أما للحصول على مقلوب المصفوفة نستخدم الأمر

D=**inv**(A)

لتقسيم مصفوفتين A/B نأخذ مقلوب B و نضربه بـ A، تابع المثال التالي...

```
» A=[3 4 5;6 7 8; 5 8 6];
» B=[3 4 7;5 6 11;7 7 8];
» inv(B)*A
ans =
   2.0833    1.5833    0.2500
  -2.4167   -0.9167   -0.2500
   0.9167    0.4167    0.7500
```

لاحظ أنه إذا ضربنا مقلوب مصفوفة بالمصفوفة نفسها نحصل على المصفوفة الواحدية ...

```
» det(A)
ans =
   15
» inv(A)
ans =
  -1.4667    1.0667   -0.2000
   0.2667   -0.4667    0.4000
   0.8667   -0.2667   -0.2000
» A*inv(A)
```

```
ans =
   1.0000         0   -0.0000
   0.0000    1.0000   -0.0000
  -0.0000    0.0000    1.0000
» inv(A)*A
ans =
   1.0000   -0.0000   -0.0000
        0    1.0000    0.0000
```

```
-0.0000   -0.0000    1.0000
```

رفع المصفوفة إلى قوة ...

إذا كانت المصفوفة **A** مربعة و **p** عدد صحيح موجب فعند رفع المصفوفة A للقوة p أي عند تنفيذ العملية (A^p)يتم ضرب المصفوفة بنفسها p مرة. إذا كانت **p** عدد صحيح سالب فإنه عند تنفيذ العملية (A^(-p)) يتم ضرب مقلوب المصفوفة inv(A) بنفسه p مرة. باستخدام المعامل (.^) يتم رفع كل عنصر من عناصر المصفوفة إلى القوة p .

```
» A=[3 4 5;6 7 8; 5 8 6];
» A^3
ans =
      1039      1408      1392
      1792      2431      2400
      1648      2240      2207
» A.^3
ans =
    27    64   125
   216   343   512
   125   512   216
» A.^(-3)
ans =
    0.0370    0.0156    0.0080
    0.0046    0.0029    0.0020
    0.0080    0.0020    0.0046
```

بعض الوظائف الأخرى ...

- **sqrt(A)** يحسب مكافئ التابع A أي A.^(1/2) .
- **sqrtm(A)** يقوم بحساب الجذر التربيعي للمصفوفة A أي مكافئ المصفوفة A^(1/2) و لكن بدقة أكبر.
- **expm(A)** يقوم هذا التابع بحساب e^A .
- **logm(A)** يقوم هذا التابع بحساب log(A).

- **trace(A)** يقوم هذا التابع بإيجاد حاصل جمع العناصر القطرية.

```
» A=[3 4 5;6 7 8; 5 8 6];
» sqrt(A)
ans =
    1.7321    2.0000    2.2361
    2.4495    2.6458    2.8284
    2.2361    2.8284    2.4495
» sqrtm(A)
ans =
   0.7724 + 0.5278i   1.0484 + 0.1047i   1.0345 - 0.4461i
   1.3326 - 0.3447i   1.8088 + 0.6073i   1.7848 - 0.4428i
   1.2265 - 0.0447i   1.6648 - 0.6937i   1.6427 + 0.7818i
» expm(A)
ans =
  1.0e+007 *
    1.0239    1.3898    1.3714
    1.7666    2.3979    2.3661
    1.6259    2.2070    2.1777
» logm(A)
ans =
  -0.0773 + 2.5671i   1.4520 - 0.7798i   0.2857 - 0.7694i
   0.8482 - 0.9911i   1.3082 + 1.7963i   1.1752 - 1.3275i
   1.3496 - 0.9122i   0.5103 - 1.2382i   1.4771 + 1.9198i
» trace(A)
Ans =
        16
```

أمر تنسيق Format Command ...

يتحكم أمر التنسيق بتنسيق ظهور القيم الناتجة عن عمل البرنامج و ينحصر تأثير الأمر في كيفية ظهور هذه الأرقام على الشاشة فقط و ليس له علاقة بطريقة حساب MATLAB لهذه القيم أو طريقة تخزينه لهم و سنبين فيما يلي بعض أوامر التنسيق المستخدمة في MATLAB …

إذا كانت لدينا مصفوفة X …

```
» X=[4/3 1.2345e-6]

X =

  1.3333    0.0000
```

1. أمر تنسيق format short يحدد للعدد خمس خانات مع فاصلة عشرية عائمة، و هو نفس أمر التنسيق الافتراضي الذي يستعمله MATLAB – لاحظ المثال السابق.

2. أمر التنسيق format short e يعطي الشكل الأسي للعدد و بتحديد خمس خانات للعدد مع فاصلة عائمة.

```
» format short e

» X

X =

  1.3333e+000  1.2345e-006
```

3. أمر التنسيق format long يحدد لعدد 15 خانة مع فاصلة عشرية عائمة.

```
» format long
» X
X =
  1.33333333333333   0.00000123450000
```

4. أمر التنسيق format long e يعطي الشكل الأسي للعدد مع تحديد 15 خانة و فاصلة عشرية عائمة.

```
» format long e
» X
X =
   1.333333333333333e+000    1.234500000000000e-006
```

لإعادة التنسيق إلى الوضع الافتراضي إما أن نكتب format أو format short. لاحظ انه يجب كتابة الأوامر السابقة بأحرف صغيرة ليتعرف عليها MATLAB لاحظ المثال التالي.

```
» format
» X
X =
   1.3333    0.0000
» Format
??? Undefined variable or capitalized internal function Format; Caps Lock may be on.
```

كثيرات الحدود Polynomials ...

