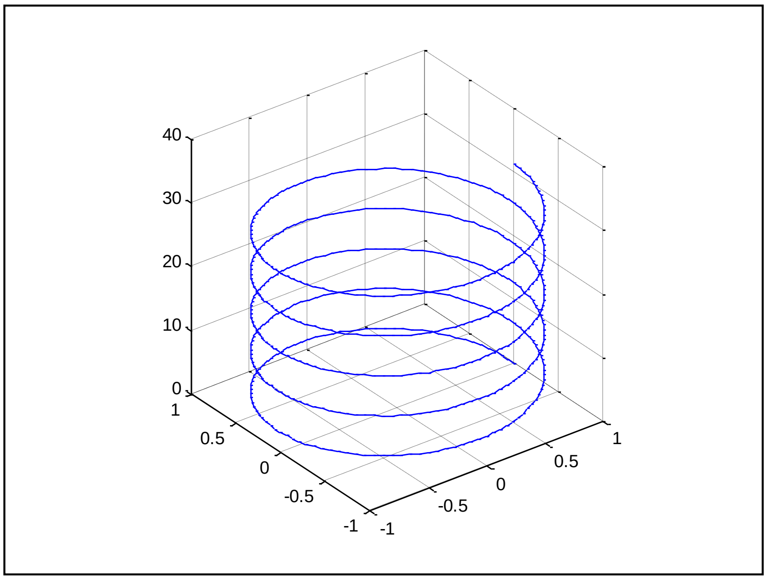
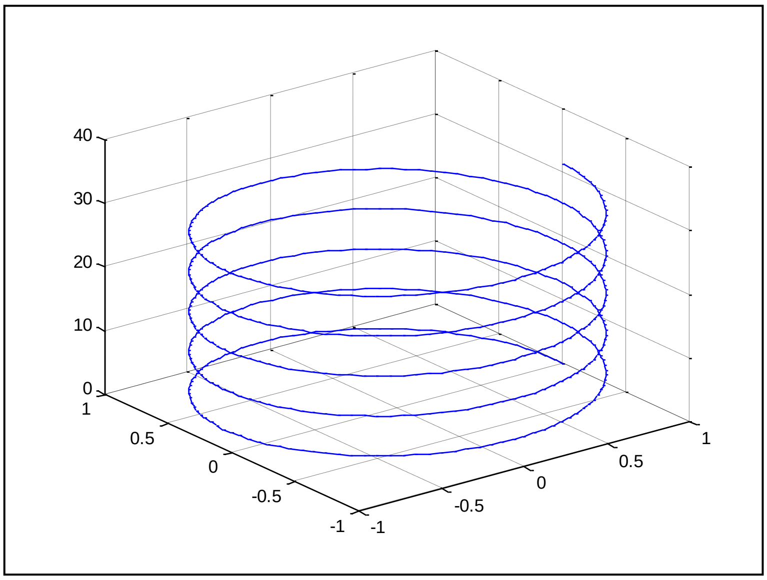
مثال:

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




الشكل ( 10-3 )




الشكل ( 11-3 )

استخدمنا axis square و ذلك لجعل تدريجات المحورين x,y متساويتين و بالتالي طولي المحورين ( 10-3 ) و بدون استخدام هذا الأمر سيظهر المخطط بالنسب الافتراضية.

مخططات الأعمدة و مخططات المساحة Bar & Area Graphs ...

تستعمل مخططات الأعمدة و المساحة لإظهار المعطيات المرتبة على شكل شعاع أو مصفوفة هذه المخططات مفيدة بشكل خاص عند ضرورة مقارنة النتائج مع بعضها. مخططات الأعمدة مناسبة لإظهار المعطيات المنفصلة بينما مخططات المساحة مناسبة لإظهار المعطيات المستمرة.

