الدرس الأول

ما هو?MatLab

هو أداة وبيئة تطوير برمجية مخصصة للمهام الحسابية، حيث تتوفر فيه الكثير من الوظائف والدوال الرياضية المبنية داخليا والتي تسهل حل مختلف أنواع المعادلات الرياضية. كما تساعد لغة برمجة على كتابة دوال وبرامج خاصة. بالإضافة للعديد من المميزات الأخرى به.

تتضمن استعمالات الـ MatLab المجالات التالية:

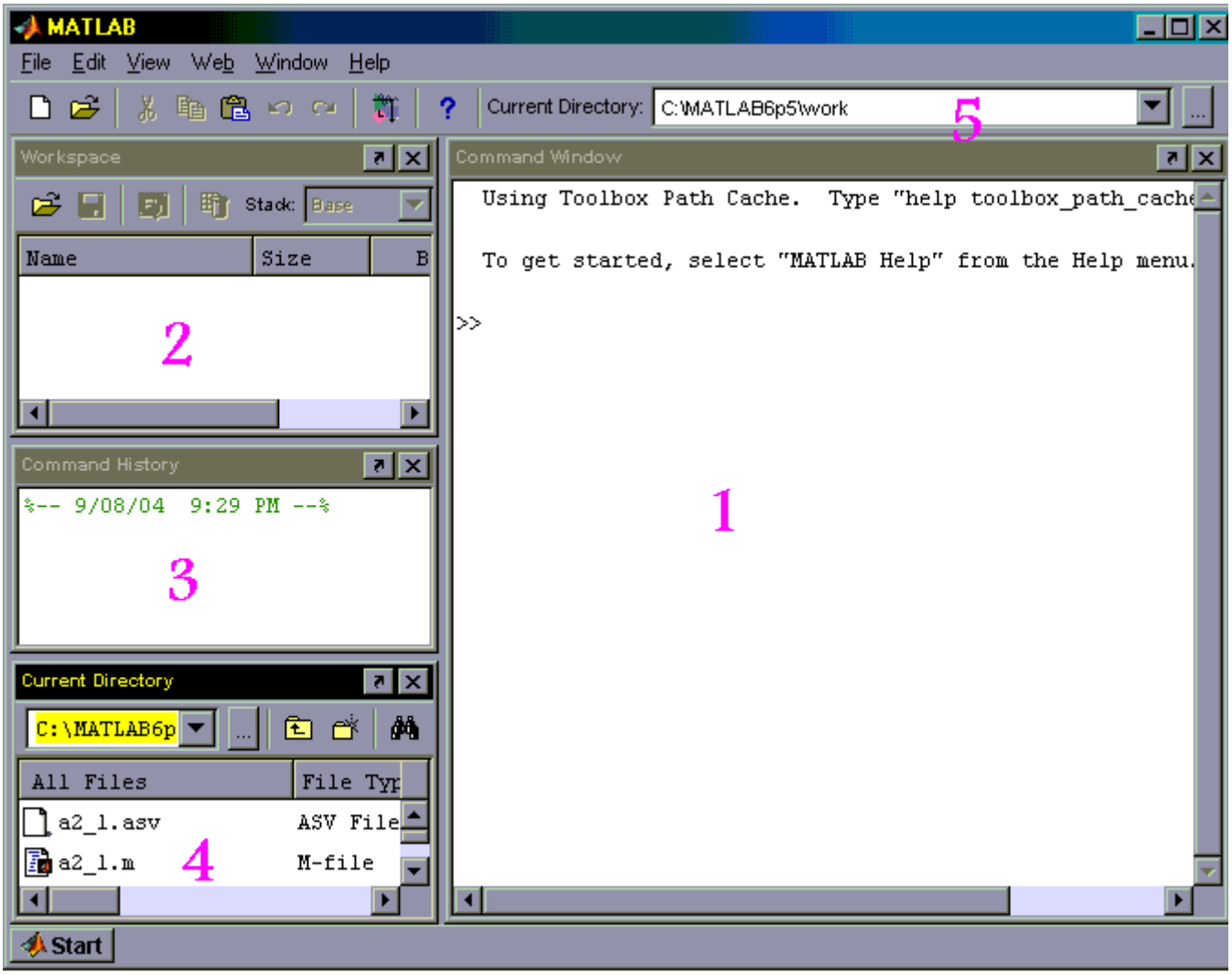
- الرياضيات و الحساب Math and computation
- تطوير الخوارزميات Algorithm development
- Data acquisition
- النمذجة والمحاكاة Modeling, simulation, and prototyping
- تحليل واستكشاف وتصوير البيانات Data analysis, exploration, and visualization
- الرسوم الهندسية والبيانية Scientific and engineering graphics
- بناء واجهات استخدام رسومية للتطبيقات المعدة Application development, including graphical user interface building