يوجد في MATLAB عدد من التوابع لإجراء العمليات على كثيرات الحدود سنستعرض بعض هذه التوابع

إدخال كثير حدود ...

يتم كتابة كثير الحدود في MATLAB على شكل صف يحتوي على أمثال الحدود مرتبة حسب القوة الأكبر ثم الأصغر و هكذا مثلاً لإدخال كثير الحدود التالي:

$$P(x)=x^3 - 2x + 5$$

نكتب في MATLAB ما يلي:

```
» P=[1 0 -2 -5]
P =
    1    0   -2   -5
```

جذور كثير الحدود ...

لإيجاد جذر كثير الحدود نستعمل التابع **rotts** ، فمثلاً لإيجاد جذور كثير الحدود P نكتب:

```
» r=roots(P)
r =
   2.0946
  -1.0473 + 1.1359i
  -1.0473 - 1.1359i
```

يخزن MATLAB بشكل افتراضي الجذور في مصفوفة عمود. لإعادة تشكيل كثير الحدود بمعرفة جذوره نستعمل التابع **poly** ، فمثلاً:

```
» P2=poly(r)
P2 =
```

```
   1.0000   -0.0000   -2.0000   -5.0000
```

و يمكن استعمال التابع poly لإيجاد كثير الحدود المميز لمصفوفة، على سبيل المثال لإيجاد كثير الحدود المميز للمصفوفة A المبينة:

```
» A=[1.2 3 -0.9;3 1.75 6;9 0 1];
» poly(A)
ans =
   1.0000   -3.9500    4.1500 -169.2750
```

يمكن حساب جذور كثير الحدود المميز هذا باستعمال التابع **roots**.

حساب قيمة كثير الحدود ...

يمكن حساب قيمة كثير الحدود عند نقطة معينة باستعمال التابع **polyval**، فمثلاً لحساب قيمة كثير الحدود P عند النقطة x=5 نكتب:

```
» polyval(P,5)
ans =
  110
```

يمكن إيجاد قيمة كثير الحدود أيضاً من أجل مصفوفة معينة x (بدلاً من نقطة واحدة) باستعمال التابع **polyvalm**، فمثلاً لحساب قيمة كثير الحدود P عند المصفوفة x نكتب كثير الحدود على الشكل:

$$P(x) = x^3 - 2\,x + 5\,I$$

حيث **I** هي المصفوفة الواحدية، فإذا كانت قيمة x :

x=[2 4 5;-1 0 3;7 1 5]

فإن:

```
» x=[2 4 5;-1 0 3;7 1 5]
x =
    2     4     5
   -1     0     3
    7     1     5
» p=[1 0 -2 5]
 P =
    1     0    -2     5
```

```
» y=polyvalm(p,x)
y =
  387   179   439
  111    91   136
  490   253   649
```

جداء كثيرات الحدود ...

لجداء كثيرات الحدود نستعمل التابع **conv**، فمثلاً لحساب جداء كثيري الحدود:

$a(s)= s^2 + 2 s + 5$

$b(s)= 4s^2 +5s + 6$

نكتب أولاً التابعين على الشكل:

```
» a=[1 2 3];
» b=[4 5 6];
» c=conv(a,b)
c =
    4    13    28    27    18
```

حيث c هي أمثال كثير الحدود الناتج عن عملية الضرب.

قسمة كثيرات الحدود ...

لتقسيم كثيرات الحدود نستعمل الأمر **deconv**. لحساب قسمة c/a كثير الحدود c على a فيجب أن يكون الناتج b نكتب:

```
» [q,r]=deconv(c,a)
q =
    4    5    6
r =
    0    0    0    0    0
```

حيث يخزن في الشعاع q نتيجة القسمة و في الشعاع r باقي القسمة.

جمع كثيرات الحدود ...

إذا كان كلا كثيري الحدود لهما نفس البعد فيتم وفق مايلي :

$a(x)=x^3 + 2x^2 + 3x +4$ $b(x) = x^3 +4x^2 +9x +16$

```
>> d = a + b
d =
   2  6  12  20
```

إذا كان كثيري الحدود من درجتين مختلفتين نقوم بتوسيع كثير الحدود ذي الدرجة الأدنى عبر إضافة معاملات صفرية وفق مايلي:

c(x) = 4x + 10

```
>> e = a + [ 0  0  c ]
e =
    1  2  7  14
```

مكاملة و اشتقاق كثيرات الحدود ...