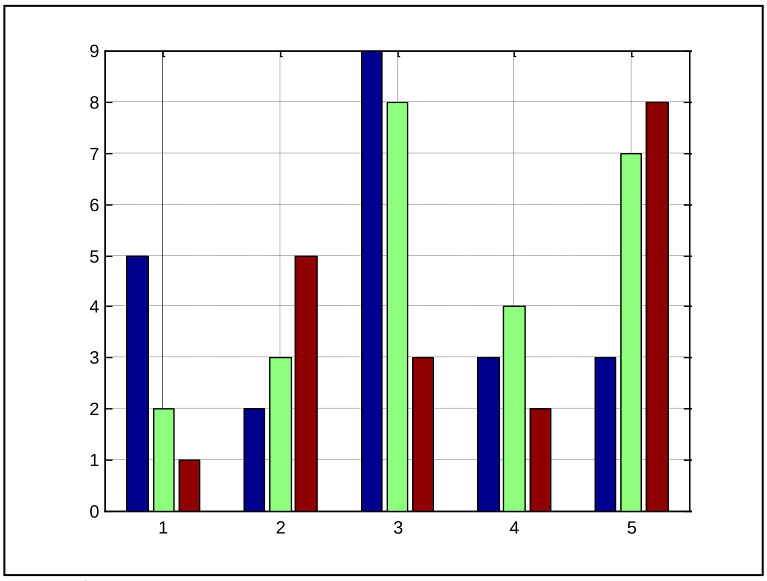
الأوامر المستخدمة لتوليد مخططات الأعمدة و المساحة ...

- **bar** بواسطة هذا الأمر يتم رسم أعمدة المصفوفة ( m × n ) على شكل مجموعة من الأعمدة عددها m تتألف كل مجموعة من n عمود شاقولي.
- **barh** بواسطة هذا الأمر يتم رسم أعمدة المصفوفة ( m × n ) على شكل مجموعة من الأعمدة عددها m تتألف كل مجموعة من n عمود أفقي.
- **bar3, bar3h** يقومان بنفس عمل bar و barh و لكن يتم رسم الأعمدة فراغياً 3D.
- **area** لرسم مخططات المساحة.

أمثلة ...

**Bar** ...

```
» y=[5 2 1;2 3 5;9 8 3;3 4 2;3 7 8];
» bar(y)
» grid
» shg
```




الشكل ( 3-13 )

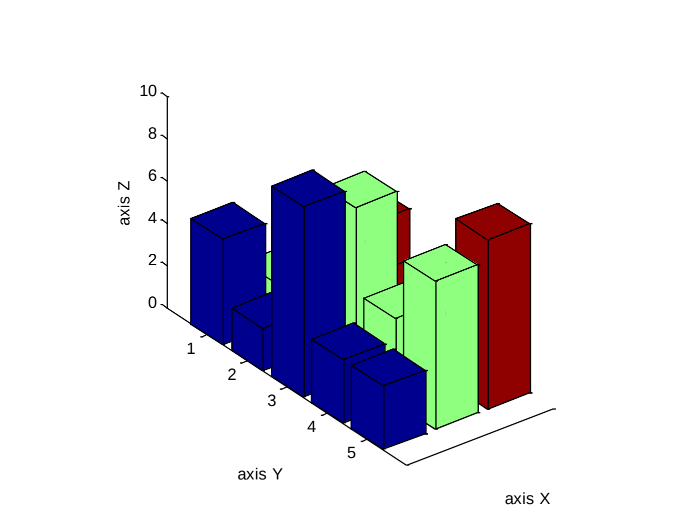
نلاحظ أن كل ثلاثة أعمدة مجمعة مع بعضها و هي تعبر عن عناصر صف من صفوف المصفوفة فمثلاً العناصر الثلاثة الأولى هي عبارة عن y(1,:) أي :

y(1,1)=5, y(1,2)=2, y(1,3)=1

**Bar3** …

بنفس الطريقة نستخدم الأمر **bar3** لرسم الأعمدة فراغياً و لكن في هذه الحالة تتوضع عناصر كل عمود من أعمدة المصفوفة على طول المحور y بحيث تكون إحداثيات مجموعة الأعمدة التابعة للعمود الأول من المصفوفة على المحور x مساوية الواحد و إحداثيات مجموعة الأعمدة التابعة للعمود الثاني من المصفوفة على المحور x مساوية 2 و هكذا …

```
» bar3(y)
» grid
» shg
»
```




