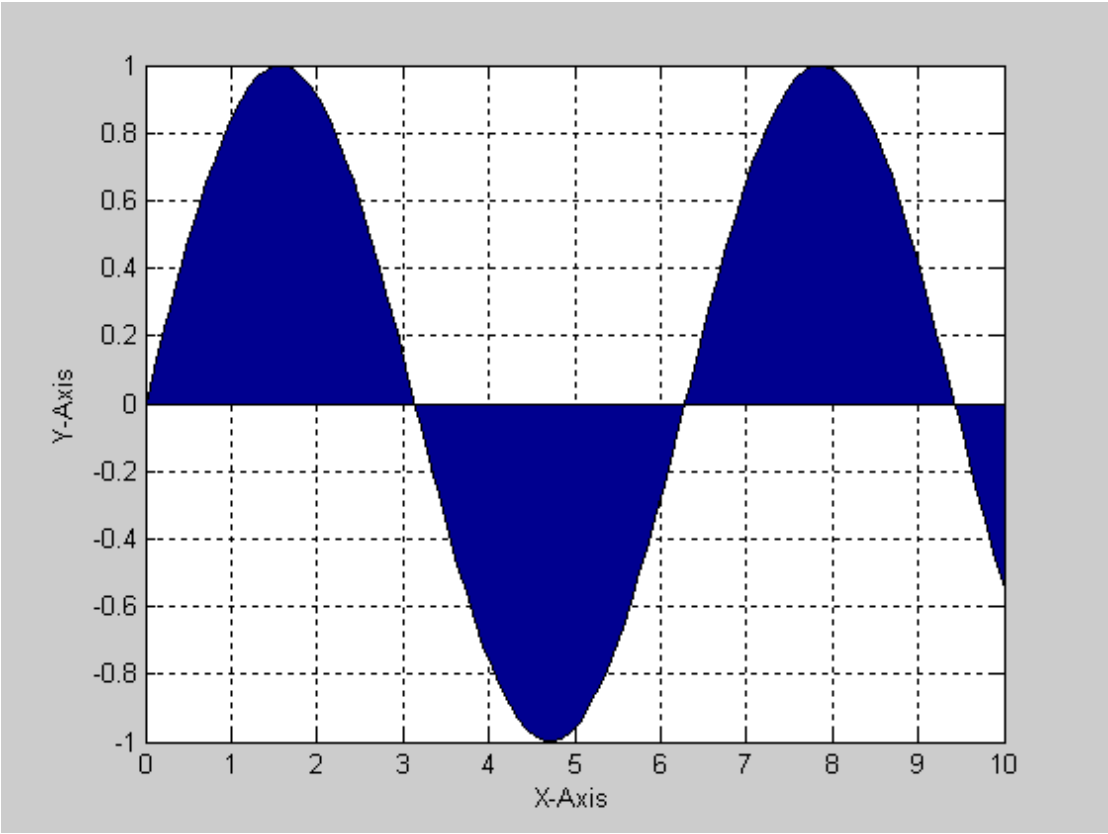
حيث يأخذ الصورة التالية

```
area( x, y )
```

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



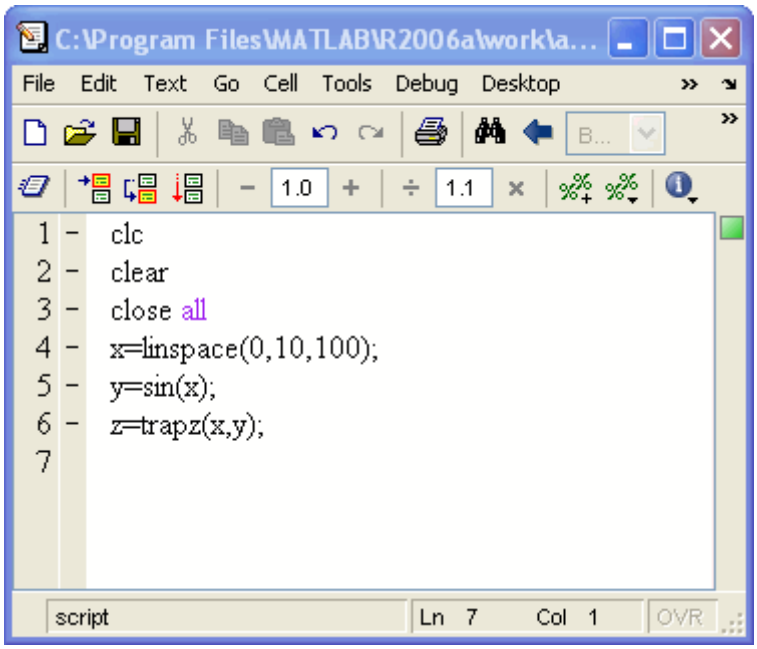
وستظهر الرسمة كالتالي



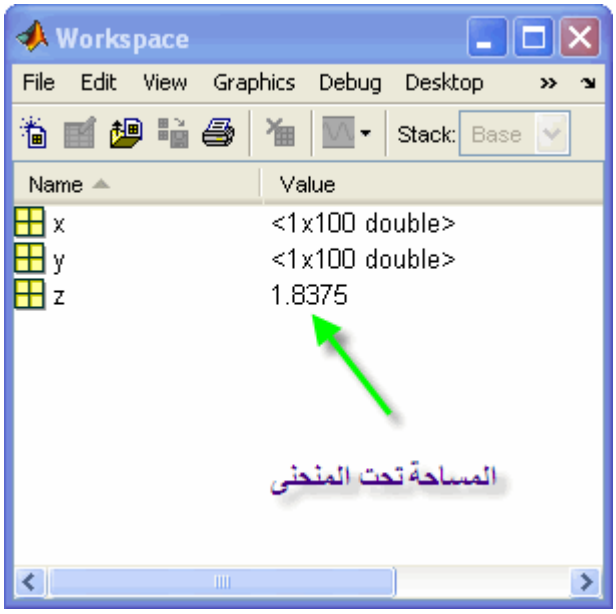
أما بخصوص الأمر **trapz** فيستخدم في إيجاد المساحة تحت المنحنى، حيث يأخذ الصورة التالية

```
trapz( x , y )
```

وسنقوم بعمل برنامج بسيط في إيجاد المساحة تحت منحنى دالة الجيب



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 outputsثم سنقوم بإختيار النقطتان من على الرسمة، ثم سنوجد المساحة تحت المنحنى بين تلك النقطتين، ثم سنقوم بتظليل المساحة بين النقطتين.

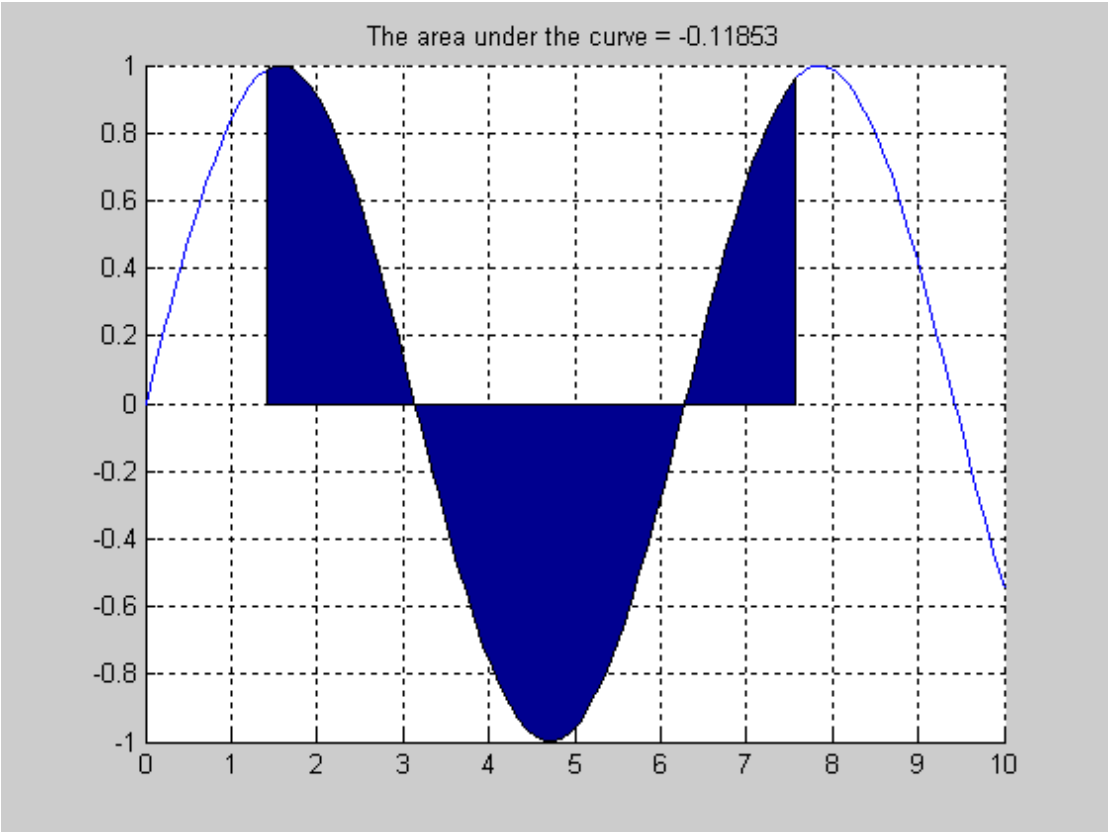
C:\Program Files\MATLAB\R2006a\work\area_under_the_curve.m

File Edit Text Go Cell Tools Debug Desktop Window Help

```
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 delecting 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);
```

script Ln 16 Col 12 OVR

وتم إختيار نقطتان عشوائيتان، وظهرت الرسمة كالتالي

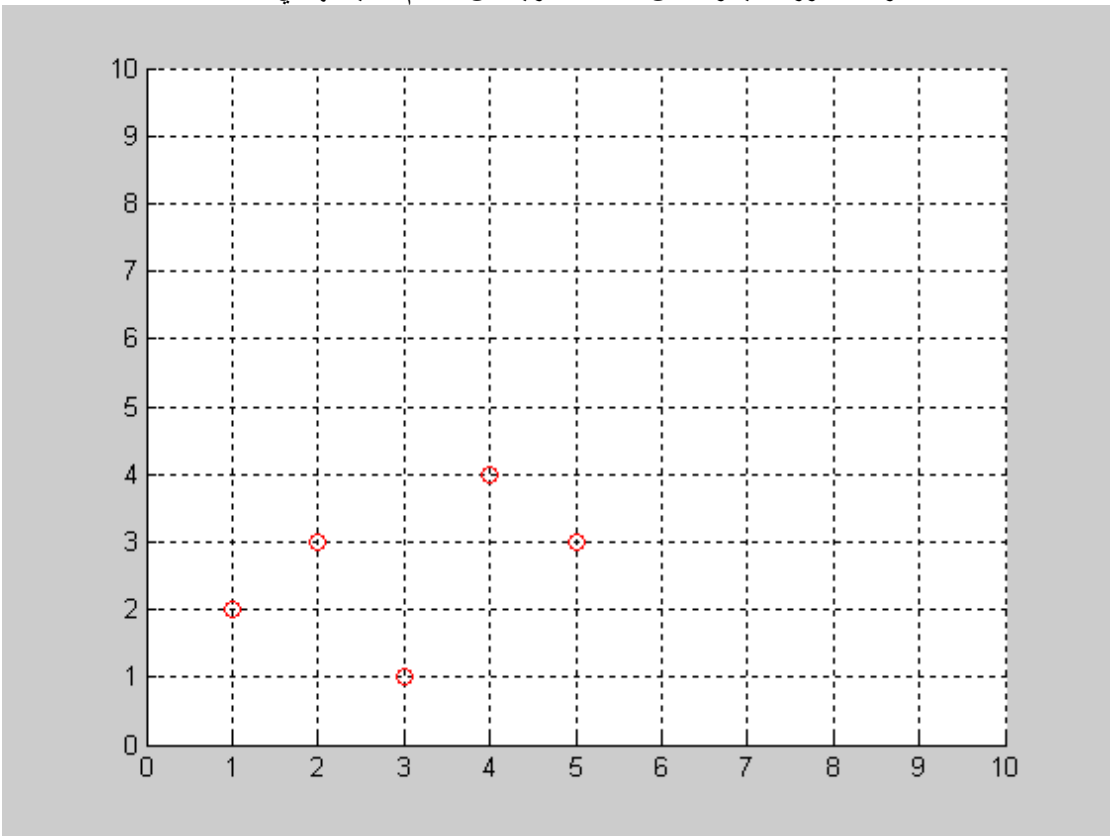


وبهذا يكون التطبيق الثالث قد تم بنجاح
التطبيق الرابع والأخير في هذه الدورة سيتم تنفيذه المشاركة القادمة

Curve Fitting

سنقوم الآن بأخذ المثال التطبقي الثالث والأخير وهو **Curve Fitting** حيث أن هذه العملية هامة جداً في إيجاد علاقة مكافأة لأي نظام، فمثلاً عند إدخال مجموعة من المدخلات **inputs** سنلاحظ أن الخرج **outputs** تأخذ مجموعة من النقاط المتشتتة التي لا تجمعها علاقة محددة، أما عند إستخدام **curve fitting** سنلاحظ تكون علاقة تقريبية لتوصيف النظام.

وهذه صورة لمجموعة من النقاط الخارجة من النظام لا تجمعها أي علاقة

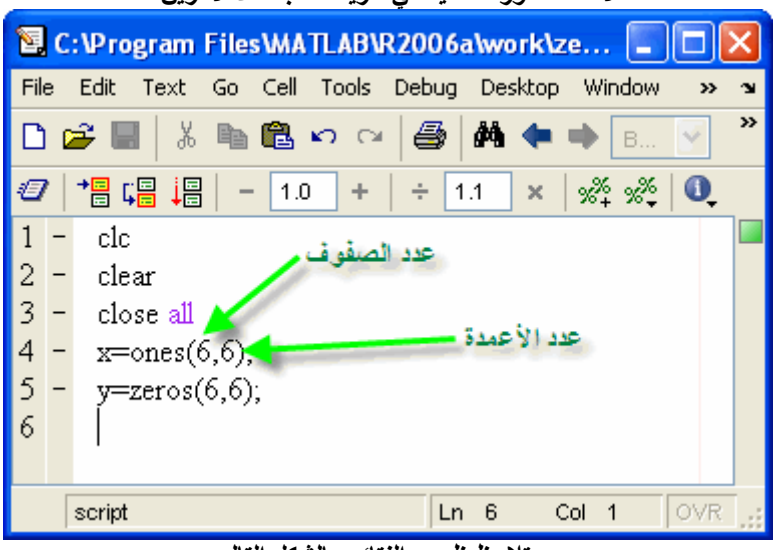


ولكن سنقوم بشرح أمرين وهما **ones** و **zeros** واللذان لهما القدرة التالية

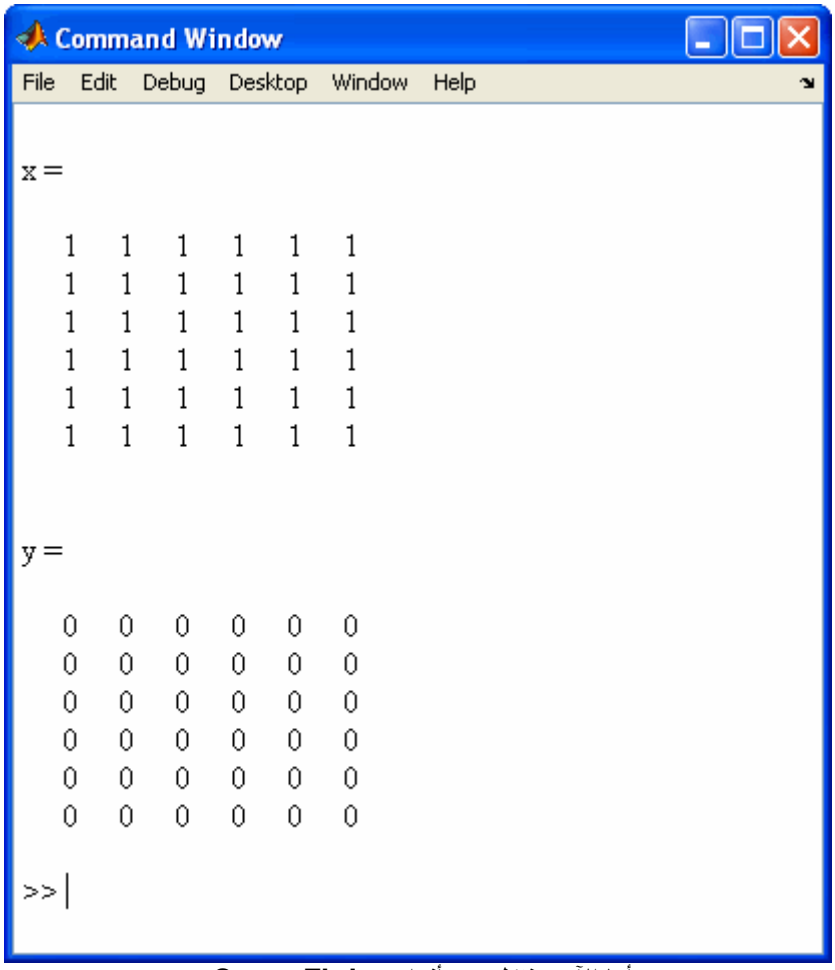
onesيستطيع أن يكون مصفوفة أو متجه جميع عناصره ١

zerosيستطيع أن يكون مصفوفة أو متجه جميع عناصره صفر

لاحظ الصورة التالية في طريقة كتابة كلا الأمرين



وستلاحظ ظهور النتائج بالشكل التالي



أما الآن سنتكلم عن أنواع Curve Fitting

هنالك أنواع عديدة منها

1- linear

2- Quadratic

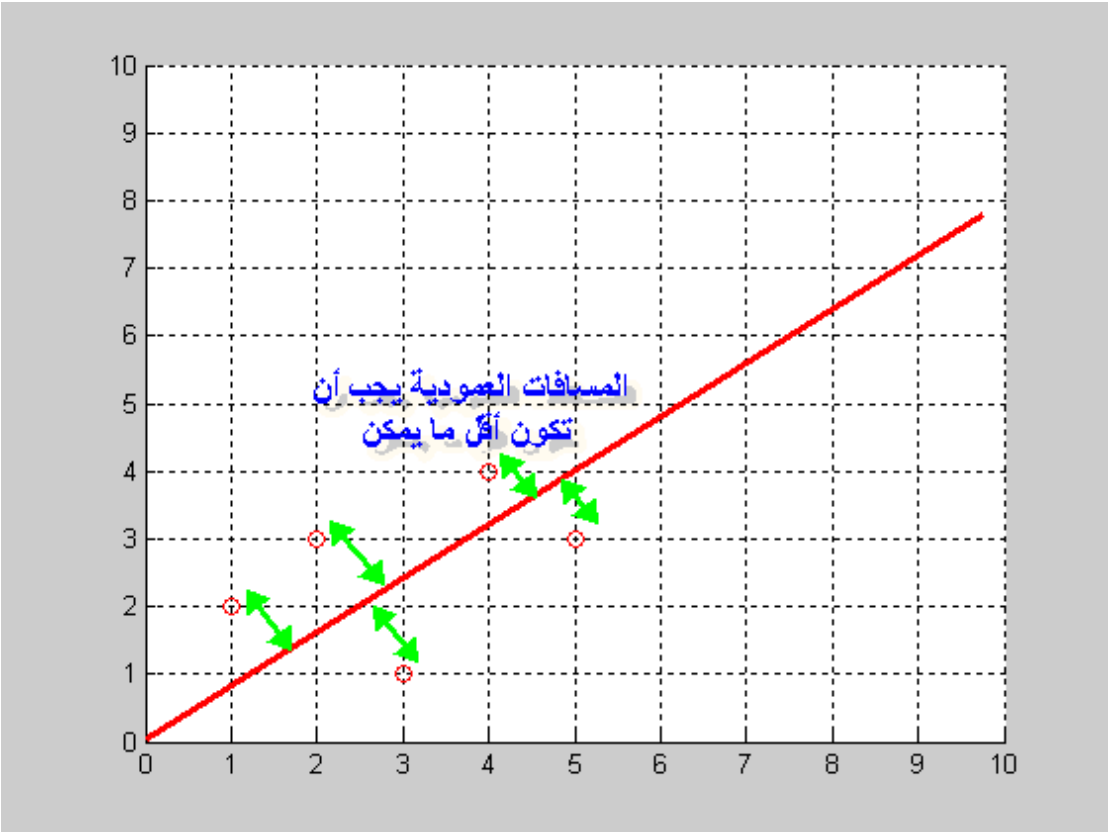
3- Sinusoidal

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

ويمكنا كما تعلمنا كتابة تلك المعادلة في الصورة التالية

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

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

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

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

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

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

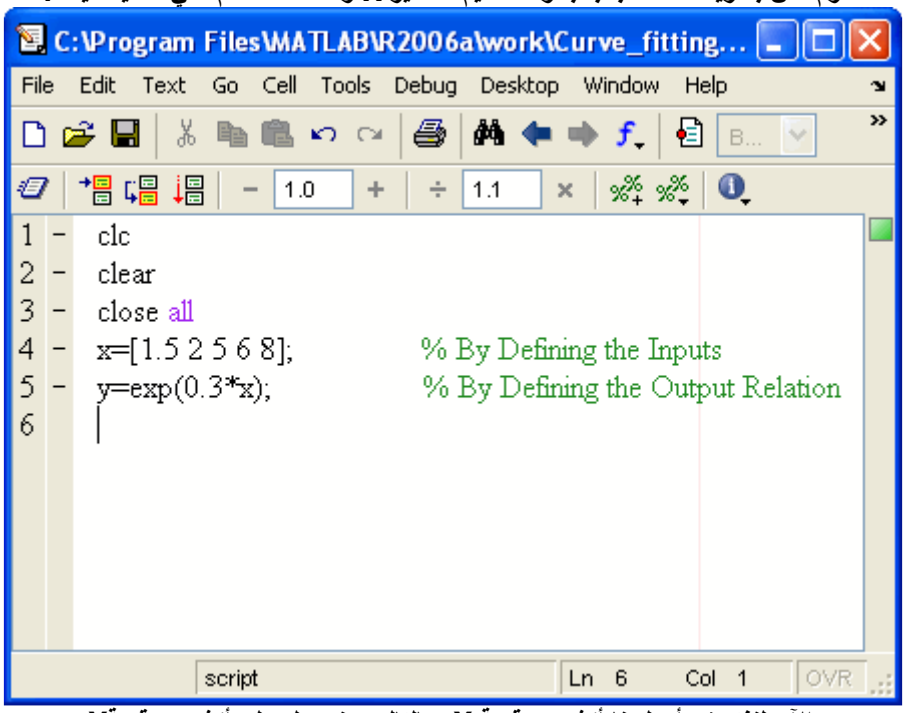
بالحصول على قيمة Y ومنها نقوم برسم العلاقة بين X & Y والتي تمثل خطاً تبعاً للمعادلة التالية

$$Y=KX+T$$

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 قيمةY



والآن لنفترض أن لدينا أكثر من قيمة X وبالتالي سنحصل على أكثر من قيمةY

وحيث أن العلاقة بين X & Y خطية كما ذكرنا مسبقاً فإننا بالتالي سيكون لدينا أكثر من معادلة يمكن كتابتها في الصورة التالية

$$Y_1 = K X_1 + T$$

$$Y_2 = K X_2 + T$$

•

•

•

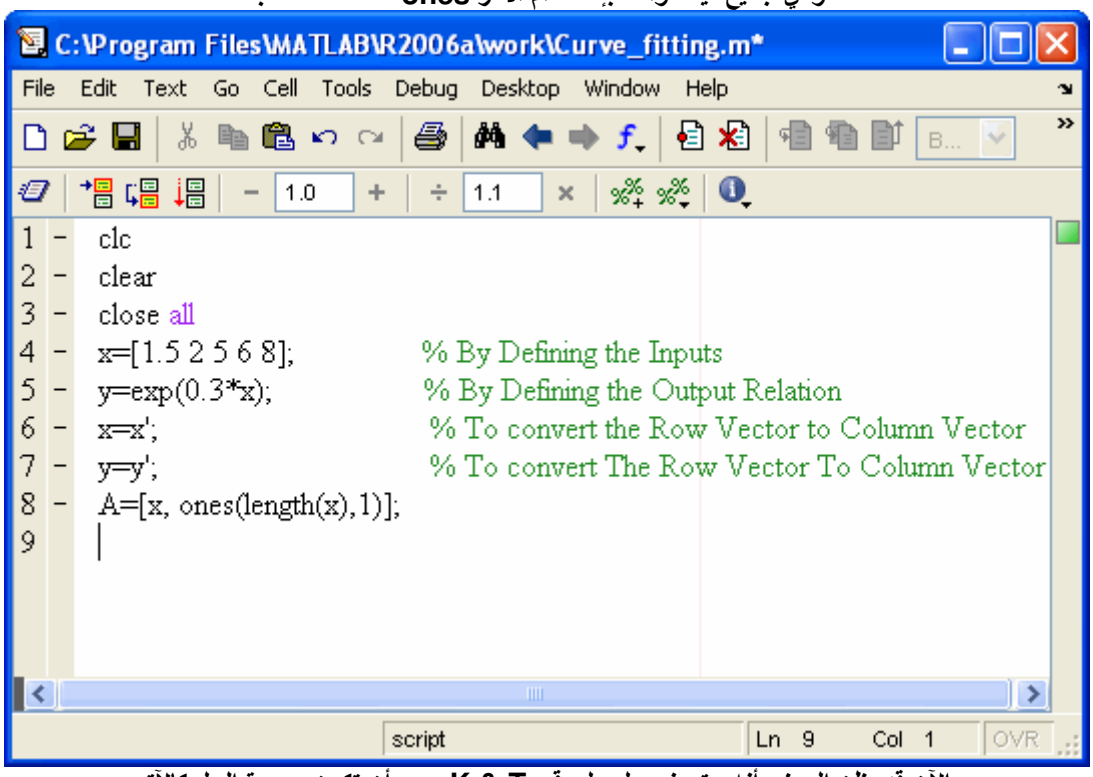
$$Y_n = K X_n + T$$

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 متجة عمودي جميع قيمه واحد بإستخدام الأمر **ones** كما تعلمنا مسبقاً



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

$$AX = B$$

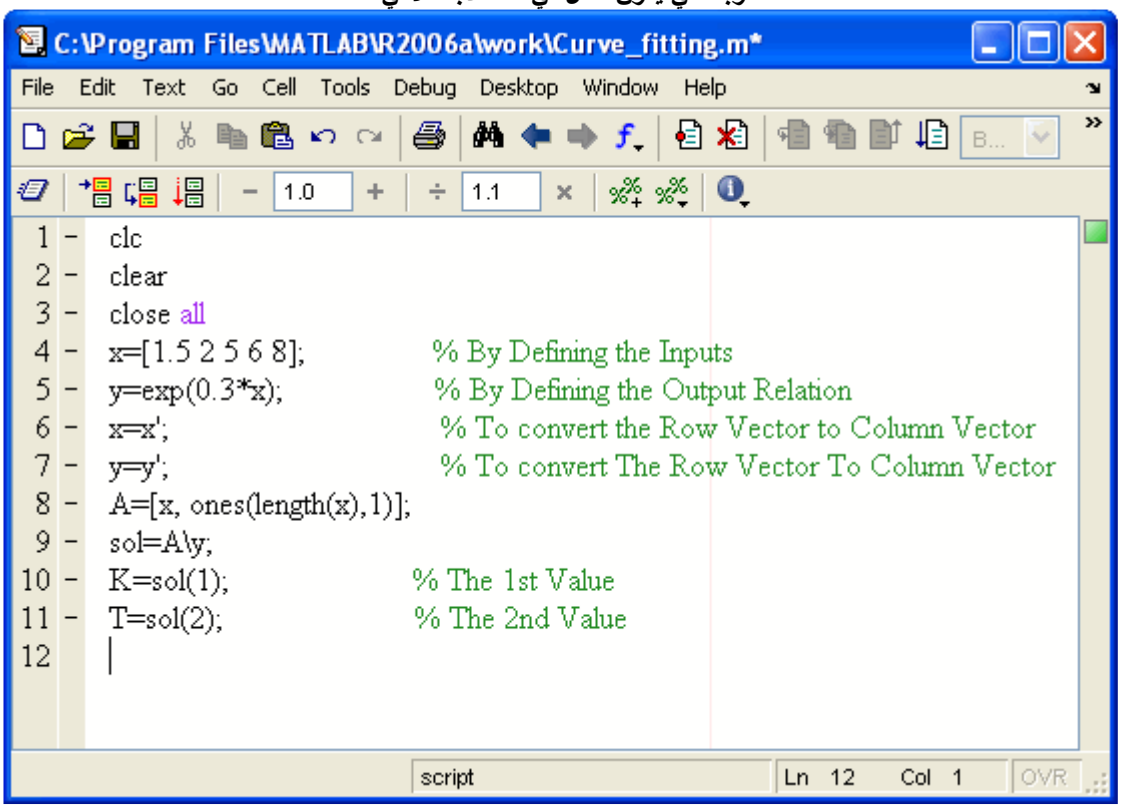
$$X = inv(A) \times B$$

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 إستخدام الأمر **inv** أي أن صورة الحل الصحيح تكون

$$AX = B$$

$$X = A \backslash B$$

وبالتالي يكون الحل في الماتلاب كالآتي



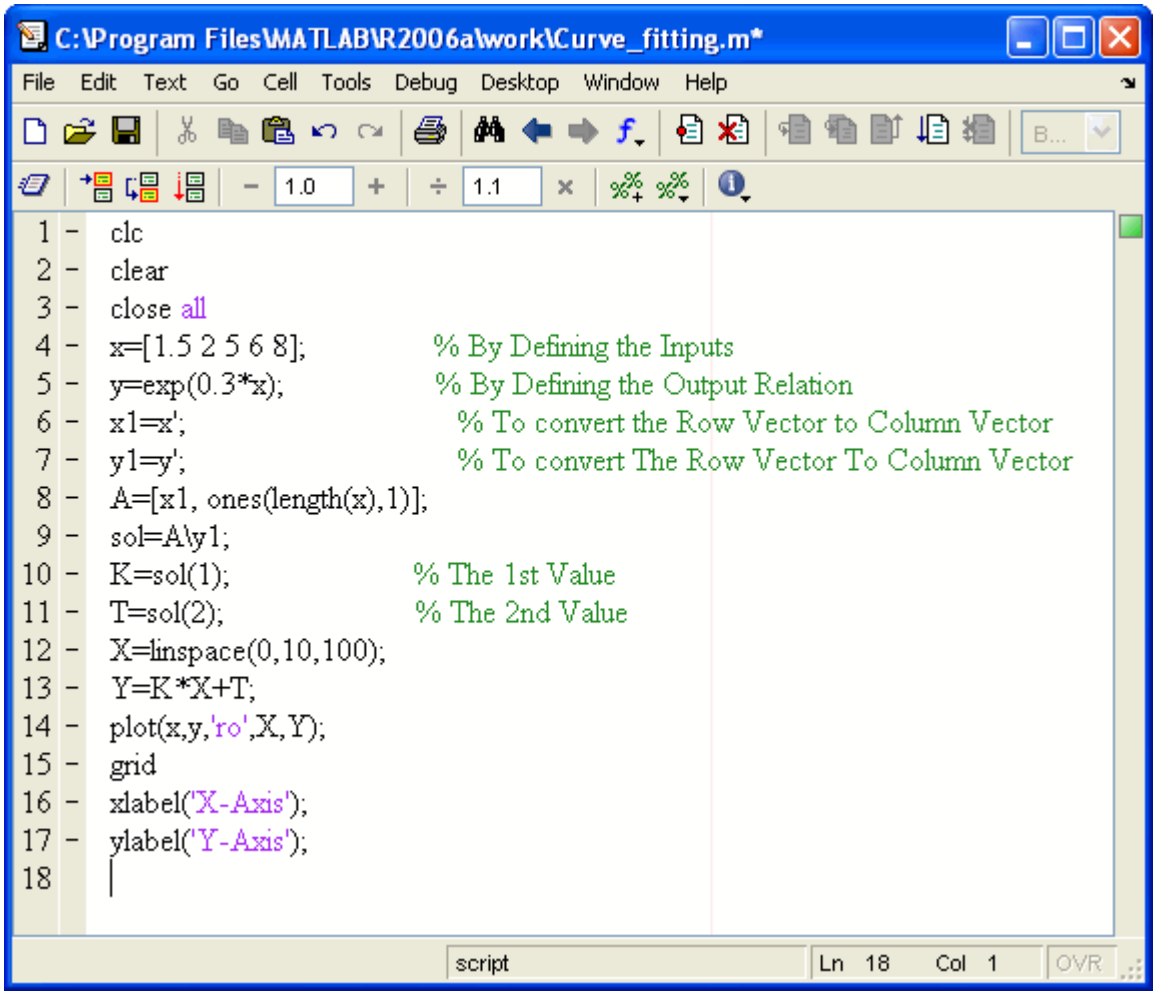
وبالتالي فإن المعادلة الناتجة والتي من خلالها سنرسم خطاً بحيث تكون المسافة العمودية بينه وبين النقاط أقل ما يمكن، تكون في الصورة التالية

$$Y=KX+T$$

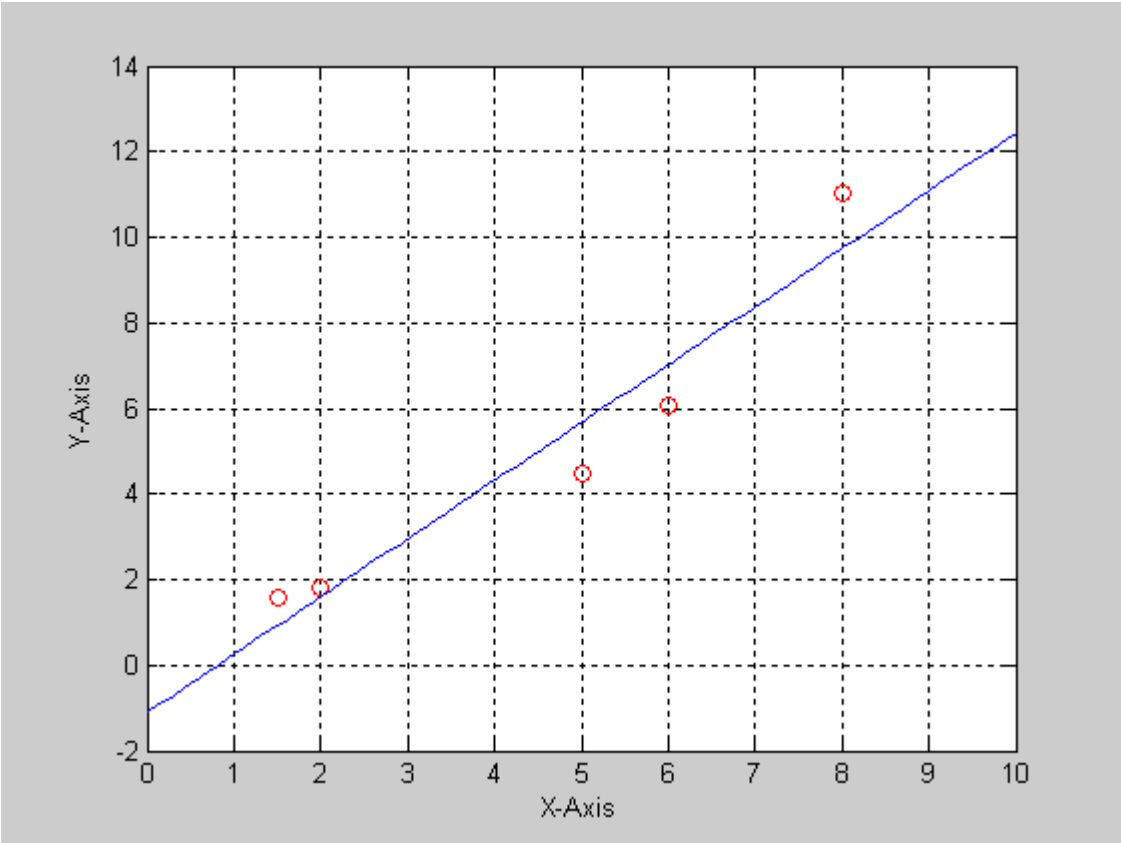
Where K & T are constant

والآن سنقوم بتعريف الماتلاب عدة نقاط بحيث نرسم ذلك الخط

!Error



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

$$Y_1 = K \times e^{X_1} + T$$
$$Y_2 = K \times e^{X_2} + T$$
$$\bullet$$
$$\bullet$$
$$Y_n = K \times e^{X_n} + T$$

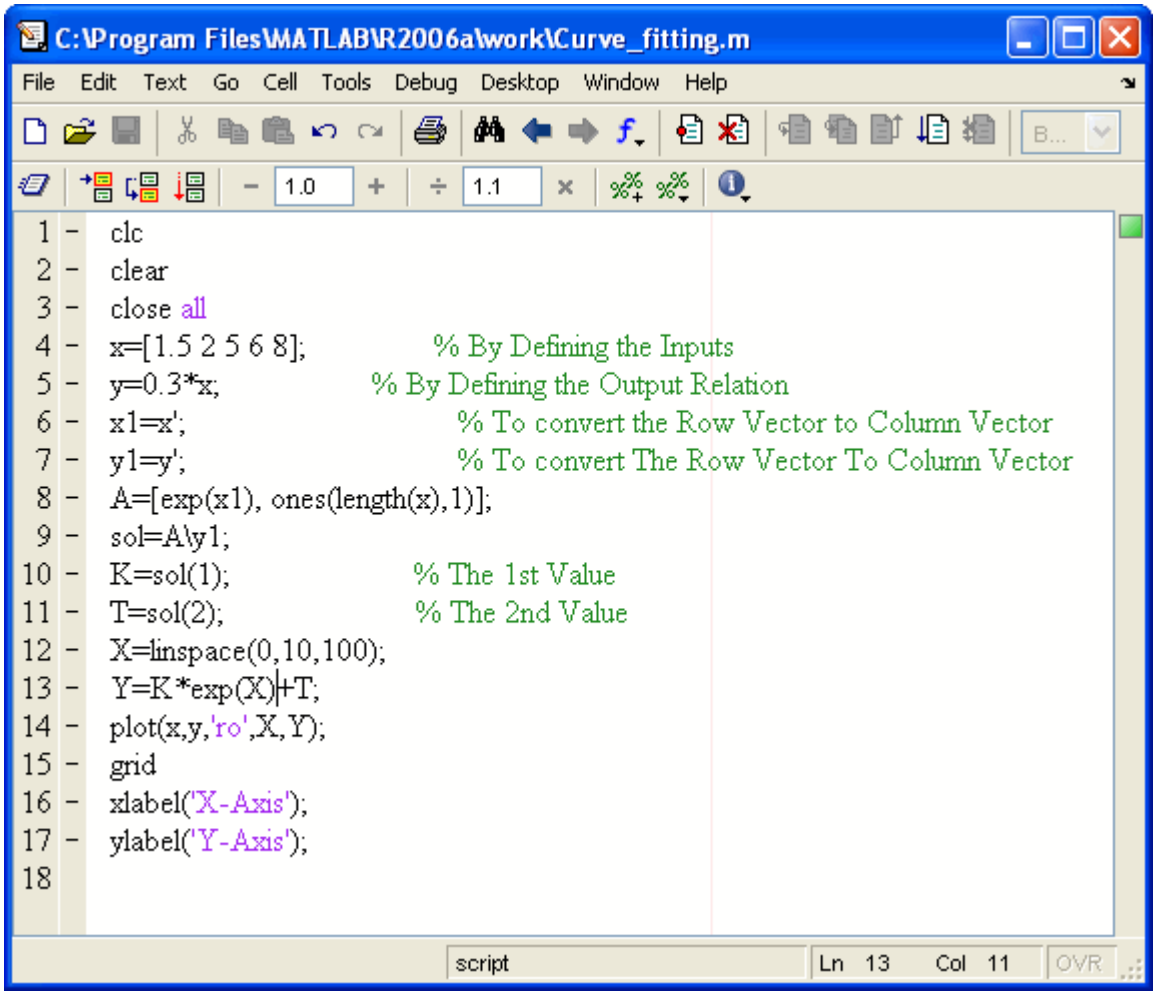
والتي يمكن كتابتها في صورة المصفوفة

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

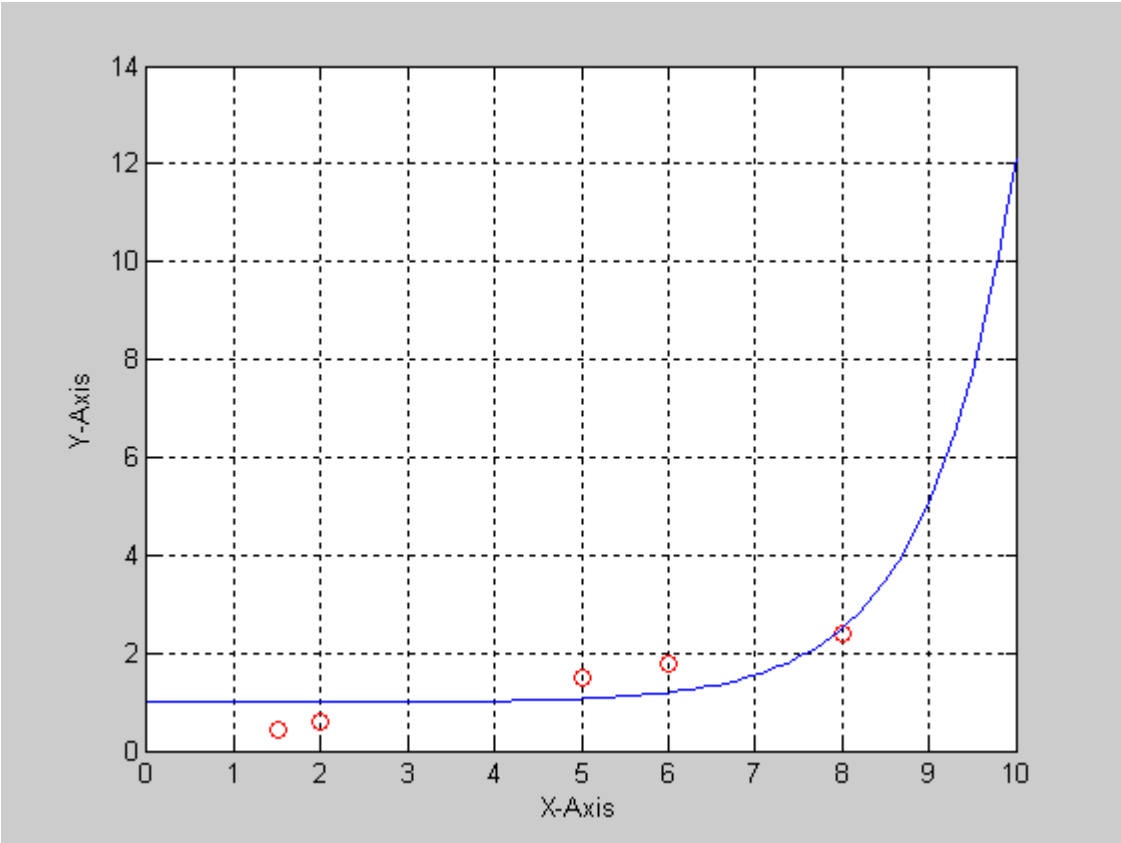
$$\begin{bmatrix} K \\ T \end{bmatrix} = \begin{bmatrix} e^{X1} & 1 \\ e^{X2} & 1 \\ e^{X3} & 1 \\ \bullet & 1 \\ e^{Xn} & 1 \end{bmatrix}^{-1} \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ \bullet \\ Y_n \end{bmatrix}$$

والآن سنقوم بوضع البرنامج على الماتلاب

!Error



وبالتالي سيظهر الناتج كالتالي



هذا ونكون قد أنهينا دورة الماتلاب بفضل الله

الخاتمة

الحمد لله الذي أنعم علي ووفقني 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

أحـمـد عـفـيـفـي

دورة مكثفة في استخدام

MATLAB

م. عمر علي أبو علة

كلية الهندسة

جامعة 7 أكتوبر- مصراتة

This document was created using SOLID CONVERTER PDF

To remove this message, purchase the product at www.SolidPDF.com

Course Contents محتويات الدورة

The Course Goal..... هدف الدورة

MATLAB Windows..... النوافذ

MATLAB Links..... المسارات والارتباطات

Numbers and Results Representation..... تمثيل الأعداد وعرض النتائج

Special Variables..... متغيرات خاصة

Examples of Expressions..... بعض الأمثلة على تمثيل الأعداد

Operation Symbols..... رموز العمليات

Vectors, Matrices, Arrays, and Cells (Manipulations and Operations).....