وللمزيد من المعلومات حول MatLab راجع موقع الشركة المنتجة للنظام:

http://www.mathworks.com/

واجهة التشغيل:

عند تشغيلك لـ MatLab سوف تظهر لك واجهة الاستعمال التالية:



تتكون الواجهة من مجموعة من الإطارات

1. إطار الأوامر Command Window

ومن خلاله يتم إدخال الأوامر للبرنامج، حيث يظهر المحث على الشكل (>>) ويتم كتابة الأمر بعده، وبما أن لغة MatLab هي لغة مفسرة Interpreted فإننا نحصل على الاستجابة فور الانتهاء من كتابة البرنامج، ولكن يمكن تجنب إظهار النتيجة لكل أمر بإلحاق الأمر بفاصلة منقوطة;

2. إطار منطقة العمل Workspace

حيث يظهر جميع المتغيرات المستعملة في جلسة العمل الحالية.

3. إطار الأوامر السابقة Command History حيث يتم عرض جميع الأوامر التي سبق إدخالها في جلسات عمل سابقة.

4. إطار المحلد الحالي Current Directory

في هذا الإطار يتم عرض جميع الملفات الموجودة في مجلد العمل الحالي والذي يكون عادة C:MATLAB6p5work حيث يوجد به البرامج التي سنقوم بتشغيلها.

يمكن تعديل هذا المجلد لأي مجلد آخر من خلال المفتاح (...) المجاور لأسم المجلد في أعلى الإطار، أو من خلال نفس المفتاح الموجود على شريط الأدوات) منطقة رقم 5 في الصورة(

أما مفتاح Start الموجود أسفل الشاشة فهو شبيه لمفتاح start في نظام ويندوز، حيث يمكن من خلاله تشغيل بقية الأدوات المرافقة لبيئة.MatLab

ملاحظة :

قد تظهر لديك واجهة الاستعمال مختلفة بعض الشيء عن المعروضة في الصورة، أو قد ترغب أنت في إخفاء بعض الأطر أو جعلها خارج الواجهةundock

لجعل أي أطار خارجيا استعمل مفتاح  الموجود على الجانب الأيمن العلوي من الإطار، ولإعادة داخل الواجهة أختار من الإطار

View -> dock (window name)

ولإغلاقه استعمل مفتاح 

توفر الوثائق المرافقة لـ MatLab الكثير من المعلومات المفيدة حول MatLab ويمكنك البدا باستعراضها من خلال اختيار MatLab Helpمن قائمة.Help

---

خلال هذا الدرس تعرفنا على بيئة تطوير Matlab وأطر واجهة الاستخدام، في الدرس الثاني سوف نتعرف على أوامر MatLab الخاصة بالتعامل مع المتجهات.

الدرس الثاني:

المتجهات في MatLab

يوفر Matlab مجموعة من الأوامر التي تجعل إدخال المتجهات والتعامل معها أكثر سهولة، حيث تشبه الأوامر المستعملة في MatLab أسلوب كتابة المتجهات في الجبر.

خلال هذا الدرس سوف نوضح هذه الأوامر، وكيفية استعمالها.

معلومة:

كلمة MatLab هي اختصار لعبارة matrix laboratory أو مختبر المصفوفات.