التابع **polyder** يحسب مشتق أي كثير حدود ما للحصول على مشتق كثير الحدود، نكتب:

```
» P=[1 0 -2 -5]
P =
   1    0    -2    -5
» q=polyder(P)
q =
   3    0    -2
```

يمكن بواسطة التابع polyder حساب مشتق كثيري حدود A*B أيضاً فمثلاً :

```
» a=[1 3 5];
» b=[2 4 6];
» c=polyder(a,b)
c =
   8    30    56    38
```

يمكن مكاملة كثير الحدود عن طريق التابع **polyint** وفق التالي:

```
>> polyint(q,5)
q =
   1   0   -2   5
```

يمكن إجراء عملية تفريق الكسور لكثير حدود كسري و التي تتم كما يلي:

$$\frac{B(X)}{A(X)} = \frac{R(1)}{X - P(1)} + \frac{R(2)}{X - P(2)} + \ldots\ldots + \frac{R(n)}{X - P(n)} + K(X)$$

```
>> A=[1 2];B=[2 4 6 1];
  [R , P , K] = RESIDUE (A,B)
R =
  -0.1924 – 0.0825i
  -0.1924 + 0.0825i
   0.3849
P =
  -0.9060 + 1.3559i
  -0.9060 – 1.3559i
  -0.1880
K =
    []
```

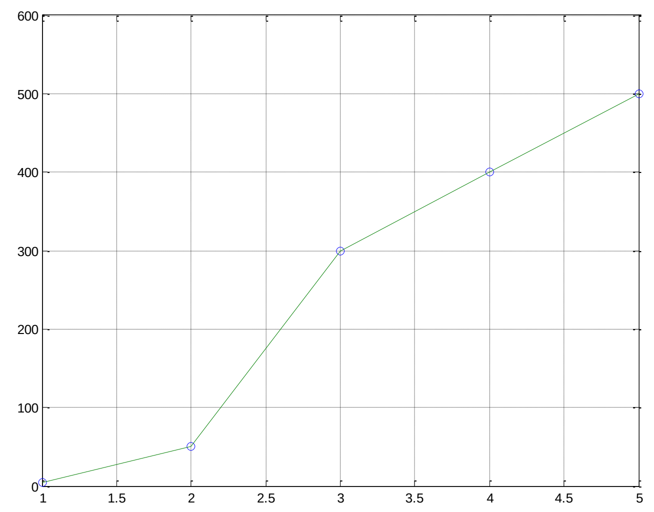
إلباس المعطيات بمنحني...

```
p = polyfit(x,y,n)
```

حيث n درجة المنحني المراد إلباسه

```
x = [1 2 3 4 5]
y = [5 50  300 400 500]
p = polyfit(x,y,5)
x2 = [1 2 3 4 5]
y2 = polyval(p,x2)
plot(x,y,'o',x2,y2)
grid on
y=
     5 50 300 400 500
P=
```

```
  7.1731  -86.5541  340.1262 -419.9787        0  164.2336
x2=
  1 2 3 4 5
= y2
     5.0000   50.0000  300.0000  400.0000  500.0000
```



المخططات ...

فـي MATLAB إمكانيـة كبيـرة لإظهـار الأشـعة و المصـفوفات بواسـطة المخططـات و يمكـن بواسـطة MATLAB طباعـة هـذه المخططـات. سنسـتعرض فيمـا يلـي بعـض التوابـع الأساسـية لتوليـد المخططـات و سنستعرض فيما بعد و بشكل مفصل توليد المخططات.

لتوليد مخطط نستعمل أمر **plot** و لهذا الأمر أشكال مختلفة تبعاً للمتحولات الداخلة في هذا الأمر. فإذا كـان y عبـارة عـن شـعاع فـإن الأمـر **plot(y)** يولـد مخطـط لعناصـر y بدلالـة دليـل هـذه العناصـر. أما إذا كان x,y عبارة عن شعاعين فإن المر **plot(x,y)** يولد مخطط لعناصر y بدلالة عناصر x.

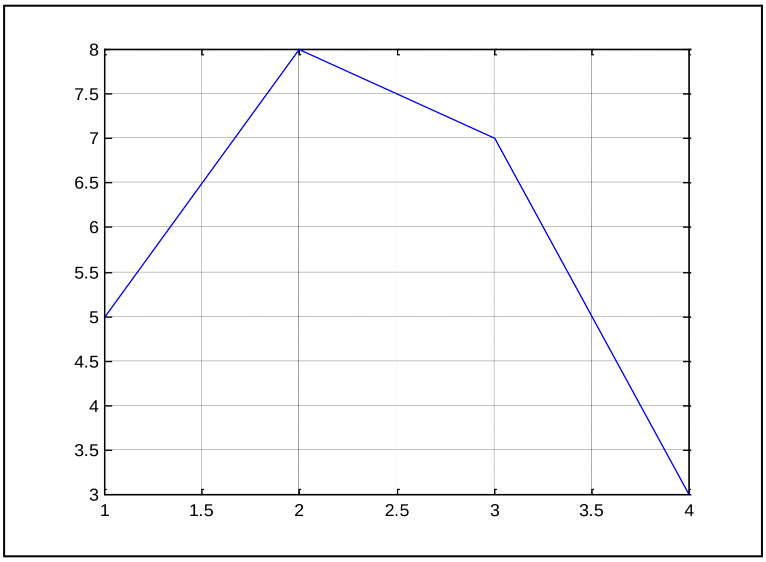
مثلاً إذا كان لدينا شعاعين x,y :

x=[3 7 5 9]   y=[5 8 7 3]

plot(y) يرسم النقاط (5,1) ، (8,2) ، (7,3) ، (3,4) أي كل نقطة بدلالة دليل هذه النقطة.

```
» x=[3 7 5 9]
```

```
x =
   3   7   5   9
» y=[5 8 7 3]
y =
   5   8   7   3
» plot(y)
» grid
» shg
```