الشكل ( 3-14

يمكن تسمية المحاور باستخدام الأوامر

xlabel(‘X_label’), ylabel(‘Y_label’), zlabel(‘Z_label’)

```
» xlabel('axis X')
» ylabel('axis Y')
» zlabel('axis Z')
» shg
```

مخطط الأعمدة ثلاثي الأبعاد المجمع ...

يمكن تجميع الأعمدة العائدة لكل صف مع بعضها كما في المخطط ثنائي الأبعاد باستخدام **‘group’** كما يلي:

```
» bar3(y,'group')
» shg
```

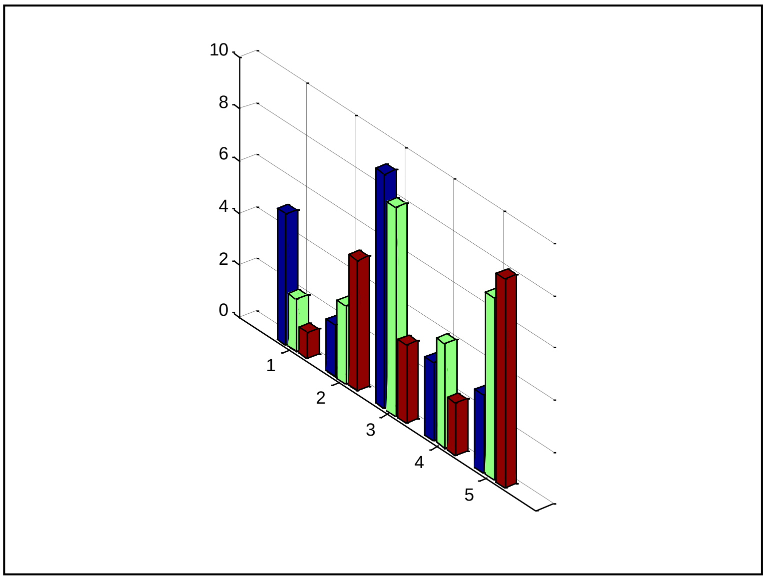
من الشكل ( 3-15 ) نلاحظ أنه يتم تجميع الأعمدة على طول المحور y.

مخطط الأعمدة الموسع ...

يمكن إظهار جميع العناصر العائدة لصف واحد من صفوف المصفوفة متوضعة فوق بعضها البعض على شكل عمود واحد طوله يساوي مجموع العناصر. نعبر في مخطط الأعمدة الموسع عن كل صف من المصفوفة بواسطة عمود واحد يقسم إلى ثلاث أقسام. يتم توليد مخطط الأعمدة الموسع باستخدام الأوامر التالية:

bar(y,’stack’)