إنشاء المتجهات:

ابسط طريقة لتعريف المتجهة هي بكتابة عناصر المتجهة يفصل بين كلا منها مسافة ومحصورة بقوسين مربعين [ ]

```
>> a = [1 2 3]

a =

     1     2     3

>>
```

لاحظ أن نتيجة الأمر الذي قمنا بكتابته قد ظهرت لنا مباشرة جرب نفس الأمر السابق ولكن أضف في نهايته فاصلة منقوطة ;

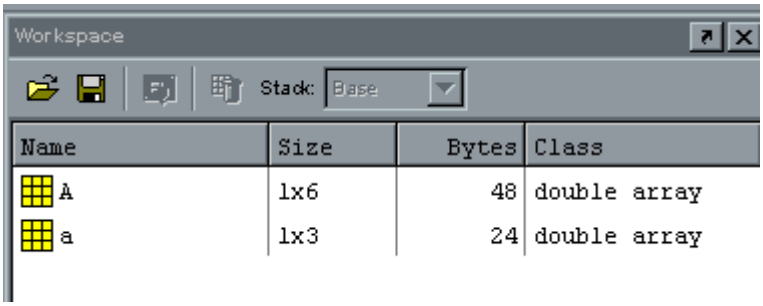
```
>> a = [1 2 3];
>>
```

هذه المرة لا تظهر نتيجة الأمر بعده.

كما يمكن تعريف المتجهة من خلال تحديد القيمة الأولى : قيمة الزيادة: القيمة الأخيرة

```
>> A = [0: 2: 10]

A =

     0     2     4     6     8    10
```

في إطار منطقة العمل Workspace لاحظ المتغيرات المعرفة حاليا في جلسة العمل



كما بالشكل أعلاه يظهر حتى الآن متغيرين هما a و A

MatLab حساس لحالة الأحرف Case-sensitive لذا فإن المتغير a مختلف تماما عن المتغير A.

ملاحظة:

يمكنك كتابة الأمر Whos لعرض المتغير المعرفة في جلسة العمل الحالية في إطار الأوامر Command Window

عرض المتجهات:

لعرض محتويات أي متجه نقوم بكتابة اسم المتجه ثم نضغط على مفتاح الإدخال Enter

```
>> a

a =

     1     2     3

>>
```

أو يمكن عرض القيمة الثانية في المتجه فقط من خلال الأمر:

```
>> a(2)

ans =

     2

>>
```

لاحظ المتغير الجديد الذي تم أنشاؤه ans. في كل مرة يتم فيها كتابة امر تنتج عنه قيمة بدون تعيين هذه القيمة إلى متغير فأن تلك القيمة سوف تحمل في المتغير ans.

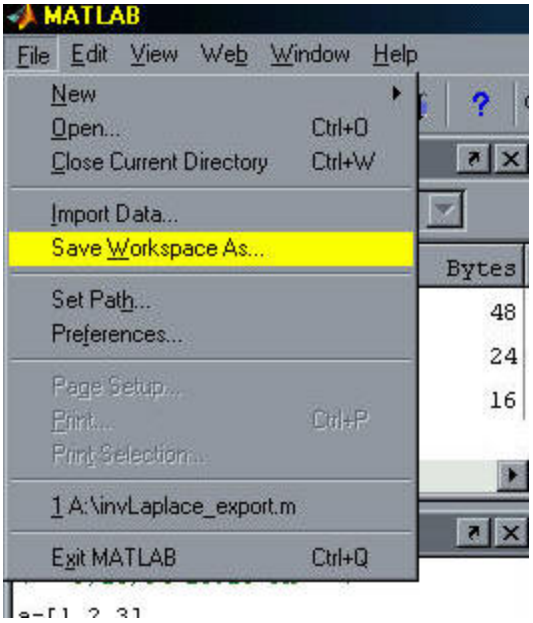
لعرض أول 4 قيم بالمتجه، أو لعرض القيمة الأولى والرابعة فقط:

```
>> A(1:4)

ans =

     0     2     4     6

>> A(1:3:4)

ans =

     0     6
```