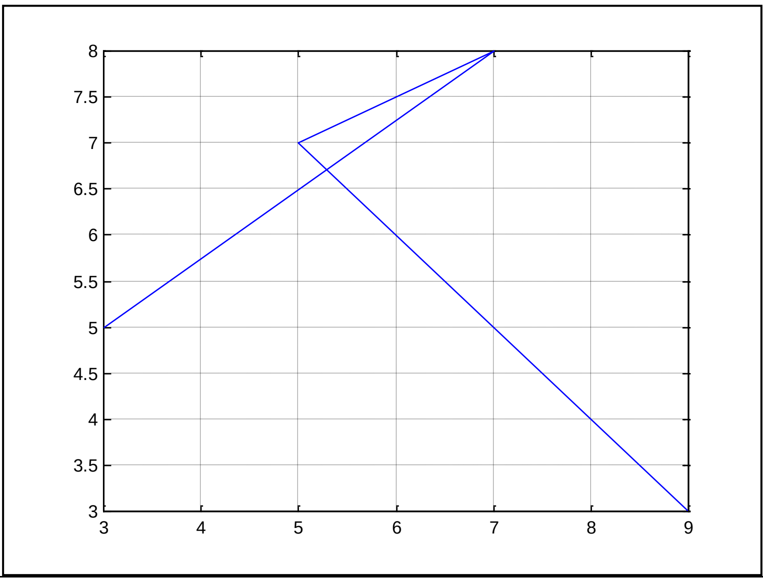


الشكل (2-2)

ملاحظة : لإظهار نافذة المخططات نستخدم الأمر **shg** و لوضع شبكة على المخطط نستخدم الأمر **grid**.

الأمر **plot(x,y)** يرسم النقاط (5,3) ، (8,7) ، (7,5) ، (3,9) أي عناصر y بدلالة عناصر x.

```
» plot(x,y)
» grid
```

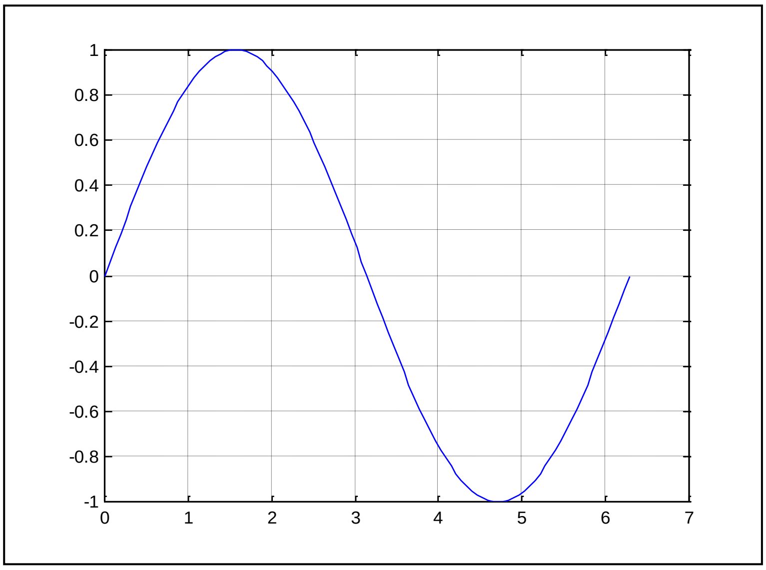


الشكل (3-2)

```
» shg
```

بشكل عام يتم توليد المخططات باستعمال تابع المجال، مثلاً لرسم التابع y بدلالة الشعاع t نكتب:

```
» t=0:pi/50:2*pi;
» y=sin(t);
» plot(t,y);
» grid;
» shg
```



الشكل (4-2)

يمكن توليد عدد من المخططات باستدعاء واحد لأمر **plot** عندها يعطي MATLAB لكل منحني من المنحنيات لون يأخذه من لائحة الألوان المعرفة مسبقاً، مثلاً بكتابة العلاقات ...

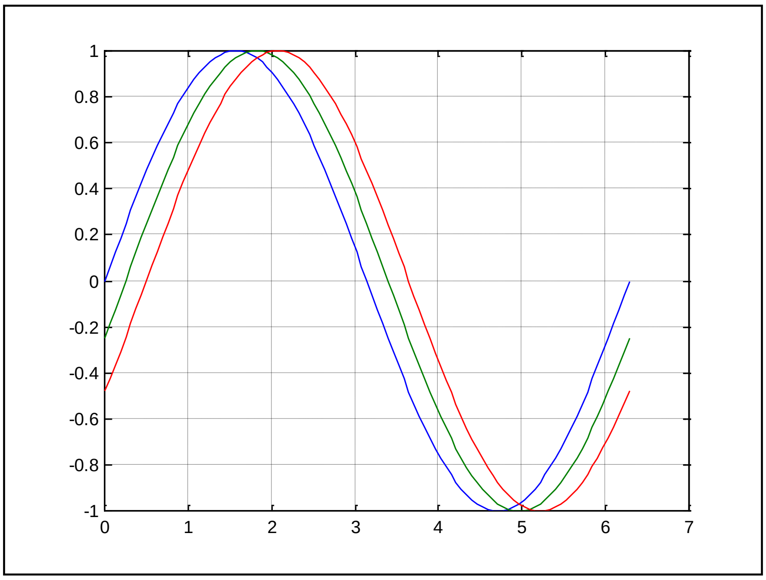
```
» t=0:pi/50:2*pi;
» y=sin(t);
» y1=sin(t-0.25);
» y2=sin(t-0.5);
» plot(t,y,t,y1,t,y2);
» grid
» shg
```