..... المتجهات والمصفوفات والترتيبات والخلايا

Linear Algebra..... الجبر الخطي

MATLAB Programming and Structures..... التركيبات البرمجية

Polynomials..... عديدة الحدود

Numerical Analysis with MATLAB..... التحليل العددي

Probability Tools in MATLAB..... أدوات الإحصاء والاحتمالات

Adding New Functions ..... تعريف وإضافة الدوال

2-D Graphics..... الرسوميات ثنائية الأبعاد

3-D Graphics..... الرسوميات ثلاثية الأبعاد

Symbolic Expressions ..... التعبيرات والصيغ الرياضية الرمزية

2-D Easy Graphics..... الرسوميات ثنائية الأبعاد المسهلة

3-D Easy Graphics..... الرسوميات ثلاثية الأبعاد المسهلة

Create Function Calculator..... استدعاء حاسبة الدوال واستخدامها

Solving Simultaneous Equations ..... حل المعادلات الآنية

Solving Differential Equations..... حل المعادلات التفاضلية

Control Systems Analysis Using MATLAB..... تحليل نظم التحكم والسيطرة

MATLAB Help..... طلب المساعدة

SIMULINK..... فرع المحاكاة

GUI (Graphical User Interface)..... البرمجة باستخدام الواجهات الرسومية

Subroutines in MATLAB by functions..... استخدام الدوال في بناء البرامج الفرعية

This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

الغرض من الدورة:

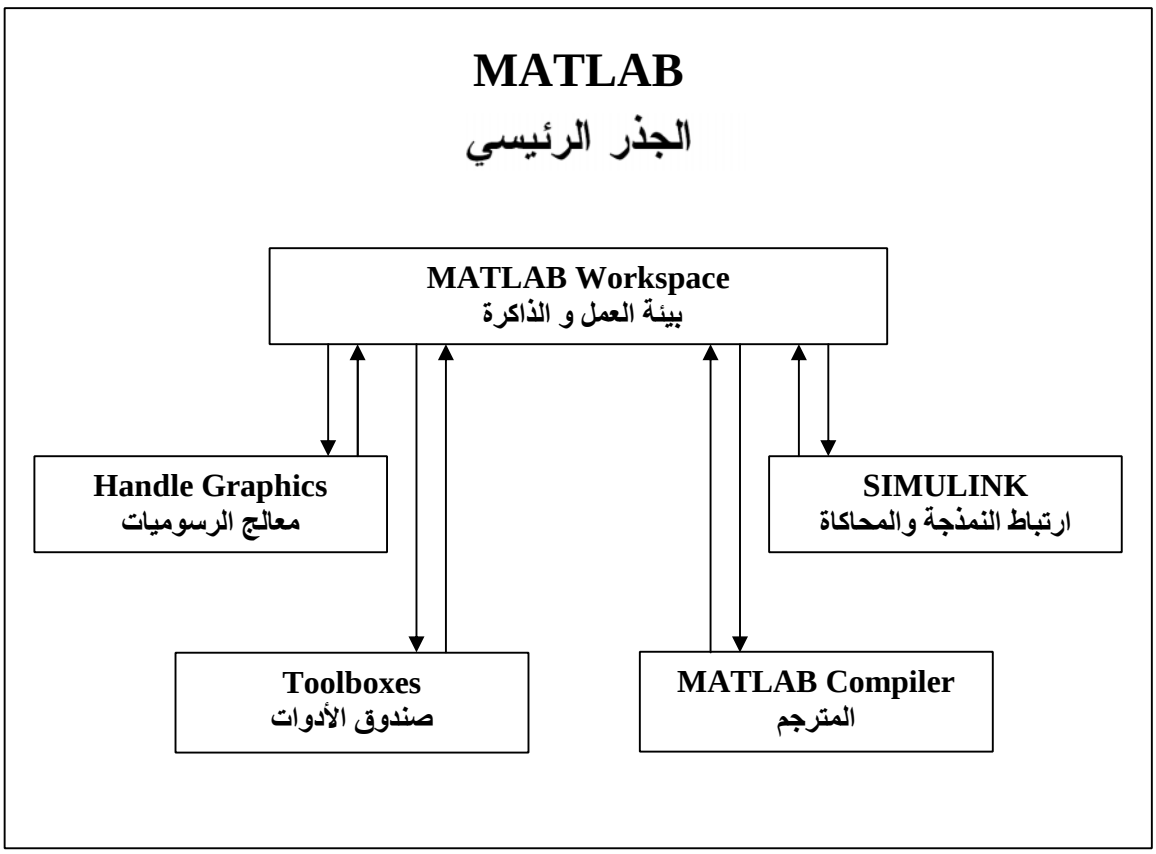
التعرف على لغة البرمجة الهندسية **MATLAB** وكيفية استخدامها في حل المسائل والمعادلات الخطية وغير الخطية والتفاضلية وذلك بتعريف المتدرب على إمكانيات **MATLAB** في التعامل مع المصفوفات والدوال الرياضية المختلفة، وكذلك التعامل مع الأعداد المركبة، والتعبيرات الرياضية المختلفة، وفهم بعض إمكانياتها في الرسوم الثنائية والثلاثية الأبعاد الثابتة والمتحركة، كذلك سنتطرق إلى استخدام هذه اللغة في التحليل العددي، وتحليل بعض التوزيعات الاحتمالية، كما تشمل الدورة تطبيقات **MATLAB** لتحليل أنظمة التحكم والسيطرة المتصلة والمتقطعة بمختلف التحويلات، كذلك تستهدف الدورة عرضا لإمكانيات **MATLAB** في برامج المحاكاة، والبرمجة باستخدام الواجهات الرسومية، وتختم الدورة بعرض يوضح للمتدرب إمكانيات هذه اللغة، وتدريبه على استخدام **MATLAB Help** ليتمكن من تطوير آدائه ويطلع بعمق على خفايا هذه اللغة.

المستهدفون بالدورة:

طلاب كلية الهندسة والملتحقين ببرامج الدراسات العليا، والجامعيين في الأقسام العلمية، وأعضاء هيئة التدريس، والمهتمين بالتحليلات الرياضية المختلفة.

This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

الارتباطات والتفرعات المهمة



MATLAB Links

This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

النوافذ المستخدمة غالباً ..... **MATLAB Windows**

نافذة الأوامر ..... ***Command Window***

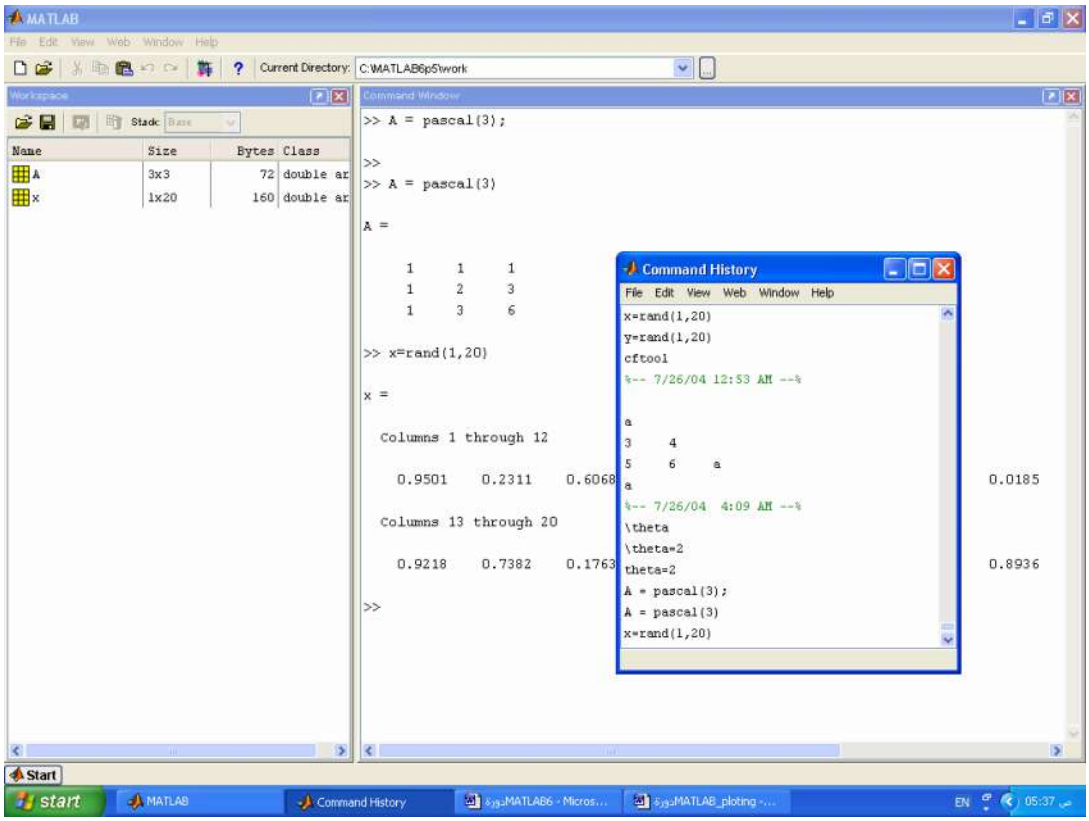
وهي الشاشة التنفيذية في**MATLAB** حيث يتم اظهار النتائج فيها كما يمكن أيضا كتابة الأوامر بهده الشاشة على هيئة سطر.

الأوامر السابقة..... ***Command History*** .....

وهو الحيز الذي يخزن فيه **MATLAB** الأوامر السابقة.

حيز العمل والذاكرة..... ***Work Space*** .....

هو جزء من داكرة الجهاز يستغله **MATLAB** في تخزين الممتغيرات المختلفة.

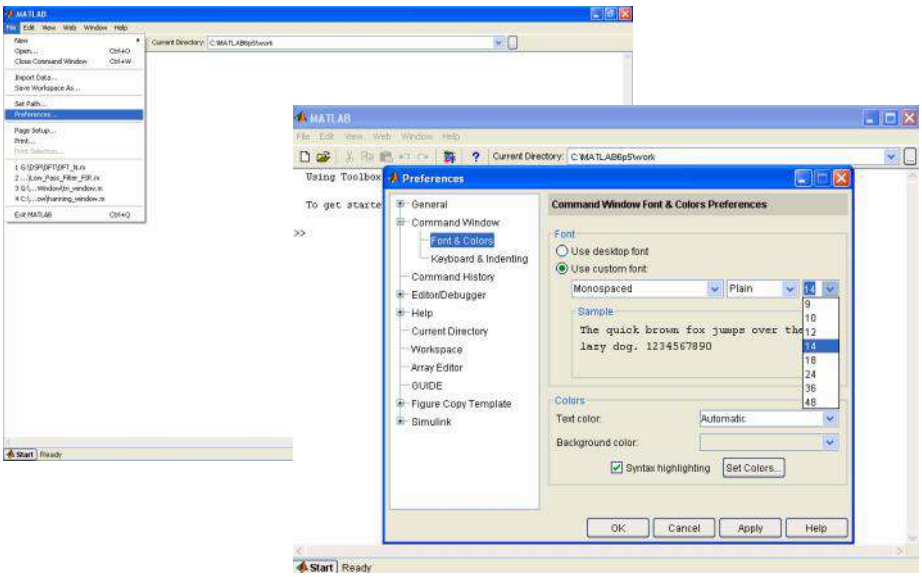


This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

Changing Font Size: تغيير حجم الكتابة:

File> Preferences> Command Window> Font & Colors> use custom font :>

Ex.: Monospaced, Plain, 14



This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

Numbers and Results Representation ..... تمثيل الأعداد وعرض النتائج:

```
s = 1 + 2
s =
3
fun = sin(pi/4)
fun =
0.7071
format long
fun
fun =
0.70710678118655
format short
fun
fun =
0.7071
realmin
ans =
2.2251e-308
i
ans =
0 + 1.0000i
```

Special Variables ..... متغيرات خاصة

الثابت	قيمه
pi	3.14159265...
i	$\sqrt{-1}$
j	مثل العدد التخيلي (i)
eps	رقم صغير جداً 2^{-52}
realmin	أصغر رقم يحسبه البرنامج 2^{-1022}
realmax	أكبر رقم يحسبه البرنامج $(2-\varepsilon)^{21023}$
inf	ما لانهاية ∞
NaN	غير معرف = $\infty - \infty$ or 0.0/0.0

Examples of Expressions ..... بعض الأمثلة لتمثيل الأعداد

3	99	0.0001
9.6397238	1.60210e20	6.02252e23
1i	3.14159j	3e5i

```
Theta = (1+sqrt(5))/2
Theta =
   1.6180
a = abs(3+4i)
a =
   5
```

This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

```
z = sqrt(besselk(4/3,Theta-i))
z =
  0.3730+ 0.3214i
huge = exp(log(realmax))
huge =
 1.7977e+308

toobig = pi*huge
toobig =
   Inf
```

Operation Symbols ..... **رموز العمليات**

Addition + ... **الجمع**

Subtraction – ..... **الطرح**

Multiplication * ..... **الضرب**

***Division / or *** ..... **القسمة**

Exponentiation ^ ≡ to the power of ..... **الأس**

أمثلة على بعض العمليات:

```
X = 47/3
X  =
15.6667

Y = 47\3
Y  =
0.0638

x = sin(1) - sin(2) + sin(3) - sin(4) + sin(5) -...
sin(6) + sin(7) - sin(8) + sin(9) - sin(10)
x  =
0.7744
```

Vectors, Matrices, Arrays, and Cells ..... **المتجهات والمصفوفات والترتيبات والخلايا**

Vectors ..... **المتجهات**

```
x=[0,0.1*pi,0.3*pi,0.5*pi,pi]
x =
      0    0.3142    0.9425    1.5708    3.1416

y=cos(x)
y =
   1.0000    0.9511    0.5878    0.0000   -1.0000

x(2)
ans =
   0.3142

y(2)
ans =
   0.9511
```

This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

Using step.....تعيين قيم المصفوفة بنقطة بداية ونقطة نهاية والانتقال بخطوة ثابتة

```
x(1:2:5)
ans =
       0    0.9425    3.1416

x(5:-1:2)
ans =
   3.1416    1.5708    0.9425    0.3142

z=y([3 1 4 5])
z =
   0.5878    1.0000    0.0000   -1.0000
```

What is the difference between [ ] and ( )? .....؟ [ ] و( ) ما الفرق بين

```
z(13)
??? Index exceeds matrix dimensions.

x=(1:2:10)
x =
    1    3    5    7    9
```

Using the step size: .....توليد قيم بين عددين بخطوة معينة

```
x= (1:0.1:2)
x =
  Columns 1 through 9
    1.0000    1.1000    1.2000    1.3000    1.4000    1.5000    1.6000    1.7000    1.8000
  Columns 10 through 11
    1.9000    2.0000


x=(0:0.4:2*pi)
x =
  Columns 1 through 9
         0    0.4000    0.8000    1.2000    1.6000    2.0000    2.4000    2.8000    3.2000
  Columns 10 through 16
    3.6000    4.0000    4.4000    4.8000    5.2000    5.6000    6.0000


y=cos(x)
y =
  Columns 1 through 9
    1.0000    0.9211    0.6967    0.3624   -0.0292   -0.4161   -0.7374   -0.9422   -0.9983
  Columns 10 through 16
   -0.8968   -0.6536   -0.3073    0.0875    0.4685    0.7756    0.9602


x=(0:(2*pi- 0)/9:2*pi)
x =
  Columns 1 through 9
         0    0.6981    1.3963    2.0944    2.7925    3.4907    4.1888    4.8869    5.5851
  Column 10
    6.2832
```

This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

تقسيم المسافات خطيا أو لوغارتميا .....

```
x=linspace(0,2*pi,10)
x =
  Columns 1 through 9
        0    0.6981    1.3963    2.0944    2.7925    3.4907    4.1888    4.8869    5.5851
  Column 10
    6.2832


x=logspace(0,2,7)

x =
    1.0000    2.1544    4.6416   10.0000   21.5443   46.4159  100.0000
```

دمج المتجهات أو تكوين المتجه من مجموعة متجهات أو عناصر .....

```
part1=[1,2,3,4,5]
part1 =
    1    2    3    4    5

part2=[5,4,3,2,1]
part2 =
    5    4    3    2    1

x=[part1,6,part2]
x =
    1    2    3    4    5    6    5    4    3    2    1

part3=(1:5)
part3 =
    1    2    3    4    5

x=[part3,6,part3(5:-1:1)]
x =
    1    2    3    4    5    6    5    4    3    2    1


p=(1:2:10)
p =
    1    3    5    7    9


q=3*p
q =
    3    9   15   21   27


q=p+1
q =
    2    4    6    8   10


q=4*p-10/2
q =
   -1    7   15   23   31
```

This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

المصفوفة Matrix .....

بناء مصفوفة .....

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


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

a =
    1    2    3
    4    5    6
    7    8    1


a(1:5)'
ans =

    1
    4
    7
    2
    5
```

ضرب المصفوفة في ثابت .....

```
a*4
ans =
    4    8   12
   16   20   24
   28   32    4
```

جمع وطرح مصفوفتين .....

```
a=[1,2;3,4];b=[3,5;2,1];
a+b
ans =
    4    7
    5    5

a-b
ans =
   -2   -3
    1    3
```

ضرب مصفوفتين .....

```
a*b

ans =
    7    7
   17   19
```

This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

قسمة مصفوفتين؟!!!.....

```
a/b
ans =
   0.4286   -0.1429
   0.7143    0.4286

a*inv(b)
ans =
   0.4286   -0.1429
   0.7143    0.4286

a\b
ans =
  -4.0000   -9.0000
   3.5000    7.0000
```

معكوس المصفوفة.....

```
inv(a)*b
ans =
  -4.0000   -9.0000
   3.5000    7.0000
```

محورة المصفوفة.....

```
z=[1+i,2-2i;1+sqrt(3)*i,sqrt(7)+4i]
z =
  1.0000 + 1.0000i   2.0000 - 2.0000i
  1.0000 + 1.7321i   2.6458 + 4.0000i

z'
ans =
  1.0000 - 1.0000i   1.0000 - 1.7321i.
  2.0000 + 2.0000i   2.6458 - 4.0000i
```

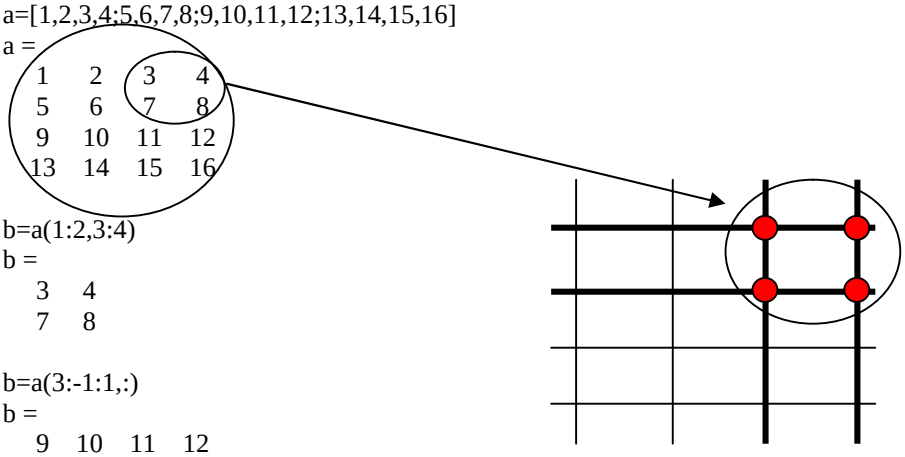
محورة غير محافظة على إشارة العدد المركب.....

```
z.'
ans =
  1.0000 + 1.0000i   1.0000 + 1.7321i
  2.0000 - 2.0000i   2.6458 + 4.0000i
```

محورة محافظة على إشارة العدد المركب.....

استعمال مؤشر المصفوفة للتعامل مع إحداثيات من عناصر المصفوفة.....

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



```
b=a(1:2,3:4)
b =
   3    4
   7    8

b=a(3:-1:1,:)
b =
   9   10   11   12
   5    6    7    8
   1    2    3    4
```

This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

```
b=a(:,[2,1,2,1])
b =
    2     1     2     1
    6     5     6     5
   10     9    10     9
   14    13    14    13


a(2,3)=19
a =
    1    2     3     4
    5    6    19     8
    9   10    11    12
   13   14    15    16


a(2,:)=[ ]
a =
    1     2     3     4
    9    10    11    12
   13    14    15    16
```

التعامل مع المصفوفة وتغيير شكلها.....

```
a(:)'
ans =
    1    9   13    2   10   14    3   11   15    4   12   16


b=reshape(a,2,6)
b =
    1   13   10    3   15   12
    9    2   14   11    4   16

rot90(a)
ans =
    4   12   16
    3   11   15
    2   10   14
    1    9   13


a = [1 2 3]
a =
1 2 3

b = [1;2;3]
b =
1
2
3
(a+i*b')'
ans =
1.0000 - 1.0000i
2.0000 - 2.0000i
3.0000 - 3.0000i
```

This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

```
 (a+i*b').'
ans =
1.0000 + 1.0000i
2.0000 + 2.0000i
3.0000 + 3.0000i
```

تعيين حجم المصفوفة.....

```
size(a)
ans =
    3     4


size(b)
ans =
    2     6


length(b)
ans =
    6
```

أنواع من ضرب ا لمصفوفات.....

dotprod = a*b

```
dotprod =
14
```

outprod = b*a

```
outprod =
1 2 3
2 4 6
3 6 9


b = [-2 1 2];


cp = cross(a,b)

cp =
1 -8 5
```

```
R = randn(4,4)
R =
   1.0668   0.2944  -0.6918  -1.4410
   0.0593  -1.3362   0.8580   0.5711
  -0.0956   0.7143   1.2540  -0.3999
  -0.8323   1.6236  -1.5937   0.6900
```

This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

```
p =
   [0.4706   0.0882   0.0588   0.3824
   0.1471   0.2941   0.3235   0.2353
   0.2647   0.1765   0.2059   0.3529
   0.1176   0.4412   0.4118   0.0294];

p^2 =

   0.2950   0.2465   0.2258   0.2327
   0.2258   0.2604   0.2673   0.2465
   0.2465   0.2673   0.2604   0.2258
   0.2327   0.2258   0.2465   0.2950
```

Arrays*: Just by using dot (.)Before operators (*, /, ^)*

الترتيبات: لجعل العملية عملية ترتيبية أي معاملة كل عنصر بالعنصر المناظر له فقط، يجب وضع نقطة قبل رمز العملية المراد إجراؤها.

```
a=[1,2;3,4];b=[3,5;2,1];


a.*b
ans =
    3   10
    6    4


a./b
ans =
   0.3333   0.4000
   1.5000   4.0000


a.^b
ans =
    1   32
    9    4


A = [1 2 3; 3 2 1] ;

A.*A

ans =
1 4 9
9 4 1
A*A
```

"??? Error using ==> *.....تنبيه لوجود خطأ
Inner matrix dimensions must agree.

This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

Building Tables ..... **بناء الجداول**

```
n = (0:9)';
pows = [n  n.^2  2.^n]
pows =
    0    0    1
    1    1    2
    2    4    4
    3    9    8
    4   16   16
    5   25   32
    6   36   64
    7   49  128
    8   64  256
    9   81  512
```

Cells: ***using { } instead of [ ]***

الخلايا يمكن بناء الخلايا بوضع العناصر بين القوسين الكبيرين بدلا من قوسي المصفوفة

```
c = {1,-2,3}
c =
[1] [-2] [3]
```

```
{'How are you ','كيف الحال'}
```

Some of Matrices Operations ..... **بعض عمليات المصفوفات**

إيجاد بعض الإحصاءات على المصفوفة.....

```
a=[1 2 -3;-9 5 1;7 6 0]
```

sort(a)..... **ترتيب العناصر تصاعديا بالنسبة للأعمدة**

sortrows(a)..... **ترتب صفوف المصفوفة بحيث يكون أول صف المبتدئ بأصغر عنصر**

sum(a)..... **مجموع عناصر كل عمود**

diff(a) ..... **الفرق بين كل صف والدي قبله**

min(a) ..... **أكبر عنصر في كل عمود**

max(a)..... **أصغر عنصر في كل عمود**

median(a)..... **الوسيط**

mean(a) ..... **المتوسط الحسابي**

std(a)..... **الانحراف المعياري**

diag(a)..... **إظهار عناصر القطر الرئيس**

This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

عمليات التراتيب والمصفوفات

عمليات المصفوفات		عمليات التراتيب	
x	1 2 3	y	4 5 6
x'	1 2 3	y'	4 5 6
x+y	5 7 9	x–y	–3 –3 –3
x + 2	3 4 5	x–2	–1 0 1
x * y	Error	x.*y	4 10 18
x'*y	32	x'.*y	Error
x*y'	4 5 6 8 10 12 12 15 18	x.*y'	Error
x*2	2 4 6	x.*2	2 4 6
x\y	16/7	x.\y	4 5/2 2
2\x	1/2 1 3/2	2./x	2 1 2/3
x/y	0 0 1/6 0 0 1/3 0 0 1/2	x./y	1/4 2/5 1/2
x/2	1/2 1 3/2	x./2	1/2 1 3/2
x^y	Error	x.^y	1 32 729
x^2	Error	x.^2	1 4 9
2^x	Error	2.^x	2 4 8

This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

استخدام ثابت يدخله المستخدم في إجراء بعض العمليات.....
```
A=[1 2 3;5 2 3;6 8 9]
k=input('k=') .....هذا الأمر يجعل البرنامج يسمح للمستخدم بإدخال قيمة للمتغير
B=k*A
```

```
a=[1 2 3;4 5 1+6*i;7 8 9];
```

مجموع عناصر القطر الرئيسي.....
```
trace(a) .....
```

المصفوفتين المثلثيتين العليا والسفلى.....
```
tril(a),triu(a) .....
```

Linear Algebra ..... الجبر الخطي

تحليل منظومة من المعادلات الخطية باستخدام المصفوفات

```
% Solve the following equations:
% x+2*y+3*z=14;
% 4*x+5*y+8*z=32;
% 7*x+8*y+z=26;
```

مصفوفة المعاملات.....
```
A=[1,2,3;4,5,6;7,8,1] .....
```

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

مصفوفة الثوابت لمعرفة ما إذا كانت المصفوفة شاذة أو غير ذلك.....
```
b=[14;32;26] .....
```

```
b =
   14
   32
   26
```

```
det(A)    محدد المصفوفة
ans =
   24
x=inv(A)*b
x =       مصفوفة الحلول
   1.0000
   2.0000
   3.0000
```

$$b / A \equiv b * inv(A)$$

$$A \backslash b \equiv inv(A) * b$$

```
x=A\b
x =          1.0000
             2.0000
             3.0000
```

This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

يجب الانتباه أثناء قسمة مصفوفة على أخرى حيث يجب مراعاة أبعاد المصفوفتين أعمدة الأولى تساوي صفوف الثانية.

```
% x = B/A
```
تعني أن B مقسومة على A (قسمة يسارية).
```
% (wrong division: improper dimensions)

%inv (A)*B; ..... لإيجاد قيم المجاهيل
%SOL=A\B; % means B divided by A (RIGHT division).

% or, inv (A)*B..... تعني أن B مقسومة على A (قسمة يمينية)

A=gallery(3)..... مصفوفة خاصة
% An especial matrix (built in MATLAB)

A =
  -149   -50  -154
   537   180   546
   -27    -9   -25

det(A)
ans =
     6
```

Badly Condition Problems (ill condition) ..... المصفوفة المريضة
```
b1=[-711;2535;-120],x=A\b1

b1 =
      -711
      2535
      -120

x =
    1.0000
    2.0000
    3.0000
```

b2=b1+ [1; 1; 0.1], x=A\b2 ..... إحداث تغيير طفيف على المصفوفة

```
b2 =
  1.0e+003 *
   -0.7100
    2.5360
   -0.1199

x =        99.6667
         -312.0667
           9.5000
```

This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

cond(A)..... مقياس اعتلال المصفوفة
ans =
2.7585e+005

Finding Eigenvalues and Eigenvector .....القيم الذاتية والمتجهات الذاتية

[V, D] = eig(A)

V =..... المتجهات الذاتية
0.5438 -0.8165 0.1938
-0.7812 -0.4082 0.4722
0.3065 0.4082 0.8599

D =..... القيم الذاتية
0.1270 0 0
0 1.0000 0
0 0 7.8730

Special Matrices :.....أمثلة على بعض المصفوفات الخاصة

A = pascal(3)
A =
1 1 1
1 2 3
1 3 6

B = inv(A)
B =
3 -3 1
-3 5 -2
1 -2 1

A = magic(3);
det(A)
ans =
-360

M=ones(2,3)
M =
1 1 1
1 1 1

Z=zeros(2,1)
Z =
0
0

This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

MATLAB Programming and Structures .....التركيبات البرمجية

for Loop % ***doesn't preferable !!!*** .....انتبه إلى أنها بطيئة في التنفيذ

Example1:

```
a=zeros (1, 10); % open a matrix and store filled by 10 zeros.
for k=1:10
   a(1,k)=k^2;
end
disp('Programme_Results:  ',num2str(a))=
    1     4     9    16    25    36    49    64    81   100
```

.....الطريقة البديلة عن الحلقة

```
 a=(1:10).^2
a =
    1     4     9    16    25    36    49    64    81   100
```

Example2:

```
a=zeros(3);
for k=1:3
   for L=1:3
      a(L,k)=1/(k+L-1);
   end
end
a =
   1.0000    0.5000    0.3333
   0.5000    0.3333    0.2500      Hilbert Matrix of Order 3 .hilb(3)
   0.3333    0.2500    0.2000
```

while Loop

```
while (condition)
statements
end
```

Example:

أكتب برنامجاً يحسب عدد السكان لبلد ما، ويضع الناتج مجدولاً حسب السنوات وذلك إلى أن يتضاعف عدد سكان هذه الدولة اعتماداً على معرفة معدل النمو السنوي لعدد السكان وكذلك التعداد الحالي للسكان.

```
format bank
pop_initial=input('Initial Population= ');
Gr=input('Growth Rate: ');
year=0;
pop_now=pop_initial;
disp('         Year    Population')
while (pop_now<2*pop_initial)
```

This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

```
    year=year+1;
    pop_now=pop_now+Gr*pop_now;
    disp([2005+year, pop_now])
end
Initial Population= 5000000
Growth Rate: 0.1
```

نتيجة المثال

Year	Population
2006.00	5500000.00
2007.00	6050000.00
2008.00	6655000.00
2009.00	7320500.00
2010.00	8052550.00
2011.00	8857805.00
2012.00	9743585.50
2013.00	10717944.05

The if-elseif-else statements

```
if (condition 1)
statements A
elseif (condition 2)
statements B
elseif (condition 3)
statements C
…
else
statement E
end
```

Example1:

أكتب برنامجاً يحسب صافي الدخل إذا علمت أن ضريبة الدخل تكون كالتالي:
9% على الدخول الأقل من 5000 دينار، 12% على الدخول الأقل من 10000 دينار، 15% على الدخول الأخرى.

```
Income=input('Enter the total income: ');
if(Income<5000)
   Tax=0.09;
elseif(Income<10000)
   Tax=0.12;
else
   Tax=0.15;
end
disp(['Net income: ',num2str(Income*(1-Tax))])
Enter the total income: 6465
Net income: 5689.2
```

Example2:

```
a=input('a=');
b=input('b=');
c=input('c=');
if((b^2-4*a*c==0)&(a~=0))
   x=-b/(2*a)
end
```

This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

Polynomials ..... **عديدة الحدود**

Polynomial storage ..... **تخزين عديدة الحدود**

$z^5 - 2z^4 + 3z^3 + 4z - 12$

z=[1,-2,3,0,4,-12]

Roots of Polynomial ..... **جذور عديدة الحدود**

$z^5 - 2z^4 + 3z^3 + 4z - 12 = 0$

```
z =
    1    -2     3     0     4   -12

soln=roots(z)
soln =

   1.1667 + 1.7265i
   1.1667 - 1.7265i
  -0.8790 + 1.0805i
  -0.8790 - 1.0805i
   1.4245
```

Finding the Polynomial from Its Roots ..... **إيجاد عديدة الحدود من الجذور**

```
p=poly(soln)
p=
1    -2     3     0     4   -12
```

$z^5 - 2z^4 + 3z^3 + 4z - 12$

Adding Polynomials ..... **جمع متعددات الحدود**

To add $z^3 - 14z^2 + 26z - 24$ to $10z^2 + 3z - 2$

```
z1=[1,-14,26,-24],z2=[0,10,3,-2]
z1 =
    1   -14    26   -24
z2 =
    0    10     3    -2
>> z1+z2

ans =
    1    -4    29   -26
```

Multiplying Polynomials ..... **ضرب متعددات الحدود**

```
z3=conv(z1,z2)
z3 =
    0    10  -137   216  -134  -124    48
```

$10z^5 - 137z^4 + 216z^3 - 134z^2 - 124z + 48$

This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

Dividing Polynomials ..... **قسمة متعددات الحدود**

$$\frac{10z^5-137z^4+216z^3-134z^2-124z+48}{10z^2+3z-2}$$

```
z3=[10,-137,216,-134,48],z2=[10,3,-2]
z3 =
   10  -137   216  -134    48

z2 =
   10     3    -2

[q,r]=deconv(z3,z2)
q =
   1.0000  -14.0000   26.0000

r =
        0    0.0000         0      -240.0000       100.0000
```

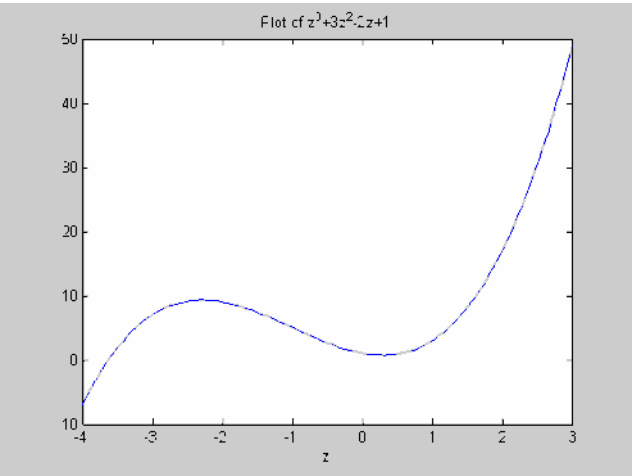
Polynomial Derivatives ..... **اشتقاق عديدة الحدود**

$$\frac{d}{dz}\left(z^4+2z^3+12z^2+11z+1\right)=4z^3+6z^2+24z+11$$

```
z1=[1, 2, 12, 11, 1]
z1 =
    1     2    12    11     1

polyder(z1)
ans =
    4     6    24    11
```

Polynomial Evaluation..... **إيجاد قيم الدالة في النطاق المطلوب**

```
z=[1,3,-2,1];
z0=linspace(-4,3,100);
value=polyval(z,z0);
plot(z0,value);
title('Plot of z^3+3z^2-2z+1')
xlabel('z')
```



This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

Partial Fractions الكسور الجزئية

$$\frac{z^2+1}{z^2-z}=\frac{2}{z-1}-\frac{1}{z}+1$$

```
z1=[1,0,1];
z2=[1,-1,0];

[r,p,k]=residue(z1,z2)
r =
     2
    -1

p =
     1
     0

k =
     1
```

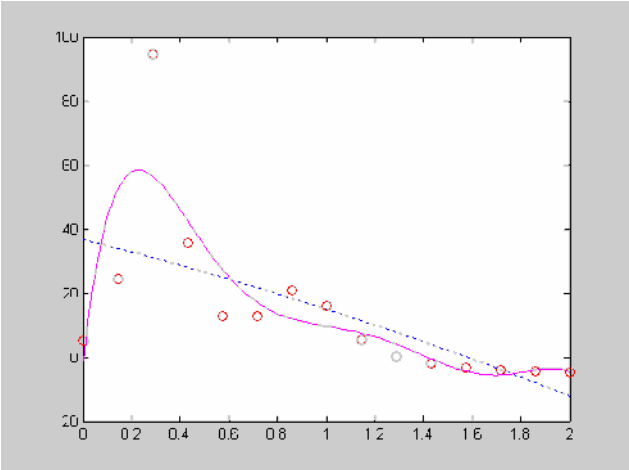
Create Polynomial Tools in MATLAB .....**أداة عديدة الحدود الجاهزة**
Polytool

This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

Numerical Analysis with MATLAB .....التحليل العددي

Curve Fitting: Least Squares ..... مطابقة المنحنيات باستخدام (المربعات الصغرى)

```
x=linspace(0,2,15);
y=humps(x);
order_2=polyfit(x,y,2);
x_2=linspace(0,2,100);
y_2=polyval(order_2,x_2);
order_6=polyfit(x,y,6);
x_6=x_2;
y_6=polyval(order_6,x_6);
plot(x,y,'ro',x_2,y_2,'b:',x_6,y_6,'m');
```



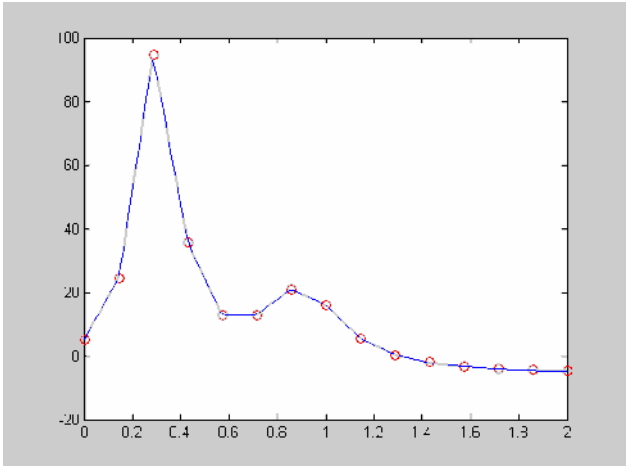
Curve Fitting Tools
الأداة الجاهزة لمطابقة المنحنيات