الآن وبعد أن تعرفنا على كيفية التعامل مع المتجهات في بيئة MatLab ، نختتم الدرس بحفظ جلسة العمل الحالية للرجوع لها في أي وقت آخر

حفظ واسترجاع جلسة العمل:

لحفظ جلسة العمل WorkSpace أختار من قائمة **File -> Save workspace As**



ومن خلال مربع حوار Save As أختار اسم مناسب لملف جلسة العمل مثلا: MyFirstMat

الملف سوف يحفظ في مجلد العمل والذي يكون عادة مجلد **C:MATLAB6p5work** ويعطى الامتداد MAT

وعند تشغيل MatLab مرة ثانية يمكن استعادة ملف جلسة العمل من خلال **File -> Open** ثم اختار الملف ذو الامتداد Mat الذي حفظت به جلسة العمل.

الدرس الثالث:

المصفوفات فيMatLab

خلال هذا الدرس سوف نستكمل دراسة المزيد من أوامر Matlabوالمتعلقة بإنشاء المصفوفات والتعامل معها.

إنشاء المصفوفات:

طريقة تعريف المصفوفات في MatLab قريبا جداً إلى طريقة تعريف المتجهات، نبدأ مباشرة مع أول مثال:

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

D =

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

لاحظ الفرق بين فصل الأعداد بمسافة أو فاصلة منقوطة، جرب هذا الأسلوب كذلك:

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

D =

1 2 3
4 5 6
7 8 9

>>
```

كما يوجد عدد من الدوال لإنشاء مصفوفات خاصة:

1. دالة pascal لإنشاء مصفوفة متناظرة symmetric
2. دالة magic لإنشاء مصفوفات يتساوى فيها مجموع كل الصفوف والاعمدة.
3. دالة zeros لإنشاء مصفوفة صفرية.
4. دالة ones لإنشاء مصفوفة كل عناصرها تساوي 1

لاحظ الامثلة التالية

```
>> P = pascal(3)

P =

1 1 1
1 2 3
1 3 6

>> M= magic(3)

M =

8 1 6
3 5 7
4 9 2

>> z= zeros(2, 3)

z =

0 0 0
0 0 0

>> o = ones(2, 4)

o =

1 1 1 1
1 1 1 1

>>
```

العمليات الحسابية على المصفوفات:

كما ذكرنا سابقا فإن MatLab يجعل التعامل مع المتجهات والمصفوفات أكثر سهولة، جرب الأمثلة التالية:

```
>> Sum = D + P
```

```
>> Sub = P - D

>> D = D + 2

>> P2 = P * 2

>> Mult1 = P * D

>> Mult2 = P .* D
```

الأمر الأول: يجمع كلا من P و D وينتج عنه المصفوفةSum

الأمر الثاني: ناتج طرح D من P في المصفوفةSub

الأمر الثالث: يضيف 2 إلى كل عنصر من عناصر المصفوفةD

الأمر الرابع: ينتج عنه مصفوفة Mult1 والتي يحفظ بها ناتج ضرب Pفي D

الأمر الخامس) :*لاحظ النقطة قبل علامة الضرب* (هذا الأمر سينتج عنه مصفوفة Mult2 والتي هي عبارة عن حاصل ضرب كل عنصر في P في العنصر المقابل له فيD

جرب أيضا الأمرين التالين ولاحظ الفرق في الناتج

```
>> M

M =

8  1  6
3  5  7
4  9  2

>> MM = M ^ 2
```

```
MM =

91  67  67
67  91  67
67  67  91

>> M2 = M .^ 2

M2 =

64   1  36
 9  25  49
16  81   4

>>
```

M ^ 2 يعني ضرب المصفوفة في نفسها

M .^ 2 يعني ضرب كل عنصر في المصفوفة في نفسه.

لإيجاد محورة المصفوفة Transpose