نجد أن MATLAB يرسم هذه المنحنيات على مخطط واحد و يعطي لكل منحني لون معين. يمكن تحديد لون الخط و شكل الخط و شكل الإشارات المحددة لكل نقطة من نقاط المعطيات باستعمال أمر:

plot(x,y,’color_style_marker’)

حيث أن:

Color هي سلسلة اللون و هي عبارة عن ‘c’ سماوي، ‘m’ الماسي، ‘g’ أخضر، ‘r’ أحمر، ‘y’ أصفر،’b’ أزرق، ‘w’ أبيض.



الشكل (5-2)

Style هي سلسلة شكل الخط و هي عبارة عن ‘-’ للخط المستمر،’--’ للخط المتقطع، ‘:’ للخط المنقط، ‘-.’ للخط المؤلف من خط و نقطة، ‘none’ لمنع ظهور الخط.

Marker الإشارات المحددة لنقاط المعطيات و هي عبارة عن ‘+’ ،’o’،’×’ .

مثال : لرسم منحني باللون الأصفر و بخط نقطي و بإشارة ‘+’ عند كل نقطة من نقاط المعطيات نكتب:

```
» plot(t,y,'y:+');
```

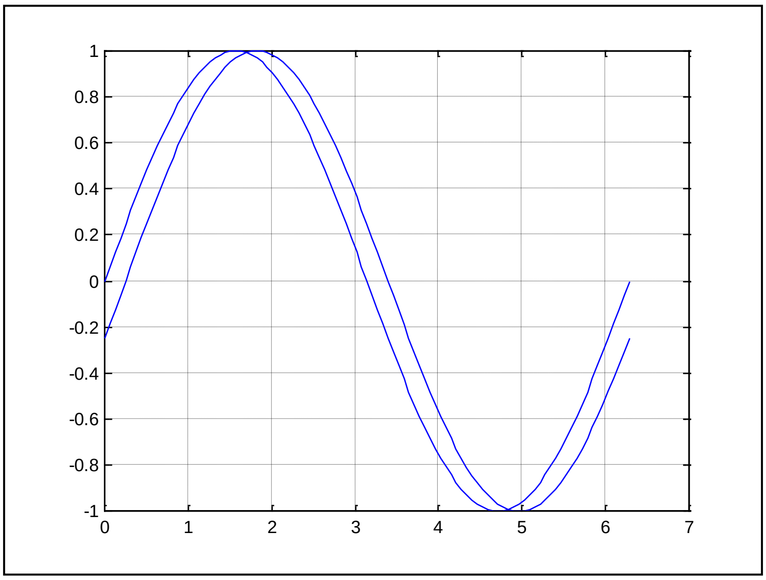
```
» grid
» shg
```

يفتح الأمر **plot** بشكل آلي نافذة صورة جديدة إذا لم تكن هناك نافذة صور مفتوحة أما إذا كانت هناك نافذة صور مفتوحة فإن MATLAB يستعمل هذه النافذة بشكل افتراضي أي يمسح المخطط الموجود و يرسم المخطط الجديد. لفتح نافذة جديدة و جعلها النافذة الفعالة نستخدم أمر **figure**. لإعادة تفعيل نافذة الصور الموجودة (أو أي نافذة إذا كان عدد النوافذ أكبر من الواحد) نستخدم الأمر **figure(n)** حيث أن n هو رقم النافذة الموجودة في شريط عنوان النافذة.

إضافة منحنيات جديدة إلى نافذة الصور ...

يمكن إضافة منحنيات جديدة إلى نافذة الصور باستخدام الأمر **hold on** عندما يصادف MATLAB هذه الأمر فإنه لا يزيل المخطط الموجود لكنه يضيف المنحنيات الجديدة و يعيد تحديد مجالات المحاور إذا لزم الأمر ، لاحظ المثال التالي:

```
» plot(t,y)
» hold on
» plot(t,y1)
» hold off
» grid
» shg
```



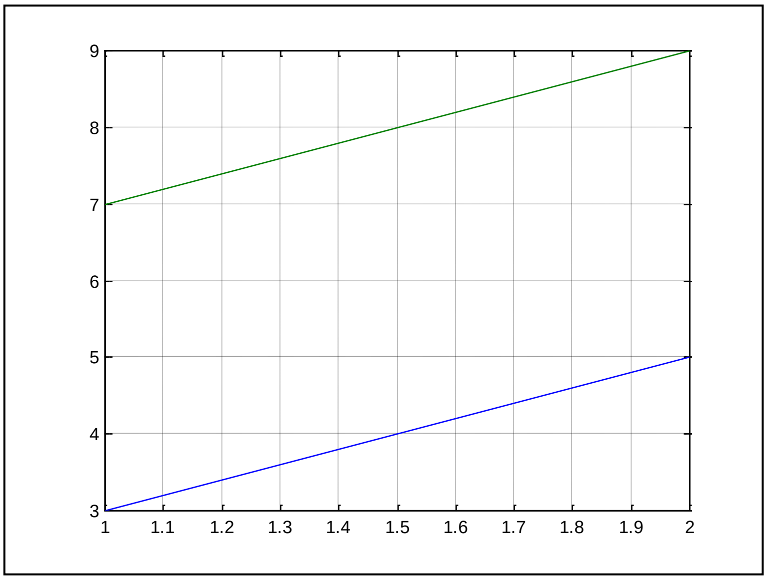
الشكل (7-2)