```
cftool
```

Interpolation .....الاستكمال

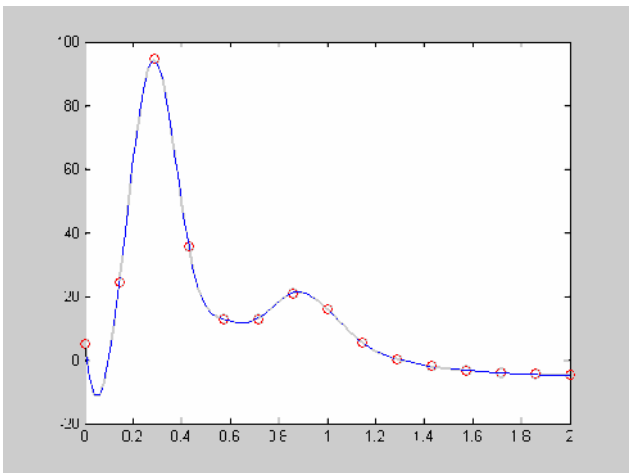
Linear Interpolation .....الاستكمال الخطي

```
Linear_x=linspace(0,2,100);
Linear_fit=interp1(x,y,Linear_x,'linear');
plot(x,y,'or',Linear_x,Linear_fit,'b')
```



Spline Interpolation (Cubic Spline)
الاستكمال (الطيّع)

```
Spline_x=linspace(0,2,100);
Spline_fit=interp1(x,y,Spline_x,'spline');
plot(x,y,'or',Spline_x,Spline_fit,'b')
```



Numerical Integration .....التكامل العددي

Trapezoidal .....أشباه المنحرفات

```
x = 0:pi/100:pi;
y = sin(x);
z = trapz(x,y)
z =
   1.9998
```

Quadrature .....المتعامد

```
F = inline('1./(x.^3-2*x-5)');
Q = quad(F,0,2)
Q =
  -0.4605
```

This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

Probability Tools in MATLAB .....الاحتمالات
Disttool .....الأداة الجاهزة للتوزيعات الاحتمالية
randtool .....الأداة الجاهزة للتوزيعات العشوائية

Adding New Functions .....إضافة وتعريف دوال جديدة

The general form of a function file .....الصيغة العامة لملف تعريف الدالة

Function [output variables] =function_name(input variables)

[My output variables]= function_name(input variables)

```
function f=bell(x,a)
f=exp(-.5*a*(x^2))/sqrt(2*pi);

bell(1,2)
ans =
   0.1468
```

ولكي نجعل هده الدالة تتعامل مع المصفوفات نجري التتعديل البسيط الآتي:

```
function f=mbell(x,a)
f=exp(-.5*a*(x.^2))/sqrt(2*pi);
% To deal with an array instead of a matrix (change x^2 to x.^2)
mbell([1 3 4 5,8]*1e-2,12)
ans =
   0.3987    0.3968    0.3951    0.3930    0.3839
```

Saving and Retrieving Data.....حفظ البيانات في ملف واسترجاعها

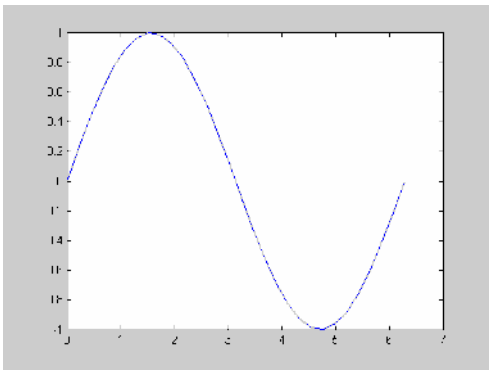
```
x=[11 12  13 5  9  8  7];
save('abc.mat','x');
load('abc')
```

This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

2-D Graphics .....الرسوميات ثنائية الأبعاد

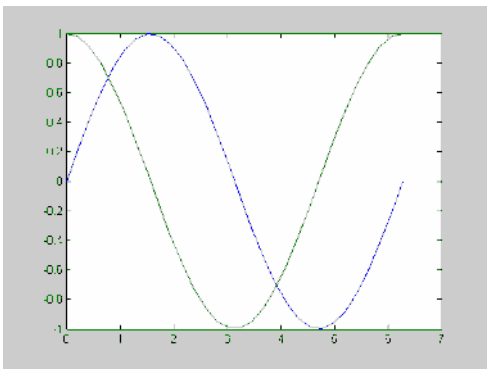
***Using plot command*.....استخدام أمر الرسم**

```
x=linspace(0,2*pi,30);
```
الأمر السابق لفتح مصفوفة بها 30 نقطة تبدأ بـ 0 وتنتهي بـ 6.28
```
y=sin(x);
plot(x,y);
```

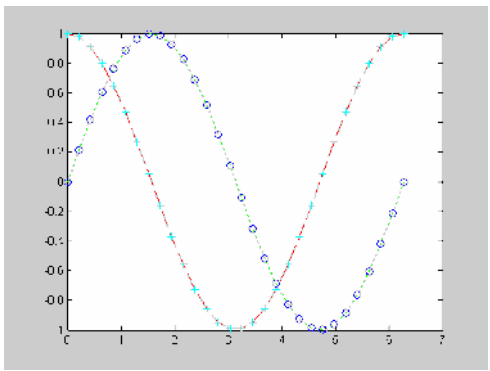


***Two Curves*.....رسم منحنيين معاً**
```
z=cos(x);
plot(x,y,x,z);
```
***Another method*.....طرق أخرى**

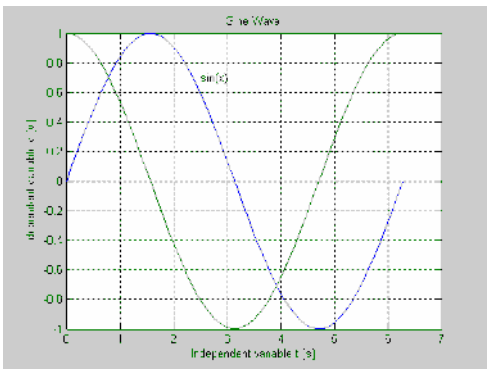
نكون مصفوفة من المصفوفة الجديدة من المصفوفتين السابقتين
```
w=[y;z];
plot(x,w);
plot(w,x);
```
ولتمييز المنحنيين أحدهما عن الآخر نستخدم الأمر التالي:
```
plot(x,y,'g:',x,z,'r--',x,y,'bo',x,z,'c+');
```



Adding grid, labels and text to the graph
إضافة شبكة وعناوين ونصوص للرسم
```
x=linspace(0,2*pi,60);
y=sin(x);
z=cos(x);
plot(x,y,x,z);
grid;
title('Sine Wave'); .....عنوان الرسم
xlabel('Independent variable t [s]'); عنوان المحور السيني
ylabel('dependent variable v [v]'); عنوان المحور الصادي
text(2.5,0.7,'sin(x)'); .....لإضافة نص على الرسم
```

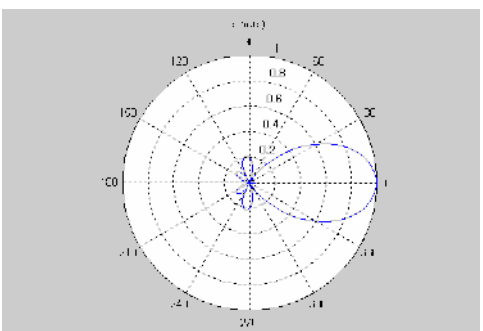


***Interactive commands*.....الأوامر التفاعلية**
```
gtext('cos(x)');....لوضع النص في المكان الذي نحدده بالفأرة
ginput;.....لإيجاد إحداثي نقطة ما على الرسم
```



***Axis properties*.....خصائص المحاور**
```
axis off;.....لإخفاء المحاور
grid off;.....لإخفاء الشبكة
axis ij;
axis('square');
axis('equal');
axis('xy','normal');
axis([pi 2*pi -1.5 1.5]);
```

***Polar plot*.....الرسم القطبي**
```
x=linspace(-pi,pi);y=sinc(x);
polar(x,y);
title('sinc(x)');
```



This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

Bar plot.....التخطيط العمودي
```
E=exp(-x.*x);
bar(x,E,'g');
title('bell graph');
```
Stem plot.....الرسم على هيئة أعمدة عينات
```
g=randn(1,20);
```
.....هذا الأمر يعطي قيماً عشوائية
```
stem(g);
```
Compass plot.....الرسم الاتجاهي
```
Z = eig(randn(20,20));
compass(Z)
```

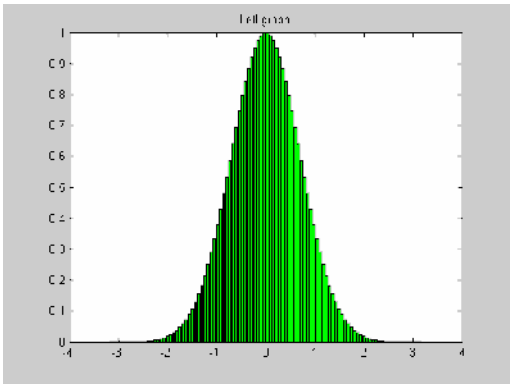
Using plotyy Command.....الرسم بمحورين عموديين
```
x=linspace(0,2*pi,50);
y1=sin(x);
y2=sinc(x);
plotyy(x,y1,x,y2,'plot','stem')
```

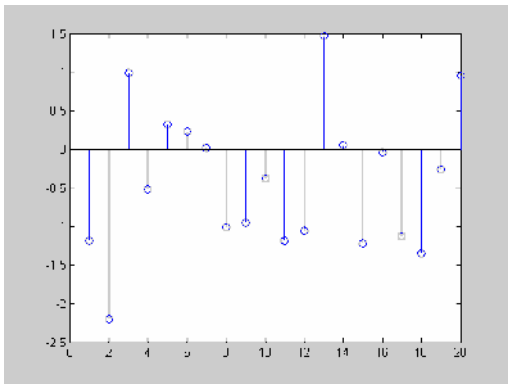
Area plot.....الرسم المساحي
```
Y = [   1, 5, 3 4;2 3 4 1; 3 2 5 2;2 1 4 3]
area(Y)
```

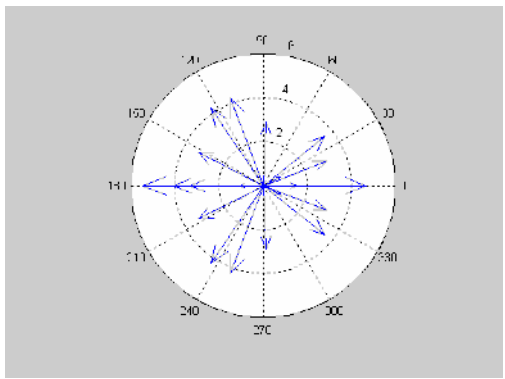
pie plot and creating two figures
التخطيط القطاعي واستدعاء شكلين للرسم
```
x=linspace(0,2*pi,20);
y1=sin(x).*exp(x);
y2=cos(x);
figure(1)
pie3(x,y1)
figure(2)
stem(x,y2)
```

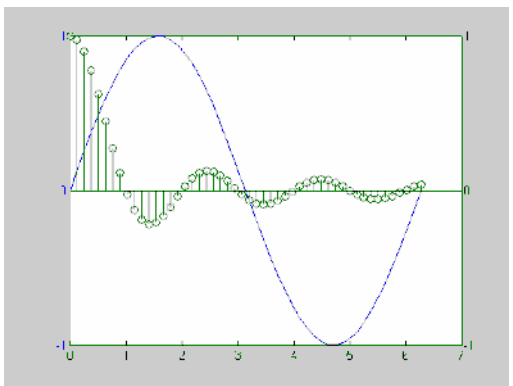
fplot.....أمر رسم الدوال
```
fplot('exp(-x)',[-2 2 0 10] );grid on
```

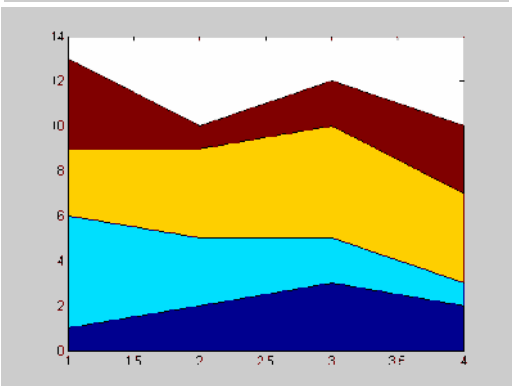
Subplots.....تجزيء الشكل إلى أشكال فرعية
```
x=linspace(-20,20);
y=sin(x);
z=cos(x);
a=2*sin(x).*cos(x);
b=sin(x)./(cos(x)+eps);
subplot(2,2,1);
plot(x,y);
subplot(2,2,2);
plot(x,z);
subplot(2,2,3);
plot(x,a);
subplot(2,2,4);
```





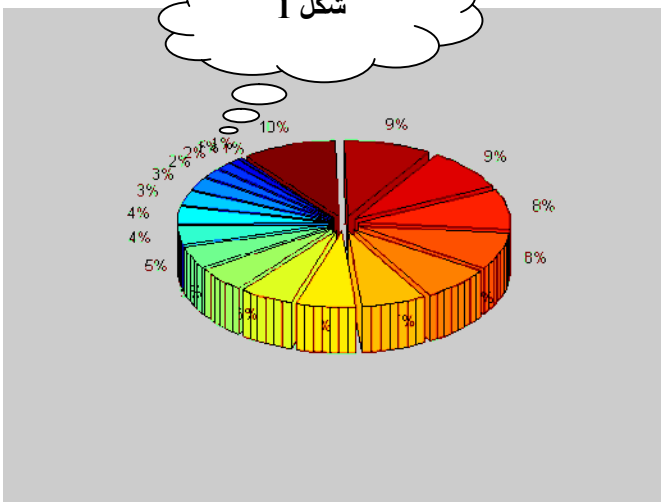




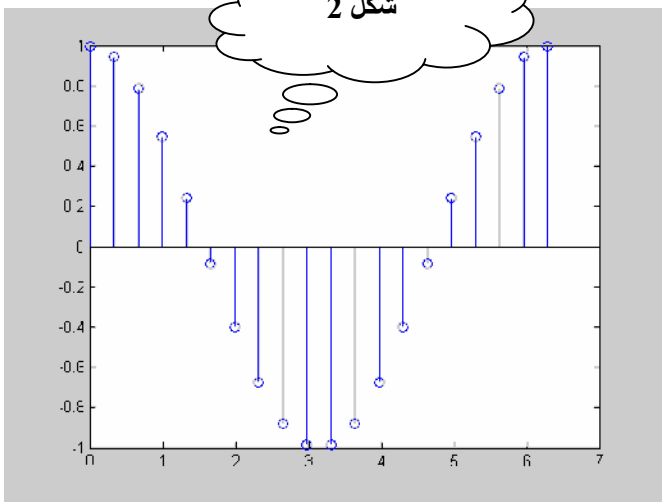


This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

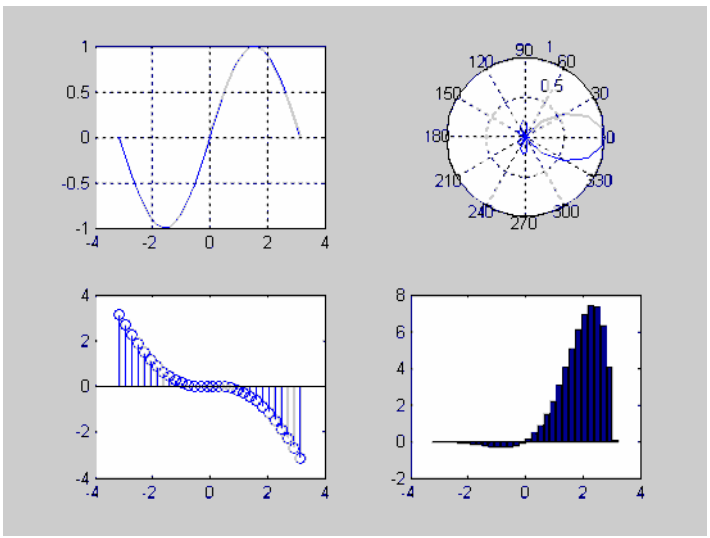
شكل 1



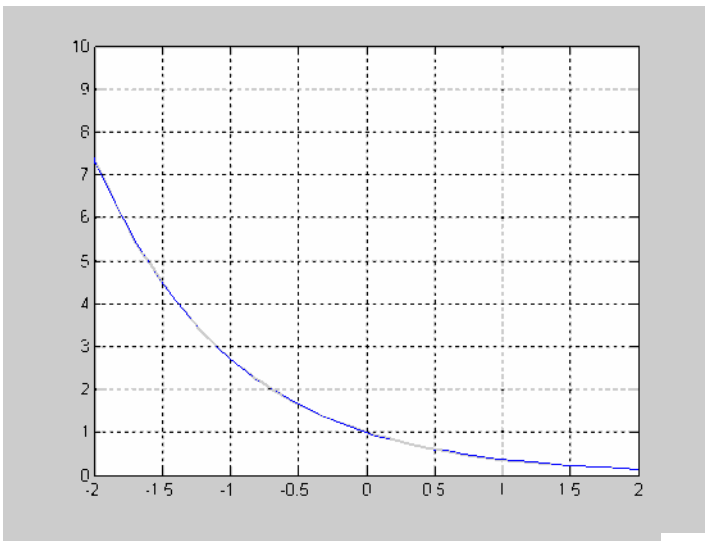
شكل 2

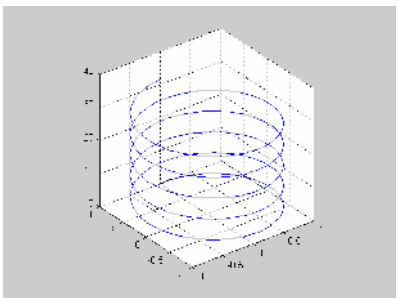


الأشكال المفرعة



رسم الدوال





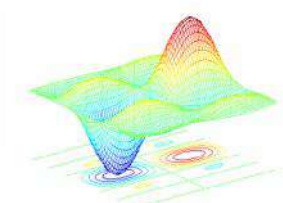
3-D Graphics.....الرسوميات ثلاثية الأبعاد

Using plot3 Command.....استخدام أمر الرسم الثلاثي

```
t = 0:pi/50:10*pi;
plot3(sin(t),cos(t),t)
grid on
axis square
```

Mesh plot and Contour plot.....الرسم الشبكي و الإسقاطي

```
 [x,y]=meshgrid(-pi:0.1:pi); .....حجز مصفوفة في بعدين
z=sin(x).*sinc(y);
mesh(z);
figure(2) ;
meshc(z) ;
```



This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

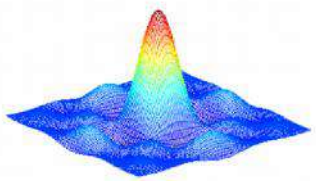
Surface plot.....رسم السطوح

```
x=-7.5 : 0.5 : 7.5;
y=x;
[X,Y]=meshgrid(x,y);
R=sqrt(X.^2+Y.^2)+eps;
Z=sin(R)./R;
mesh(X,Y,Z);
surf(X,Y,Z);
shading interp;
%colormap([0.6 0.6 0.6]);
colormap(copper);
```

Adding Title labels and text to the graph

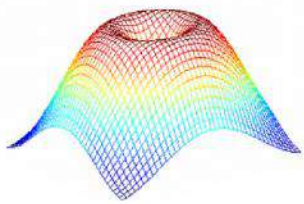
إضافة عنوان رئيسي للشكل وتمييز المحاور وإضافة نصوص للرسم

```
x=linspace(-pi,pi);
y=x;
[X,Y]=meshgrid(x,y);
%A=sinc(X);
%B=sinc(Y);
%mesh(X);
%mesh(Y);
%Z=cos(X).*exp(-Y);
Z=sinc(X).*sinc(Y);
mesh(X,Y,Z);
title('Sinc Function in 3-D')
xlabel('X axis');
ylabel('Y axis');
zlabel('Z axis');
```



Motion, view point and camera position.....الحركة وزاوية النظر وموقع (الكاميرا الافتراضية)

```
[x,y]=meshgrid(-1:0.05:1);
r=sqrt(x.^2+y.^2)+eps;
z=sin(pi*r)./(1+r);
%r=2*cos(x).*sin(y);
%z=sinc(r);
mesh(z)
```



<u>Coordinate System Conversion</u>

التحويل من إحداثي إلى آخر

```
[THETA,PHI,R] = cart2sph(X,Y,Z)

[THETA,RHO,Z] = cart2pol(X,Y,Z)
[THETA,RHO] = cart2pol(X,Y)

[X,Y] = pol2cart(THETA,RHO)
[X,Y,Z] = pol2cart(THETA,RHO,Z)
[X,Y,Z] = sph2cart(THETA,PHI,R)
```

This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

Control Systems Analysis Using MATLAB .....تحليل نظم التحكم والسيطرة
Building Transfer Function .....بناء دالة الانتقال

```
num=[1 1];
den=[1 2 3];
sys1=tf(num,den);
```

Other methods .....طرق أخرى

```
h=tf([1 0],[1 2 10])
s=tf('s')
H=s/(s^2+2*s+10)
```

Blocks diagram.....المخطط الكتلي
Unity feedback.....التغذية العكسية بدون كسب

```
G=tf([2 1],[1 0]);
sys1=feedback(G,1,1);
```
.....الرقم (1) الأخير يعني أن إشارة التغذية العكسية موجبة والعكس لو كانت(-1)

Series Connection .....الربط المتوالي

```
sys1=tf([2 1],[1 2])
sys2=tf([-3 1],[4 -5 1])
s1=sys1*sys2
```

Another equivalent method (Series) .....طريقة أخرى للربط المتوالي

```
sys=series(sys1,sys2)
sys3=tf([-2 1],[1 0 3])
```

Parallel Connection .....الربط المتوازي

```
T1=Parallel(sys1,sys2)
T2=Parallel(T1,sys3)
```

Example

```
disp('Title of Program')
s=tf('s');
sys1=(s^2+2)/(4*s^2-3)
G=s/(1+s^2)
H=G*sys1
FIL=H+G
```

Discrete transfer Function..... دالة التحويل المتقطعة

```
Nums={1 [1 0];[-1 2] 3}
Ts = 0.2
H= tf(Nums,[1 0.3],Ts)
size(H)
```

This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

State Variables (State Space) ..... **متغيرات الحالة (فضاء الحالة)**

```
A= [-2 -1;1 -2];
B=[1 1;2 -1];
C=[1 0];
D=[0 1];
sys2=ss(A,B,C,D)
sys3=tf(sys2);
size(sys2);
size(sys3);
g=tf([1,2],[-2,3,4]);
sg=ss(g);
size(g);
g=tf([1 1],[1 2 3],0.1);
h=tf([1 1],[1 2 3],0.1,'variable','z^-1');
```

Zero pole gain ..... **تعيين الصفر والقطب والكسب**

```
sys=zpk(2,-3,4);
```

Convert from Continuous to Discrete ..... **التحويل من النطاق المستمر إلى المتقطع**

```
H=tf([1 -1],[1 4 5],'inputdelay',0.75);
hd=c2d(H,0.05,'foh');
step(H,':r',hd,'--');
```

Convert from Discrete to Continuous ..... **لتحويل من النطاق المتقطع إلى المستمر**

```
Ts =0.1;
H=zpk(-0.2,-0.5,1,Ts)*tf(1,[1 1 0.4],Ts)
Hc=d2c(H)
HD=c2d(Hc, Ts)
step(H,'-r',Hc,'-',HD,'g:');
```

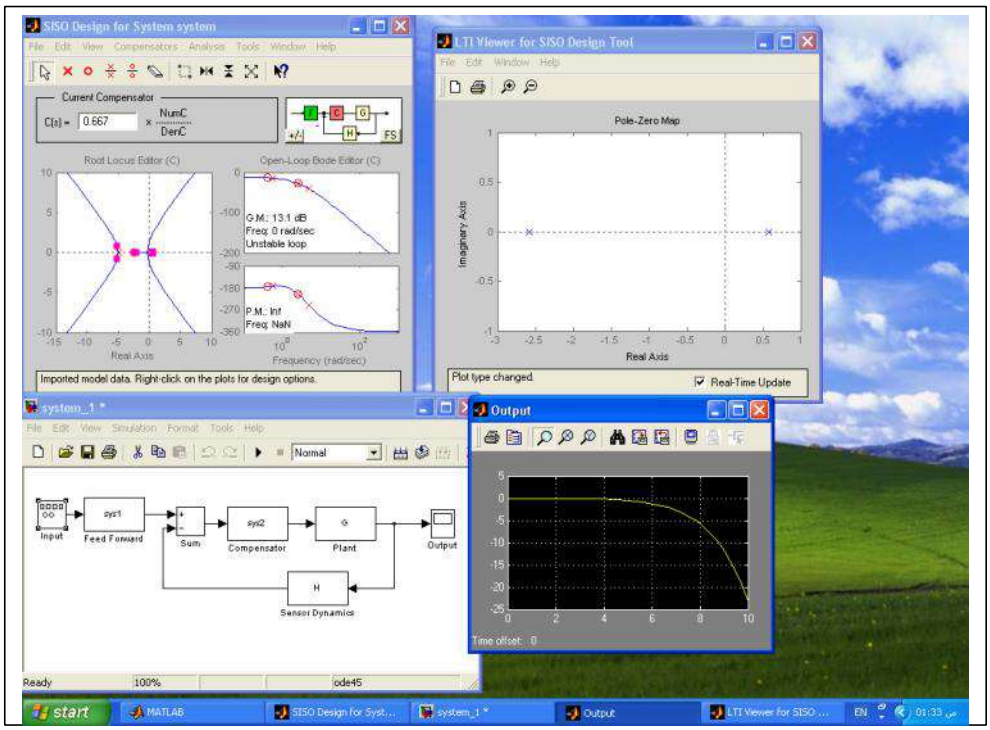
Example (step, impulse, root locus, and bode plot)

أمثلة على استجابة الخطوة والومضة ومحل الجذور والاستجابة الترددية

```
k=input('Enter the Gain k:')
G=tf([k],[1 2 2])
H=tf([5],[1 5])
sys=Feedback(G,H,-1)
subplot(2,2,1)
step(sys)
subplot(2,2,2)
impulse(sys);
subplot(2,2,3)
rlocus(sys)
subplot(2,2,4)
bode(sys)
```

This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

Control Systems Analysis Tool in MATLAB .....أداة تحليل نظم التحكم والسيطرة
sisotool



Fourier Tranform .....محولة فوريير
f=sym('t^2+t+3')
F=fourier(f)
SF=simple(F)
symvar('t^2+x^2+s')
s=sym('cos(t)')
F=fourier(s)
SS=simple(F)

Laplace Tranform .....محولة لابلاس
syms t
y=laplace(t^2+sin(t))
ilaplace(y)
ifourier(F)
f= sym('1');
laplace(f)
s=sym('Heaviside(t)')
laplace(s)

Z Tranform .....محولة(*Z*)
syms n, f = n^4, ztrans(f)
f = n^4

ans =
z*(z^3+11*z^2+11*z+1)/(z-1)^5

This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

Symbolic Expressions .....التعابير والصيغ الرمزية
Expression manipulation.....التعامل مع الصيغ

```
clc
clear
syms x y z a b r
(x+4)*(x-4)*(x-6)*x^2
```

.....التعامل مع الدوال وتبسيطها

```
expand((x+4)*(x-4)*(x-6)*x^2)
pretty((x+4)/(x-4)*(x-6)*x^2)
simple((x+4)*(x-4)*(x-6)*x^2)
f=simple((x+2)/(x-4)*sin(x)^2-tan(x))
pretty(f)
```

Symbolic-to-numeric polynomial conversion

.....تحويل التعبير إلى مصفوفة والعكس

```
y=[2 3 4 -2]
f=poly2sym(y)
pretty(f)
g=sym2poly(f)
```

Simple mathematical operations of function.....العمليات الرياضية البسيطة للدوال
.....جمع دالتين

```
symadd('sqrt(x)+cos(x)','1+2*cos(x)-sin(pi/6)')
```

.....طرح دالتين

```
symsub('sqrt(x)+cos(x)','1+2*cos(x)-sin(pi/6)')
```

.....قسمة دالتين

```
symdiv('sqrt(x)+cos(x)','1+2*cos(x)-sin(pi/6)')
```

.....ضرب دالتين

```
symmul('sqrt(x)+cos(x)','1+2*cos(x)-sin(pi/6)')
```

Summation, Limits, Differentiation and Integration of functions
إيجاد المجموع والنهايات والتفاضل والتكامل للدوال
Summation.....المجموع

```
syms x y k
symsum(k^2,0,10)
symsum(x^k/sym('k!'), k, 0,inf)
z=symsum((x+2)*y,'y',a,10),z=simple(z)
```

Substiution.....التعويض عن قيم المجاهيل بقيم ثابتة

```
subs(z,a,3)
subs(z,{a,x},[3,2])
```

Limits.....النهايات
limit(sin(x)/x).....بدون تحديد يعني تؤول الى الصفر
.....النهايات المحدودة

```
limit(sin(x)/x,pi/6)
limit(1/x,x,0,'left')
limit(1/x,x,0,'right')
```

This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

Differentiation .....التفاضل
diff(sin(x^2))
.....التفاضل السادس مثلا
diff(t^6,6)
.....التفاضل بالنسبة لمتغير لدالة تحتوي على أكثر من متغير
diff(x*y^2,'y',2)

Integration .....التكامل

int(-2*x/(1+x^2))
int(-2*x/(1+x^2)^2)
int(x/(1+z^2),z)
pretty(int(x/(1+z^2),x))
.....التكامل المحدد
pretty(int(x/(1+z^2),z,1,3))
.....التكامل الثنائي
syms x y z t h a b
int(int(y*x^2,x,0,pi),y,0,2*pi)
.....التكامل الثلاثي
int(int(int(y*x^2,x,0,pi),y,0,2*pi),z,0,pi)

طريقة أخرى لتعريف المتغيرات وذلك باستخدام علامة التنصيص المفردة

int('k+2*k^2')

2-D Easy Graphics .....الرسم ثنائي الأبعاد المسهل
.....رسم دوال على هيئة منحنيات متصلة ولاداعي للنقاط
syms x y t
ezplot(2*x+sin(x),[0,5*pi]);
ezpolar('sin(t)+exp(t)')
ezpolar('sin(t)',[0,.9*pi])
ezmesh('x*exp(-x^2-y^2)')
colormap([1,0.5,0.7])
ezplot(2*x+sin(x));
fplot('yx',[-pi,pi]);

3-D Easy Graphics.....الرسم ثلاثي الأبعاد المسهل
ezmesh('x*exp(-x^2-y^2)');
.....الأمر التالي للتحكم في الألوان
ezmesh('x*exp(-x^2-y^2)');colormap([.11,.10,.5])

Create Function Calculator .....استدعاء الآلة الحاسبة المستخدمة في عمليات الدوال

funtool

This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

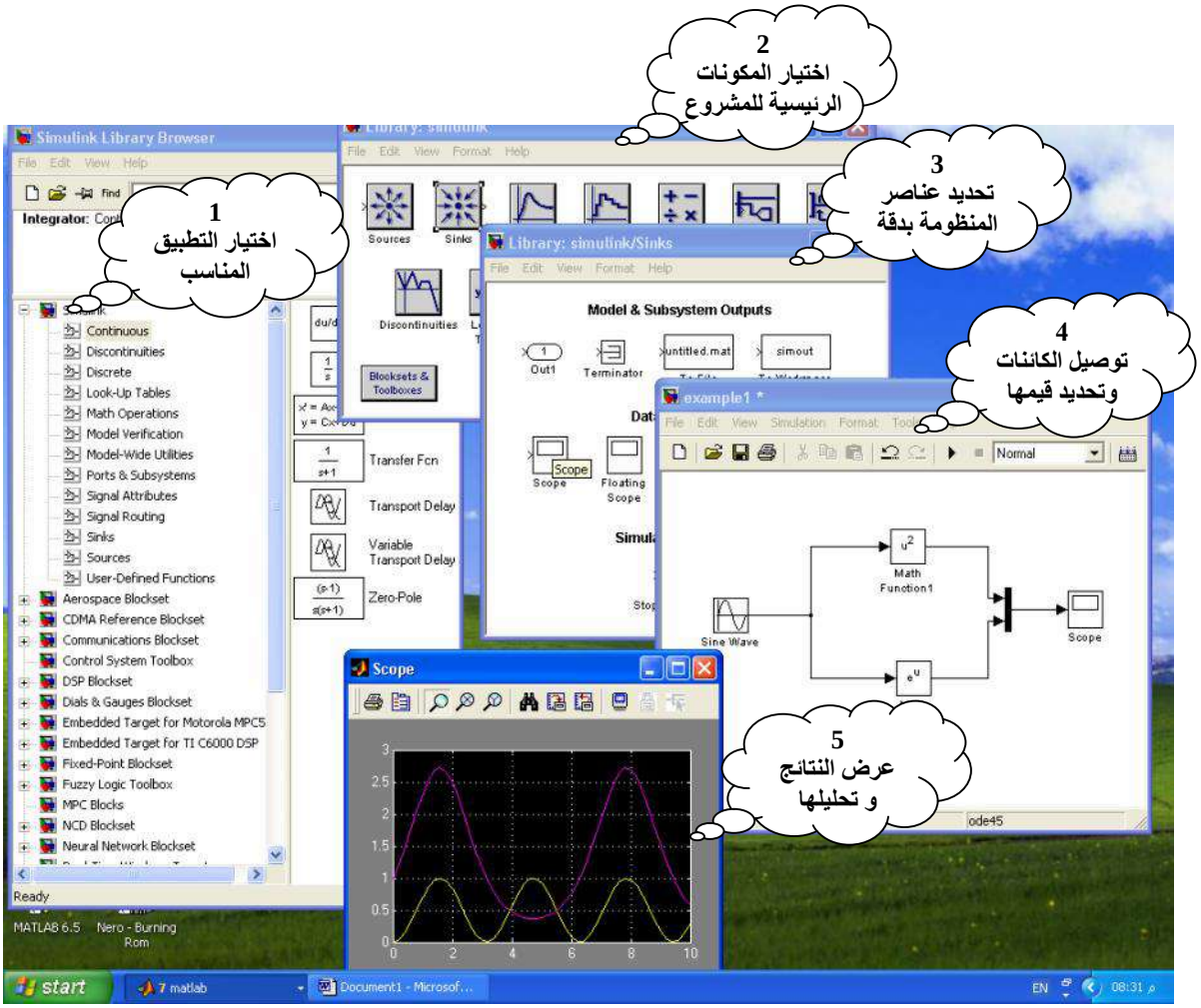
Solving Simultaneous Equations ..... **حل المعادلات الآنية**

```
syms x a b c
pretty(solve('a*x^2+b*x+c'))
[x,y]=solve('x+y=1','x-11*y=5')
```

Solving Differential Equations ..... **حل المعادلات التفاضلية**