```
>> M'

ans =

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

لإيجاد معكوس المصفوفة Inverse

```
>> inv(M)

ans =

0.1472   -0.1444    0.0639
```

```
-0.0611  0.0222  0.1056
-0.0194  0.1889 -0.1028

>>
```

<u>للتعرف على حجم المصفوفة</u>

```
>> size(z)

ans =

2  3


>> size(o)

ans =

2  4

>>
```

العدد الأول يمثل عدد الأسطر والثانيالدرس الرابع:

كثير الحدود فيMatLab

<u>أهداف الدرس:</u>

التعرف على كيفية تمثيل كثير الحدود في MatLab ، وكيفية التعامل معها.

يوفر Matlab عدد من الدوال المبنية داخليا لتسهيل التعامل مع كثير الحدود Polynomials ، حيث يتم تمثيلها كمتجه، مثلا لتمثيل معادلة كثير الحدود التالية:

$$S^4 + 3S^3 - 15S^2 - 2S + 9$$

نعرف المتجه التالي:

```
>> x = [1 3 -15 -2 9]

x =

1 3 -15 -2 9
```

كذلك لتمثيل $S^4 - 2$

```
>> Z = [1 0 0 0 -2]

Z =

1 0 0 0 -2
```

حساب قيمة كثير الحدود عند قيمة محددة:

لكي نحسب قيمة كثير الحدود الأول x عند قيمة s=3 مثلا، يمكن استعمال دالةpolyval

```
x =

     1     3   -15    -2     9

>> polyval(x, 3)

ans =

    30

>>
```

احسبها وتأكد من الناتج(:

إيجاد جذور كثير الحدود:

يقصد بالجذور قيم المتغير s التي تجعل القيمة الكلية للمعادلة تساوي 0

```
>> roots(x)

ans =

   -5.5745
    2.5836
   -0.7951
    0.7860

>>
```

والعكس:

يعني لاكتشاف معادلة كثير الحدود لجذور معلومة، الدالة هنا هيpoly

```
>> poly(ans)

ans =

    1.0000    3.0000  -15.0000   -2.0000    9.0000

>>
```

ضرب وقسمة كثير الحدود:

لضرب معادلتين كثير حدود في بعضهما استعمال دالة conv وللقسمة الدالةdeconv

```
>> x

x =

     1     3   -15    -2     9

>> z

z =

     1     0     0     0    -2

>> mu = conv(x, z)

mu =

     1     3   -15    -2     7    -6    30     4   -18

>> [d, r] = deconv(mu, x)

d =

     1     0     0     0    -2


r =

     0     0     0     0     0     0     0     0     0
```

عند استعمال deconv لقسمة كثيري حدود فأنه ينتج متجهين:

- الأول d ناتج القسمة.
- الثاني r باقي القسمة) وفي المثال السابق كان الباقي من القسمة متجه صفري.(

يمثل عدد الأعمدة

الدرس الخامس:

أوامر مفيدة فيMatLab

وقفة قصيرة من الأمور الرياضية ودوالها التي تكلمنها عنها في الدروس السابقة، لنتعلم المزيد عن كيفية استعمال matlab والأوامر الأساسية به .

مسح إطار الأوامر:

أثناء عملنا قد نرغب من وقت لآخر في مسح كل ما هو موجود على إطار الأوامر. يوجد طريقتين لذلك:

1. إذا كنت من محبي استعمال الفأرة أختار Edit-> Clear Command Window
2. أما إذا كنت تفضل استعمال لوحة المفاتيح فأكتب `clc` ثم Enter .