نلاحظ في هذه الحالة أننا استعملنا أوامر منفصلة لرسم المنحنيات y,y1 و بدون استعمال hold on سوف يقوم MATLAB عند رسم المنحني y1 بإزالة المنحني y. و أيضاً استعملنا الأمر **hold off** عند الانتهاء من الرسم على المخطط.

رسم المصفوفة الثنائية ...

إذا كان لدينا مصفوفة ثنائية y باستعمال الأمر **plot(y)** يقوم MATLAB برسم خط لكل عمود من أعمدة المصفوفة و بدلالة عدد أسطر المصفوفة.أي يقسم المحور x إلى عدد من الأقسام يساوي عدد أسطر المصفوفة.

```
» y=[3 7;5 9]
y =
    3    7
    5    9
» plot(y)
» grid
» shg
```



الشكل (2-8)

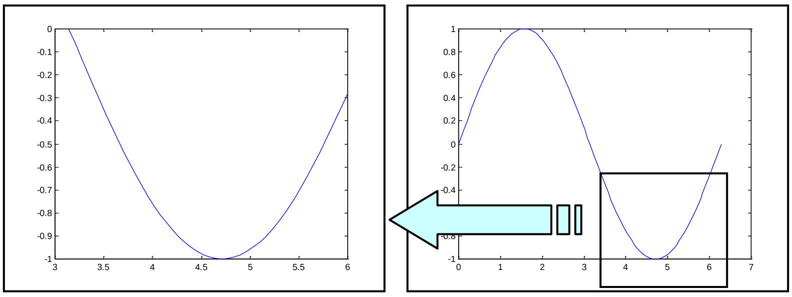
تحديد مجالات المحاور ...

لتحديد مجالات المحاور نستخدم الأمر **axis** بالشكل التالي:

axis([X_{min} ,X_{max} ,Y_{min} ,Y_{max}])

```
» t=0:pi/50:2*pi;
```

```
» y=sin(t);
» plot(t,y)
» axis ([3 ,6 ,-1 ,0])
```



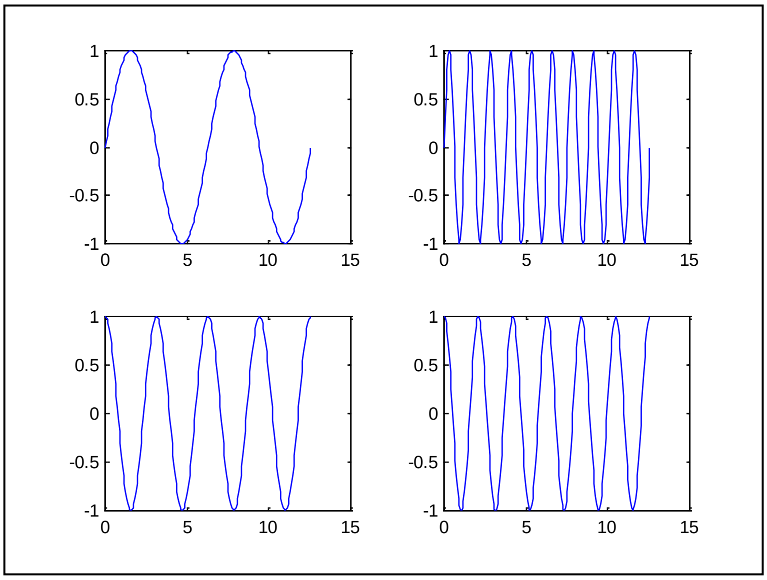
الشكل (9-2)

إظهار اكثر من مخطط على نافذة الصور ...

لإظهار أكثر من مخطط على نافذة الصور نستخدم الأمر **subplot(m,n,i)**. حيث يقسم هذا الأمر الصورة إلى مصفوفة من الصور الجزئية m,n و يتم اختيار الصورة المقابلة بواسطة الدليل i .