```
syms x y f a
y=dsolve('Dy=a*y')
f=dsolve('Df=f+sin(t)')
y=dsolve('Dy=a*y','y(0)=b')
[x,y]=dsolve('Dy-x=exp(t)','Dx+y=exp(-t)','y(0)=0','x(0)=0')
y=dsolve('D2y=-a^2*y','y(0)=1','Dy(pi/a)=0')
الحل الخاص والحل العام%
y=(dsolve('D2y+6*Dy+13*y=10*sin(5*t)','y(0)=0','Dy(0)=0','t'))
pretty(simple(dsolve('D2y+6*Dy+13*y=10*sin(5*t)','y(0)=0','Dy(0)=0','t')))
```

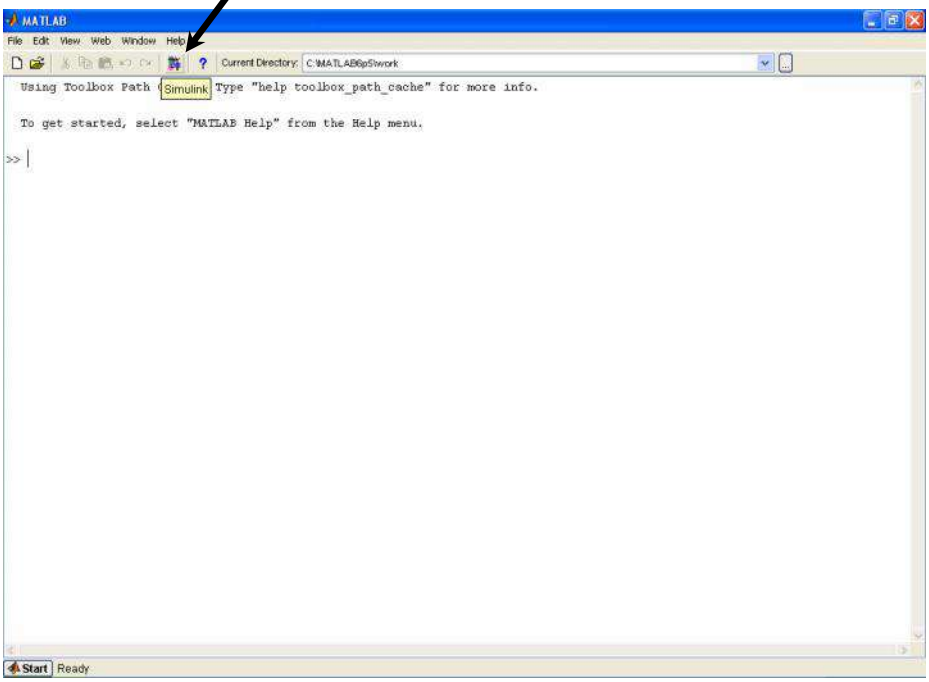
This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com



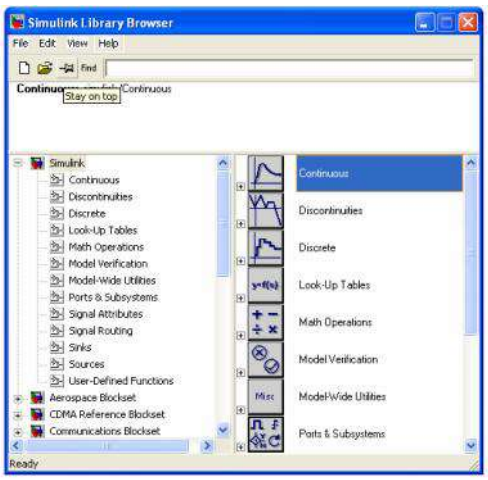
كيفية اختيار العناصرلبناء برنامج محاكاة

This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

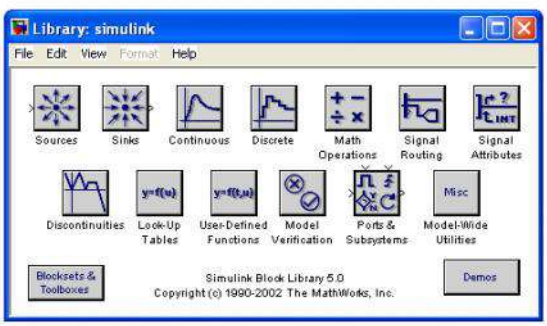
سنبدأ برنامج المحاكاة بفتح Simulink Library Browser وذلك بالنقر على الزر Simulink الموضح بالشكل التالي:



عندها ستظهر النافذة التالية:



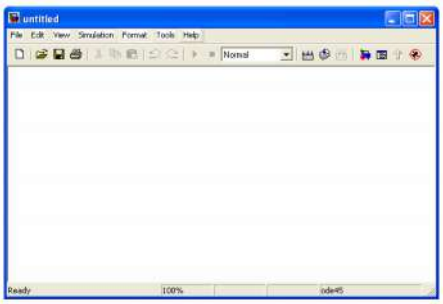
في هذه النافذة تظهر أول مجموعة من الأدوات وهي تمثل العناصر الأساسية لبرامج المحاكاة وتشمل العديد من المكونات والتي من بينها المصادر Sources، أدوات عرض النتائج Sinks، العمليات الرياضية Math Operations، النظم المتصلة Continuous، والمتقطعة Discrete، والكثير من أدوات تحليل النظم الأخرى كما بالشكل التالي:



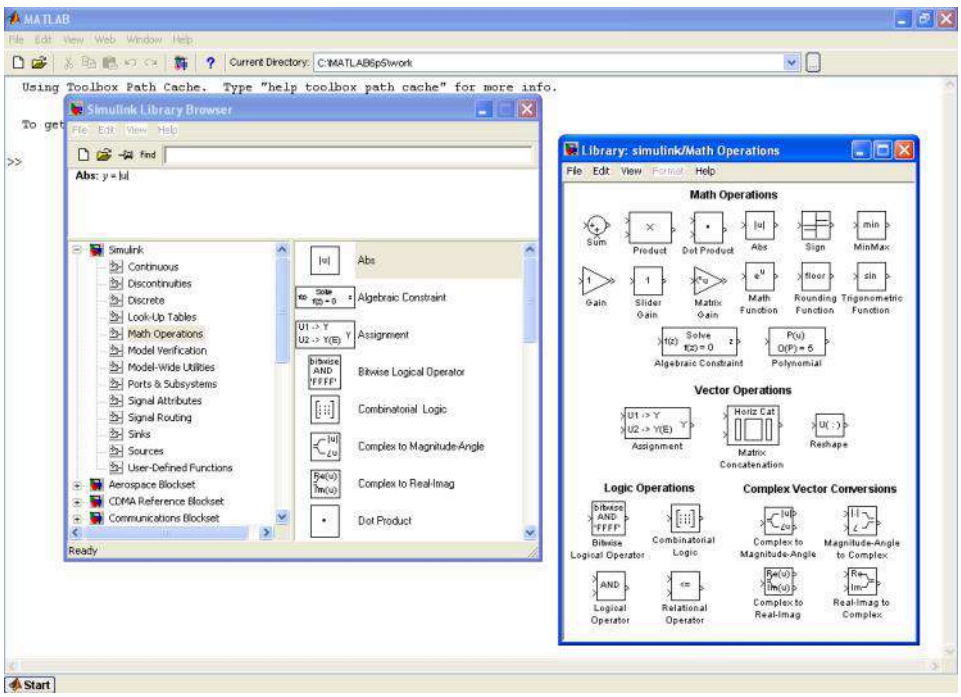
This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

وكمثال لاستخدام هذه المجموعة من برامج المحاكاة سنقوم ببناء منظومة بسيطة لإجراء بعض العمليات على إشارة الدخل وتظهر هذه النتائج في صورة رسومات على جهاز العرض Scope.

افتح برنامج المحاكاة كما عرفنا سابقا، ثم قم بفتح ملف جديد بالنقر على زر التسريع الذي على هيئة صفحة بيضاء في القائمة الرئيسية لـــ Simulink Library Browser، عند ذلك سيتكون ملف جديد غير معنون Untitled كما بالشكل الآتي:

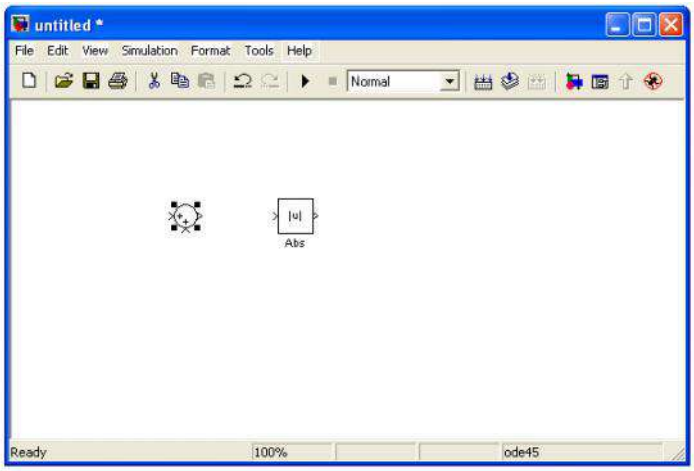


أنقر على الخيار Math Operation، أو اضغط على هذا الخيار باستخدام الزر الأيمن وافتحه، حينها ستظهر لك النوافذ كما تبدو في الشكل التالي:



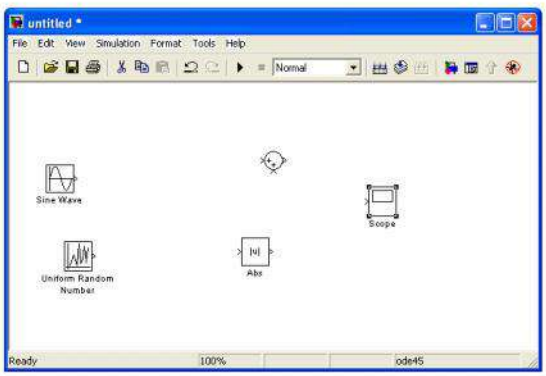
This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

اختر الأيقونة المعنونة بـــ Abs وهي مخصصة لإيجاد القيمة المطلقة، انقر عليها ومستمراً بالغط على زر الفأرة الأيسر اسحب هذه الأيقونة إلى الملف الجديد، وبنفس الكيفية اختر الأيقونة Sum وانقلها إلى الملف، ستحصل على الشكل التالي:



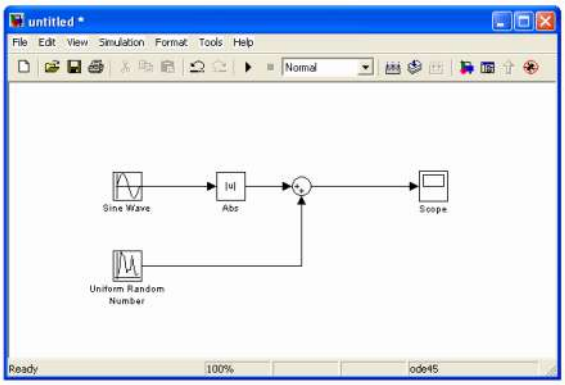
أما الآن فسنختار إشارة الدخل ولتكن Sine Wave ولنتحصل على هذه الإشارة قم بالرجوع إلى Simulink Library Browser واختر من القائمة التي على اليسار Sources واختر من القائمة التي ستظهر الأيقونة Sine Wave وأدرجها في الملف، عد إلى القائمة Sources واختر Uniform Random Number، حاول أن تنقر عليها بزر الفأرة الأيسر ستتظهر لك الجملة Add to Untitled انقر عليها ولاحظ أن الأيقونة انتقلت إلى الملف بهذه الطريقة أيضا.

انتقل مجددا إلى Simulink Library Browser واختر Sinks وبالنقر عليها ستفتح وتظهر لك أدوات عرض النتائج التي يمكنك أن تستخدمها، سنختار هذه المرة Scope، وبعد أن تدرج هذه الأيقونة إلى الملف المفتوح ستكون الصورة عندك مشابهة للشكل التالي:

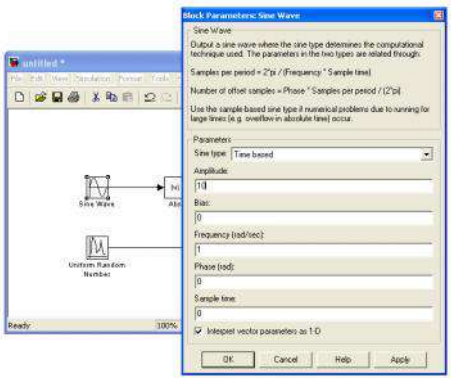


This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

رتب الايقونات الظاهرة في النافذة، لكي يتسنى لك توصيلها مع بعض وذلك بالنقر على المثلث الصغير الموجود في الأيقونة، ومع ابقاء اصبعك ضاغطا على الزر حاول أن توصله بطرف الأيقونة المناسبة.

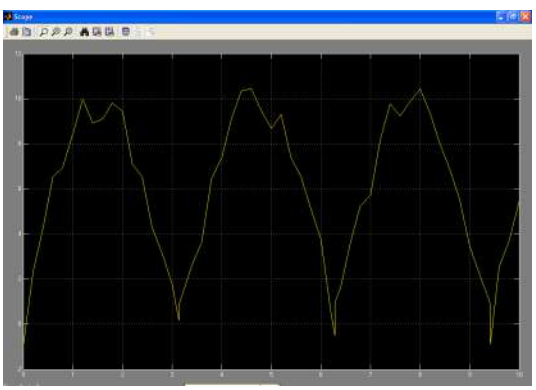


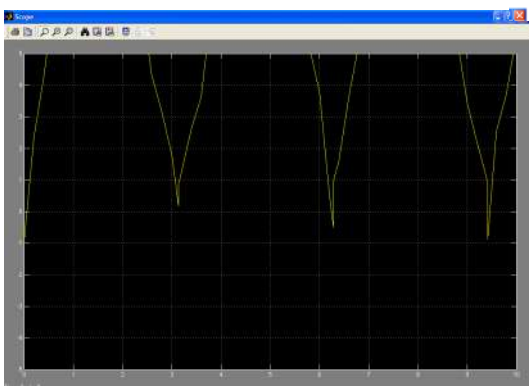
سنقوم في هذه الخطوة بتغيير معامل من معاملات Sine Wave وذلك بالنقر عليها نقرا مزدوجا فتفتح النافذة التالية:



غير اتساع الإشارة وذلك بتغيير القيمة Amplitude إلى 10 ويمكنك أيضا تغيير معاملات أي من الأيقونات بنفس الطريقة.

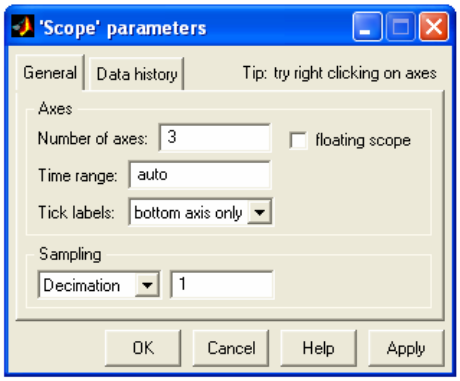
في الخطوة الآتية احفظ الملف باسم مناسب وليكن Simulation_Programme1، ولتقوم بتنفيذ البرنامج بطريقة سريعة وبسيطة انقر زر التسريع Start Simulation وهو على سكل مثلث أسود في أعلى النافذة، ولعرض النتيجة انقر نقرا مزدوجا على Scope، قد تكون شاشة العرض غير مناسبة عندها قم بالنقر على Auto scale وهو الزر الشبه بالمنظار في شاشة عرض Scope.





This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

كما يمكن أيضاً أن نجعل Scope يبين خرجين أو أكثر وذلك بالنقر على الزر Parameters في أعلى Scope وهو الزر الموجود بجانب زر الطباعة، عندها ستظهر لك النافذة الصغيرة التالية:

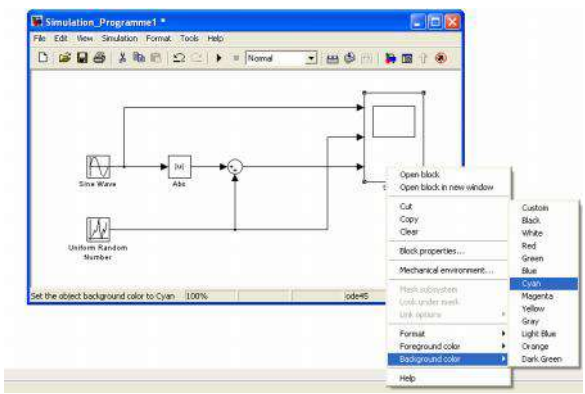


ومن خلال تغيير Number of axes إلى 3 ستتحول شاشة Scope إلى شاشة بها ثلاث فضاءات لعرض النتائج، كذلك سنجد أن الأيقونة Scope في الملف قد صار لها ثلاث منافذ للدخل مما يسمح لنا من قياس إشارتين إضافيتين باستخدام عارض واحد.

سنضيف مسارا جديد للإشارة من Sine Wave إلى Scope وذلك كالتالي:

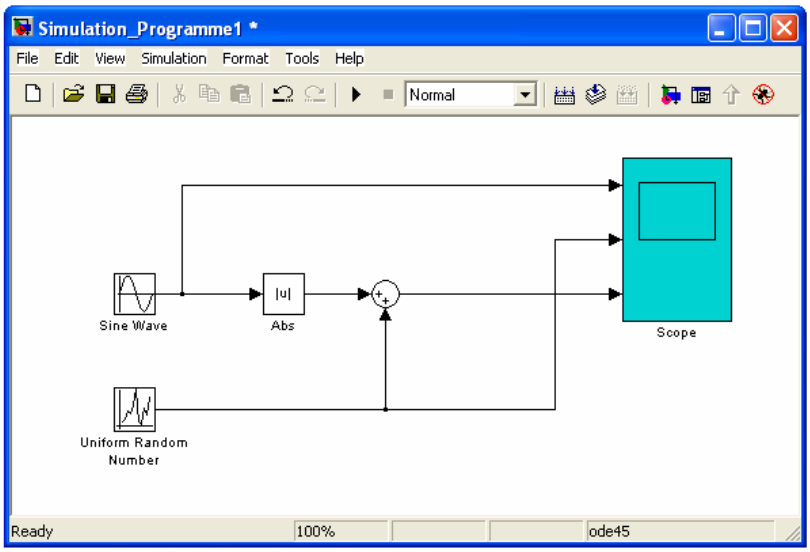
أنقر أولا على المسار بين Sine Wave و Abs، اضغط على المفتاح Ctrl في لوحة المفاتيح، ومستمرا باضغط على المفتاح حاول ان ترسم المسار الجديد بين Sine Wave و Scope. أرسم المسار الجديد كذلك بين Uniform Random Number و Scope.

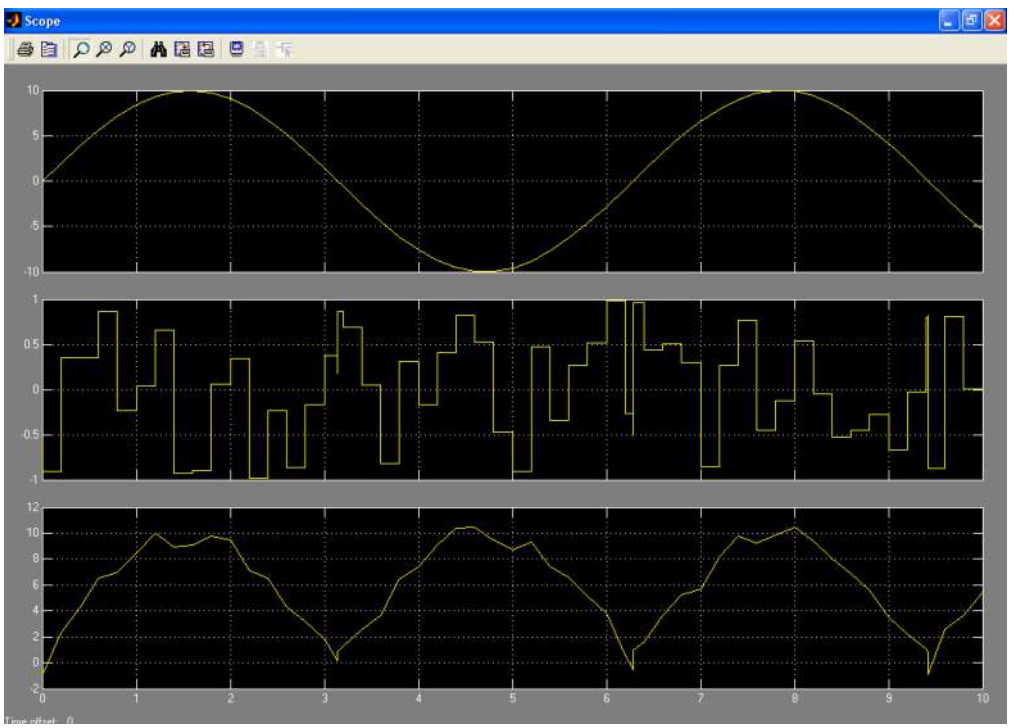
يمكننا التحكم في حجم الايقونات بسهولة من خلال النقر عليها مرة واحدة ثم جذبها من أحد أطرافها، كذلك يمكن تلوين الأيقونات وغيرها من خلال النقر عليها بالزر الأيمن الذي سيظهر لنا القنافذة التالية:



This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

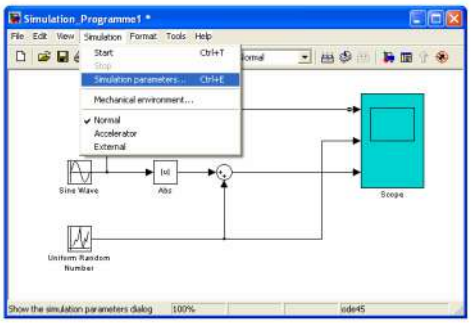
أخيراً سيكون سيكون الشكل النهائي لمنظومتنا وإشاراتها كالتالي:





This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

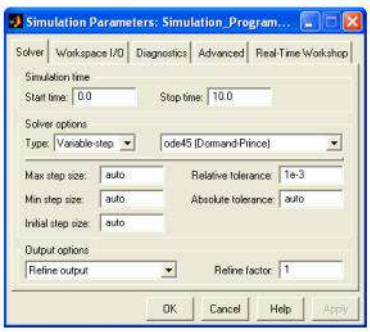
حاول أن تغير بعض الإعدادت لبرنامج المحاكاة عن طريق النقر على Simulation، ثم اختر Simulation Parameters

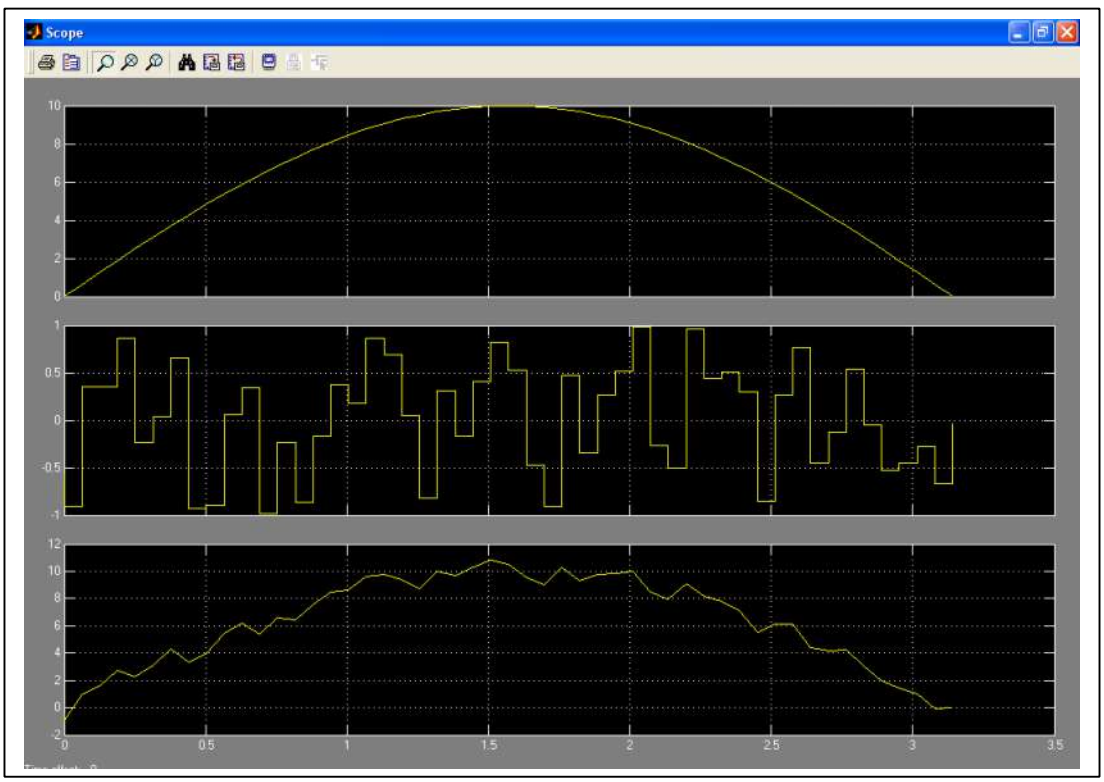


عندها ستظهر النافذة الآتية:

غير Stop time إلى π (pi)

ولاحظ النتيجة.



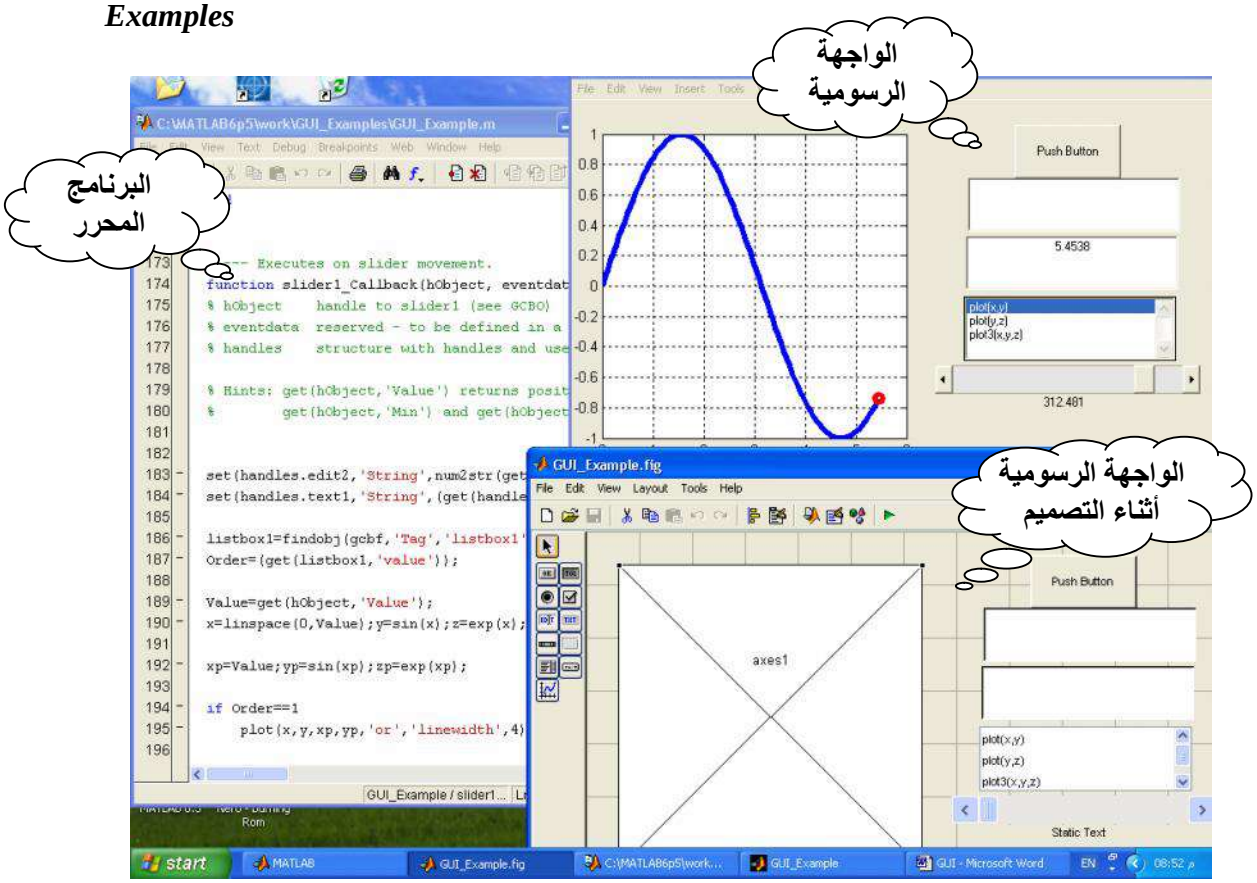


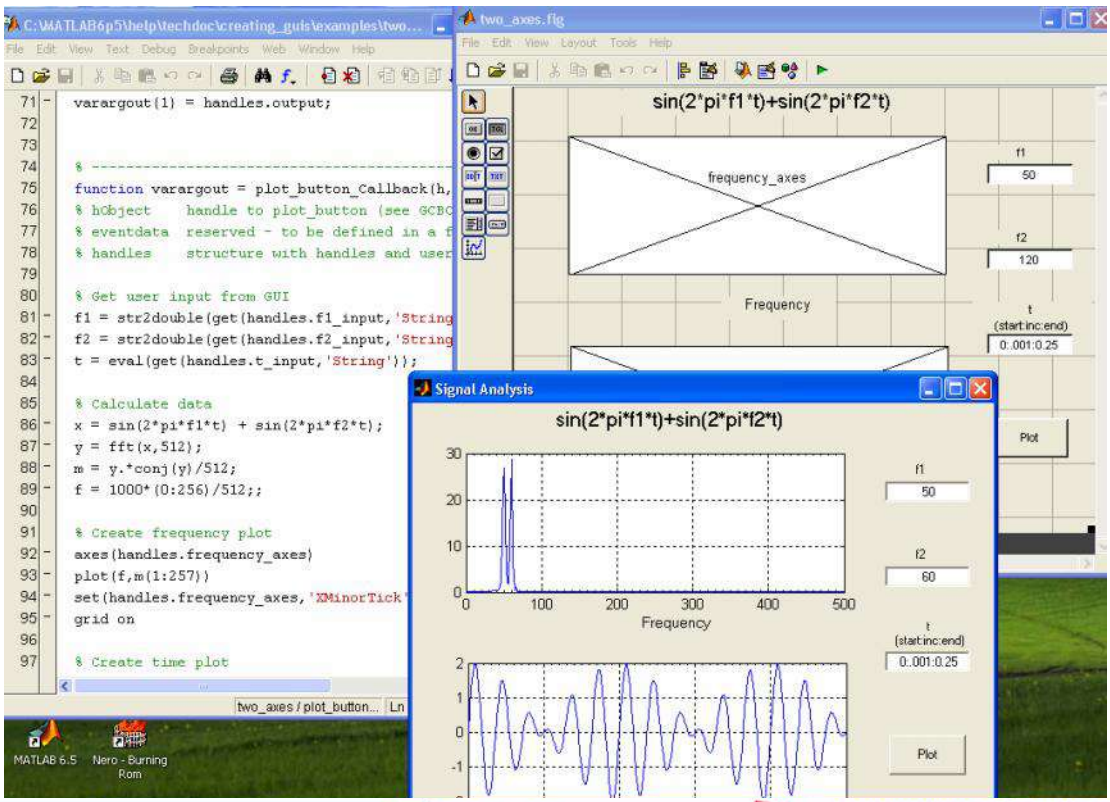
This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

GUI ..... **الوجهات الرسومية التفاعلية**

Who should create GUIs- and Why? ..... **ما سبب اللجوء للواجهات الرسومية كطريقة للبرمجة؟**

Examples



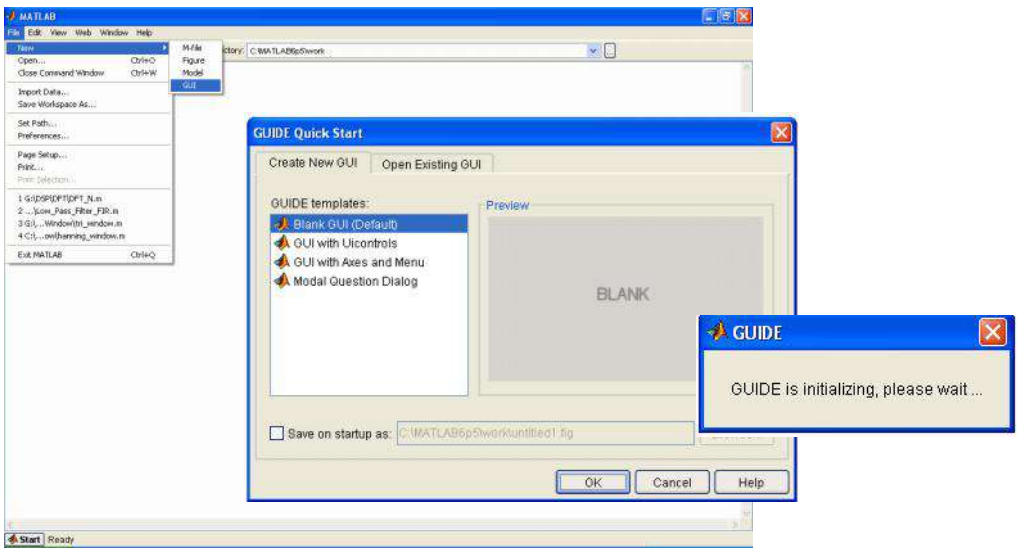


This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

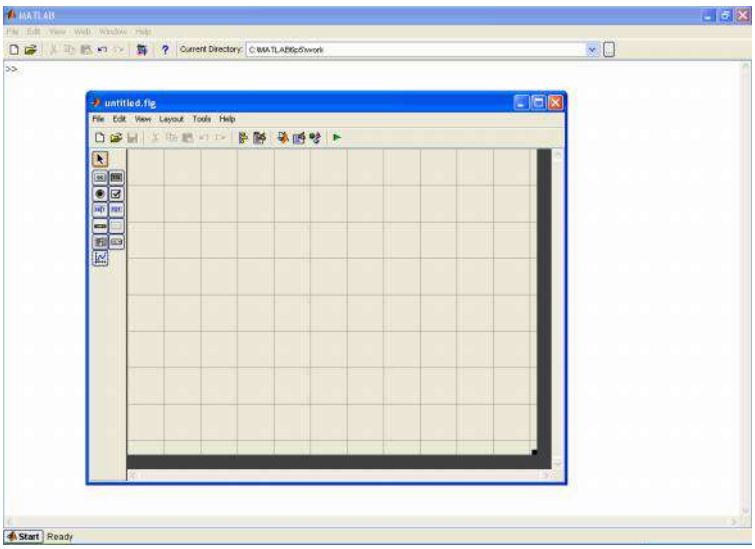
Graphical User Interface(GUI): **البرمجة باستخدام الوجهات الرسومية:**

- لتبدأ استخدام الوجهات الرسومية في MATLAB اتبع التسلسل التالي وكما هو مبين بالأشكال:

File> GUI> Blank GUI (Default)



عندها ستظهر لك نافذة كما بالشكل التالي:

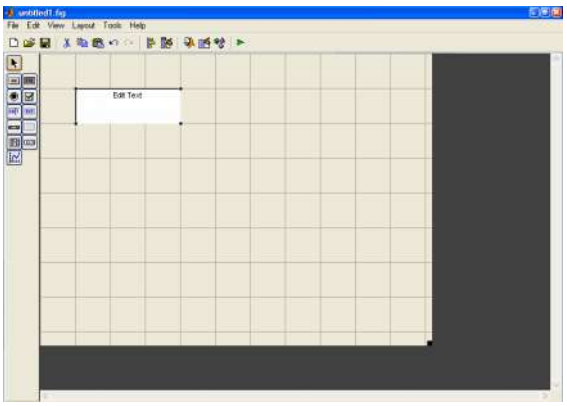


- كما يمكنك استدعاء الواجهات الرسومية في MATLAB بكتابة الأمر (guide) في Command Window.

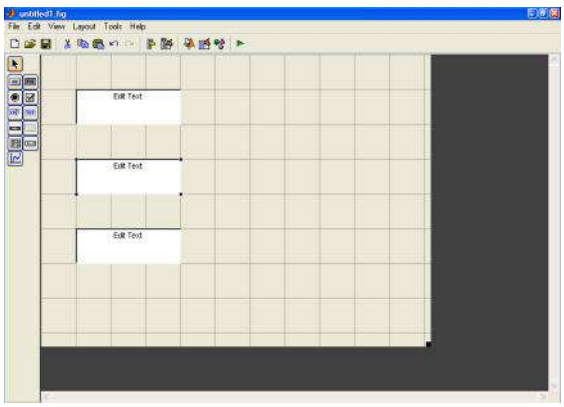
يمكنك الآن اختيار بعض الكائنات الموجودة على يسار النافذة لتكوين برنامج واجهة رسومية بسيط، سنستعمل هذا البرنامج لجمع او طرح عددين نقوم بادخالهما عن طريق مربعي تحرير Edit Text1 و Edit Text2 ويوضع الناتج في مربع تحرير ثالث Edit Text3.

This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

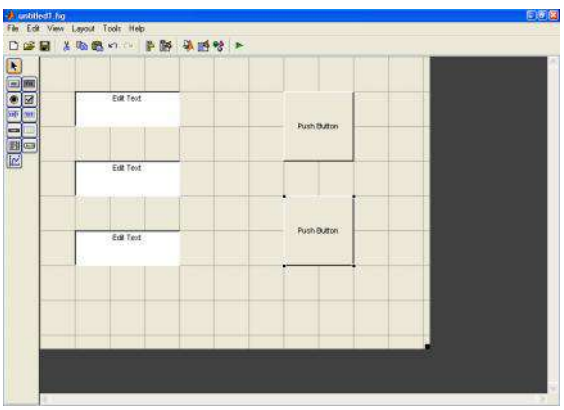
اختر مربع تحرير Edit Text من القائمة التي على يسار النافذة وانقر عليه بالزر الأيسر للفأرة، ثم انتقل إلى المساحة الفارغة من النافذة وبضغطة مستمرة على الزر الأيسر للفأرة ارسم الحجم الذي تريده لمربع التحرير.



يمكنك أن تنشأ مربع التحرير الاول وتنسخ عنه نسختين للحصول على ثلاث مربعات تحرير متشابهة في الحجم.

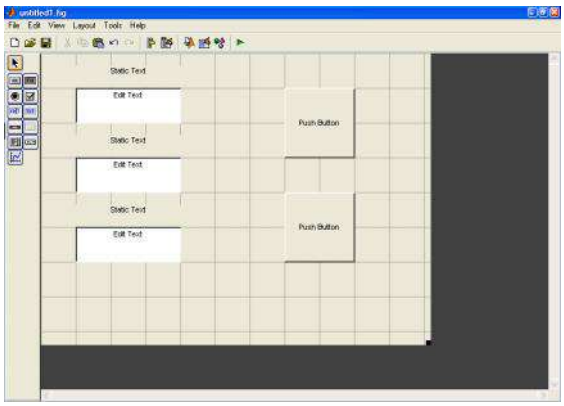


بنفس الخطوات السابقة اتجه إلى القئمة اليسرى واختر زر التنفيذ OK وارسم الحجم المناسب للزر، ثم انسخ عنه زرًا ثانيا لتبدو لك النافذة كما بالشكل التالي:



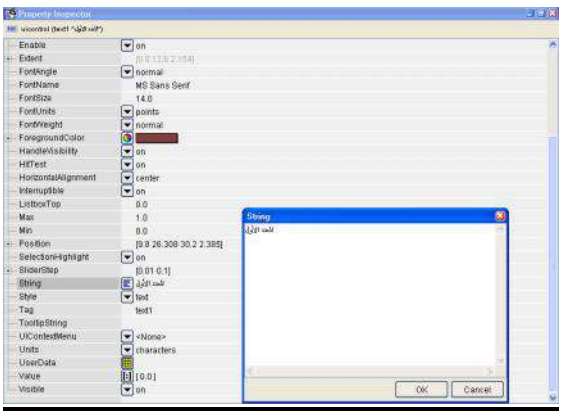
This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

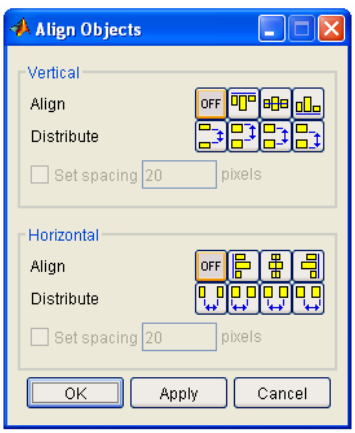
سنضيف إلى الواجهة في هذه الخطوة مربع نص ثابت فوق كل مربع تحرير ليتمكن مستخدم البرنامج من معرفة ما سيكون في هذا المربع.



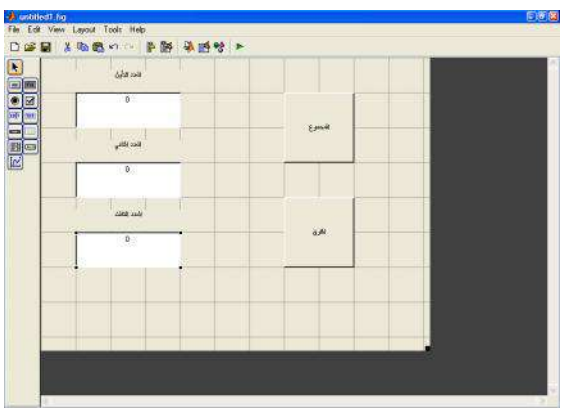
لتغيير النصوص المكتوبة على الكائنات التي شكلنا منها واجهة البرنامج وكذلك لون الخط وحجمه وبعض الخصائص الأخرى، اضغط على الزر Property Inspector الموجود في أعلى الشاشة، ستظهر في الحال نافذة تبين الخصائص التي يمكن تغييرها للكائن المفعل.

يمكنك التحكم في حجم هذه الكائنات ومواضعها مفردة أو مجموعة باستخدام الفأرة، أو باستخدام ازرار التسريع الموجودة في أعلى النافذة، حيث يمكنك محاذاة الكائنات بالشكل والمسافة المرغوبة باستخدام Align Objects.



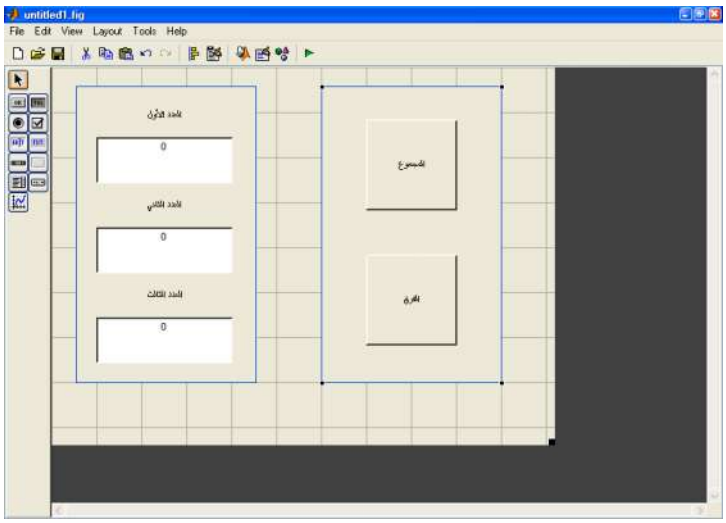


ستكون النافذة كما بالشكل التالي بعد تغيير خصائص الكائنات.

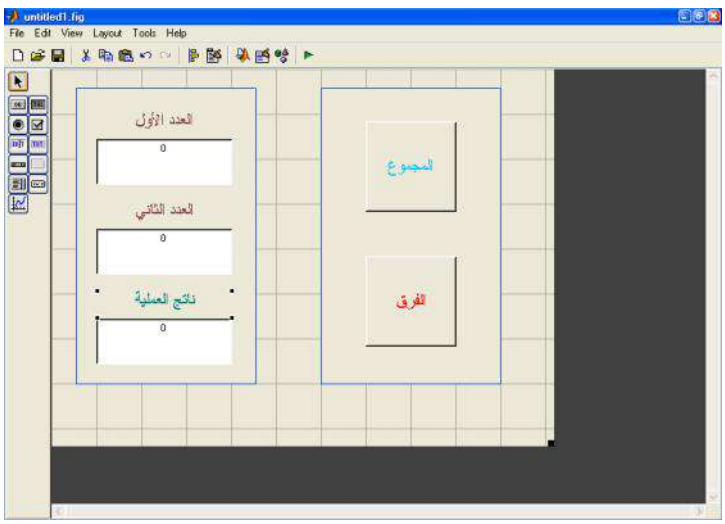


This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

في الخطوة التالية انتقل إلى قائمة الكائنات على اليسار واختر Frame إطار، ارسم الإطار مظللا به أزرار الضغط التي على يمين النافذة، وبضغطة واحدة على الزر الأيمن للفأرة ستظهر لك قائمة منسدلة اختر منها Send to Back، سيظهر لك الزران المختفيان وراء الإطار، انسخ عن هذا الإطار إطارا آخر وانتقل به إلى الجهة اليسرى من النافذة، وبنفس الكيفية انقله خلف الكائنات باستخدام Send to Back. عندها ستظهر لك نافذة كالنافذة التالية:



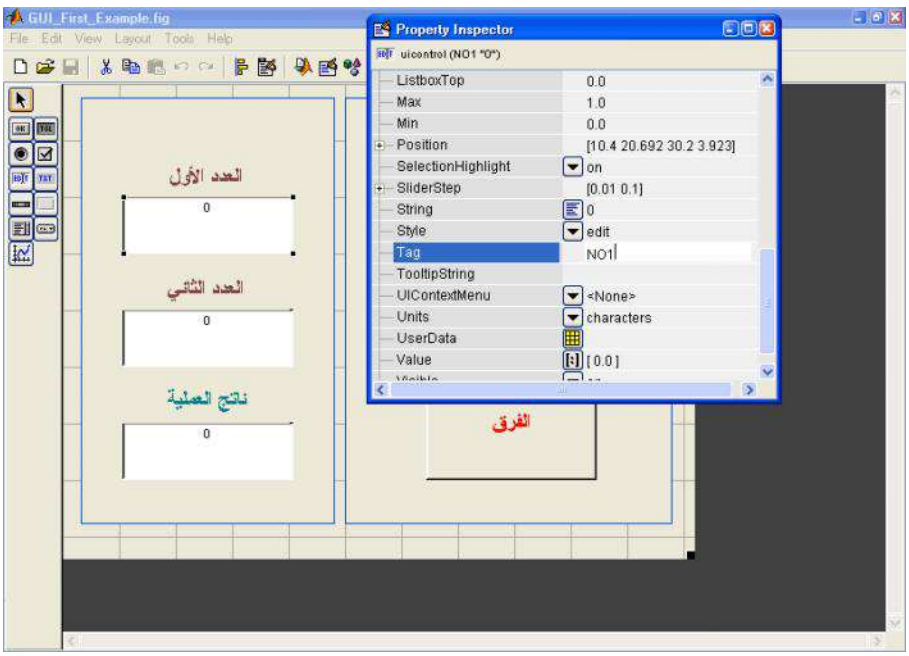
أما الآن فسنحاول أن نجري بعض التعديلات الأخيرة على الوجهة ونغير ألوان الخطوط وحجمها لإكسابها مزيدا من التمييز.



This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

إن من أهم الخصائص التي يجب الانتباه لها عند استعمال الواجهات الرسومية، الخاصية المسماه Tag وهي تستخدم للإشارة إلى الكائن من داخل البرنامج المشفر لهذه الواجهة البرمجية.

افتح خصائص مربع التحرير الأول واكتب أمام الخاصية **Tag**: NO1 ، ثم افتح خصائص مربع التحرير الثاني واكتب أمام الخاصية **Tag**: NO2 ، بعد ذلك انتقل إلى مربع التحرير الثالث واكتب أمام الخاصية **Tag**: Result، انتقل الآن إلى زر المجموع وغير خاصية الإشارة له إلى Sum_button، أخيرا انتقل إلى زر الفرق اضغط عليه ضغطتين سريعتين بالزر الأيسر للفأرة وغير تأشيره إلى Diff_button .



اضغط على الزر المرمز بشكل أيقونة MATLAB الموجود في أعلى النافذة، سيظهر لك مباشرة ملف نصي يحوي شفرة البرنامج، ولكي يشتغل البرنامج كما نريد سنضيف بعض الأسطر إليه كما بالخطوات التالية:

انتقل إلى زر التسريع File واختر Save As عندها ستظهر لك نافذة تخزين البرنامج، أكتب اسم البرنامج باللغة الانجليزية غير مبتدإ برقم ولا تاركا لفراغات بين مقاطع الاسم، سنقترح اسما لهذا البرنامج وليكن(GUI_First_Example).

اضغط الرمز على شكل حرف f الموجود في أعلى البرنامج، ستظهر لك في الحال قائمة منسدلة بها مجموعة دوال الاستدعاء الموجودة في البرنامج، عندها اختر Sum_Button_Callback ، سينتقل المؤشر في الحال إلى حيث هذه الدالة. انتقل إلى ما بعد الأسطر غير المفعّلة – خضراء اللون افتراضيا– واكتب الأسطر التالية:

```
x1=str2double(get(handles.NO1,'String'));

x2=str2double(get(handles.NO2,'String'));

y=x1+x2;
```

This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

```
set(handles.Result,'string',num2str(y));
```

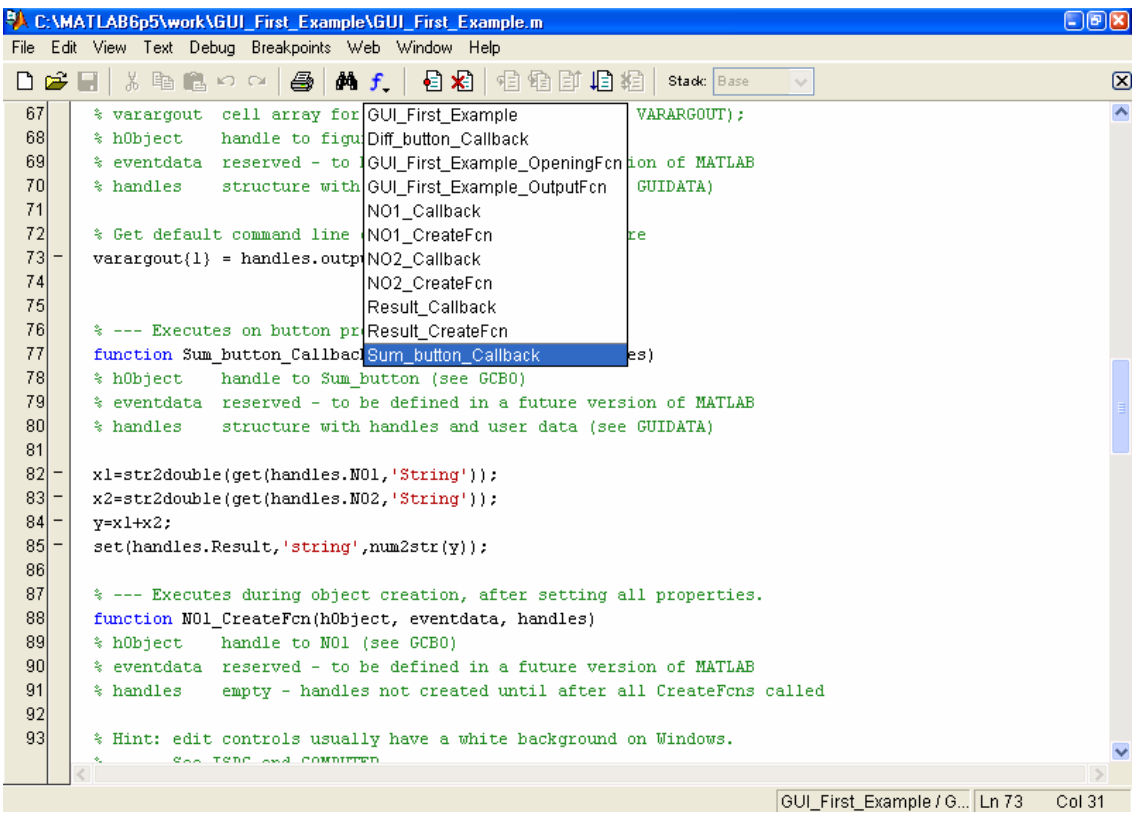
السطر الأول يستخلص النص المكتوب في مربع التحرير المشار إليه بـــNO1، ثم يحول هذا النص إلى رقم ذو دقة مضاعفة، ويخزنه في متغير اسمه x1 .

السطر الأول يستخلص النص المكتوب في مربع التحرير المشار إليه بـــNO2، ثم يحول هذا النص إلى رقم ذو دقة مضاعفة، ويخزنه في متغير اسمه x2 .

السطر الثالث يجمع العددين ويخزن الناتج في متغير اسمه y .

السطر الرابع يحول قيمة المتغير y إلى نص، ثم يضعه في مربع التحرير المشار إليه بـــ Result .

* لاحظ أن إشارة الكائن تكون عن طريق تغيير خاصية الكائن Tag.



This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

وبنفس الكيفية اضغط الرمز على شكل حرف f الموجود في أعلى البرنامج، ستظهر لك في الحال قائمة منسدلة بها مجموعة دوال الاستدعاء الموجودة في البرنامج، عندها اختر Diff_Button_Callback ، سينتقل المؤشر في الحال إلى حيث هذه الدالة.

انتقل إلى ما بعد الأسطر غير المفعّلة – خضراء اللون افتراضيا– واكتب الأسطر التالية:

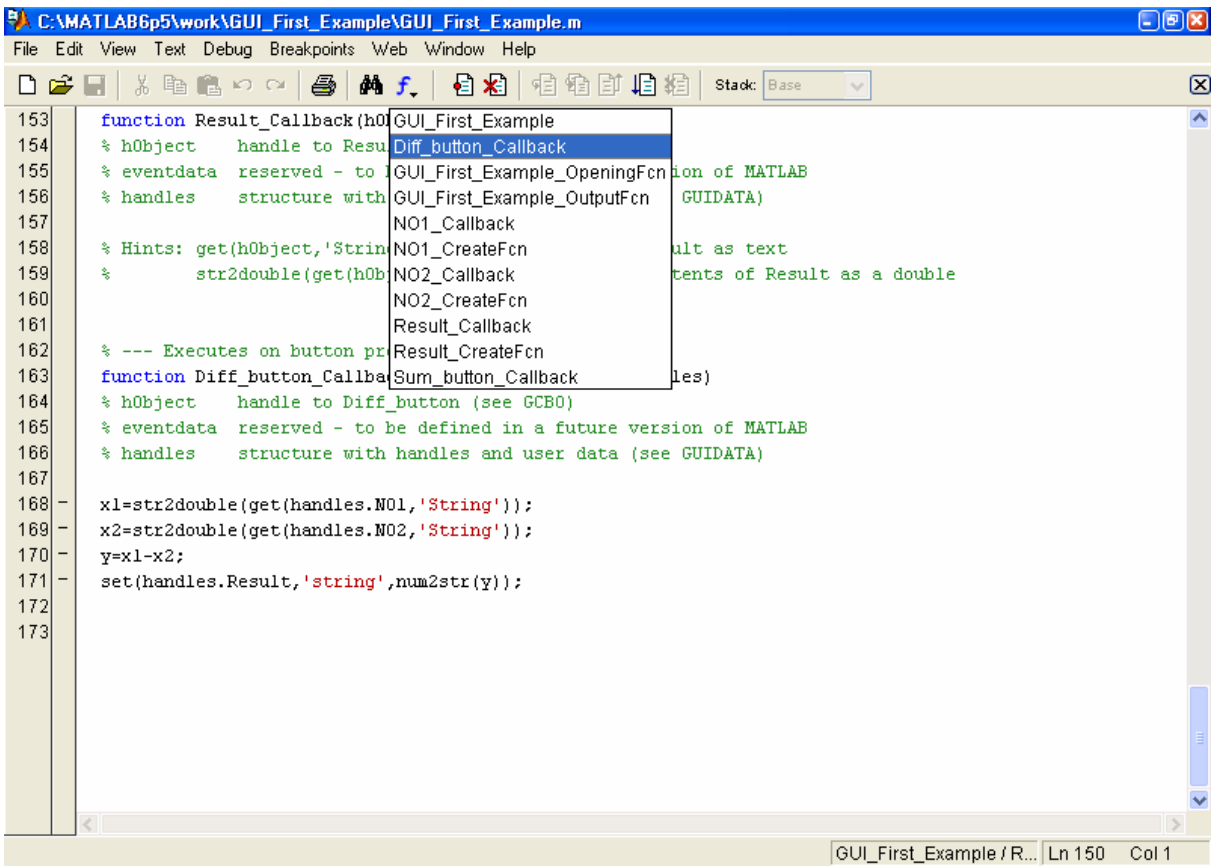
```
x1=str2double(get(handles.NO1,'String'));

x2=str2double(get(handles.NO2,'String'));

y=x1-x2;

set(handles.Result,'string',num2str(y));
```

السطر الأول يستخلص النص المكتوب في مربع التحرير المشار إليه بـــNO1، ثم يحول هذا النص إلى رقم ذو دقة مضاعفة، ويخزنه في متغير اسمه x1 .

السطر الأول يستخلص النص المكتوب في مربع التحرير المشار إليه بـــNO2، ثم يحول هذا النص إلى رقم ذو دقة مضاعفة، ويخزنه في متغير اسمه x2 .

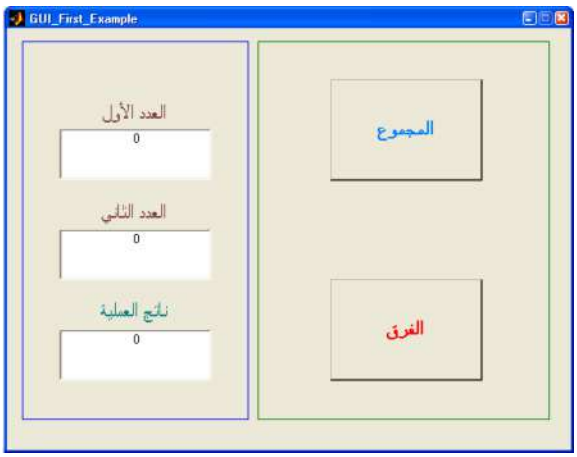
السطر الثالث يطرح العدد الثاني من العدد الأول ويخزن الناتج في متغير اسمه y .

السطر الرابع يحول قيمة المتغير y إلى نص، ثم يضعه في مربع التحرير المشار إليه بـــ Result .



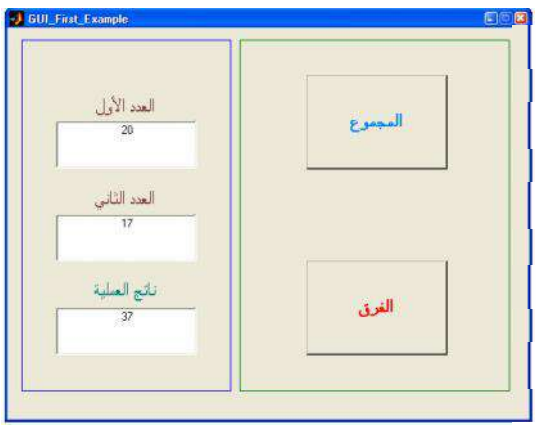
This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

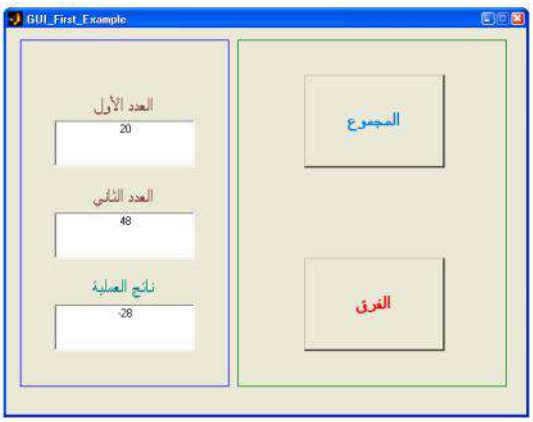
أخيراً: اضغط على زر التسريع **Run** ستظهر لك الواجهة التنفيذية للبرنامج، أو اضغط على الزر الأخضر المثلث الشكل في أعلى النافذة Run وستظهر لك النافذة التالية:



حاول أن تدخل قيمتين في مربعي التحرير الأول والثاني، ونفذ عملية الجمع بضغط زر المجموع وكذلك نفذ عملية الطرح بالضغط على زر الفرق.

في الشكلين المقابلين تنفيذ للبرنامج لكلتا العمليتين:





This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

استخدام الدوال في بناء البرامج الفرعية:..... Subroutines in MATLAB by functions:

نبدأ البرنامج الفرعي بكلمة function، وتلحق هذه الكلمة بفراغ يأتي بعده اسم البرنامج الفرعي، أول سطر في هذا البرنامج الفرعي يجب أن يبدأ بتعريف المتغيرات المتبادلة بين البرامج الفرعية والبرنامج الرئيسي، يتم تعريف هذه المتغيرات والمعاملات عن طريق الأمر global يعقبه أسماء المتغيرات، بعد ذلك نكتب خوارزمية البرنامج الفرعي أو الدالة لتنفيذ أمر أو مجموعة من الأوامر التي سيتم تنفيذها في هذا البرنامج الفرعي عند استدعائه في البرنامج الرئيسي.

وكمثال على استخدام البرامج الفرعية في MATLAB سنكتب برنامجا رئيسيا مقسما إلى ثلاث برامج فرعية كالتالي:

- برنامج فرعي للاستنهاض (Initialization) ومن خلاله ستعطى قيم ابتدائية لبعض المتغيرات، ويطلب البرنامج إدخال قيم متغيرات أخرى.
- برنامج فرعي للحسابات (Calculations) وهو المسئول عن الحسابات في البرنامج الرئيسي.
- برنامج التقرير (Report) هذا البرنامج الفرعي أو الدالة هي التي سنحدد فيها طريقة عرض النتائج.

البرنامج الرئيسي:

```
% Main Programme

global K xi yi h f Y xii Y_Euler R

Initialization
Calculations
Report
```

البرنامج الفرعي الأول (برنامج الاستنهاض):

```
function Initialization

clear
clc

global K xi yi h f Y xii Y_Euler R

xi=input('Initial value of x hint x=0: ');;
yi=input('Initial value of y hint y=1: ');
h=0.1;
```

This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

البرنامج الفرعي الثاني (برنامج الحسابات):

```
function Calculations

global K xi yi h f Y xii Y_Euler R

for K=1:51
    f=0.5*yi; % dy/dx=.5y
    Y_Euler=yi+h*f;
    xii=xi+h;
    Y_true=exp(xii/2);
    R(K,[1:6])=[K-1,xi,yi,f,Y_Euler,Y_true]; % Matrix of Results
 %    New Values:
    xi=xii;
    yi=Y_Euler;
end
```

البرنامج الفرعي الثالث (برنامج التقرير):

```
function Report

global K xi yi h f Y xii Y_Euler R

format short

disp('Euler Method Results: ')
disp('   ')
disp('      K       xi       yi     f(xi,yi)  y(i+1)Eul. True(i+1)   ')
disp('   ------------------------------------------------------------')
disp(R)
plot(R(:,2),R(:,5:6))
grid on
```

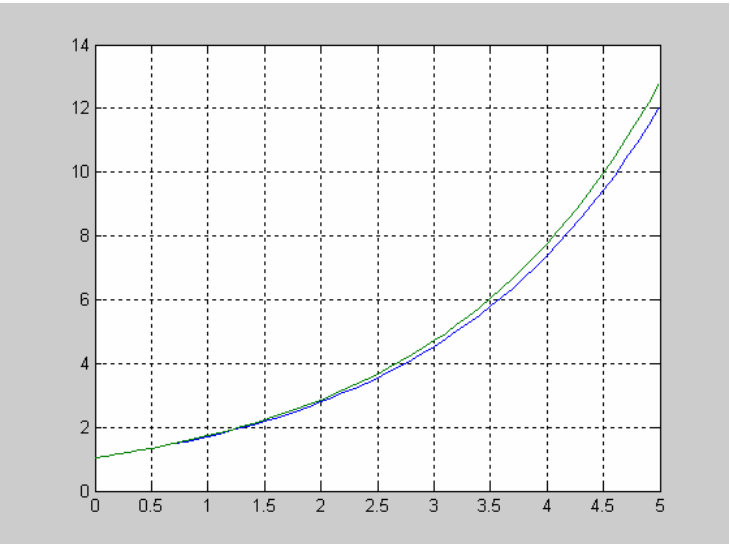
This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

Euler Method Results: نتيجة البرنامج:

K	xi	yi	f(xi,yi)	y(i+1)Eul.	True(i+1)
0	0	1.0000	0.5000	1.0500	1.0513
1.0000	0.1000	1.0500	0.5250	1.1025	1.1052
2.0000	0.2000	1.1025	0.5513	1.1576	1.1618
3.0000	0.3000	1.1576	0.5788	1.2155	1.2214
4.0000	0.4000	1.2155	0.6078	1.2763	1.2840
5.0000	0.5000	1.2763	0.6381	1.3401	1.3499
6.0000	0.6000	1.3401	0.6700	1.4071	1.4191
7.0000	0.7000	1.4071	0.7036	1.4775	1.4918
8.0000	0.8000	1.4775	0.7387	1.5513	1.5683
9.0000	0.9000	1.5513	0.7757	1.6289	1.6487
10.0000	1.0000	1.6289	0.8144	1.7103	1.7333
11.0000	1.1000	1.7103	0.8552	1.7959	1.8221
12.0000	1.2000	1.7959	0.8979	1.8856	1.9155
13.0000	1.3000	1.8856	0.9428	1.9799	2.0138
14.0000	1.4000	1.9799	0.9900	2.0789	2.1170
15.0000	1.5000	2.0789	1.0395	2.1829	2.2255
16.0000	1.6000	2.1829	1.0914	2.2920	2.3396
17.0000	1.7000	2.2920	1.1460	2.4066	2.4596
18.0000	1.8000	2.4066	1.2033	2.5270	2.5857
19.0000	1.9000	2.5270	1.2635	2.6533	2.7183
20.0000	2.0000	2.6533	1.3266	2.7860	2.8577
21.0000	2.1000	2.7860	1.3930	2.9253	3.0042
22.0000	2.2000	2.9253	1.4626	3.0715	3.1582
23.0000	2.3000	3.0715	1.5358	3.2251	3.3201
24.0000	2.4000	3.2251	1.6125	3.3864	3.4903
25.0000	2.5000	3.3864	1.6932	3.5557	3.6693
26.0000	2.6000	3.5557	1.7778	3.7335	3.8574
27.0000	2.7000	3.7335	1.8667	3.9201	4.0552

This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

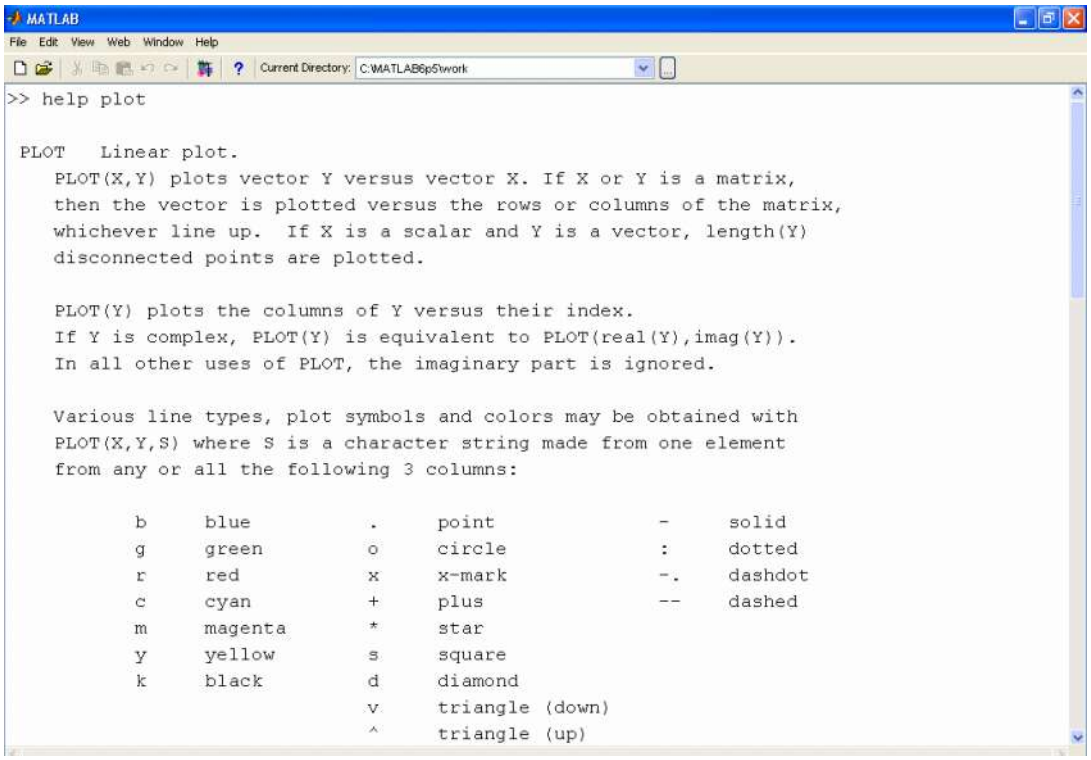
28.0000	2.8000	3.9201	1.9601	4.1161	4.2631
29.0000	2.9000	4.1161	2.0581	4.3219	4.4817
30.0000	3.0000	4.3219	2.1610	4.5380	4.7115
31.0000	3.1000	4.5380	2.2690	4.7649	4.9530
32.0000	3.2000	4.7649	2.3825	5.0032	5.2070
33.0000	3.3000	5.0032	2.5016	5.2533	5.4739
34.0000	3.4000	5.2533	2.6267	5.5160	5.7546
35.0000	3.5000	5.5160	2.7580	5.7918	6.0496
36.0000	3.6000	5.7918	2.8959	6.0814	6.3598
37.0000	3.7000	6.0814	3.0407	6.3855	6.6859
38.0000	3.8000	6.3855	3.1927	6.7048	7.0287
39.0000	3.9000	6.7048	3.3524	7.0400	7.3891
40.0000	4.0000	7.0400	3.5200	7.3920	7.7679
41.0000	4.1000	7.3920	3.6960	7.7616	8.1662
42.0000	4.2000	7.7616	3.8808	8.1497	8.5849
43.0000	4.3000	8.1497	4.0748	8.5572	9.0250
44.0000	4.4000	8.5572	4.2786	8.9850	9.4877
45.0000	4.5000	8.9850	4.4925	9.4343	9.9742
46.0000	4.6000	9.4343	4.7171	9.9060	10.4856
47.0000	4.7000	9.9060	4.9530	10.4013	11.0232
48.0000	4.8000	10.4013	5.2006	10.9213	11.5883
49.0000	4.9000	10.9213	5.4607	11.4674	12.1825
50.0000	5.0000	11.4674	5.7337	12.0408	12.8071



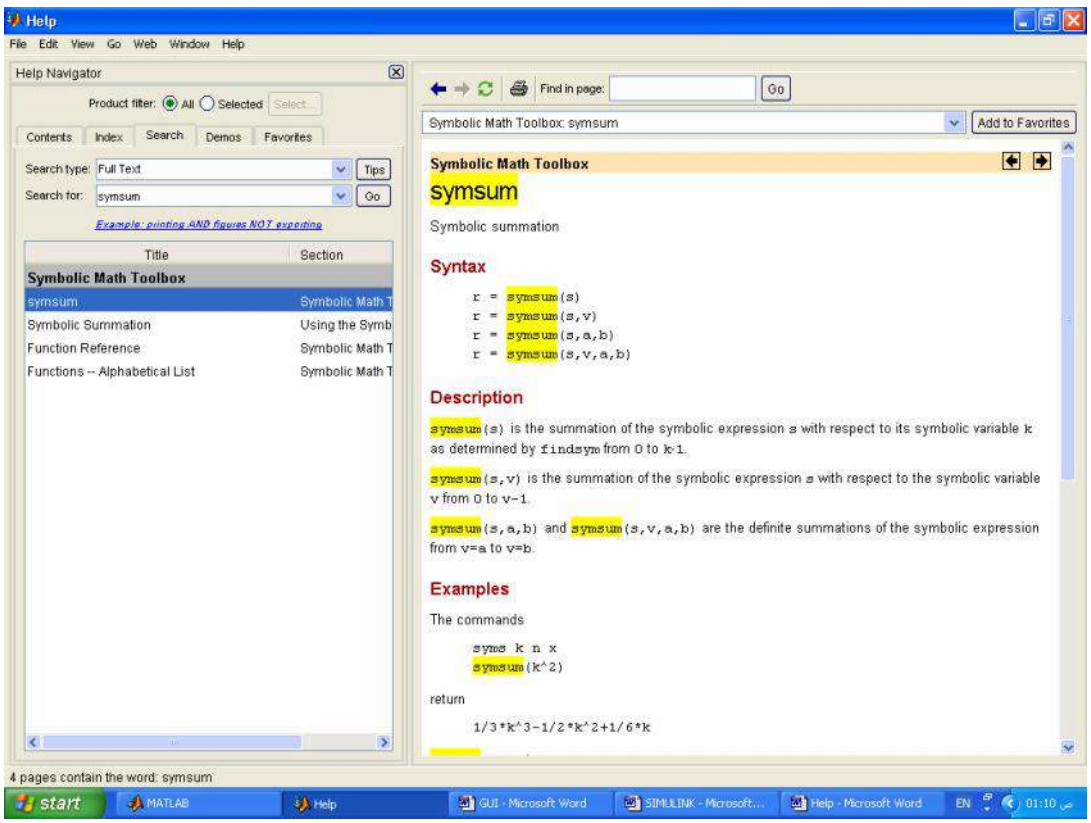
This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

طلب المساعدة ..... **MATLAB Help**

المساعدة المباشرة.....***Online help***



المساعدة باستخدام الملفات المخزنة.....***MATLAB Help (search in disk files)***



This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

References.....**المراجع**

- MATLAB Help.
- Demonstration Programs Included with MATLAB.
- The MathWorks Web Site: http://www.mathworks.com/
- MATLAB Central: http://www.mathworks.com/matlabcentral_redirect
- MATLAB Tutorials.
- Technical Support Knowledge Base: http://www.mathworks.com/support

This document was created using SOLID CONVERTER PDF
To remove this message, purchase the product at www.SolidPDF.com

MATLAB
تعليـم الماتلاب خطوة بخطـــــــوة

اعداد وتقديم

المهندس:- احمد محمد الفلاح الرابطى.

التخصص:- الهندسة الكهربية.

القسم:- التحكم الألى.

الجامعة:- جامعة الجبل الغربى.

المهنة:- معيد فى كلية الهندسة.

البريد الألكترونى:- ahmad_engineer21@yahoo.com

المستوى التعليمى:- بكالوريوس فى الهندسة الكهربية شعبة التحكم الألى من جامعة الجبل الغربى ودوبلوما فى الدراسات العليا فى الهندسة الكهربية شعبة التحكم الألى من جامعة الفاتح.

السنة الدراسية:- 2009 – 2010 ف.

الهواية:- المطالعة والشطرنج وكرة القدم.

العنوان:- ليبيا – غريان – الرابطة.

Numerical Integration

1- Trapezoidal Rule

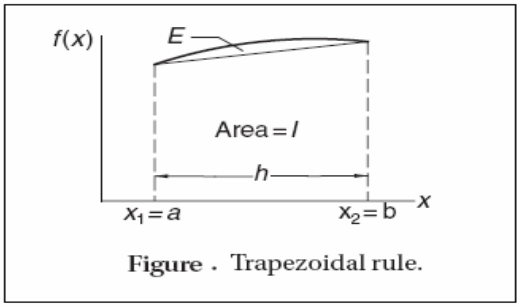


Figure . Trapezoidal rule.

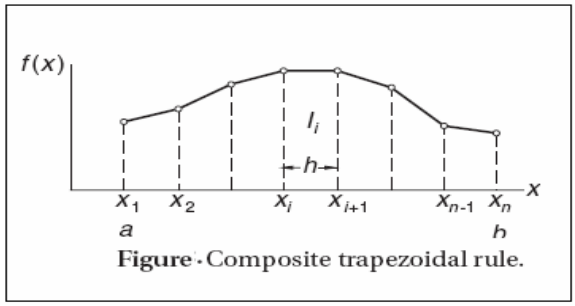


Figure . Composite trapezoidal rule.

The composite trapezoidal rule.

$$I = \sum_{i=1}^{n-1} I_i = [f(x_1) + 2f(x_2) + 2f(x_3) + \cdots + 2f(x_{n-1}) + f(x_n)]\frac{h}{2}$$

Example

Suppose we wished to integrate the function trabulated the table below for $f(x)=e^{x}$ over the interval from x=1.8 to x=3.4 using n=8

$$Am=\int_{a}^{b} f(x)dx=\int_{1.8}^{3.4} (e^{x})dx$$

x	1.6	1.8	2	2.2	2.4	2.6	2.8	3	3.2	3.4	3.6	3.8
f(x)	4.953	6.050	7.389	9.025	11.023	13.464	16.445	20.086	24.533	29.964	36.598	44.701

Solution

```
%---Trapezoidal Rule---------------
clc
a=1.8;
b=3.4;
h=0.2;
n=(b-a)/h
f=0;
x=2;
for i=1:n;
    %c=a+(i-1/2)*h;
    %f=f+(c^2+1);
    f=(f+exp(x))
    x=x+h;
end
Am_approx=h/2*(exp(a)+2*f+exp(b))
syms t
Am_exact=int(exp(t),1.8,3.4)
error=Am_exact-Am_approx
E_t=(error/(Am_approx+error))*100
```

```
E_a=((Am_approx-Am_exact)/Am_approx)*100
%--------------------------------
```

2- *Simpson's 1/3 rule*

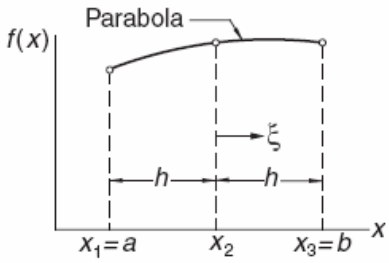


Figure . Simpson's 1/3 rule.

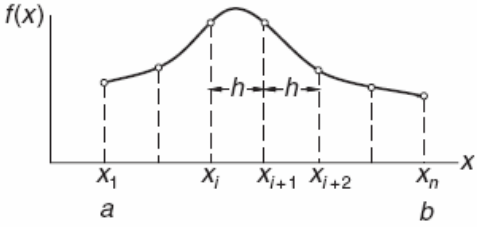


Figure. Composite Simpson's 1/3 rule.

The composite Simpson's 1⁄3 rule

$$\int_a^b f(x)\,dx \approx I = [f(x_1) + 4f(x_2) + 2f(x_3) + 4f(x_4) + \cdots + 2f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)]\frac{h}{3}$$

Example

Suppose we wished to integrate the function using Simpson's 1⁄3 rule and Simpson's 3⁄8 rule the table below for $f(x)=e^x$ over the interval from x=1.8 to x=3.4 using n=8

$$Am=\int_{a}^{b} f(x)dx=\int_{1.8}^{3.4} (e^x)dx$$

x	1.6	1.8	2	2.2	2.4	2.6	2.8	3	3.2	3.4	3.6	3.8
f(x)	4.953	6.050	7.389	9.025	11.023	13.464	16.445	20.086	24.533	29.964	36.598	44.701

Solution

```
%---Simpson's 1/3 rule -------------------------------
clc
   a=1.8;
   b=3.4;
   h=0.2;
   n=(b-a)/h
   f=0;
   m=0;
for x=2:(h+h):3.2;
      f=(f+exp(x));
end

for x=2.2:(h+h):3;
      m=(m+exp(x));
end
  Am_approx=h/3*(exp(a)+4*f+2*m+exp(b))
syms t
  Am_exact=int(exp(t),1.8,3.4)
  error=Am_exact-Am_approx
```

```
  E_t=(error/(Am_approx+error))*100
  E_a=((Am_approx-Am_exact)/Am_approx)*100
%-------------------------------------------------
```

3-Simpson's 3/8 rule

The composite Simpson's 3⁄8 rule

$$\int_a^b f(x)\,dx \approx I = [f(x_1) + 3f(x_2) + 3f(x_3) + 2f(x_4) + \cdots + 3f(x_{n-2}) + 3f(x_{n-1}) + f(x_n)]\frac{3h}{8}$$

```
%--------------------Simpson's 3/8 rule --------------
clc
   a=1.8;
   b=3.4;
   h=0.2;
   n=(b-a)/h;
   f=0;
   m=0;
%-------------------------------------------------
    for x=2:h:2+h;
        f=f+exp(x)
    end
%-------------------------------------------------
    x=x+h;
    m=exp(x);
%-------------------------------------------------
     for x=2.6:h:2.6+h;
        f=f+exp(x);
     end
%-------------------------------------------------
    x=x+h;
    m=m+exp(x);
    x=x+h;
    f=f+exp(x);
%-------------------------------------------------
    Am_approx=((3*h)/8)*(exp(a)+3*f+2*m+exp(b))
%-------------------------------------------------
syms t
   Am_exact=int(exp(t),1.8,3.4)
   error=Am_exact-Am_approx
   E_t=(error/(Am_approx+error))*100
   E_a=((Am_approx-Am_exact)/Am_approx)*100
%-------------------------------------------------
```

```
%--------------------Simpson's 3/8 rule --------------
clc
   a=1.8;b=3.4;h=0.2;n=(b-a)/h;f=0;m=0;
%-------------------------------------------------------
   for x=2:h:3.2;
      switch x
        case {2,2.2}
             f=f+exp(x)
        case {2.4}
             m=exp(x);
        case {2.6,2.8}
            f=f+exp(x);
        case {3}
            m=m+exp(x);
        otherwise
           f=f+exp(x);
      end
   end
%-------------------------------------------------------
    Am_approx=((3*h)/8)*(exp(a)+3*(f)+2*(m)+exp(b))
%-------------------------------------------------------
syms t
   Am_exact=int(exp(t),1.8,3.4)
   pretty(Am_exact)
   error=Am_exact-Am_approx
   pretty(error)
   E_t=(error/(Am_approx+error))*100
   pretty(E_t)
   E_a=((Am_approx-Am_exact)/Am_approx)*100
   pretty(E_a)
%-------------------------------------------------------
```

4-Lagrange Interpolating Polynomial Method

Lagrange's interpolation method uses the formula

$$f(x) = \frac{(x-x_1)(x-x_2)\dots(x-x_n)}{(x_0-x_1)(x_0-x_2)\dots(x_0-x_n)}f(x_0) + \frac{(x-x_0)(x-x_2)\dots(x-x_n)}{(x_1-x_0)(x_1-x_2)\dots(x_1-x_n)}f(x_1)$$

$$+ \frac{(x-x_0)(x-x_1)\dots(x-x_{n-1})}{(x_n-x_0)(x_n-x_2)\dots(x_n-x_{n-1})}f(x_n)$$

EXAMPLE

Given the data points

x	0	2	3
y	7	11	28

use Lagrange's method to determine y at $x = 1$.

Solution

$$\ell_1 = \frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)} = \frac{(1-2)(1-3)}{(0-2)(0-3)} = \frac{1}{3}$$

$$\ell_2 = \frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)} = \frac{(1-0)(1-3)}{(2-0)(2-3)} = 1$$

$$\ell_3 = \frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)} = \frac{(1-0)(1-2)}{(3-0)(3-2)} = -\frac{1}{3}$$

$$y = y_1\ell_1 + y_2\ell_2 + y_3\ell_3 = \frac{7}{3} + 11 - \frac{28}{3} = 4$$

```
%----------Lagrange's interpolation method ---------
clc
x=1;
%syms x
%-----------------------------------------------------
x1=0;
x2=2;
x3=3;
%-----------------------------------------------------
y0=7;
y1=11;
y2=28;
%-----------------------------------------------------
l0=((x-x2)*(x-x3))/((x1-x2)*(x1-x3))
l1=((x-x1)*(x-x3))/((x2-x1)*(x2-x3))
l2=((x-x1)*(x-x2))/((x3-x1)*(x3-x2))
%-----------------------------------------------------
y=y0*l0+y1*l1+y2*l2
%-----------------------------------------------------
```

Example 2

Construct the polynomial interpolating the data by using Lagrange polynomials

X	1	1/2	3
F(x)	3	-10	2

Solution

```
%-----------Lagrange's interpolation method ---------
clc
syms x
%-----------------------------------------------------
x1=1;
x2=0.5;
x3=3;
%-----------------------------------------------------
y0=3;
y1=-10;
y2=2;
%-----------------------------------------------------
l0=((x-x2)*(x-x3))/((x1-x2)*(x1-x3))
l1=((x-x1)*(x-x3))/((x2-x1)*(x2-x3))
l2=((x-x1)*(x-x2))/((x3-x1)*(x3-x2))
%-----------------------------------------------------
y=y0*l0+y1*l1+y2*l2;
collect(y)
%-----------------------------------------------------

%-------------Lagrange's interpolation method--------

clc
syms x
p=0;
s=[1 1/2 3];
f=[3 -10 2];
n=length(s);
for i=1:n;
    l=1;
    for j=1:n;
        if (i~=j);
            l=((x-s(j))/(s(i)-s(j)))*l;
        end
      end
  p=l.*f(i)+p;
end
p=collect(p)

%-----------------------------------------------------
```

Example 2

Construct the polynomial interpolating the data by using Lagrange polynomials

X	1	1/2	3
F(x)	3	-10	2

Solution

```
%-------------Lagrange's interpolation method--------
clc
x=input(' enter value of x:')
p=0;
s=[1 1/2 3];
f=[3 -10 2];
n=length(s);
for i=1:n;
    l=1;
    for j=1:n;
        if (i~=j);
            l=((x-s(j))/(s(i)-s(j)))*l;
        end
      end
  p=l.*f(i)+p;
end
p;
fprintf('\n p(%3.3f)=%5.4f',x,p)
%-----------------------------------------------------
syms x
p=0;
for i=1:n;
    l=1;
    for j=1:n;
        if (i~=j);
            l=((x-s(j))/(s(i)-s(j)))*l;
        end
      end
  p=l.*f(i)+p;
end
p=collect(p)
%-----------------------------------------------------
```

p = -283/10 -53/5 *x^2 + 419/10 *x

enter value of x:5

x =5

p(5.000)=-83.8000

```
%-----------------------------------------------------
```

Example 3

Find the area by lagrange polynomial using 3 nodes

X	1.8	2.6	3.4
F(x)	6.04964	13.464	29.964

Solution

```
%-----------Lagrange's interpolation method ---------
clc
syms x
%-------------------------------------------------------
x1=1.8;
x2=2.6;
x3=3.4;
%-------------------------------------------------------
F0=6.04964;
F1=13.464;
F2=29.964;
%-------------------------------------------------------
l0=((x-x2)*(x-x3))/((x1-x2)*(x1-x3))
A0=int(l0,1.8,3.4)
l1=((x-x1)*(x-x3))/((x2-x1)*(x2-x3))
A1=int(l1,1.8,3.4)
l2=((x-x1)*(x-x2))/((x3-x1)*(x3-x2))
A2=int(l2,1.8,3.4)
%-------------------------------------------------------
F=F0*A0+F1*A1+F2*A2
collect(F)
%-------------------------------------------------------
```

```
%-------------Lagrange's interpolation method---
clc
syms x
format long
p=0;
s=[1.8 2.6 3.4];
f=[6.04964 13.464 29.964];
n=length(s);
for i=1:n;
    l=1;
    for j=1:n;
        if (i~=j);
            l=((x-s(j))/(s(i)-s(j)))*l;
        end
    end
  A=int(l,s(1),s(n))
  p=A*f(i)+p;
end
 p
%-------------------------------------------------
```

5-Mid Point Rule

Example

Find the mid point approximation for

$$Am=\int_{a}^{b} f(x)dx=\int_{-1}^{2}(x\text{^}2+1)dx$$

using n=6

Solution

```
%---Mid Point Rule---------------
clc
a=-1;
b=2;
n=6;
h=(b-a)/n;
f=0;
for i=1:n;
    c=a+(i-1/2)*h;
    f=f+(c^2+1);
end
Am=h*f
%--------------------------------
```

6- Taylor series

A function $f(x)$ which possesses all derivatives up to order n at a point $x = x_0$ can be expanded in a *Taylor series* as

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \ldots + \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n$$

If $x_0 = 0$, reduces to

$$f(x) = f(0) + f'(0)x + \frac{f''(0)}{2!}x^2 + \ldots + \frac{f^{(n)}(0)}{n!}x^n$$

Example

Compute the first three terms of the Taylor series expansion for the function

$$y = f(x) = \tan x$$

at $a = \pi/4$.

Solution:

The Taylor series expansion about point a is given by

$$f_n(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \ldots$$

and since we are asked to compute the first three terms, we must find the first and second derivatives of $f(x) = \tan x$.

From math tables, $\frac{d}{dx}\tan x = \sec^2 x$, so $f'(x) = \sec^2 x$. To find $f''(x)$ we need to find the first derivative of $\sec^2 x$, so we let $z = \sec^2 x$. Then, using $\frac{d}{dx}\sec x = \sec x \cdot \tan x$, we get

$$\frac{dz}{dx} = 2\sec x \frac{d}{dx}\sec x = 2\sec x(\sec x \cdot \tan x) = 2\sec^2 x \cdot \tan x$$

Next, using the trigonometric identity

$$\sec^2 x = \tan^2 x + 1$$

and by substitution , we get,

$$\frac{dz}{dx} = f''(x) = 2(\tan^2 x + 1)\tan x$$

Now, at point $a = \pi/4$ we have:

$$f(a) = f\left(\frac{\pi}{4}\right) = \tan\left(\frac{\pi}{4}\right) = 1 \quad f'(a) = f'\left(\frac{\pi}{4}\right) = 1 + 1 = 2 \quad f''(a) = f''\left(\frac{\pi}{4}\right) = 2(1^2 + 1)1 = 4$$

and by substitution into (6.125),

$$f_n(x) = 1 + 2\left(x - \frac{\pi}{4}\right) + 2\left(x - \frac{\pi}{4}\right)^2 + \ldots$$

We can also obtain a Taylor series expansion with the MATLAB **taylor(f,n,a)** function where **f** is a symbolic expression, **n** produces the first n terms in the series, and **a** defines the Taylor approximation about point a.

The following MATLAB script computes the first 8 terms of the Taylor series expansion of $y = f(x) = \tan x$ about $a = \pi/4$.