ملاحظة:

مسح إطار الأوامر لن يحذف المتغيرات التي تم تعريفها خلال جلسة العمل، أنظر لإطار جلسة العمل Workspace ستجد أن المتغيرات لم تتغير أو تحذف. يمكنك أيضا استعمال الأمر `Whos` *لعرض المتغير المعرفة في جلسة العمل الحالية للتأكد، إذا كان إطار جلسة العمل غير ظاهر لديك.*

حذف جميع المتغيرات المعرفة في جلسة العمل:

وهنا أيضا لدينا طريقتين:

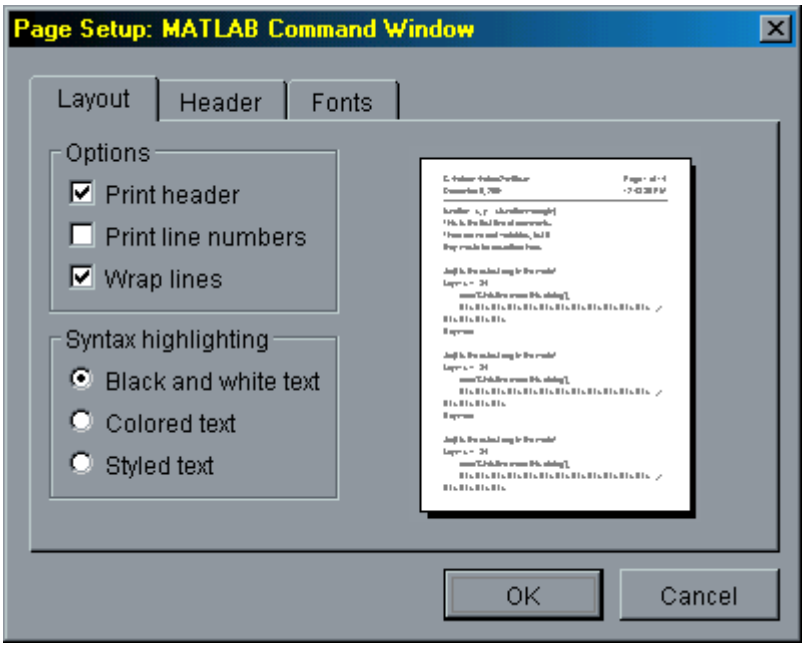
1. إذا كنت من محبي استعمال الفأرة أختار Edit-> Clear Workspace
2. أما إذا كنت تفضل استعمال لوحة المفاتيح فأكتب `clear` ثم Enter.

طباعة محتويات إطار الأوامر:

لطباعة محتويات إطار الأوامر كاملة اختارPrint -> File

ولطباعة الجزء المحدد فقط من الإطار أختار Print -> File selection

وللتحكم في تنسيق المخرجات من الطباعة اختار Page -> File Setupحيث تظهر لك مربع حوار page setup الذي يمكن من خلاله التحكم في تنسيق الصفحة مثل ظهور رأس الصفحة Headerأو لا ومحتويات هذا الرأس، ظهور أرقام للأسطر، والخطوط fonts المستعملة إثناء الطباعة.



ملاحظات مفيدة:

- خلال عملك على matlab تذكر أنه حساس لحالة الأحرف case sensitive، لذلك فإن `Clear` ليست مثل `clear` على سبيل المثال .
- يمكن أن تكتب أكثر من أمر على سطر واحد في MatLab شرط أن تفصل بينهما بفاصلة منقوطة.

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

```
C =

7  9  11  13  15

>>
```

كما يمكن كتابة الأمر الواحد على سطرين منفصلين،) خاصة إذا كان عرض الشاشة لا يتسع له) بأن نضع ثلاث نقاط (...) عند نهاية السطر الأول.

```
>> D = [ 2 5 2 4 1 66 8 44 88 66 ...
5 7 44 88 44 787 56 66 4]

D =

Columns 1 through 12

2  5  2  4  1  66  8  44  88  66  5  7

Columns 13 through 19

44  88  44  787  56  66  4

>>
```

خلال العمل على MatLab فإن الأوامر التي تكتبها في أطار الأوامر تحفظ في حافظة الـ History وقد تسأل ما الفائدة من هذا؟

الفائدة منه أنه يمكنك إعادة استدعاء إي من هذه الأوامر السابقة وتنفيذها من جديد، وذلك من خلال الضغط المتكرر على مفتاح السهم للأعلى حتى تصل إلى الأمر الذي تريد تكراره، وذلك دون الحاجة إلى إعادة كتابته مرة ثانية.