مثال :

```
» t=0:pi/50:4*pi;
» y=sin(t);
» y1=sin(5*t);» y2=cos(2*t);
» y3=cos(3*t);
» subplot(2,2,1);
» plot(t,y);
» subplot(2,2,2);
» plot(t,y1);
» subplot(2,2,3);
» plot(t,y2);
» subplot(2,2,4);
» plot(t,y3);
» shg
```



الشكل (10-2)

تسمية المخططات و المحاور و إدراج نص في المخطط …

نستخدم الأوامر التالية:

- لتسمية المخطط …

title('plot_name')

- لتسمية المحور x …

xlabel('X_label')

- لتسمية المحور y …

ylabel('Y_label')

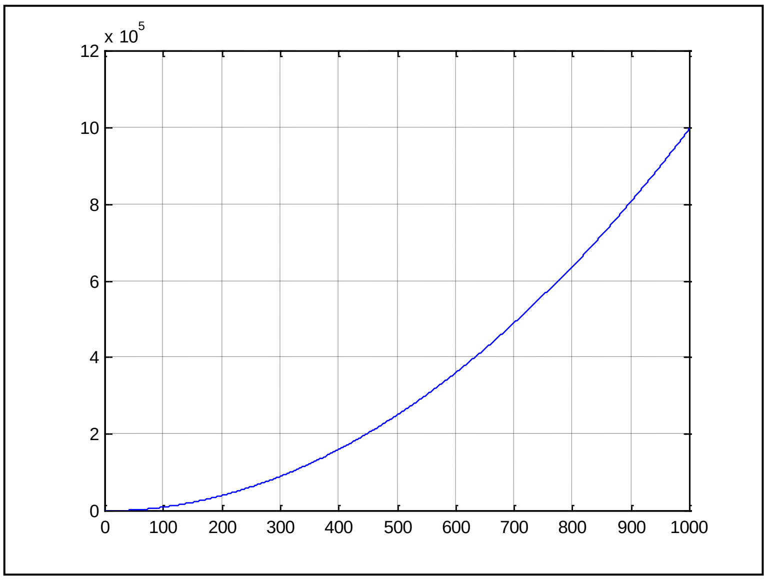
يستخدم MATLAB عدد من الأوامر لتوليد المخططات ثنائية الأبعاد و طباعتها و متحول الدخل لجميع هذه الأوامر هو إما شعاع أو مصفوفة. من هذه الأوامر:

Plot, Plot3, loglog, semilogx, semilogy, plotyy.

سابقا تكلمنا عن الأمر **plot** و الذي له أشكال مختلفة حسب متحولات الدخل plot(y)، plot(x,y) ، سنشرح فيما يلي بقية الأوامر. مثال:

```
» x=0:0.2:1000;
» y=x.^2+3;
» plot(x,y)
» grid
» shg
```

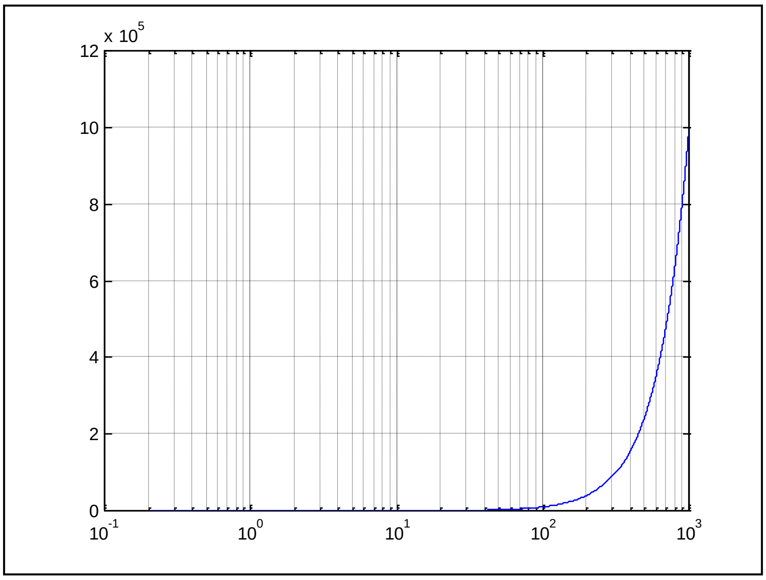
عند تطبيق المثال السابق سيعطي MATLAB المنحني الموضح بالشكل (4-3).



الشكل (4-3)

```
» semilogx(x,y)
» grid
» shg
```

سينتج عن التعليمات السابقة الرسم المبين بالشكل (5-3) حيث نلاحظ أن تقسيمات المحور x أصبحت بالواحدات اللوغاريتمية.



الشكل (5-3)

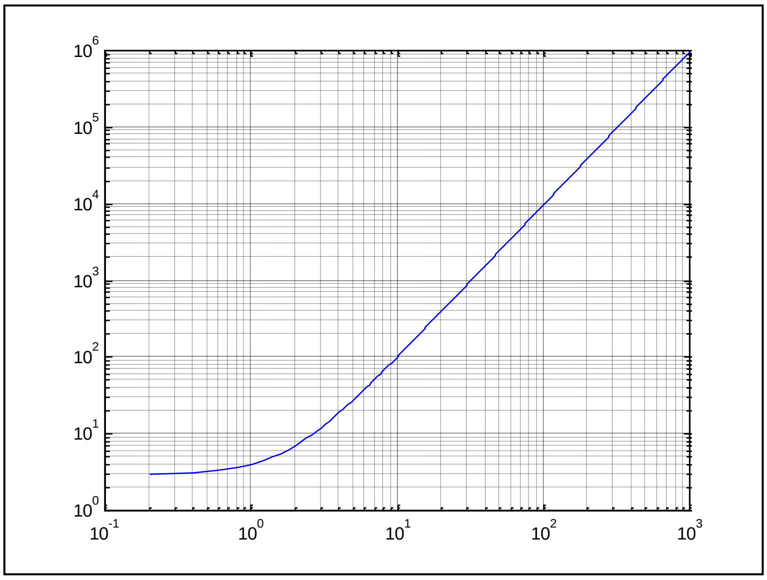
ويمكن جعل تقسيمات المحور Y بالواحدات اللوغاريتمية.

```
» semilogy(x,y)
```

```
» grid
» shg
```