```
%-------------------- Taylor series -------------
clc
    a=pi/4;
    sym x
    y=tan(x);
    z=taylor(y,8,a);
    pretty(z)
%-----------------------------------------------
```

Example

Express the function

$$y = f(t) = e^t$$

in a Maclaurin's series.

Solution:

A Maclaurin's series has the form, that is,

$$f(x) = f(0) + f'(0)x + \frac{f''(0)}{2!}x^2 + \ldots + \frac{f^{(n)}(0)}{n!}x^n$$

For this function, we have $f(t) = e^t$ and thus $f(0) = 1$. Since all derivatives are e^t, then, $f'(0) = f''(0) = f'''(0) = \ldots = 1$ and therefore,

$$f_n(t) = 1 + t + \frac{t^2}{2!} + \frac{t^3}{3!} + \ldots$$

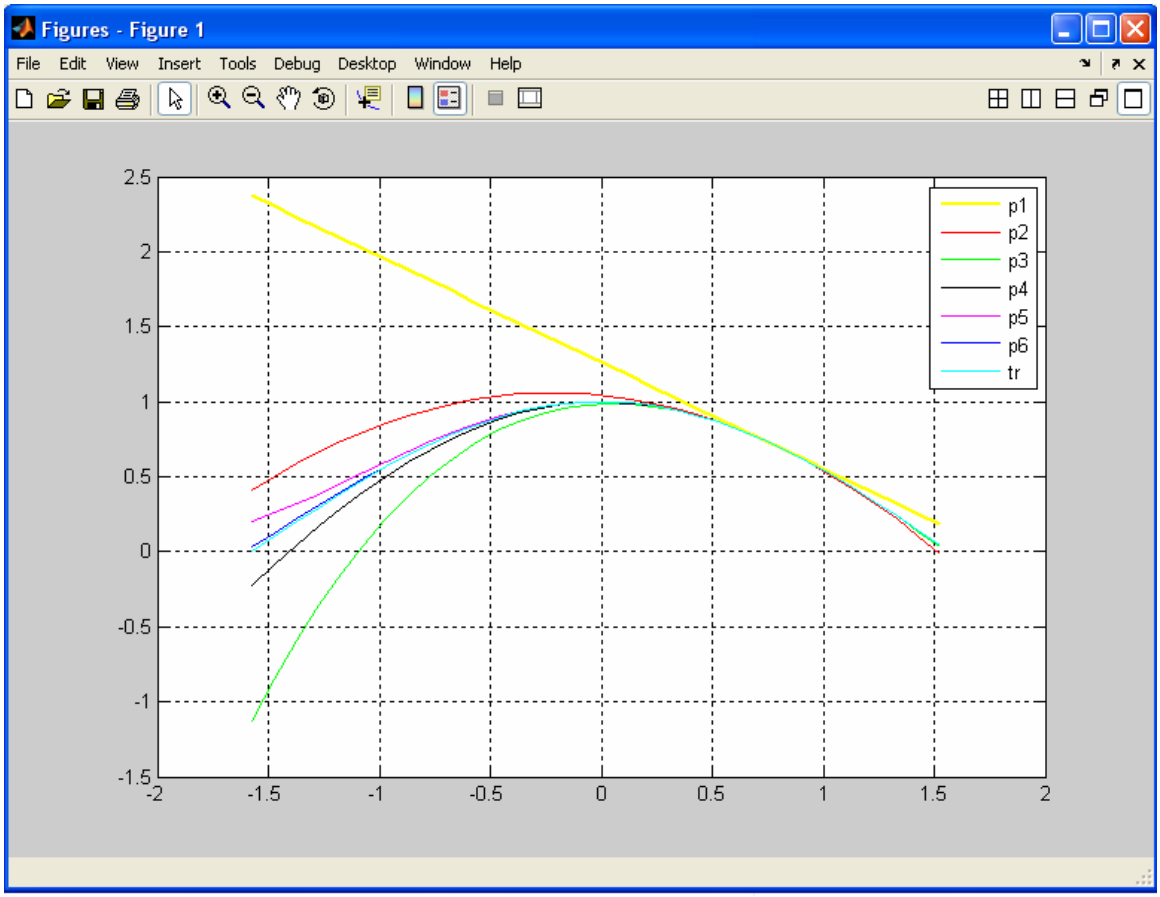
MATLAB displays the same result.

```
%-------------------- Taylor series -------------
clc
  syms t
  fn=taylor(exp(t));
  pretty(fn)
%-----------------------------------------------
```

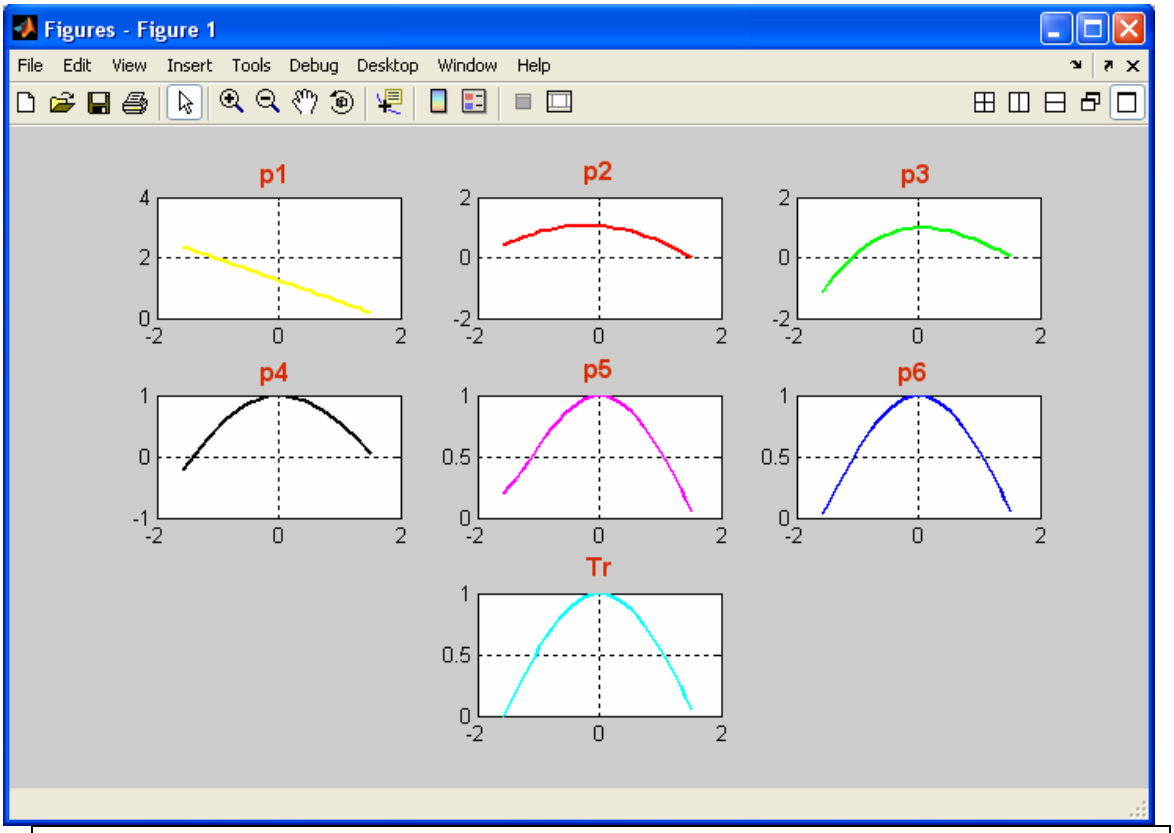
```
%-------------------------------------------------------------------
 clc
 clear
%syms x
%p2=taylor(cos(x),7,pi/4)
format long
x=pi/3
true=cos(pi/3)
p1=1/2*2^(1/2)-1/2*2^(1/2)*(x-1/4*pi)
et_1=((true-p1)/true)*100
p2=1/2*2^(1/2)-1/2*2^(1/2)*(x-1/4*pi)-1/4*2^(1/2)*(x-1/4*pi)^2
et_2=((true-p2)/true)*100
p3=1/2*2^(1/2)-1/2*2^(1/2)*(x-1/4*pi)-1/4*2^(1/2)*(x-
1/4*pi)^2+1/12*2^(1/2)*(x-1/4*pi)^3
et_3=((true-p3)/true)*100
p4=1/2*2^(1/2)-1/2*2^(1/2)*(x-1/4*pi)-1/4*2^(1/2)*(x-
1/4*pi)^2+1/12*2^(1/2)*(x-1/4*pi)^3+1/48*2^(1/2)*(x-1/4*pi)^4
et_4=((true-p4)/true)*100
p5=1/2*2^(1/2)-1/2*2^(1/2)*(x-1/4*pi)-1/4*2^(1/2)*(x-
1/4*pi)^2+1/12*2^(1/2)*(x-1/4*pi)^3+1/48*2^(1/2)*(x-1/4*pi)^4-
1/240*2^(1/2)*(x-1/4*pi)^5
et_5=((true-p5)/true)*100
p6=1/2*2^(1/2)-1/2*2^(1/2)*(x-1/4*pi)-1/4*2^(1/2)*(x-
1/4*pi)^2+1/12*2^(1/2)*(x-1/4*pi)^3+1/48*2^(1/2)*(x-1/4*pi)^4-
1/240*2^(1/2)*(x-1/4*pi)^5-1/1440*2^(1/2)*(x-1/4*pi)^6
et_6=((true-p6)/true)*100
%-------------------------------------------------------------------
```

```
x =
   1.047197551196598
true =
   0.500000000000000
p1 =
   0.521986658763282
et_1 =
  -4.397331752656441
p2 =
   0.497754491403425
et_2 =
   0.449101719315004
p3 =
   0.499869146930044
et_3 =
   0.026170613991194
p4 =
   0.500007550810613
et_4 =
  -0.001510162122553
p5 =
   0.500000304000373
et_5 =
   -6.080007448616696e-005
p6 =
   0.499999987798625
et_6 =
    2.440274993187329e-006
```

```
%-------------------------------------------------------------------
clc
clear
x=-pi/2:0.1:pi/2;
p1=1/2*2.^(1/2)-1/2*2.^(1/2)*(x-1/4*pi);
p2=1/2*2.^(1/2)-1/2*2.^(1/2)*(x-1/4*pi)-1/4*2.^(1/2)*(x-1/4*pi).^2;
p3=1/2*2.^(1/2)-1/2*2.^(1/2)*(x-1/4*pi)-1/4*2.^(1/2)*(x-
1/4*pi).^2+1/12*2.^(1/2)*(x-1/4*pi).^3;
p4=1/2*2.^(1/2)-1/2*2.^(1/2)*(x-1/4*pi)-1/4*2.^(1/2)*(x-
1/4*pi).^2+1/12*2.^(1/2)*(x-1/4*pi).^3+1/48*2.^(1/2)*(x-1/4*pi).^4;
p5=1/2*2.^(1/2)-1/2*2.^(1/2)*(x-1/4*pi)-1/4*2.^(1/2)*(x-
1/4*pi).^2+1/12*2.^(1/2)*(x-1/4*pi).^3+1/48*2.^(1/2)*(x-1/4*pi).^4-
1/240*2.^(1/2)*(x-1/4*pi).^5;
p6=1/2*2.^(1/2)-1/2*2.^(1/2)*(x-1/4*pi)-1/4*2.^(1/2)*(x-
1/4*pi).^2+1/12*2.^(1/2)*(x-1/4*pi).^3+1/48*2.^(1/2)*(x-1/4*pi).^4-
1/240*2.^(1/2)*(x-1/4*pi).^5-1/1440*2.^(1/2)*(x-1/4*pi).^6;
tr=cos(x);
plot(x,p1,'y'),pause(1)
hold on
plot(x,p2,'r'),pause(1)
hold on
plot(x,p3,'g'),pause(1)
hold on
plot(x,p4,'k'),pause(1)
hold on
plot(x,p5,'m'),pause(1)
hold on
plot(x,p6,'b'),pause(1)
hold on
plot(x,tr,'c'),pause(1)
hold on
legend('p1','p2','p3','p4','p5','p6','tr')
grid
%-------------------------------------------------------------------
```


Figures - Figure 1
File Edit View Insert Tools Debug Desktop Window Help
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
p1
p2
p3
p4
p5
p6
tr

```
%-------------------------------------------------------------------
clc
clear
x=-pi/2:0.1:pi/2;
p1=1/2*2.^(1/2)-1/2*2.^(1/2)*(x-1/4*pi);
p2=1/2*2.^(1/2)-1/2*2.^(1/2)*(x-1/4*pi)-1/4*2.^(1/2)*(x-1/4*pi).^2;
p3=1/2*2.^(1/2)-1/2*2.^(1/2)*(x-1/4*pi)-1/4*2.^(1/2)*(x-
1/4*pi).^2+1/12*2.^(1/2)*(x-1/4*pi).^3;
p4=1/2*2.^(1/2)-1/2*2.^(1/2)*(x-1/4*pi)-1/4*2.^(1/2)*(x-
1/4*pi).^2+1/12*2.^(1/2)*(x-1/4*pi).^3+1/48*2.^(1/2)*(x-1/4*pi).^4;
p5=1/2*2.^(1/2)-1/2*2.^(1/2)*(x-1/4*pi)-1/4*2.^(1/2)*(x-
1/4*pi).^2+1/12*2.^(1/2)*(x-1/4*pi).^3+1/48*2.^(1/2)*(x-1/4*pi).^4-
1/240*2.^(1/2)*(x-1/4*pi).^5;
p6=1/2*2.^(1/2)-1/2*2.^(1/2)*(x-1/4*pi)-1/4*2.^(1/2)*(x-
1/4*pi).^2+1/12*2.^(1/2)*(x-1/4*pi).^3+1/48*2.^(1/2)*(x-1/4*pi).^4-
1/240*2.^(1/2)*(x-1/4*pi).^5-1/1440*2.^(1/2)*(x-1/4*pi).^6;
tr=cos(x);
subplot(331)
plot(x,p1,'y'),pause(4)
title('p1')
grid on
subplot(332)
plot(x,p2,'r'),pause(4)
title('p2')
grid on
subplot(333)
plot(x,p3,'g'),pause(4)
title('p3')
grid on
subplot(334)
plot(x,p4,'k'),pause(4)
title('p4')
grid on
subplot(335)
plot(x,p5,'m'),pause(4)
title('p5')
grid on
subplot(336)
plot(x,p6,'b'),pause(4)
title('p6')
grid on
subplot(338)
plot(x,tr,'c')
title('Tr')
grid on
%-------------------------------------------------------------------
```



Example

Find first and second derivatives for F(x)=x^2+2x+2

Solution

```
%------To find first and second derivatives of Pn(x)-----
--
clc
a=[1 2 3];
syms x
p=a(1);
for i=1;
    p=a(i+1)+x*p;
end
disp('First derivative')
  p2=p+x*diff(p)
disp('Second derivative')
  p22=diff(p2)
```

```
%-----------------------------------------------
First derivative

  p2 =

       2+2*x

Second derivative

  p22 =

        2
```

Example

P4(x)=3x^4-10x^3-48x^2-2x+12 at r=6 deflate the polynomial with Horners algorithm Find P3(x).

Solution

```
%-------------Horner alogorithm------------------
clc
a=[3 -10 -48 -2 12];
r=6;
b(1)=a(1);
p=0;
n=length(a);
for i=2:n;
    b(i)=a(i)+r.*b(i-1);
end
syms x
for i=1:n;
    p=p+b(i)*x^(4-i);
end
disp('P3(x)=')
p
%-----------------------------------------------
```

```
P3(x)=
        3*x^3+8*x^2-2
```

Numerical Differentiation

1-Finite Difference Approximations

The derivation of the finite difference approximations for the derivatives of $f(x)$ are based on forward and backward Taylor series expansions of $f(x)$ about x, such as

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2!}f''(x) + \frac{h^3}{3!}f'''(x) + \frac{h^4}{4!}f^{(4)}(x) + \cdots \quad \text{(a)}$$

$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2!}f''(x) - \frac{h^3}{3!}f'''(x) + \frac{h^4}{4!}f^{(4)}(x) - \cdots \quad \text{(b)}$$

$$f(x+2h) = f(x) + 2hf'(x) + \frac{(2h)^2}{2!}f''(x) + \frac{(2h)^3}{3!}f'''(x) + \frac{(2h)^4}{4!}f^{(4)}(x) + \cdots \quad \text{(c)}$$

$$f(x-2h) = f(x) - 2hf'(x) + \frac{(2h)^2}{2!}f''(x) - \frac{(2h)^3}{3!}f'''(x) + \frac{(2h)^4}{4!}f^{(4)}(x) - \cdots \quad \text{(d)}$$

We also record the sums and differences of the series:

$$f(x+h) + f(x-h) = 2f(x) + h^2 f''(x) + \frac{h^4}{12}f^{(4)}(x) + \cdots \quad \text{(e)}$$

$$f(x+h) - f(x-h) = 2hf'(x) + \frac{h^3}{3}f'''(x) + \cdots \quad \text{(f)}$$

$$f(x+2h) + f(x-2h) = 2f(x) + 4h^2 f''(x) + \frac{4h^4}{3}f^{(4)}(x) + \cdots \quad \text{(g)}$$

$$f(x+2h) - f(x-2h) = 4hf'(x) + \frac{8h^3}{3}f'''(x) + \cdots \quad \text{(h)}$$

First Central Difference Approximations

The solution of Eq. (f) for $f'(x)$ is

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} - \frac{h^2}{6}f'''(x) - \cdots$$

Keeping only the first term on the right-hand side, we have

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + \mathcal{O}(h^2)$$

which is called the *first central difference approximation* for $f'(x)$. The term $\mathcal{O}(h^2)$ reminds us that the truncation error behaves as h^2.

From Eq. (e) we obtain

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + \frac{h^2}{12}f^{(4)}(x) + \cdots$$

or

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + \mathcal{O}(h^2)$$

Central difference approximations for other derivatives can be obtained from Eqs. (a)–(h) in a similar manner. For example, eliminating $f'(x)$ from Eqs. (f) and (h) and solving for $f'''(x)$ yield

$$f'''(x) = \frac{f(x+2h) - 2f(x+h) + 2f(x-h) - f(x-2h)}{2h^3} + \mathcal{O}(h^2)$$

The approximation

$$f^{(4)}(x) = \frac{f(x+2h) - 4f(x+h) + 6f(x) - 4f(x-h) + f(x-2h)}{h^4} + \mathcal{O}(h^2)$$

First Noncentral Finite Difference Approximations

These expressions are called *forward* and *backward* finite difference approximations.

Noncentral finite differences can also be obtained from Eqs. (a)–(h). Solving Eq. (a) for $f'(x)$ we get

$$f'(x) = \frac{f(x+h) - f(x)}{h} - \frac{h}{2}f''(x) - \frac{h^2}{6}f'''(x) - \frac{h^3}{4!}f^{(4)}(x) - \cdots$$

Keeping only the first term on the right-hand side leads to the *first forward difference approximation*

$$f'(x) = \frac{f(x+h) - f(x)}{h} + \mathcal{O}(h)$$

Similarly, Eq. (b) yields the *first backward difference approximation*

$$f'(x) = \frac{f(x) - f(x-h)}{h} + \mathcal{O}(h)$$

Note that the truncation error is now $\mathcal{O}(h)$, which is not as good as the $\mathcal{O}(h^2)$ error in central difference approximations.

We can derive the approximations for higher derivatives in the same manner. For example, Eqs. (a) and (c) yield

$$f''(x) = \frac{f(x+2h) - 2f(x+h) + f(x)}{h^2} + \mathcal{O}(h)$$

Second Noncentral Finite Difference Approximations

Finite difference approximations of $\mathcal{O}(h)$ are not popular due to reasons that will be explained shortly. The common practice is to use expressions of $\mathcal{O}(h^2)$. To obtain noncentral difference formulas of this order, we have to retain more terms in the Taylor series. As an illustration, we will derive the expression for $f'(x)$. We start with Eqs. (a) and (c), which are

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f'''(x) + \frac{h^4}{24}f^{(4)}(x) + \cdots$$

$$f(x+2h) = f(x) + 2hf'(x) + 2h^2 f''(x) + \frac{4h^3}{3}f'''(x) + \frac{2h^4}{3}f^{(4)}(x) + \cdots$$

We eliminate $f''(x)$ by multiplying the first equation by 4 and subtracting it from the second equation. The result is

$$f(x+2h) - 4f(x+h) = -3f(x) - 2hf'(x) + \frac{2h^2}{3}f'''(x) + \cdots$$

Therefore,

$$f'(x) = \frac{-f(x+2h) + 4f(x+h) - 3f(x)}{2h} + \frac{h^2}{3}f'''(x) + \cdots$$

or

$$f'(x) = \frac{-f(x+2h) + 4f(x+h) - 3f(x)}{2h} + \mathcal{O}(h^2)$$

This Equation is called the *second forward finite difference approximation*.

EXAMPLE

Use forward difference approximations of oh to estimate the first

% derivative of

fx = -0.1.*x.^4-0.15.*x.^3-0.5.*x.^2-0.25.*x+1.2

solution

```
%-----------------------------------------------------------------
% Use forward difference approximations to estimate the first
% derivative of fx=-0.1.*x.^4-0.15.*x.^3-0.5.*x.^2-0.25.*x+1.2
clc
h=0.5;
x=0.5;
x1=x+h
fxx=[-0.1 -0.15 -0.5 -0.25 1.2]
fx=polyval(fxx,x)
fx1=polyval(fxx,x1)
tr_va=polyval(polyder(fxx),0.5)
fda=(fx1-fx)/h
et=(tr_va1-fda)/(tr_va1)*100
%-----------------------------------------------------------------
```

EXAMPLE

Comparison of numerical derivative for backward difference and central difference method with true derivative and with standard deviation of 0.025

x = [0:pi/50:pi];

yn = sin(x)+0.025

True derivative=td=cos(x)

solution

```
%-----------------------------------------------------------
clc
% Comparison of numerical derivative algorithms
x = [0:pi/50:pi];
n = length(x);
% Sine signal with Gaussian random error
yn = sin(x)+0.025*randn(1,n);
% Derivative of noiseless sine signal
td = cos(x);
% Backward difference estimate noisy sine signal
dynb = diff(yn)./diff(x);
subplot(2,1,1)
plot(x(2:n),td(2:n),x(2:n),dynb,'o')
xlabel('x')
ylabel('Derivative')
axis([0 pi -2 2])
legend('True derivative','Backward difference')
% Central difference
dync = (yn(3:n)-yn(1:n-2))./(x(3:n)-x(1:n-2));
subplot(2,1,2)
plot(x(2:n-1),td(2:n-1),x(2:n-1),dync,'o')
xlabel('x')
ylabel('Derivative')
axis([0 pi -2 2])
legend('True derivative','Central difference')
%-----------------------------------------------------------
```

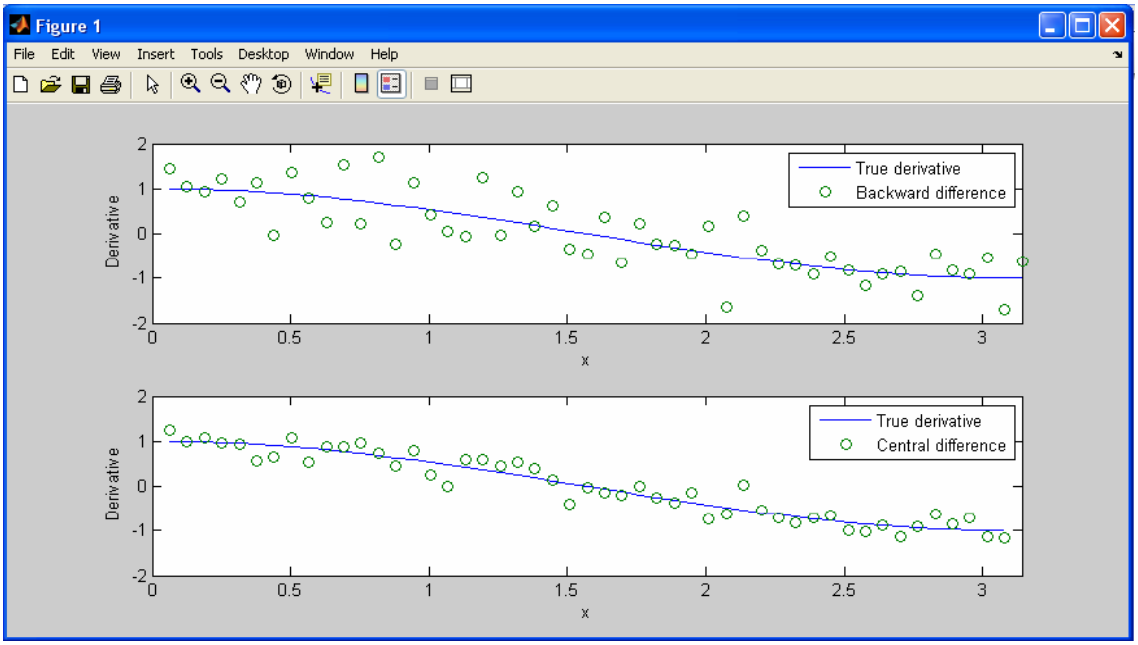


Figure. Comparison of backward difference and central difference methods

Example

Consider a `Divided Difference table` for points following

x	0	0.5	1	1.5
$f(x)$	0.0000	1.1487	2.7183	4.9811

Solution

x_k	$f[x_k]$	$f[x_k, x_{k+1}]$	$f[x_k, .., x_{k+2}]$	$f[x_k, .., x_{k+3}]$
0.0	0.0000			
		2.2974		
0.5	1.1487		0.8418	
		3.1392		0.36306
1.0	2.7183		1.3864	
		4.5256		
1.5	4.9811			

$$
\begin{aligned}
p(x) &= f(x_0) + x - x_0 f[x_0, x_1] + (x - x_0)(x - x_1) f[x_0, x_1, x_2] + (x - x_0)(x - x_1)(x - x_2) f[x_0, x_1, x_2, x_3] \\
&= 0.00 + (x - 0.0)2.2974 + (x - 0.0)(x - 0.5)0.8418 + (x - 0.0)(x - 0.5)(x - 1.0)0.36306 \\
&= 2.05803x + 0.29721x^2 + 0.36306x^3
\end{aligned}
$$

```
%----------------Divided Difference table algorithm-------------
clc
disp('******** divided difference table *********')
x=[2 4 6 8 10]
y=[4.077 11.084 30.128 81.897 222.62]
     f00=y(1);
    for i=1:4
        f1(i)=(y(i+1)-y(i))/(x(i+1)-x(i));
        f01=f1(1);
    end
 f1=[f1(1) f1(2) f1(3) f1(4)]
    for i=1:3
        f2(i)=(f1(i+1)-f1(i))/(x(i+2)-x(i));
        f02=f2(1);
    end
 f2=[f2(1) f2(2) f2(3)]
    for i=1:2
        f3(i)=(f2(i+1)-f2(i))/(x(i+3)-x(i));
        f03=f3(1);
    end
 f3=[f3(1) f3(2)]
    disp('************************************')
y=input('enter value of y:')
p4x=f00+((y-x(1))*f01)+((y-x(1))*(y-x(2))*f02+((y-x(1))*(y-
x(2))*f02))
fprintf('\np4(%3.3f)=%5.4f',y,p4x)
syms y
p4x=f00+((y-x(1))*f01)+((y-x(1))*(y-x(2))*f02+((y-x(1))*(y-
x(2))*f02))
%-----------------------------------------------------------------
f1 =  3.5035    9.5220   25.8845   70.3615

f2 =       1.5046    4.0906   11.1193

f3 =              0.4310    1.1714

p4x = -293/100+7007/2000*y+12037/4000*(y-2)*(y-4)
enter value of y:8
y = 8
p4(8.000)=97.3200

% ----------------------------------------------------------------
```

Example { H.W }

Find the divided differences (newten's Interpolating) for the data and compare with lagrange interpolating.

X	1	1/2	3
F(x)	3	-10	2

Solution

```
************** divided difference table *****************

f1 =

      26.000000000000000   4.800000000000000


f2 =

               -10.600000000000000

----------------Divided Difference table algorithm-------
----------------{ newtens Interpolating }----------------
enter value of y:5

p4(5.000)=-83.8000
px = -283/10-53/5*y^2+419/10*y


----------------------compare with ----------------------
-------------Lagranges interpolation method-------------
 enter value of x:5

 p(5.000)=-83.8000
 p = -283/10-53/5*m^2+419/10*m
```

```
%------------------------Solve H.W-------------------------------
%----------------Divided Difference table algorithm-------------
%----------------{ newten's Interpolating }--------------------
clc
disp('******** divided difference table *********')
x=[1 0.5 3];
y=[3 -10 2];
     f00=y(1);
    for i=1:2;
        f1(i)=(y(i+1)-y(i))/(x(i+1)-x(i));
        f01=f1(1);
    end
 f1=[f1(1) f1(2)]
    for i=1;
        f2(i)=(f1(i+1)-f1(i))/(x(i+2)-x(i));
        f02=f2(1);
    end
 f2=f2(1)
disp('----------------Divided Difference table algorithm-----------')
disp('----------------{ newtens Interpolating }--------------------')
y=input('enter value of y:');
px=f00+((y-x(1))*f01)+((y-x(1))*(y-x(2))*f02);
fprintf('\npx(%3.3f)=%5.4f',y,px)
syms y
px=f00+((y-x(1))*f01)+((y-x(1))*(y-x(2))*f02);
px=collect(px)
%---------------------compare with --------------------------
%-------------Lagrange's interpolation method-----------------
disp('----------------------compare with -------------------------')
disp('-------------Lagranges interpolation method-----------------')
m=input(' enter value of x:');
p=0;
s=[1 1/2 3];
f=[3 -10 2];
n=length(s);
for i=1:n;
    l=1;
    for j=1:n;
        if (i~=j);
            l=((m-s(j))/(s(i)-s(j)))*l;
        end
      end
  p=l.*f(i)+p;
end
p;
fprintf('\n p(%3.3f)=%5.4f',m,p)
syms m
p=0;
for i=1:n;
    l=1;
    for j=1:n;
        if (i~=j);
            l=((m-s(j))/(s(i)-s(j)))*l;
        end
      end
  p=l.*f(i)+p;
end
p=collect(p)
%-------------------------------------------------------------
```

Example { H.W }

Estimate the In(3) for

Xi	2	4	6
F(x)	In(2)	In(4)	In(6)

a) Linear Interpolation.
B) Quardratic Interpolation
compare between a&b

Solution

a)Linear Interpolation.

```
F1(x)=f(x0)+((f(x1)-f(x0))/(x1-x0))*(x-x0)
```

b)Quardratic Interpolation

```
f2(x)=b0+b1*(x-x0)+b2*(x-x0)*(x-x1)
```

b0= f(x0) = 0.693147180559945;
b1= (f(x1)-f(x0))/(x1-x0) = 0.346573590279973
b2= ((f(x2)-f(x1))/(x2-x1))-b1/(x2-x0) = -0.035960259056473;

---------a) Linear Interpolation----------

fx1 = 0.693147180559945-0.346573590279973 (*X*-2)

inter value x:3

fx1 = 1.039720770839918

---------b) Quardratic Interpolation----------

fx2 = 0.346573590279973*X*+(-0.035960259056473*X*+0.071920518112945)*(*X*-4)

inter value x:3

fx2 = 1.075681029896391

---------- compare between a&b----------
---------a) Linear Interpolation----------

Et1 =5.360536964281382 %

---------b) Quardratic Interpolation----------

Et2 = 2.087293124994937 %

Quardratic Interpolation is better than Linear Interpolation

```
%---------  Solve H.W----------------------------------------
%---------a) Linear Interpolation-------------------------
%---------b) Quardratic Interpolation---------------------
%---------  compare between a&b-------------------------
clc
x=input('inter value x:');
format long
xi=[2  4  6];
fx=[log(2)  log(4)  log(6)];
disp('---------a) Linear Interpolation----------------------')
fx1=fx(1)+((fx(2)-fx(1))/(xi(2)-xi(1)))*(x-xi(1))
disp('---------b) Quardratic Interpolation-----------------')
b0=fx(1);
b1=(fx(2)-fx(1))/(xi(2)-xi(1));
b2=(((fx(3)-fx(2))/(xi(3)-xi(2)))-b1)/(xi(3)-xi(1));
fx2=b0+b1*(x-xi(1))+b2*(x-xi(1))*(x-xi(2));
% pretty(fx2)%expand(fx2)%collect(fx2)
disp('---------  compare between a&b----------------------')
Tv=log(3);
disp('---------a) Linear Interpolation----------------------')
Et1=abs((Tv-fx1)/Tv)*100
disp('---------b) Quardratic Interpolation-----------------')
Et2=abs((Tv-fx2)/Tv)*100
if Et1>Et2;
   disp('Quardratic Interpolation is better than Linear Interpolation')
else
   disp('Linear Interpolation is better than Quardratic Interpolation')
end
syms x
disp('---------a) Linear Interpolation----------------------')
fx1=fx(1)+((fx(2)-fx(1))/(xi(2)-xi(1)))*(x-xi(1))
disp('---------b) Quardratic Interpolation-----------------')
b0=fx(1);
b1=(fx(2)-fx(1))/(xi(2)-xi(1));
b2=(((fx(3)-fx(2))/(xi(3)-xi(2)))-b1)/(xi(3)-xi(1));
fx2=b0+b1*(x-xi(1))+b2*(x-xi(1))*(x-xi(2))
```

The Bisection Method for Root Approximation

we can compute the midpoint x_m of the interval $x_1 \le x \le x_2$ with

$$x_m = \frac{x_1 + x_2}{2}$$

Knowing x_m, we can find $f(x_m)$. Then, the following decisions are made:

1. If $f(x_m)$ and $f(x_1)$ have the same sign, their product will be positive, that is, $f(x_m) \cdot f(x_1) > 0$. This indicates that x_m and x_1 are on the left side of the *x–axis* crossing as shown in Figure. In this case, we replace x_1 with x_m.

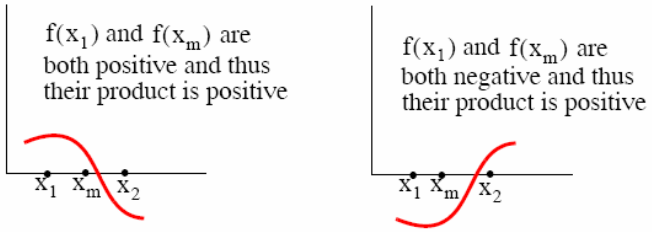


Figure . Sketches to illustrate the bisection method when $f(x_1)$ *and* $f(x_m)$ *have same sign*

2. If $f(x_m)$ and $f(x_1)$ have opposite signs, their product will be negative, that is, $f(x_m) \cdot f(x_1) < 0$. This indicates that x_m and x_2 are on the right side of the *x–axis* crossing as in Figure. In this case, we replace x_2 with x_m.

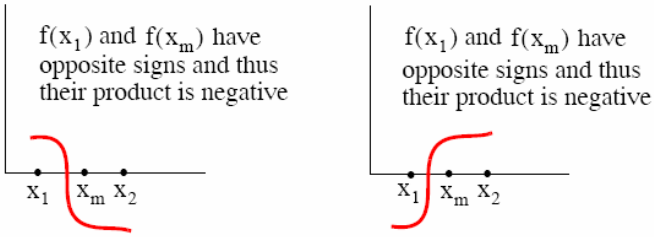


Figure . Sketches to illustrate the bisection method when $f(x_1)$ *and* $f(x_m)$ *have opposite signs*

After making the appropriate substitution, the above process is repeated until the root we are seeking has a specified tolerance. To terminate the iterations, we either:

a. specify a number of iterations

b. specify a tolerance on the error of $f(x)$

Example

Use the Bisection Method with MATLAB to approximate one of the roots of

$$y = f(x) = 3x^5 - 2x^3 + 6x - 8$$

by

a. by specifying ***16*** iterations, and using a for end loop MATLAB program

b. by specifying ***0.00001*** tolerance for f(x), and using a while end loop MATLAB program

Solution:

```
%--------------------------------------------------------------------
function y= funcbisect01(x);
y = 3 .* x .^ 5 - 2 .* x .^ 3 + 6 .* x - 8;
% We must not forget to type the semicolon at the end of the line
above;
% otherwise our script will fill the screen with values of y
%--------------------------------------------------------------------
```

call for function under name funcbisect01.m

```
%--------------------------------------------------------------------
clc
x1=1;
x2=2;
disp('-------------------------')
disp('   xm              fm')  % xm is the average of x1 and x2, fm is
f(xm)
disp('-------------------------')        % insert line under xm and
fm
for k=1:16;
f1=funcbisect01(x1); f2=funcbisect01(x2);
xm=(x1+x2) / 2; fm=funcbisect01(xm);
fprintf('%9.6f %13.6f \n', xm,fm)        % Prints xm and fm on same
line;
  if (f1*fm<0)
    x2=xm;
   else
    x1=xm;
  end
end
disp('-------------------------')
x=1:0.05:2;
y = 3 .* x .^ 5 - 2 .* x .^ 3 + 6 .* x - 8;
plot(x,y)
grid
%--------------------------------------------------------------------
```

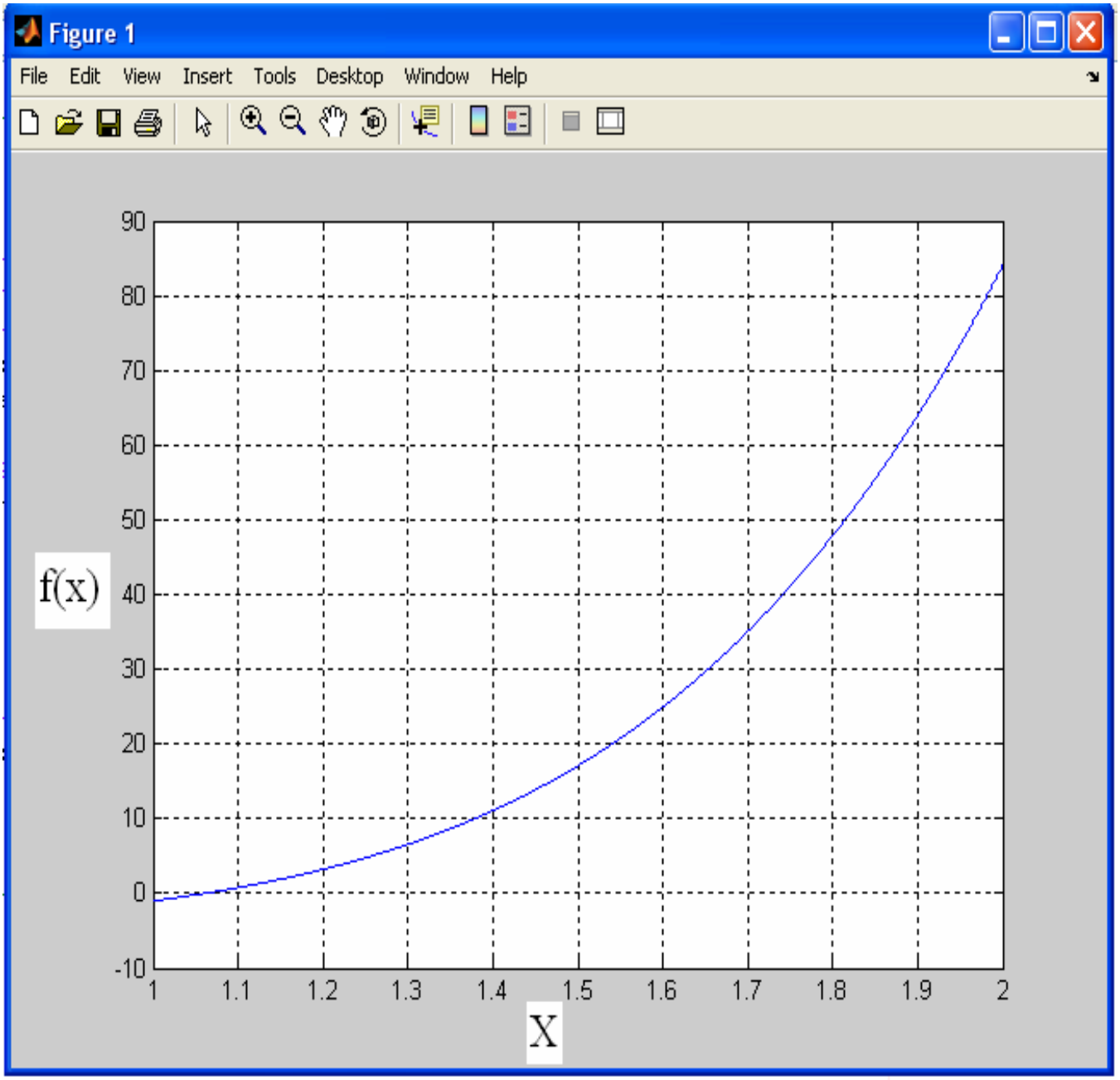


Figure 1
File Edit View Insert Tools Desktop Window Help
90
80
70
60
50
40
30
20
10
0
-10
f(x)
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
X