- للحصول على المساعدة حول أي أمر أو دالة في MatLab مباشرة في إطار الأوامر أكتب help ثم اسم الأمر أو الدالة وسوف تظهر لك كل المعلومات التي تريدها حول ذلك الامر، جرب مثلاhelp sin

الدرس السادس:

البرمجة فيMatLab

كما ذكرنا في الدرس الأول من هذه السلسلة فإن MatLab هو بيئة تطوير برمجية تحوى العديد من الدوال الجاهزة، بالإضافة إلى إمكانية كتابة برامج ودوال خاصة بنا حسب الحاجة. خلال هذا الدرس سوف نتعرف على الأوامر البرمجة في.MatLab

---

الحملة الشرطية: if

تستخدم للاختيار بين أمرين حسب شرط محدد

الصيغة العامة:

```
                                    if <condition>

   <program1>

else

   <program2>

end
```

في حالة تحقق الشرط condition يتم تنفيذ الكود في program1وإذا لم يتحقق الشرط يتم تنفيذ الكود في program2

مثال:

```
>> if n < 0
  disp('n is negative')
else
  disp('n is positive')
end
n is positive
>> n
```

```
n =

71

>>
```

يمكن أن تأخذ جملة if شكلا أكثر تداخلا باستعمال أكثر من مستوي لـelseif

```
if expression1
  statements1
elseif expression2
  statements2
else
  statements3
end
```

أو يمكن استعمال جملة switch التي لها نفس العمل

جملة switch

الصيغة العامة:

```
switch switch_expr
  case case_expr
      statement,...,statement
  case {case_expr1,case_expr2,case_expr3,...}
      statement,...,statement
  ...
  otherwise
      statement,...,statement

End
```

حيث:

switch_expr هو المتغير (أو التعبير) الذي سيتم اختبار قيمته.

`case_expr`أحد القيم التي يمكن أن يأخذها المتغيريمكن أن تتضمن الحالة الواحدة أكثر من قيمة، وإذا كانت القيمة للـ `switch_expr`غير مدرجة في اي حالة ينتقل التنفيذ للقسم `otherwise`

الحلقات التكرارية:

عندما نرغب في تكرار أمر معين (أو أكثر) عدة مرات، فإن أفضل طريقة لعمل ذلك هو بوضع هذا الأمر داخل حلقة تكرارية.

في MatLab يوجد نوعين فقط من الحلقات التكرارية:

1. حلقة for

وتستخدم عندما يكون المطلوب هو التكرار لعدد محدد من المرات.

الصيغة العامة

```
for variable = expression
  statement
  ...
  statement
end
```

مثال: حلقة بسيطة سوف تتكرر 4 مرات

```
>> for j=1:4
j
end

j =

1


j =
```

```
2


j =

3


j =

4

>>
```

2. حلقة while

حيث يكون التكرار هنا مرتبط بتحقق شرط ما، فإذا لم يعد الشرط محقق تنتهي الحلقة

الصيغة العامة:

```
while expression
   statements
end
```

مثال :هذا البرنامج يوجد أول عدد صحيح مضروبه !n مكون من 100 خانة عشرية

```
>> n = 1;
while prod(1:n) < 1e100
n = n + 1;
end
>> n
```

```
n =

70
```

ملاحظة:

لغة MatLab هي لغة مفسرة Interpreted أي أن كل أمر يتم ترجمته للحاسوب قبل تنفيذه مباشرة، لذا فإن استعمال الحلقات التكرارية سوف يجعل البرنامج أكثر بطأ، ويفضل استعمال الاوامر والدوال الجاهزة لـ MatLab كلما أمكن ذلك.

break :

يستخدم هذا الأمر لإيقاف تنفيذ حلقة تكرارية وإعادة التحكم للبرنامج أو للحلقة الخارجية عند وجود حلقات متداخلة.

continue:

يقوم هذا الأمر بوقف التكرار الحالي للحلقة iteration ويبدأ في التكرار التالي له.