سينتج عن التعليمات التالية الرسم المبين بالشكل (7-3) حيث نلاحظ أن تقسيمات المحورين x ، y أصبحت بالوا حدات اللوغاريتمية.

```
» loglog(x,y)
» grid
» shg
```



الشكل (7-3)

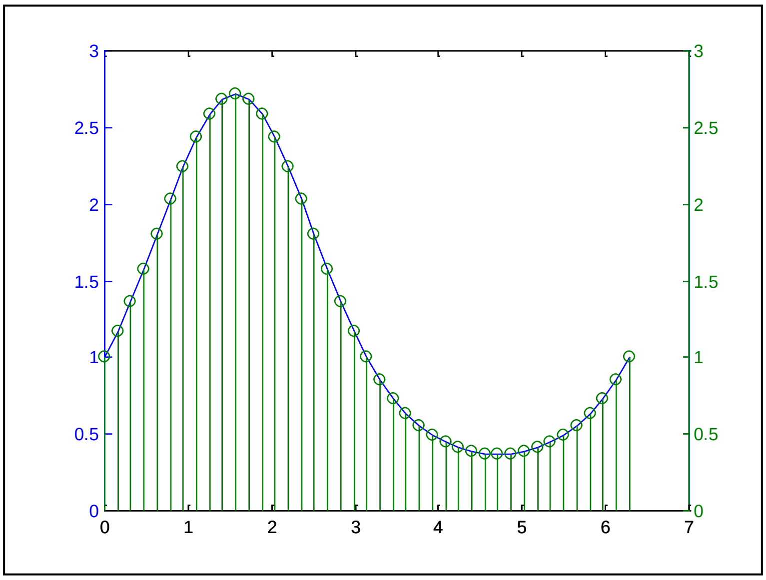
الرسم على مخطط واحد باستخدام محورين منفصلين لـ **y ...**

نستخدم الأمر **plotyy** لرسم المخططات باستعمال محورين y :

```
» t=0:pi/20:2*pi;
» y=exp(sin(t));
» plotyy(t,y,t,y,'plot','stem')
» shg
```

أمر plot هنا يقوم برسم الشعاع y بالنسبة إلى الشعاع t و يظهرهما بشكلين مختلفين، الشكل العادي (عبارة عن خط مستمر plot) و الشكل الثاني (المقطع stem) و سنأتي بذكر هذا النوع من المخططات في المحاضرة التالية. لاحظ الشكل (8-3).

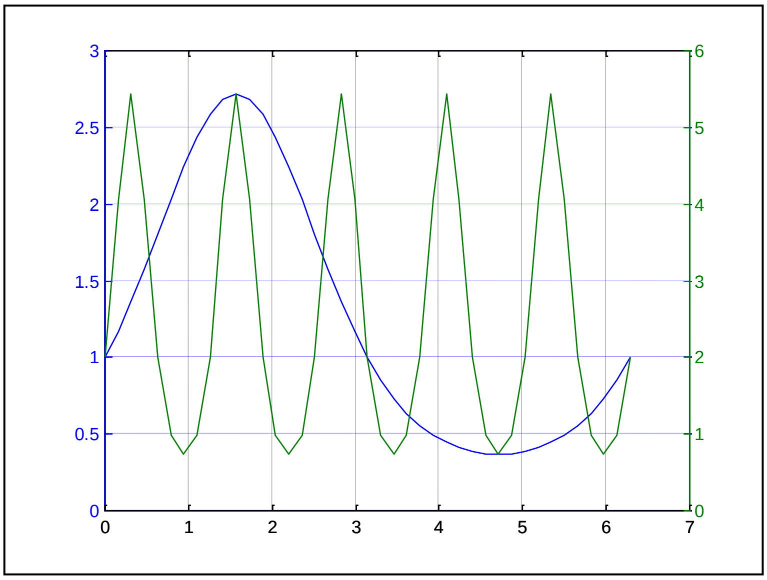
الشكل (8-3)



و نلاحظ أيضاً أن MATLAB قد قام برسم محورين لـ y بلونين مختلفين و قد قام برسم المنحنيات بنفس لون المحاور y و كمثال آخر إذا تابعنا كتابة الأوامر التالية:

```
» y1=2*exp(sin(5*t));
» plotyy (t,y,t,y1)
» grid
» shg
```

الشكل (9-3)



نجد أن MATLAB قد رسم محورين y , y1 بلونين مختلفين و قام برسم المنحنيين بنفس لون المحورين.

رسم المخططات ثلاثية الأبعاد ...

يمكن رسم ثلاثية الأبعاد باستعمال عدة أوامر و أحد هذه الأوامر **plot3(x,y,z)**، حيث يرسم هذا الأمر خط يمر من النقاط التي تحدد إحداثياتها الأشعة x,y,z أي النقاط : ......$(x_1,y_1,z_1); (x_2,y_2,z_2)$.