```
%------------------------------------------------------------------------
function y= funcbisect01(x);
y = 3 .* x .^ 5 - 2 .* x .^ 3 + 6 .* x - 8;
% We must not forget to type the semicolon at the end of the line
above;
% otherwise our script will fill the screen with values of y
%------------------------------------------------------------------------
```

call for function under name funcbisect01.m

```
%------------------------------------------------------------------------
%------------------------------------------------------------------------
clc
x1=1;
x2=2;
tol=0.00001;
disp('-------------------------')
disp(' xm                  fm');
disp('-------------------------')
while (abs(x1-x2)>2*tol);
f1=funcbisect01(x1);
f2=funcbisect01(x2);
xm=(x1+x2)/2;
fm=funcbisect01(xm);
fprintf('%9.6f %13.6f \n', xm,fm);
if (f1*fm<0);
x2=xm;
else
x1=xm;
end
end
disp('-------------------------')
%------------------------------------------------------------------------
-------------------------
 xm                fm
-------------------------
 1.500000     17.031250
 1.250000      4.749023
 1.125000      1.308441
 1.062500      0.038318
 1.031250     -0.506944
 1.046875     -0.241184
 1.054688     -0.103195
 1.058594     -0.032885
 1.060547      0.002604
 1.059570     -0.015168
 1.060059     -0.006289
 1.060303     -0.001844
 1.060425      0.000380
 1.060364     -0.000732
 1.060394     -0.000176
 1.060410      0.000102
-------------------------
```

Example

Use the Bisection Method with MATLAB to approximate one of the roots of (to find the roots of)

Y=f(x)= x.^3-10.*x.^2+5;

That lies in the interval (0.6,0.8) by specifying ***0.00001*** tolerance for f(x), and using a while end loop MATLAB program

Solution:

```
%-----------------------------------------------------------------------
function y= funcbisect01(x);
y = x.^3-10.*x.^2+5;
% We must not forget to type the semicolon at the end of the line
above;(% otherwise our script will fill the screen with values of y)
%-----------------------------------------------------------------------
```

call for function under name funcbisect01.m

```
%-----------------------------------------------------------------------
clc
x1=0.6; x2=0.8;tol=0.00001;
disp('-------------------------')
disp(' xm                  fm');
disp('-------------------------')
while (abs(x1-x2)>2*tol);
f1=funcbisect01(x1);
f2=funcbisect01(x2);
xm=(x1+x2)/2;
fm=funcbisect01(xm);
fprintf('%9.6f %13.6f \n', xm,fm);
if (f1*fm<0);
x2=xm;
else
x1=xm;
end
end
disp('-------------------------')
%-----------------------------------------------------------------------
```

```
--------------------------
 xm              fm
--------------------------
 0.700000     0.443000
 0.750000    -0.203125
 0.725000     0.124828
 0.737500    -0.037932
 0.731250     0.043753
 0.734375     0.002987
 0.735938    -0.017453
 0.735156    -0.007228
 0.734766    -0.002120
 0.734570     0.000434
 0.734668    -0.000843
 0.734619    -0.000204
 0.734595     0.000115
 0.734607    -0.000045
--------------------------
```

Newton–Raphson Method

The Newton–Raphson formula can be derived from the Taylor series expansion of $f(x)$ about x:

$$f(x_{i+1}) = f(x_i) + f'(x_i)(x_{i+1} - x_i) + O(x_{i+1} - x_i)^2 \qquad \text{(a)}$$

If x_{i+1} is a root of $f(x) = 0$, Eq. (a) becomes

$$0 = f(x_i) + f'(x_i)\,(x_{i+1} - x_i) + O(x_{i+1} - x_i)^2 \qquad \text{(b)}$$

Assuming that x_i is a close to x_{i+1}, we can drop the last term in Eq. (b) and solve for x_{i+1}. The result is the Newton–Raphson formula

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \qquad \text{(c)}$$

If x denotes the true value of the root, the error in x_i is $E_i = x - x_i$. It can be shown that if x_{i+1} is computed from Eq. (c), the corresponding error is

$$E_{i+1} = -\frac{f''(x_i)}{2f'(x_i)}E_i^2$$

indicating that the Newton–Raphson method converges *quadratically* (the error is the square of the error in the previous step). As a consequence, the number of significant figures is roughly doubled in every iteration, provided that x_i is close to the root.

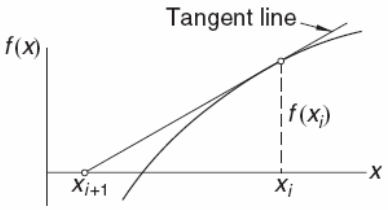


Figure (a) Graphical interpretation of the Newton–Raphson formula.

A graphical depiction of the Newton–Raphson formula is shown in Fig. (a) The formula approximates $f(x)$ by the straight line that is tangent to the curve at x_i. Thus x_{i+1} is at the intersection of the x-axis and the tangent line.

The algorithm for the Newton–Raphson method is simple: it repeatedly applies Eq. (c), starting with an initial value x_0, until the convergence criterion

$$|x_{i+1} - x_1| < \varepsilon$$

is reached, ε being the error tolerance. Only the latest value of x has to be stored. Here is the algorithm:

1. Let x be a guess for the root of $f(x) = 0$.
2. Compute $\Delta x = -f(x)/f'(x)$.
3. Let $x \leftarrow x + \Delta x$ and repeat steps 2-3 until $|\Delta x| < \varepsilon$.

EXAMPLE

A root of $f(x) = x^3 - 10x^2 + 5 = 0$ lies close to $x = 0.7$. Compute this root with the Newton–Raphson method.

Solution

The derivative of the function is $f'(x) = 3x^2 - 20x$, so that the Newton–Raphson formula in Eq. (c) is

$$x \leftarrow x - \frac{f(x)}{f'(x)} = x - \frac{x^3 - 10x^2 + 5}{3x^2 - 20x} = \frac{2x^3 - 10x^2 - 5}{x\,(3x - 20)}$$

It takes only two iterations to reach five decimal place accuracy:

$$x \leftarrow \frac{2(0.7)^3 - 10(0.7)^2 - 5}{0.7\,[3(0.7) - 20]} = 0.735\,36$$

$$x \leftarrow \frac{2(0.735\,36)^3 - 10(0.735\,36)^2 - 5}{0.735\,36\,[3(0.735\,36) - 20]} = 0.734\,60$$

Example

Use the Newton–Raphson Method to estimate the root of f(x)=e^(-x)-x, employing an initial guess of x0=0

$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$	$E_{i+1} = -\frac{f''(x_i)}{2f'(x_i)}E_i^2$

Solution

```
%------Newton-Raphson Method-------------------------
clc
x=[0];
tol=0.0000000007;
format long
for i=1:5;
      fx=exp(-x(i))-x(i);
      fxx=-exp(-x(i))-1 ;
      fxxx=exp(-x(i));
      x(i+1)=x(i)-(fx/fxx);
      T.V(i)=(abs((x(i+1)-x(i))/x(i+1)))*100;
end
for i=1:5;
      e(i)=x(6)-x(i);
      fxx=-exp(-x(6))-1 ;
      fxxx=exp(-x(6));
      e(i+1)=(-fxxx/2*fxx)*(e(i))^2;
end
if abs(x(i+1)-x(i))<tol
     disp(' enough to here')
     disp('------------')
     disp('  X(i+1) ')
     disp('-------------')
     x'
     disp('------------')
     disp('   T.V   ')
     disp('-------------')
     T.V'
     disp('------------')
     disp('   E(i+1)   ')
     disp('-------------')
     e'
     disp('-------------')
end
%-----------------------------------------------------
```

enough to here

X(i+1)

0
0.500000000000000
0.566311003197218
0.567143165034862
0.567143290409781
0.567143290409784

T.V

1.0e+002 *

1.000000000000000
0.117092909766624
0.001467287078375
0.000000221063919
0.000000000000005

E(i+1)

0.567143290409784
0.067143290409784
0.000832287212566
0.000000125374922
0.000000000000003
0.000000000000000

The secant Formula Method

A popular method of hand computation is the *secant formula* where the improved estimate of the root (x_{i+1}) is obtained by linear interpolation based two previous estimates (x_i and x_{i-1}):

$$x_{i+1} = x_i - \frac{x_i - x_{i-1}}{f(x_i) - f(x_{i-1})} f(x_i)$$

Example

Use the The secant Formula Method to estimate the root of f(x)=e^(-x)-x, employing an initial guess of x(i-1)=0 & x(0)=0

$$x_{i+1} = x_i - \frac{x_i - x_{i-1}}{f(x_i) - f(x_{i-1})} f(x_i)$$

Solution

```
%------The secant Formula Method ------------------------
clc
x=[0 1];
TV=0.567143290409784;
format long
for i=2:6;
    fx=exp(-x(i-1))-x(i-1);
    fxx=exp(-x(i))-x(i);
    x(i+1)=x(i)-((x(i)-x(i-1))*fxx)/(fxx-fx);
    E_T(i)=(abs((TV-x(i+1))/TV))*100;
end
    disp('------------')
    disp('  X(i+1) ')
    disp('-------------')
    x'
    disp('------------')
    disp('   E_T    ')
    disp('-------------')
    E_T'
    disp('------------')

%-----------------------------------------------------
```

```
----------------------
       X(i+1)
----------------------

                    0
   1.000000000000000
   0.612699836780282
   0.563838389161074
   0.567170358419745
   0.567143306604963
   0.567143290409705

---------------------
        E_T
---------------------

                    0
   8.032634281467328
   0.582727734700312
   0.004772693324310
   0.000002855570996
   0.000000000013997

--------------------
```

Example

Use N.R. Quadratically Method to estimate the multiple root of f(x)=x^3-5x^2+7x-3, initial guess of x(0)=0

$$x_{i+1} = x_i - \frac{f(x_i)\, f'(x_i)}{f'(x_i)^2 - f(x_i)\, f''(x_i)}$$

Solution

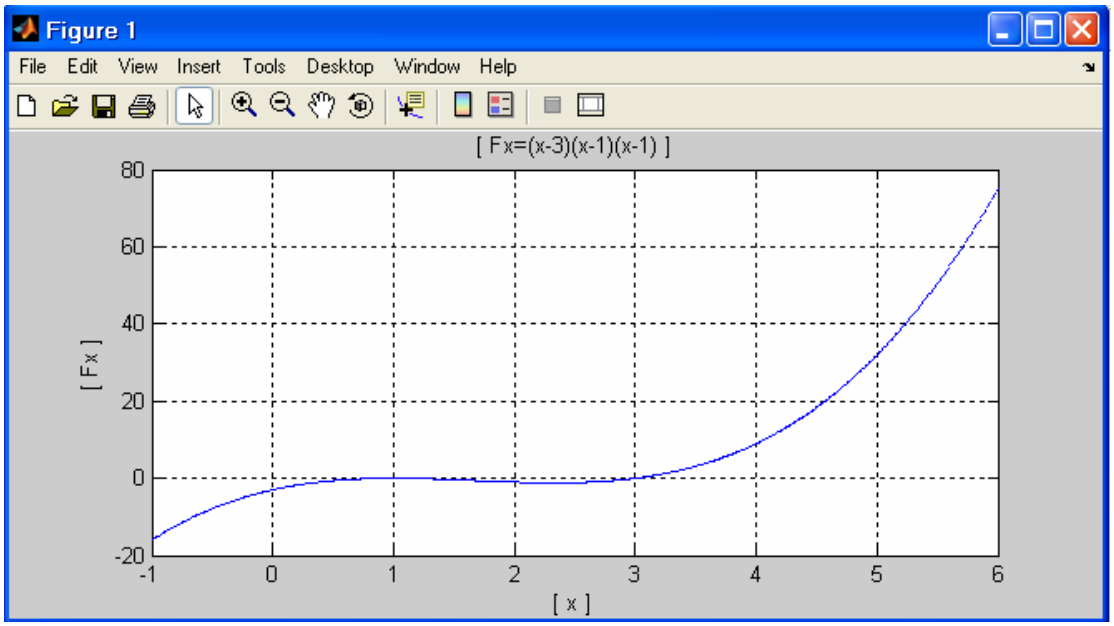
```
%------The N.R. Quadratically  Method   -----------------
clc
TV=1;
x=[0];
format long
for i=1:6;
    fx=x(i)^3-5*x(i)^2+7*x(i)-3
    fxx=3*x(i)^2-10*x(i)+7
    fxxx=6*x(i)-10
    x(i+1)=x(i)-(fx*fxx)/((fxx)^2-fx*fxxx);
    E_T(i)=(abs((TV-x(i+1))/TV))*100;
end
    disp('------------')
    disp('  X(i+1)  ')
    disp('-------------')
    x'
    disp('------------')
    disp('   E_T    ')
    disp('-------------')
    E_T'
    disp('------------')

%-------------------------------------------------
%------Multiple Roots---------
%--fx=(x-3)(x-1)(x-1)---------
clc
for x=-1:0.01:6;
   fx=x.^3-5.*x.^2+7.*x-3
   plot(x,fx)
   hold on
end
grid
title('(x-3)(x-1)(x-1)')
xlabel('x')
ylabel('fx')
%---------------------------
```

X(i+1)

0
1.105263157894737
1.003081664098603
1.000002381493816
1.000000000037312
1.000000000074625
1.000000000074625

E_T

10.526315789473696
0.308166409860333
0.000238149381548
0.000000003731215
0.000000007462475
0.000000007462475



Example

Use the Newton–Raphson Method to estimate the root of f(x)=x^3-5x^2+7x-3, initial guess of x(0)=4

$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$	$E_{i+1} = -\frac{f''(x_i)}{2f'(x_i)}E_i^2$

Solution

```
%------Newton-Raphson Method-------------------------
clc
x=[4];
tol=0.0007;
TV=3;
format long
for i=1:5;
     fx=x(i)^3-5*x(i)^2+7*x(i)-3;
     fxx=3*x(i)^2-10*x(i)+7;
     x(i+1)=x(i)-(fx/fxx);
     E_T(i)=(abs((TV-x(i+1))/TV))*100;
end
for i=1:5;
     e(i)=x(6)-x(i);
     fx=x(i)^3-5*x(i)^2+7*x(i)-3;
     fxx=3*x(i)^2-10*x(i)+7;
     fxxx=6*x(i)-10;
     e(i+1)=(-fxxx/2*fxx)*(e(i))^2;
end
if abs(TV-x(i+1))<tol
    disp(' enough to here')
    disp('------------')
    disp('  X(i+1) ')
    disp('-------------')
    x'
    disp('------------')
    disp('   T.V   ')
    disp('-------------')
    E_T'
    disp('------------')
    disp('   E(i+1)   ')
    disp('-------------')
    e'
    disp('-------------')
end
%-----------------------------------------------------
```

enough to here

X(i+1)

4.000000000000000
3.400000000000000
3.100000000000000
3.008695652173913
3.000074640791192
3.000000005570623

T.V

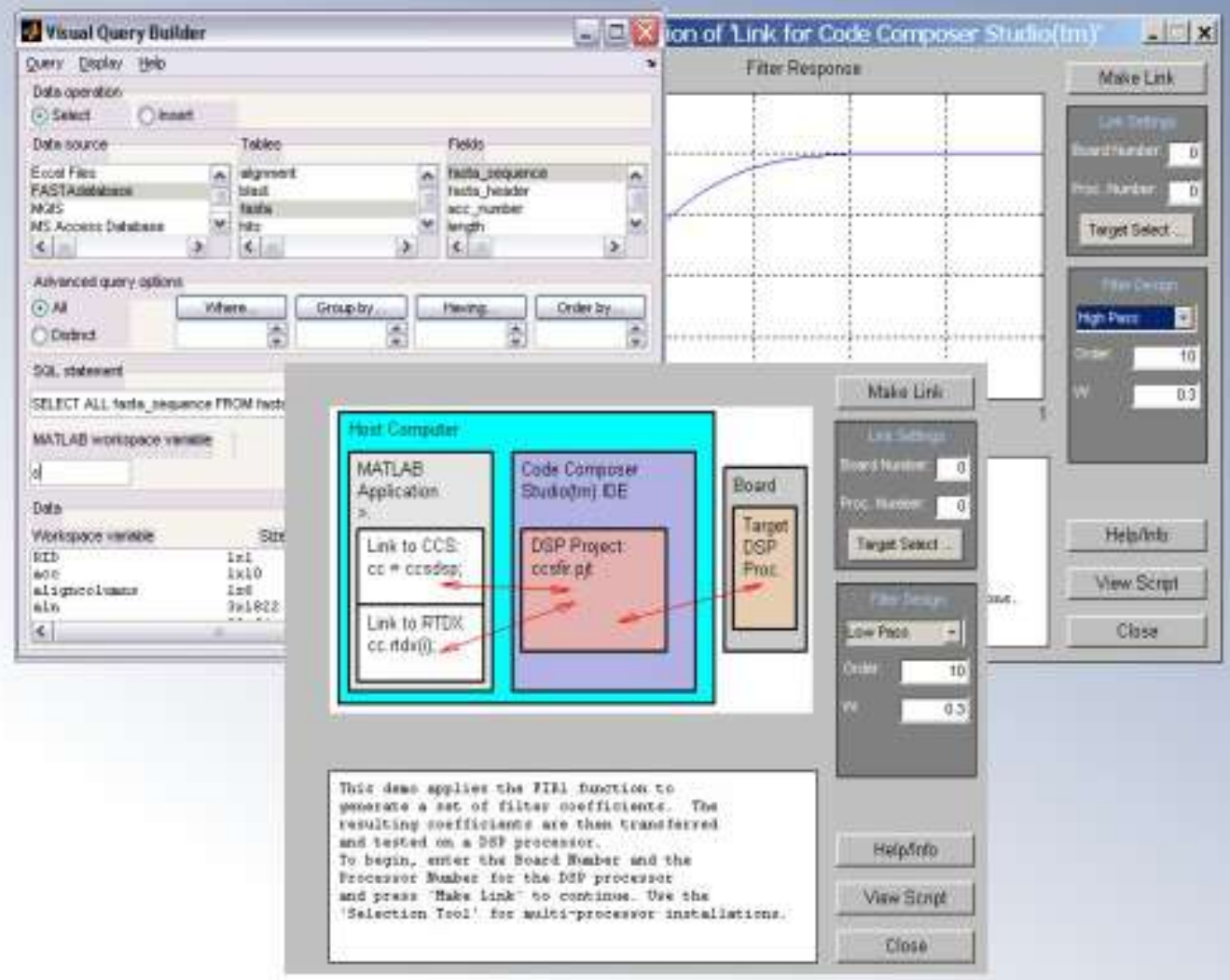
13.333333333333330
3.333333333333322
0.289855072463781
0.002488026373060
0.000000185687436
0.000000007462475

E(i+1)

-0.999999994429377
-0.399999994429377
-0.099999994429377
-0.008695646603290
-0.000074635220569
-0.000000089144954

ترقبوا المزيد من الشروحات للأمثلة فى التحليل العددى والرياضيات والتحكم الألى والأتصالات وإلكترونات القدرة ونظم التشغيل والدارات التماثلية والنظم الرقمية واسس الألكترونات وغيرها من المواد فى اغلب التخصصات راجين من الله سبحانه وتعالى التوفيق فلله الحمد والمنة وبه التوفيق والعصمة.

تصميم الواجهات الرسومية بالـ Matlab



بن العبد

كافة الحقوق المتعلقة بهذا الكتاب ملك للجميع

بســـم الله الـــرحمـــن الرحيـــم

"و لا تحسبن الله غافلا عما يعمل الظالمون إنما يؤخرهم ليوم تشخص فيه الأبصـــــار."

صدق الله العظيم

إبراهيم 42

في "أمتنا" شعب في التراب و شعوب في السحاب, شعب فلسطيني يستحلب الصخر و شعوب خليجية تتمرغ و تتمرغ النعمة في أعتابهم و يأبون إلا أن يدوسوها بأقدامهم.

الفهرس

1. مقدمة 4
2. الجزء الأول 4
3. الجزء الثاني 8
1.3. تصميم الواجهة..... 8
2.3. إدراج المكونات 12
3.3.التفاعل بين المكونات 16
4.3. نصوص المساعدة..... 19
4. خاتمة 27

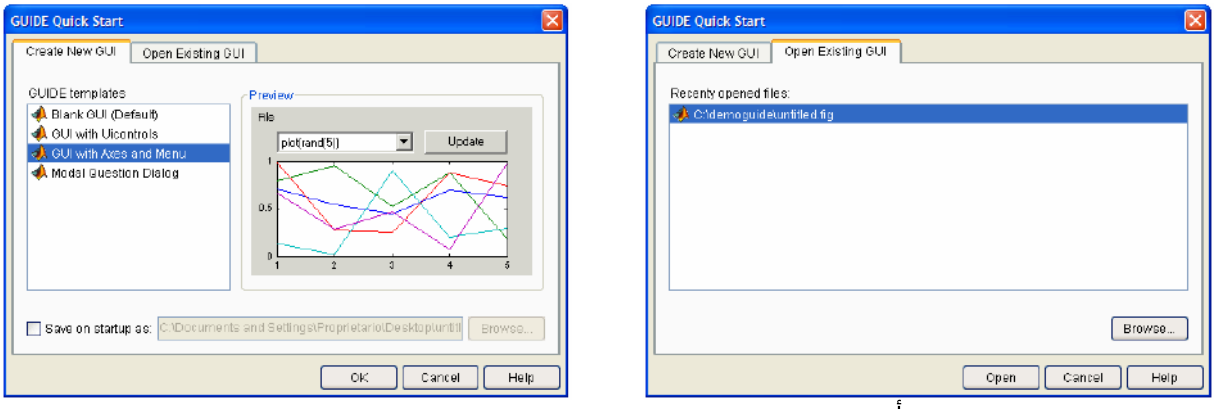
1. مقدمة

هذا الدرس موجه لمن لهم خلفية في البرمجة بالـ Matlab و يهدف للمساعدة في تطوير برامج متكاملة ذات واجهات رسومية تفاعلية. و ينقسم إلى جزئين منفصلين. يخص الجزء الأول التصميم بإستخدام الأداة guide في حين أن الجزء الثاني يهتم بالتصميم البرمجي للواجهات الرسومية.

2. الجزء الأول

في هذا الجزء سأحاول تقديم شرح مختصر لتصميم الواجهات بإستعمال guide الذي يوفره الـ Matlab. الـ guide عبارة عن أداة تحتوي على عناصر جاهزة توفر مجموعة من المكونات واوجهة رسومية جاهزة تضاف لها هذه العناصر عبر السحب و الإفلات[1].

عند تنفيد التعليمة guide نتحصل على الواجهة المجسدة في الصورة رقم 1. تحتوي هذه الواجهة على قائمة من أنواع الواجهات الرسومية التي يمكن تصميمها. كما توفر إمكانية فتح تصميم سابق.

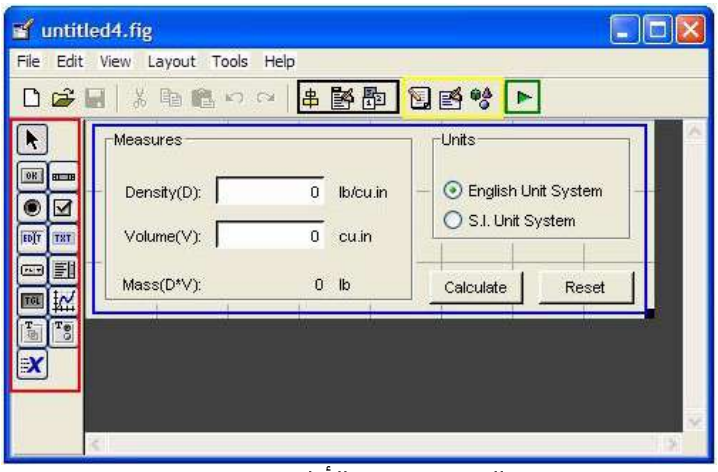


أ ب

الصورة رقم 1: إنشاء و فتح مشروع

عند إختيار تصميم واجهة رسومية جديدة من نوع Blank GUI نتحصل على ما هو مجسد في الصورة رقم 2. الواجهة تضم المكونات الرئيسية و الأكثر إستخداما و هو ما ييسر للمبرمج تعديلها والإستفادة من الشفرة الأولية المولدة تلقائيا. و بالإضافة للمكونات المتواجدة في الواجهة الرسومية، توجد عدة مكونات أخرى في علبة الأدوات[2] مما يتيح للمبرمج تصميم واجهة أكثر تناسبا مع احتياجاته.

[1] السحب و الإفلات: Drag and drop
[2] علبة الأدوات: ToolBox



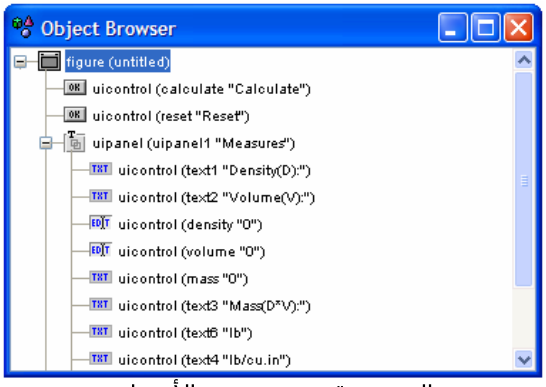
الصورة رقم 2: الأداة guide.

قبل الشروع في التصميم و تفاصيله نتوقف لفهم محتويات الواجهة الأولية لتصميم الواجهات الرسومية و عناصرها.

- ✓ يجسد المستطيل الأزرق و الأكثر أهمية بالنسبة لنا مجال التصميم و ما سيظهر لاحقا في الواجهة النهائية, و تهدف كافة العناصر المحيطة يه للتحكم في كل ما نضيفه للواجهة.
- ✓ يجمع المستطيل الأحمر المكونات التي يمكن إضافتها للواجهة الرسومية.
- ✓ المستطيل الأخضر و بكل بساطة زر الترجمة.
- ✓ يجمع المستطيل الأصفر مجموعة من العناصر الهامة في تصميم المكونات و تعديل خصائصها و وضائفها و سنفهمها سويا إن شاء الله إبتداء من اليمين إلى اليسار:

- و ضيفة العنصر الأول و هو متصفح الكائنات[3], الإطلاع على قائمة المكونات المدرجة في الواجهة بترتيب. و أعني هنا بعبارة "ترتيب" هو إبراز المكونات بكيفية مستقلة إن كانت كذلك في الواجهة. أما إن كانت متواجدة داخل حاوية[4] فإنها لن تظهر إلا إذا تصفحنا محتويات هذه الأخيرة.
- تجسد الصورة رقم 3 متصفح الكائنات حيث تحتوي أربعة عناصر رئيسية. الزران والحاويتان, و في كل حاوية مجموعة العناصر التابعة لها. يمكن التعديل على هذه العناصر عبر النقر المزدوج فيتم فتح واجهة جديدة ألا و هي مراقب الخصائص[5] (الصورة رقم 4).

[3] متصفح الكائنات: Object Browser.
[4] حاوية: Uipanel.
[5] مراقب الخصائص: Property Inspector.

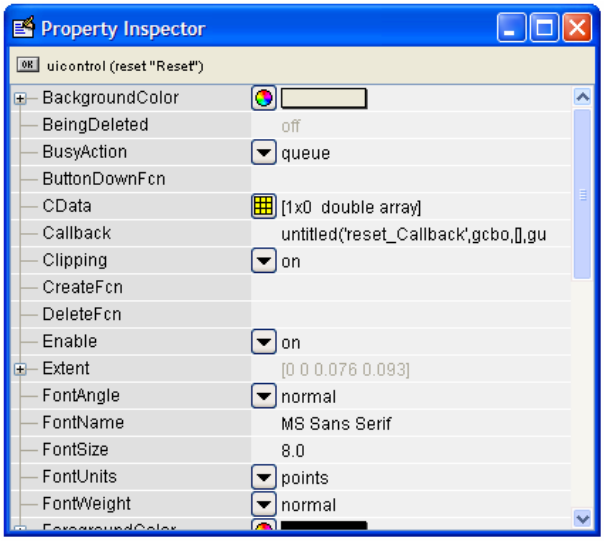


الصورة رقم 3: متصفح الأشياء.

- كل ما قد تحتاجه من تعديل في تصميم المكونات و وضائفها متوفر في مراقب الخصائص.
- يحملنا العنصر الثالث في المربع الأصفر إلى الشفرة الأولية للواجهة حيث يمكن تعديلها وفق متطلباتنا. في ما يلي جزء من الشفرة الأولية للزرين الرئيسين في الواجهة حيث يمكن تعديلهما سوى مباشرة أو عبر إستدعاء دوال أخرى كم هو الحال في إستدعاء الدالة initialize_gui.

```
% --- Executes on button press in calculate.
function calculate_Callback(hObject, eventdata, handles)
% ...
mass = handles.metricdata.density * handles.metricdata.volume;
set(handles.mass, 'String', mass);

% --- Executes on button press in reset.
function reset_Callback(hObject, eventdata, handles)
% ...
initialize_gui(gcbf, handles, true);
```

- كل ما قد تحتاجه من تعديل في تصميم المكونات و وضائفها متوفر في مراقب الخصائص.



الصورة رقم 4: مراقب الخصائص.

- ✓ يضم المستطيل الأسود و المتبقي بعض الوضائف الثانوية التي قد نحتاجها في التصميم:
  - مهمة العنصر الأول في القائمة ترتيب العناصر, حيث إذا تنقل المستخدم بين مكونات الواجهة عبر الزر Tab في لوحة المفاتيح, يكون هناك توافق بين أماكنها و ترتيبها.
  - نلاحظ في الصورة رقم 1 غياب قائمة الإختيارات أو ما يسمى باللغة الإنجليزية Menu. هذا لا يعني إستحالة إضافة هذا العنصر للواجهة, إنما ذلك ممكن عبر تفعيل الإختيار الثاني في المستطيل المعني. بذلك نجد كل اللوازم لتصميم قائمة الإختيارات و القوائم المسندلة[6] أيضا.
  - وضيفة العنصر الأخير في المستطيل الأسود في الصورة رقم 2 تحديد توزع العناصر في الواجهة خاصة عند تغير حجمها.

قبل أن أختم هذا الجزء أذكر بإمكانية الوصول لأغلب الإختيارات المذكورة أنفا عبر النقر بيمين الفأرة على المكون الذي نود تعديل خصائصه.

يولد حفظ الواجهة المتحصل عليها ملفين بإمتدادين مختلفين وهما .m و fig. يتكون الأول من شفرة الدوال التي سيتم إستدعائها لاحقا، في حين أن الثاني يضم شفرة تصميم الواجهة الرسومية, لا يمكن تعديل نصه يدويا و لابد من إعادة فتحه في برنامج التصميم guide كلما إقتضت الضرورة تعديله.

خاتمة

بعد هذه الجولة السريعة و الموجزة لكيفية إستخدام guide ننتقل إلى الطريقة الثانية, التي قد تكون أصعب نظرا لتصميم الواجهة برمجيا من الألف إلى الياء دون إستخدام واجهات رسومية أو أدوات جاهزة. و لكن إستعمال العبارة "قد" يفيد الإحتمال و الفرضية و لا يمكن التأكد إلا في نهاية الجزء الثاني.

[6] القوائم المسدلة: ContextMenu.

3. الجزء الثاني

سنتدرج في هذا الجزء في التصميم البرمجي للواجهات إنطلاقا من تصميم النافذة وصولا إلى إدراج نصوص المساعدة. علما أنني إستخدمت في هذا الدرس الإصدار 7.1 للـ matlab

و ربما تكون هناك بعض الإختلافات عند تجربة الشفرات المدرجة في هذا الدرس في إصدارات أخرى من البرنامج.

1.3. تصميم الواجهة

سنبدأ تصميم واجهة رسومية انطلاقا من بعض التعليمات البسيطة التي ربما نعلمها جميعنا. أول هاته التعليمات:

```
F_MainFigure=figure
```

هاته التعليمة ستولد واجهة رسومية فارغة و سيتم إسناد قيمة 1 كمخرج عند نجاح العملية.

إستعملنا هنا الدالة بدون أي مدخلات, و السؤال الذي يطرح نفسه هنا هو هل لهذه الدالة مدخلات؟ فربما إذا عرفنا هاته المدخلات ستصبح المهمة أسهل. لذلك سنبحث قليلا في نصوص المساعدة بإستعمال:

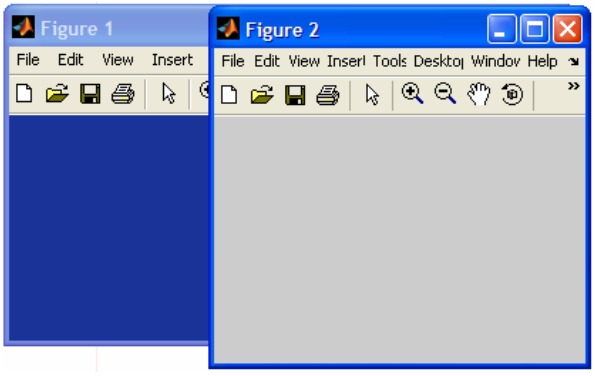
```
help figure
```

للأسف أن نص المساعدة لا يكفي لمعرفة طرق تصميم واجهة رسومية تماثل تلك التي توفرها لغات البرمجة. إذا كل ما عليك هو الإنتظار حتى نهاية الدرس.

أول خطوة في تغير الواجهة الرسومية الأصلية التي يوفرها الـ matlab تتمثل في تغير لون خلفية, جرب تنفيذ التعليمات التالية مباشرة في نافذة التعليمات:

```
F_MainFigure = figure('Color',[0.1 0.2 0.6]), figure
```

النتيجة مجسدة في الصورة رقم 3, تغير لون خلفية الواجهة من اللون الأصلي المسند لها إلى اللون الأزرق. نلاحظ في التعليمة التي ولدت هذا التغير أننا أضفنا مدخلين للدالة figure; في الحقيقة هما ليس متغيرين إنما متغير و القيمة المسندة له.

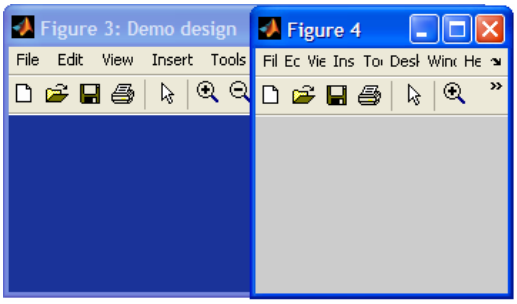


صورة رقم 1: تغير لون خلفية الواجهة

القيمة المسندة لمتغير تغير الخلفية تتمثل في مصفوفة أحادية الأبعاد متكونة من ثلاثة أرقام متراوحة بين 0 و 1. الأرقام الثلاثة تمثل الألوان الأحمر, الأخضر و الأزرق بالترتيب لذلك إذا أردت أن يطغى لون من هذه الألوان على البقية يجب أن تكون القيمة المسندة له الأكبر. نلاحظ هنا أن الـ matlab يعتمد نظام RGB[7] في التعامل مع الألوان.

نعيد الآن النظر مجددا للصورة رقم 1. نلاحظ أن الـ matlab أسند تلقائيا إسم لكل نافذة; ماذا لو أردنا وضع أسماء نختارها نحن لتكون مناسبة أكثر لمحتوى الواجهة الرسومية أو لتحمل إسم التطبيق أو غير ذلك من الإعتبارات؟ الإجابة تختصرها التعليمات التالية:

```
F_MainFigure = figure('Color',[0.1 0.2 0.6], 'Name', 'Demo design'), figure
```

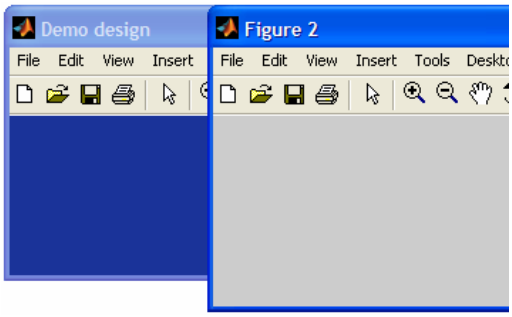


صورة رقم 2: تغير إسم الواجهة

في التعليمة السابقة و كما تلاحظون قمت بإضافة مدخلين للدالة figure, وهما متغير إسم الواجهة والقيمة المسندة له. لكن المشكل هنا هو أننا لم نتخلص بعد من تلك التسمية التلقائية التي يسندها الـ matlab و كل ما قمنا به هو إضافة تسمية أخرى للموجودة. لذلك فالخطوة الموالية في تصميمنا للواجهة الرسومية ستتمثل في التخلص كليا من التسمية التلقائية التي يسندها الـ matlab. كل ما عليك الآن هو تجربة هاته التعليمات:

```
F_MainFigure = figure('Color',[0.1 0.2 0.6], 'Name', 'Demo
design','NumberTitle', 'off'), figure
```

تم تعديل إسم الواجهة بالكيفية التي يريدها. كما يوحي له السابق, لم أعدل مجددا التسمية إنما عمت فقط بتعطيل خاصية الإضافة التلقائية لرقم الواجهة.



الصورة رقم 3: حذف ترقيم الواجهة

[7] Red, Green Blue:RGB

نتوقف مع الشفرة السابقة التي بدأت تطول و لا تزال قائمة مدخلات الدالة figure تطول, لأذكر بإمكانية إستعمال الرمز "..." كدلالة على إمتداد التعليمات على أكثر من سطر.

سنقوم هذه المرة بتعطيل صلاحية تغير حجم الشاشة. للقيام بذلك يكفي إضافة متغير Resize لقائمة مدخلات الدالة figure وإسناد القيمة off لها, كما يلي:

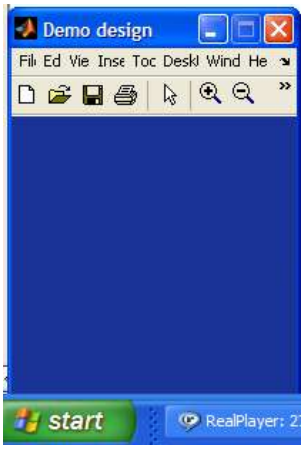
```
F_MainFigure = figure('Color',[0.1 0.2 0.6], ...
                      'Name', 'Demo design', ...
                      'Resize', 'off',...
                      'NumberTitle', 'off')
 figure
```

الغاية من حذف صلاحيات تغير حجم الواجهة الحفاظ على توزع العناصر المكونة للواجهة و عدم إنحصارها في ركن واحد عند تكبيرها. يمكن أن تحدث هذه المشاكل عندما تكون الإحداثيات المسندة للمكونات ثابتة و مستقلة عن حجم الواجهة الذي تحدده قيمة المتغير ScreenSize.

نواصل تصميمنا للواجهة بإضافة متغير جديد وهام وهو Position. هذا المدخل سيمنحنا حرية تحديد مكان ظهور الواجهة و حجمها لذلك فإن القيمة المسندة لها من نوع مصفوفة أحادية الأبعاد متكونة من أربعة عناصر; الأول و الثاني لتحديد الإحداثيتين الأفقية و العمودية و الثالث و الرابع لتحديد الطول و العرض. إذا الكود السابق يصبح كما يلي:

```
F_MainFigure = figure('Color',[0.1 0.2 0.6], ...
                      'Name', 'Demo design', ...
                      'Position', [10 60 200 200], ...
                      'Resize', 'off',...
                      'NumberTitle', 'off')
```

حددت قيمة المتغير Position بكيفية تجعل الواجهة تظهر فوق شريط المهام في الركن الأيسر للشاشة كما هو مبرز في الصورة رقم 5.



الصورة رقم 5: تحديد مكان و حجم الواجهة.

كان التعريف المسبق لقيمة المتغير Position بطريقة "بدائية". لأنه لم يكن هناك أخذ بعين الإعتبار حجم الشاشة المتغير من حاسوب لآخر, و لتجاوز هذا الإشكال يمكننا إستغلال المتغير ScreenSize

```
SCREENSIZE = get(0,'ScreenSize')
```

الذي يوفره الـ matlab و الذي يرجع حجم الشاشة في شكل مصفوفة عند إستدعائه بالكيفية التالية:

النتيجة حسب مقاييس حاسوبي:

```
SCREENSIZE =
           1           1        1440         900
```

جرب الآن الكود التالي. كيف صار تصميم الواجهة؟

```
SCREENSIZE = get(0,'ScreenSize');
F_MainFigure = figure('Color',[0.1 0.2 0.6], ...
 'Name', 'Demo design', ...
 'Position', [SCREENSIZE(1)  SCREENSIZE(2) SCREENSIZE(3) SCREENSIZE(4)], ...
 'Resize', 'off',...
 'NumberTitle', 'off')
```

ملأت الشاشة!!! هذه معلومة جديدة ربما تحتاجها عند تصميم لعبة أو غير ذلك من التطبيقات. نعود الآن لجعل الواجهة تظهر في نفس المكان الذي إخترناه سابقا ولكن هذه المرة وفق مقاييس الشاشة و بكيفية أكثر إحترافية:

```
SCREENSIZE = get(0,'ScreenSize');
FigureSize= [SCREENSIZE(1)  60*SCREENSIZE(2) round(SCREENSIZE(3)/5)
round(SCREENSIZE(4)/5)];

F_MainFigure = figure('Color',[0.1 0.2 0.6], ...
  'Name', 'Demo design', ...
  'Position',FigureSize, ...
  'Resize', 'off',...
  'NumberTitle', 'off')
```

إستعملت الدالة round لتحويل ناتج القسمة لعدد طبيعي و يمكن أيضا إستعمال الدالة floor. هاتان الدالتان غير متماثلتين و تختلف النتيجة حسب ما بعد الفاصلة. لمعرفة الفرق بينهما جرب مثلا التعليمات التالية:

```
floor(5.93), round(5.93)
```

يمكن للمطور إضافة عدة تعديلات أخرى على الواجهة الرسومية عبر إدراج خاصيات أخرى. يبرز الجدول رقم 1 قائمة بهاته الخصائص و وضيفة كل منها.

الخاصية	الوضيفة
Color	تعديل لون الواجهة
Menubar	إبقاء أو حذف شريط المهام
Name	تسمية الواجهة الرسومية
Numbertitle	تعديل ترقيم الواجهة
Parent	تحديد الواجهة الأم
Position	تحديد المكان و المقاييس
Resize	صـــلاحية تعديل حجم الشاشة
Tag	تحديد المؤشر
Toolbar	إبقاء أو حذف شريط الأدوات
Userdata	بيانات المستخدم
Visibile	إظهار أو إخفاء الواجهة

الجدول رقم 1: خصائص الواجهة.

بعد التعرَف على جملة من الإختيارات التي تمكننا من تصميم واجهة رسومية في الـ matlab بالمواصفات التي نريدها, سننتقل إن شاء الله إلى الجزء الخاص بإضافة الأزرار و الإطارات و غيرها من المكونات التي نعرفها في باقي لغات البرمجة.

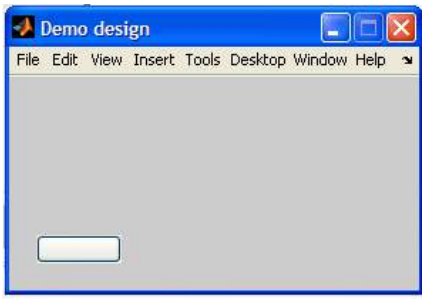
2.3. إدراج المكونات

الدالة الرئيسية في الجزء الموالي هي الدالة uicontrol و التي سيعتمد عليها باقي تصميمنا. سنبدأ إذا التصميم بإستدعاء هذه الدالة في نافذة الأوامر كانت النتيجة واجهة رسومية تحتوي زر دون نص في ركنها الأسفل الأيسر. لكن مجددا نص المساعدة حول كيفية إستعمال هذه الدالة غير كاف. إذا سنبدأ في فهم كيفية إستعمالها لجعل هذا الزر ذى وضيفة.

في باقي لغات البرمجة مثل C# و Delphi و غيرها نعرّف الواجهة ثم نضيف لها الأزرار و باقي المكونات. بحيث أن كل مكون ينتمي لواجهة دون غيرها عبر تعريفه داخل القسم المعرف للواجهة أو تحديد الواجهة التي ينتمي لها. يرث الـ matlab أيضا أحد هذه المبادئ و يفرض تحديد الواجهة المنتمي لها المكون. لذلك نعود إلى الشفرة التي كتبناها سابقا لتصميم الواجهة و نضيف لها الجزء المتعلق بالزر كما يلي:

```
SCREENSIZE = get(0,'ScreenSize');
FigureSize= [5*SCREENSIZE(1)  70*SCREENSIZE(2) round(SCREENSIZE(3)/5)
round(SCREENSIZE(4)/6)];
F_MainFigure = figure('Color',[0.8 0.8 0.8], ...
                      'Name', 'Demo design', ...
                      'Position',FigureSize, ...
                      'Resize', 'off',...
                      'NumberTitle', 'off')
b = uicontrol('Parent', F_MainFigure)
```

و هكذا تمت إضافة الزر إلى الواجهة التي عرفناها مسبقا كما هو مجسد في الصورة رقم 6.



الصورة رقم 6: إضافة عنصر للواجهة.

ملاحظة: في الصورة السابقة إستعملت عبارة "مكون" عوض عن "زر" لأن الدالة uicontrol لا تخص فقط الأزرار إنما باقي المكونات أيضا, و المكون التلقائي الذي تضيفه هو زر. سنتعرف لاحقا على كيفية إستعمال هذه الدالة في إضافة بقية أنواع المكونات.

نضيف مجموعة من الإختيارات الإضافية لتعريف شفرة الزر لتصبح كما يلي:

```
b = uicontrol('Parent', F_MainFigure,...
  'BackgroundColor',[0.701961 0.701961 0.701961], ...
  'Callback','Function1', ...
  'Interruptible', 'off', ...
  'Position',[222 120 60 20], ...
  'String','Open')
```

تمكن الخاصية Callback من تحديد إسم الدالة التي نريد إستدعائها. كتبت Function1 كمثال بحيث أنه سبق تعريف هذه الدالة. و يمكن أيضا كتابة التعليمات مباشرة كقيمة للخاصية Callback إذا كان عدد التعليمات محدودا, بالطريقة التالية:

```
'Callback','sprintf(''Assalamou Alaykom wa rahmatou ALLAH wa
baraketouhou'')', ...
```

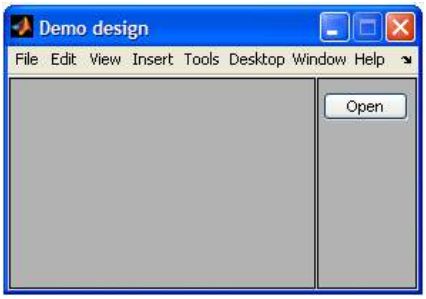
نلاحظ هنا تكرار الرمز ' و ذلك لأنه محجوز في الـ matlab لتحديد النصوص. لذلك إذا اردنا إستعماله داخل نص فلابد من سبقه بمثيله.

الخاصية الثانية التي ربما تحتاج الشرح هي Interruptible; في توفر إمكانية تعطيل الوضيفة المسندة للمكون, كما يوحي إسمها. يعد هذا المتغير ضروري و مستعمل في محله عند تنفيذ عمليات تطلب مدة زمنية كبيرة أو لا متناهية. إثر التعديل الأخير يصبح شكل الواجهة التي صممناها ومخرجاتها كما هو مبرز في الصورة رقم 7.

نواصل تصميمنا بإضافة إطار أو ما يسمى Frame في بعض لغات البرمجة للواجهة التي صممناها. الدالة التي سنستعملها هنا هي ذاتها الدالة السابقة و لكن بتعديل مختلف لمدخلاتها. إذا صارت الشفرة الجديدة التي سنضيفها كما يلي:

```
h_f_background = uicontrol('Parent', F_MainFigure, ...
  'BackgroundColor',[0.701961 0.701961 0.701961], ...
  'Position',[2 2 215 148], ...
  'Style','frame')
```

تعريف مشابه لشفرة الزر و لكن بإدراج خاصية جديدة و هي Style. تمكننا هذه الأخيرة من تحديد نوع المكون وفق القيمة المسندة له و التي تنتمي لمجموعة من القيم المعرفة مسبقا في الـ matlab (الجدول رقم 3). أذكر هنا أنه لابد من تعريف الإطار قبل تعريف باقي المكونات و بعد كتابة الشفرة الخاصة بالواجهة. بهذه الكيفية سنحدد الواجهة المنتمي لها الإطار و سنتجنب حجب باقي المكونات من قبل هذا الأخير. قمت بإضافة شفرة الإطار و آخر مماثل لها مع تعديل الإحداثيات و الحجم و كانت النتيجة الصورة رقم 8.

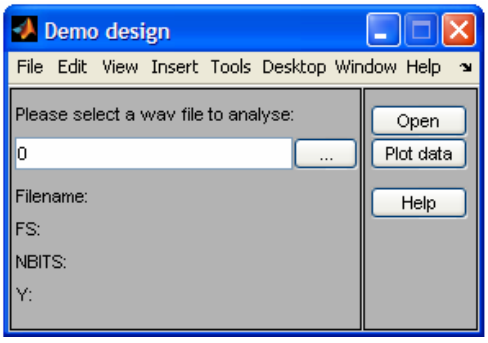


الصورة رقم 8: إضافة إطارات للواجهة

واصلت تصميم الواجهة لأحصل في نهاية المطاف ماهو مجسد في الصورة رقم 9. الجديد في هذه الأخيرة تواجد مكونات في الواجهة الرسومية دون تقديم مسبق. إضافة نص أو حقل نص أو غير ذلك يكون عبر تغير الخاصية style للدالة uicontrol المعرفة للمكونات. مثال:

```
h_f_background = uicontrol('Parent', F_MainFigure, ...
  'BackgroundColor',[1 1 1], ...
  'Position',[2 2 215 148], ...
  'Style','edit', …
  'Callback','Method_To_Call', ...
  'String','0', ...)
```

يمكننا إضافة عدة مكونات أخرى، تعادل ما يوفره guide على سبيل الذكر edit, popupmenu, text.



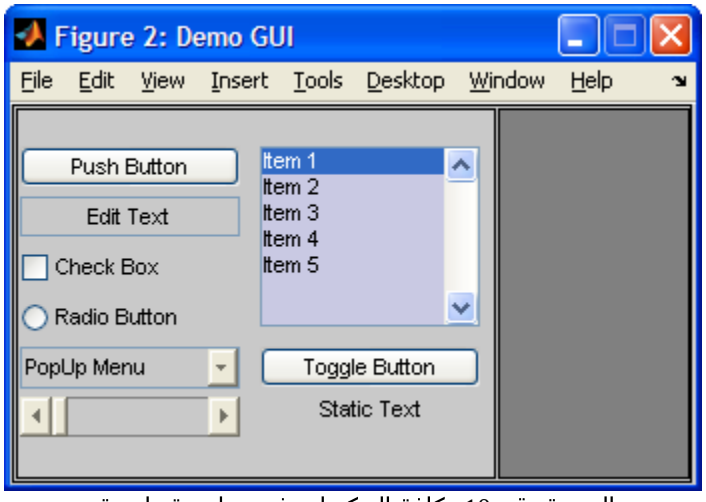
الصورة رقم 9: الواجهة النهائية.

تتشارك أغلب المكونات في الخصائص و تتماثل من حيث التصميم مع إختلافات طفيفة في الخصائص. يحوصل الجدول رقم 2 أغلب الخصائص التي يمكن إستغلالها في تصميم المكونات وسنرى في الفقرة التالية كيفية الحصول على قيمة مكون ما.

الخاصية	الوضيفة
Parent	تحديد الواجهة الأم
BackgroundColor	تحديد لون الخلفية
Position	تحديد المكان و المقاييس
Style	تحديد المكون
Tag	تحديد المؤشر
String	تحديد النص
FontWeight	تحديد نوع الكتابة
Callback	الإستدعاء
Interruptible	التعطيل
Value	تحديد القيمة

الجدول رقم 2: خصائص المكونات.

لا أنصح بإدراج كافة المكونات في واجهة واحدة إلا عند الضرورة القصوى, لأن ذلك يؤثر في إخراج الواجهة و يضفي عليها تعقيد نحن في غنى عنه. تمثل الصورة رقم 10 إستثناء لأن الغاية منها إبراز جميع المكونات التي يمكن إدراجها في الواجهة. علما أن المكون الذي يمثل الخلفيات هو المكون Frame.



الصورة رقم 10: كافة المكونات في و اجهة واحدة.

لإدراج هذه المكونات يكفي إسناد الأسماء التالية للخاصية style لكل مكون:

Push Button	Radio Button	Slider	Check Box	Frame
Static Text	PopUp Menu	ListBox	Edit Text	Toggle Button

الجدول رقم 3: أسماءالمكونات.

3.3. التفاعل بين المكونات

كل ما سبق ذكره في تصميم الواجهات الرسومية و المكونات كاف لتصميم واجهات غير تفاعلية, ونادرا ما نحتاج لذلك كمشروع مستقل. لذلك وجب الحديث عن كيفية تمرير القيم بين المكونات والربط فيما بينها لتوفير أكثر صلاحيات للمستخدم تتجاوز مشاهدة نتيجة العمليات الرياضية.

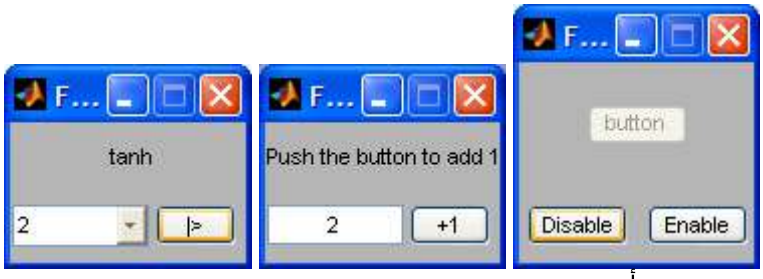
نبدأ بمثال بسيط يتمثل في إظهار صورة على الواجهة حسب إختيار المستخدم. واجهة التطبيق مجسدة في الصورة رقم 11, و لتصميم تلك الواجهة يكفي تكرار ما تم ذكره سابقا مرار. علما أن الجديد هنا يكمن في كيفية إظهار الصورة عبر النقر على الزر المسمى بإسم الصورة.

```
function ShowImage(ImageName)
switch ImageName
    case 'cameraman'              % 'ShowImage cameraman'
     imshow('cameraman.tif');
    case 'peppers'                % 'ShowImage peppers'
     imshow('peppers.PNG');
    case 'saturn'                 % 'ShowImage saturn'
     imshow('saturn.png');
end
```

تمثل التعاليق المضافة في الشفرة السابقة القيمة التي ينبغي إسنادها للخاصية Callback للأزرار الثلاثة, كما يجبذ تعريف الدالة ShowImage في ملف مستقل محفوظ في نفس مسار تواجد شفرة تصميم الواجهة.

الصورة رقم 11: التفاعل بين المكونات

يهدف المثالي التالي لإبراز كيفية التحكم في خصائص مكون في الواجهة من قبل مكون آخر كالتفعيل و التعطيل. هذه العمليات تكون عن طريق الخاصية Tag, و هي عبارة عن مؤشر للمكون يمكن التحكم فيه من خلالها كما هو مجسد في الصورة 11.أ.



ج ب أ

الصورة رقم 12: التفاعل بين المكونات

نلاحظ غياب لائحة الإختيارات من الواجهة و ذلك يعود لإدراج الخاصية MenuBar عند تصميم الواجهة.

```
Mainfigure=figure('Color',Bckgrnd, ...
                  'MenuBar', menubar,...
                  'PaperType','a4letter', ...
                  'Resize', 'Off',...
                  'Position',[150,150, 100,70],...
                  'Name', 'Demo GUI');
```

تحتوي الواجهة 12.أ ثلاثة أزرار, إثنان منها لتفعيل و تعطيل الزر الثالث الذي أسندت القيمة ControlTagللخاصية Tag التابعة له.

```
function Disable_Enable(action)
switch action
    case 'Disable'
     set(findobj('Tag','ControlTag'),'Enable','off');
    case 'Enable'
     set(findobj('Tag','ControlTag'),'Enable','on');
    case 'Show'
        imshow('cameraman.tif');
end
```

المثال الموالي (الصورة 12. ب) تجسيد لكيفية قراءة و تعديل نص مكون ما.

كلما نقر المستخدم على الزر يتم قراءة نص خانة الكتابة, تحويل نصها لرقم, إضافة 1 له ثم تعديل نصها, علما أن مؤشر خانة النص هو t_2.

في المثال السابق إستخدمت أمرين في أمر واحد, و هما البحث عن مؤشر المكون و تعديل خاصيته. في هذا المثال جزأت الأوامر أكثر لتيسر فهمها. كخطوة أولى, ينبغي البحث ما إن وجد المؤشر t_2. إذا تكللت عملية البحث بالنجاح و كانت قيمة المؤشر غير فارغة, نتمكن عندها من قراءة أو تعديل إحدى خصائصه عبر إستخدام الدالتان get و set.

```
function Increment
    handle = findobj('Tag','t_2');
    if ~isempty(handle)
       CurrentVal = str2num(get(handle, 'String'));
       CurrentVal=CurrentVal+1;
       set(handle,'String',int2str(CurrentVal));
    end
end
```

في المثال المجسد في الصورة 12.ج إستخدمت المكون popup-menu. وهو أصعب من الأزرار وحقول النصوص من ناحية التعامل لكونه يضم بيانات في شكل مصفوفات, واحدة ظاهرة للمستخدم و الثانية تضم القيم المقابلة لكل نص. لتيسير التعامل مع هذا المكون يمكن إستخدام متغيرات عامة مشتركة بين شفرة الواجهة و الشفرة النواة التي تضم الوضائف الرئيسية و الدوال. بهذه الكيفية يتحقق التوافق بين الإختيار الذي فعله المستخدم و القيمة المقابلة له.

```
function PopuModifier
% Shared parameters
global popup_strD;
global popup_strV;
global popupValue;
global p_popup;

    handle = findobj('Tag','popup_tag');
    if ~isempty(handle)
     eval(['g_str = popup_strV'])
     old_g = deblank(g_str(popupValue,:))
     SelectedIndex = get(p_popup, 'Value')
     set(findobj('Tag','t_text'),'String', deblank(g_str(SelectedIndex,:)));
    end
```

في ما سبق إعتمدنا الدالة الرئيسية uicontrol. و إسمها إختصار للنص User Interface Control. لهذه الدالة عدة "أخوات" مفيدة في تصميم الواجهات الرسومية التفاعلية, جمعت جلها في الجدول التالي:

>>ls C:\MATLAB71\toolbox\matlab\uitools\ui*			
uitoolbar.m	uisetcolor.m	uigridcontainer.m	uibuttongroup.m
uitoolfactory.m	uisetfont.m	uiload.m	uiclearmode.m
uitree.m	uisetpref.m	uiopen.m	uicontainer.m
uitreenode.m	uistack.m	uipanel.m	uiflowcontainer.m
uiundo.m	uisuspend.m	uipushtool.m	uigetdir.m
uiwait.m	uitab.m	uiputfile.m	uigetfile.m
uitoggletool.m	uitabgroup.m	uirestore.m	uigetpref.m
uisave.m	uitable.m	uiresume.m	uigettool.m
uigettoolbar.m			

كما نلاحظ, هذه الدوال مجموعة في مجلد واحد وهو مجلد الأدوات لتصميم الواجهات التابع للـ matlab. أترك لكم إكتشاف وضائف هذه الدوال.

4.3. نصوص المساعدة

ليكون التطبيق متكامل و إحترافي فلا بد من إضافة نصوص مساعدة و معلومات حول المطور, الإصدار و غير ذلك من البيانات التي قد نحتاج لإبرازها لمستخدمي التطبيق. يمكننا إضافة نصوص المساعدة بعدة طرق سوى بتصميم واجهات رسومية مخصصة لنصوص المساعدة أو بكل بساطة نستغل ما يوفره الـ matlab من أدوات مخصص لهذا الغرض. لكن قبل المرور لصلب الموضوع أذكر بإمكانية إدراج نصوص مساعدة داخل أي شفرة نكتبها و يظهرها البرنامج بنفس كيفية نصوص المساعدة التابعة لدواله.للقيام بذلك لا بد من كتابة نص المساعدة مباشرة بعد تعريف الدالة شرط أن لا يحتوي سطر فارغ. و إذا إحتجت لإدراج سطر فارغ للفصل بين فقرات نص المساعدة فلابد من إدراج سطر فارغ في شكل تعليق.

```
function [out1, out2]=DemoHelp(varargin)
% This is a the help text of the method DemoHelp...
%
%   See also ShowHelp

% 14.7.2009
% IBen Laiid

if nargin < 1
MsgBox('Invalid Input Arguments.','Demo GUI Design');
end
% Continue your code here
```

في المثال أعلاه تجسيد لما سبق ذكره خاصة في ما يخص ترك سطر فارغ داخل نص المساعدة. كما أدرجت سطر فارغ كليا و ليس في شكل تعليق و ذلك لإدراج معلومات حول كاتب الشفرة و تاريخ كتابتها دون أن تظهر هذه المعلومات في نص المساعدة عند القيام بذلك في نافذة الأوامر.

```
>> help DemoHelp
This is a the help text of the method DemoHelp

See also ShowHelp
```

نعود لتصميم الواجهات الرسومية و إدراج نصوص المساعدة بطريقة الـ matlab, أعني الواجهة التي تظهر عند النقر على الزر F1 في أي واجهة للبرنامج. بإيجاز تمكن الدالة helpwin إمكانية إظهار نص كما لو أنه تابع لنصوص المساعدة للـ matlab.

تعتمد هذه الدالة على مدخلين رئيسين و هما نص المساعدة و عنوانه, شرط أن تكون أسطر النص محررة بصفة مستقلة بعضها عن بعض و لها نفس الطول لأن الدالة تتعامل مع النص على أنه مصفوفة.

تصور الواجهة الرسومية رقم 13 مثال لإدراج نص مساعدة و نص لإظهار معلومات حول التطبيق. شفرة التطبيق مقسمة إلى ملفين أحداهما مخصص لتصميم الواجهة و إستدعاء الدالة المعرفة في الملف الثاني.

الصورة رقم 13: تطبيق نصوص المساعدة.

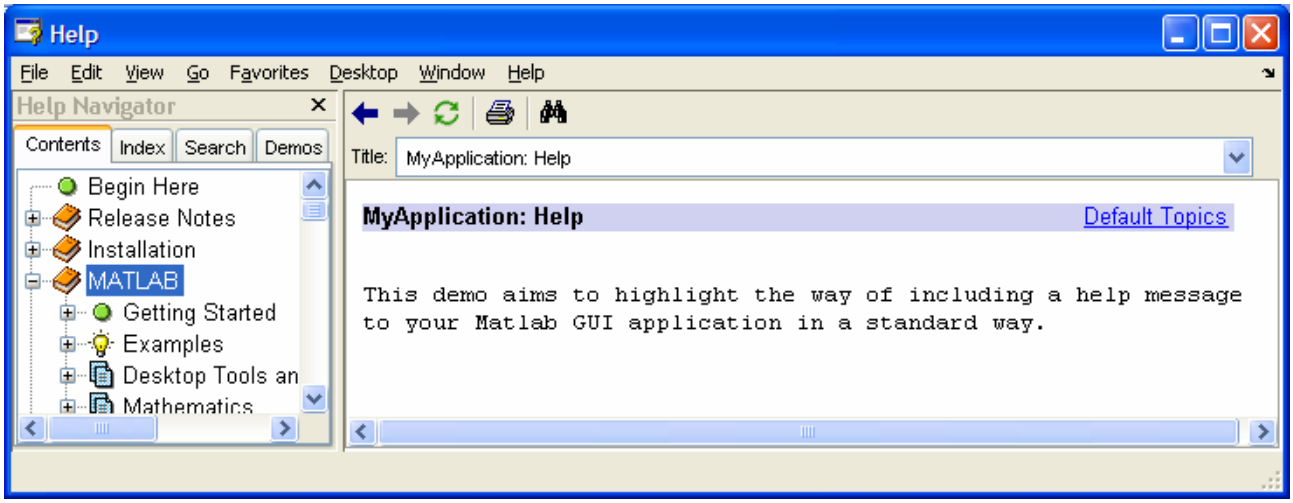
لا أرى في هذا المثال ما يستوجب الشرح خاصة أنه سبق شرح مختلف أجزاءه سابقا.

```
function InterfaceManager
%=======================================
Y_Pos=10; X_Pos=25; Width=70; Bckgrnd=[0.789 0.789 0.789];
%=======================================
Mainfigure=figure('Color',[0.8 0.8 0.8], ...
                  'PaperType','a4letter', ...
                  'Resize', 'Off',...
                  'Position',[150,150, 200,100],...
                  'Name', 'Demo GUI');
Text   =uicontrol('Parent', Mainfigure,...
                  'BackgroundColor',Bckgrnd, ...
                  'String', 'TODO:Add your controls here.',...
                  'Position', [X_Pos Y_Pos+50 Width+80 20],...
                  'Style', 'text');
%=======================================
button1=uicontrol('Parent', Mainfigure,...
                  'BackgroundColor',Bckgrnd, ...
                  'String', 'Help',...
                  'Position', [X_Pos Y_Pos Width 20],...
                  'Style', 'Push Button',...
                  'Callback', 'ShowHelp Help ');
button2=uicontrol('Parent', Mainfigure,...
                  'BackgroundColor',Bckgrnd, ...
                  'String', 'About',...
                  'Position', [X_Pos+80 Y_Pos Width 20],...
                  'Style', 'pushbutton',...
                  'Callback', 'ShowHelp About');
end
```

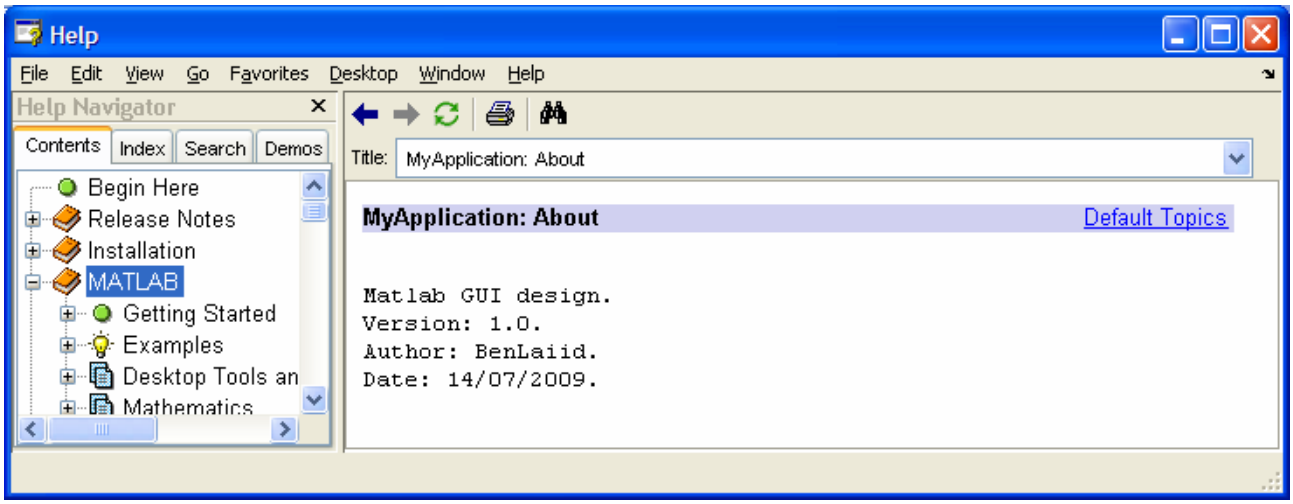
يمكن جمع مختلف نصوص المساعدة في شفرة واحدة يتم التميز بينها عبر إستخدام الجملة الشرطية . و يكفي إستدعاء الدالة و تحديد عنوان نص المساعدة لتقوم الدالة المعرفة في آخر الشفرة باللازم.

```
function ShowHelp(which_help)
 switch which_help
 case 'Help'
  helptitle = 'MyApplication: Help';
  helptext=[ ...
    'This demo aims to highlight the way of including a help message    '
    'to your Matlab GUI application in a standard way.                  '];

 case 'About'
    helptitle = 'MyApplication: About';
    helptext=[ ...
    'Matlab GUI design.                                                 '
    'Version: 1.0.                                                      '
    'Author: BenLaiid.                                                  '
    'Date: 14/07/2009.                                                  '];
end
helpwin(helptext, helptitle);
```

أخيرا نتحصل على واجهات مثل الصورة 14 و 15. مثل هذه الواجهات تفتح الشهية لمزيد تعلم الـ matlab والسعي لتطوير تطبيقات أكثر فأكثر تعقيدا و إحترافية طبعا.



الصورة رقم 14: واجهة نصوص المساعدة.



الصورة رقم 15: الواجهة "حول البرنامج".

المفيد

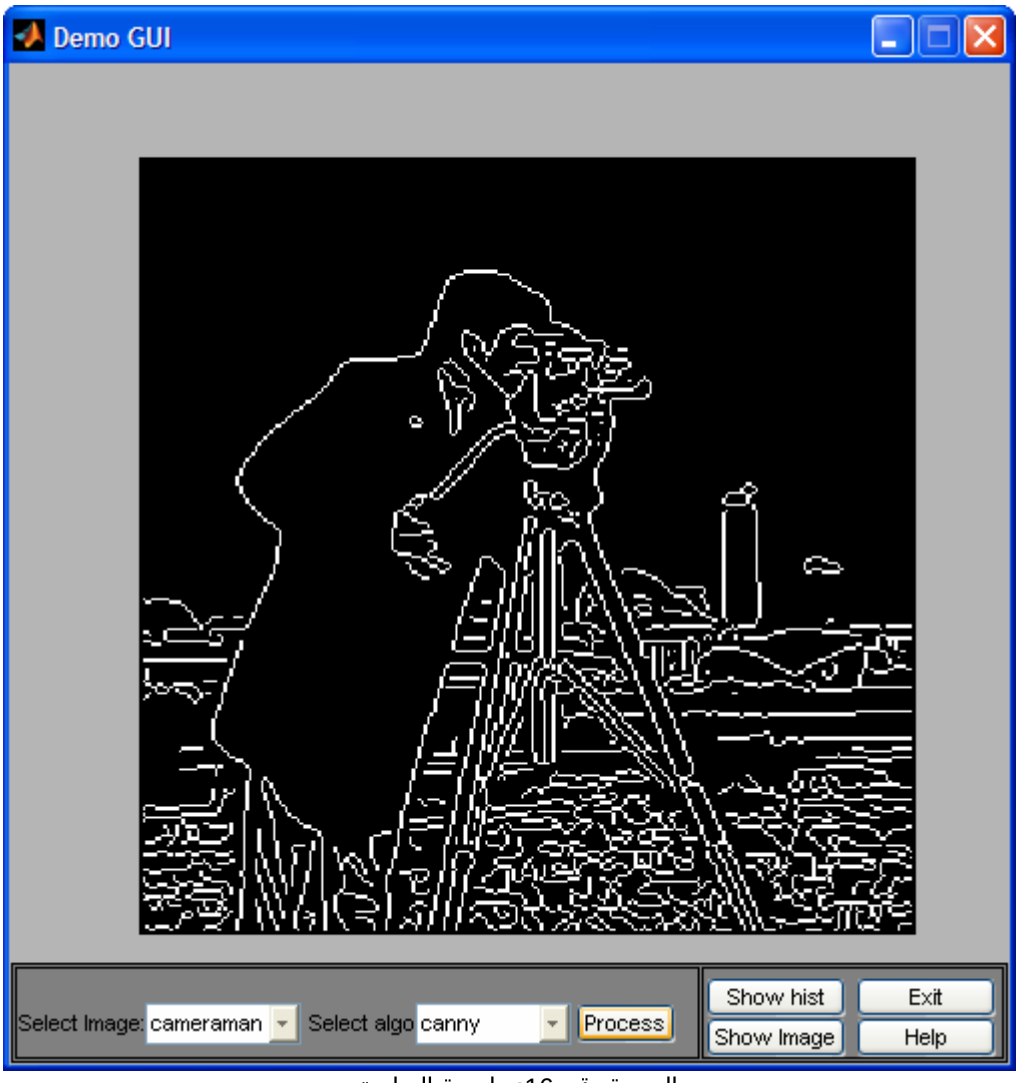
أعلم أنني ثرثرت كثيرا في ما سبق في هذا الدرس. سندخل إذا في صلب الموضوع مع تطبيقات متكاملة لتجسيد ما سبق ذكره.

التطبيق: معالج الصور.

الغاية من هذا التطبيق

- المقارنة بين أربعة خوارزميات مختلفة Canny, Sobel, Roberts, Prewitt عبر تطبيقها على أربعة صور مختلفة تابعة للـ matlab.
- كيفية تمرير البيانات بين عدة واجهات رسومية عبر تعريف المتغيرات من نوع global.
- إستدعاء واجهة رسومية ثانية مصممة بطريقتنا و تظهر نص المساعدة عوض عن إستخدام الأداة helpwin التابعة للـ matlab.

و العديد من النقاط الأخرى التي أترك لكم الفرصة لإكتشافها.



الصورة رقم 16: واجهة التطبيق.

في ما يلي شفرة التطبيق, علما أنها تمتد على ملفين منفصلين يحملان نفس إسم الدالة الرئيسية.

```
function InterfaceManager
% Shared parameters
global popup_strD;
global popup_strV;
global popupValue;
global p_popup;
global p_popupM;
global popup_ModeValue;
global popup_ModestrD;
global popup_Mode_strV;
% Initialize the default value of the popup
popupValue = 1;
popup_ModeValue=1;
% Popup-menu items texts
popup_strD = 'cameraman|circuit|kids|trees';
popup_ModestrD='prewitt|sobel|roberts|canny|all';
% Popup-menu items values
popup_strV = [ 'cameraman'; 'circuit  '; 'kids     '; 'trees    '];
popup_Mode_strV=['prewitt';'sobel  ';'roberts';'canny  ';'all    '];
%=======================================
Y_Pos=10; X_Pos=4; Width=50;
Figure_Width=500; Figure_Height=500;
Bckgrnd=[0.71 0.71 0.71];
%=======================================
Mainfigure=figure('Color',Bckgrnd, ...
                  'MenuBar', menubar,...    % remove the menubar
                  'Numbertitle','off',...
                  'PaperType','a4letter', ...
                  'Position',[150,150, Figure_Width,Figure_Height],...
                  'Resize', 'off',...
                  'Name', 'Demo GUI',...
                  'Tag', 'f_Mainfigure');
MainFrame=uicontrol('Parent', Mainfigure,...
                  'BackgroundColor',[0.5 0.5 0.5], ...
                  'Position', [2 2 Figure_Width-2 Figure_Height/10],...
                  'Style', 'frame',...
                  'Tag','popup_tag');
SideFrame=uicontrol('Parent', Mainfigure,...
                  'BackgroundColor',[0.5 0.5 0.5], ...
                  'Position', [347 4 151 46],...
                  'Style', 'frame',...
                  'Tag','popup_tag');
SideFrame=uicontrol('Parent', Mainfigure,...
                  'BackgroundColor',[0.5 0.5 0.5], ...
                  'Position', [4 4 342 46],...
                  'Style', 'frame',...
                  'Tag','popup_tag');
%=======================================
txt_box=uicontrol('Parent', Mainfigure,...
                'BackgroundColor',[0.5 0.5 0.5], ...
                 'String', 'Select Image:
Select algo:',...
                 'Position', [X_Pos+1 Y_Pos+3 Width+150 15],...
                 'Style', 'text',...
                 'Tag','t_text');
%=======================================
%...
```

```
%=======================================
p_popup=uicontrol('Parent', Mainfigure,...
                  'BackgroundColor',[1 1 1], ...
                  'Position', [X_Pos+65 Y_Pos+2 Width+28 20],...
                  'Style', 'popupmenu',...
                  'String',popup_strD ,...
                  'Value',popupValue,...
                  'Tag','popup_tag');
p_popupM=uicontrol('Parent', Mainfigure,...
                  'BackgroundColor',[1 1 1], ...
                  'Position', [X_Pos+200 Y_Pos+2 Width+28 20],...
                  'Style', 'popupmenu',...
                  'String',popup_ModestrD ,...
                  'Value',popup_ModeValue,...
                  'Tag','popup_Mode_tag');
%=======================================
btn_Proc=uicontrol('Parent', Mainfigure,...
                  'BackgroundColor',[0.5 0.5 0.5], ...
                  'String', 'Process',...
                  'Position', [X_Pos+280 Y_Pos+2 Width 20],...
                  'Style', 'pushbutton',...
                  'Callback', 'kernel Process');
btn_Orig=uicontrol('Parent', Mainfigure,...
                  'BackgroundColor',[0.5 0.5 0.5], ...
                  'String', 'Show Image',...
                  'Position', [X_Pos+345 Y_Pos-5 Width+20 20],...
                  'Style', 'pushbutton',...
                  'Callback', 'kernel show');
btn_hist=uicontrol('Parent', Mainfigure,...
                  'BackgroundColor',[0.5 0.5 0.5], ...
                  'String', 'Show hist',...
                  'Position', [X_Pos+345 Y_Pos+15 Width+20 20],...
                  'Style', 'pushbutton',...
                  'Callback', 'kernel hist');
btn_Help=uicontrol('Parent', Mainfigure,...
                  'BackgroundColor',[0.5 0.5 0.5], ...
                  'String', 'Help',...
                  'Position', [X_Pos+420 Y_Pos-5 Width+20 20],...
                  'Style', 'pushbutton',...
                  'Callback', 'kernel Help');
btn_exit=uicontrol('Parent', Mainfigure,...
                  'BackgroundColor',[0.5 0.5 0.5], ...
                  'String', 'Exit',...
                  'Position', [X_Pos+420 Y_Pos+15 Width+20 20],...
                  'Style', 'pushbutton',...
                  'Callback', 'kernel Exit');
%=======================================
% Show default image
imshow('cameraman.tif')
```

هذه شفرة الملف الثاني و هي النواة التي تحتوي الوضائف الرئيسية التي يتم إستدعائها من قبل مكونات الواجهة.

```
function kernel(action)
global I; % Read The image and store it in the global variable I
switch action
    case 'Process'
        Process()
    case 'show'
        imshow(I)
    case 'hist'
        figure, imhist(I);
    case 'Help'
       Help();
    case 'Exit'
       App_Exit();
end
%=======================================
function ProcessImage(I, Mode)
Per = edge(I,'prewitt');
Sob = edge(I,'sobel');
Rob = edge(I,'roberts');
Can = edge(I,'canny');
switch Mode
    case 1 % 'Per'
     imshow(Per);
    case 2 % 'Sobel'
     imshow(Sob);
    case 3 %'Roberts'
     imshow(Rob);
    case 4 %'Canny'
     imshow(Can);
    case 5  % 'All'
     figure
     subplot(2,2,1), subimage(Per), title('prewitt')
     subplot(2,2,2), subimage(Sob), title('sobel')
     subplot(2,2,3), subimage(Rob), title('roberts')
     subplot(2,2,4), subimage(Can), title('canny')
end
%=======================================
function Process()
% Shared parameters
global popup_strD;
global popup_strV;
global popupValue;
global p_popup;
global p_popupM;
global popup_ModeValue;
global popup_ModestrD;
global popup_Mode_strV;
global I;
% Find if there is an object having the tag popup_tag
handle = findobj('Tag','popup_tag');
if ~isempty(handle)
      eval(['g_str = popup_strV;']);
      SelectedIndex = get(p_popup, 'Value');
      ImageName= deblank(g_str(SelectedIndex,:));

  switch ImageName
    case 'kids'
     I=imread('kids.tif');
    case 'cameraman'
     I=imread('cameraman.tif');
%...
```

```
    case 'trees'
     I=imread('trees.tif');
    case 'circuit'
     I=imread('circuit.tif');
  end
end % end if

handle1=findobj('Tag','popup_Mode_tag');
if ~isempty(handle1)
      eval(['g_str = popup_Mode_strV;']);
      ProcessImage(I, get(p_popupM, 'Value'));
end % end if
%=======================================
function Help
HelpFigure=figure('Color',[0.71 0.71 0.71], ...
                  'MenuBar', menubar,...            % remove the menubar
                  'PaperType','a4letter', ...       %'Resize', 'Off',...
% disable resize property
                  'Position',[150,150, 200,200],...
                  'Resize', 'off',...
                  'Numbertitle','off',...
                  'Name', 'Demo GUI: Help',...
                  'Tag','f_HelpFigure');
SideFrame=uicontrol('Parent', HelpFigure,...
                    'BackgroundColor',[0.5 0.5 0.5], ...
                    'Position', [2 2 198 198],...
                    'Style', 'frame',...
                    'Tag','popup_tag');
txt_box=uicontrol('Parent', HelpFigure,...
                 'BackgroundColor',[0.5 0.5 0.5], ...
                  'String', 'Write some blabla here ...',...
                  'Position', [4 20 150 150],...
                  'Style', 'text',...
                  'Tag','t_text');
%=======================================
function App_Exit
Tags = ['f_HelpFigure'
        'f_Mainfigure'];
 for i=1:size(Tags,1)
   handle = findobj('Tag', deblank(Tags(i,:)));
   if ~isempty(handle)
     close(handle);
   end
 end
```

4. خاتمة

و صلنا إلى ختام هذا الدرس و قد إطلعنا فيه على أساسيات تصميم الواجهات الرسومية في الـ matlab بطريقتين مختلفتين. الخلاصة أن الأداة يسيرة الإستخدام و لكنها لا توفر الحرية الكافية و المرونة التي قد نجدها عند التصميم البرمجي. يبقى الكثير من التفاصيل و المعلومات حول كيفية تصميم الواجهات و لم أذكر في هذا الدرس إلا الأساسيات التي قد نحتاجها لتعلم هذا النوع من البرمجة المثير و الممتع في الآن ذاته و للمزيد من التفاصيل يمكن مراجعة الرابط [1].

أخيرا أسئل الله عز و جل أن يجعل هذا العمل خالصا لوجهه الكريم و أن يكون حافزا لكل ذا علم و إن كان متواضع ليعمل على نشر علمه و مساعدة غيره لإنشاء جيل جديد لا يحني الجبين إلا لله عز وجل. فلقد ورثنا الذل عن آبائنا و لا داعي لنورِّثه لأبنائنا.

المراجع

[1]
www.mathworks.com/access/helpdesk/help/techdoc/index.html?/access/helpdesk/help/techdoc/ref/figure_props.html

لكل الإستفسارات, الإقتراحات و التصويب المتعلق بمحتويات هذا الكتاب بالإمكان الإتصال بي عبر البريد الإلكتروني التالي:

Ben_Laiid@laposte.net