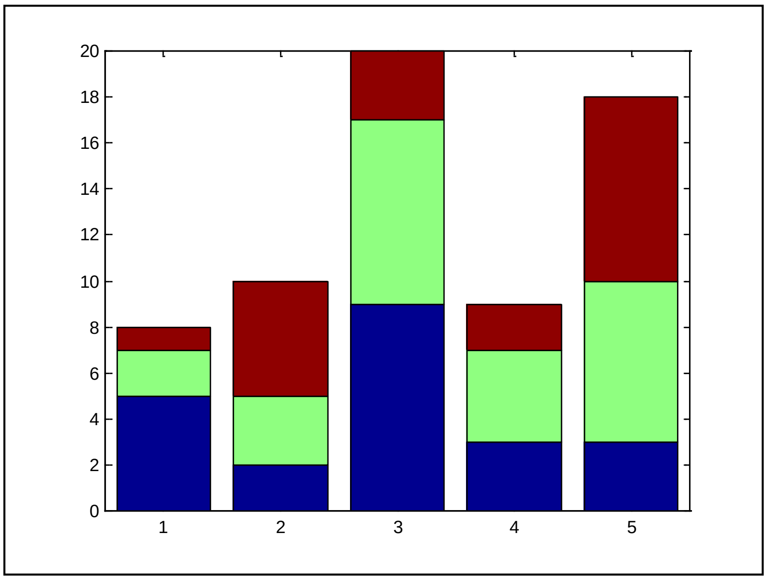
bar3(y,’stack’)




الشكل ( 3-15 )

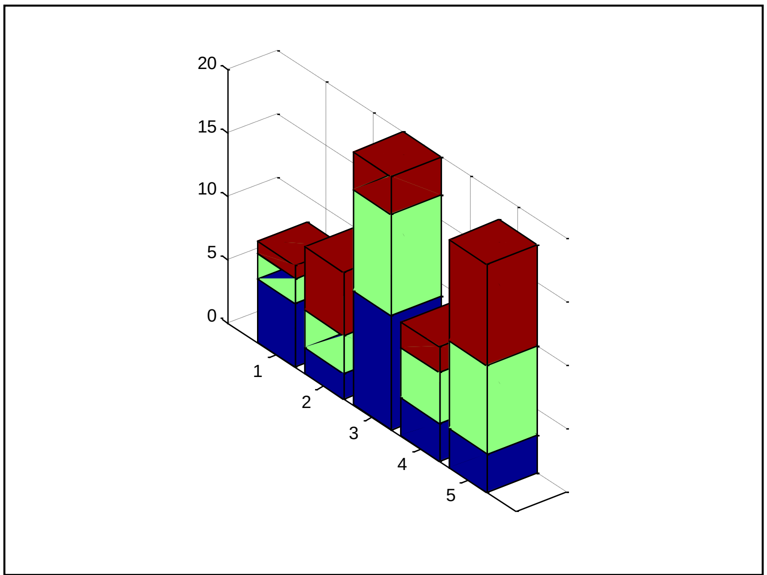
```
» bar(y,'stack')
» shg
```

الشكل ( 16-3 )




```
» bar3(y,'stack')
» shg
```

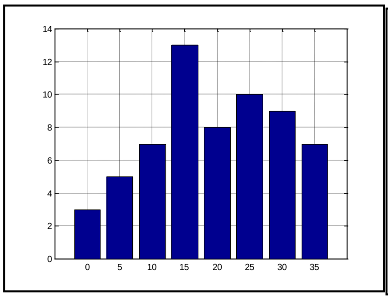
الشكل ( 17-3 )

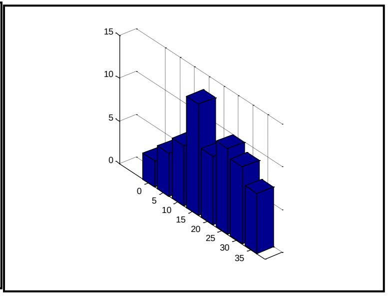



إظهار العلاقة بين شعاعين بواسطة مخططات الأعمدة ...

يمكن إظهار العلاقة بين الأشعة x,y بواسطة مخطط الأعمدة باستعمال الأمر **bar(x,y)**. حيث يظهر كل عنصر من عناصر الشعاع y بشكل عمود يتوضع عند القيمة المقابلة للشعاع x.

```
» x=0:5:35;
» y=[3 5 7 13 8 10 9 7];
» bar(x,y)
» grid
» shg
» bar3(x,y)
» shg
```





الشكل ( 3-20 )

مخططات الشرائح pie graghs ...

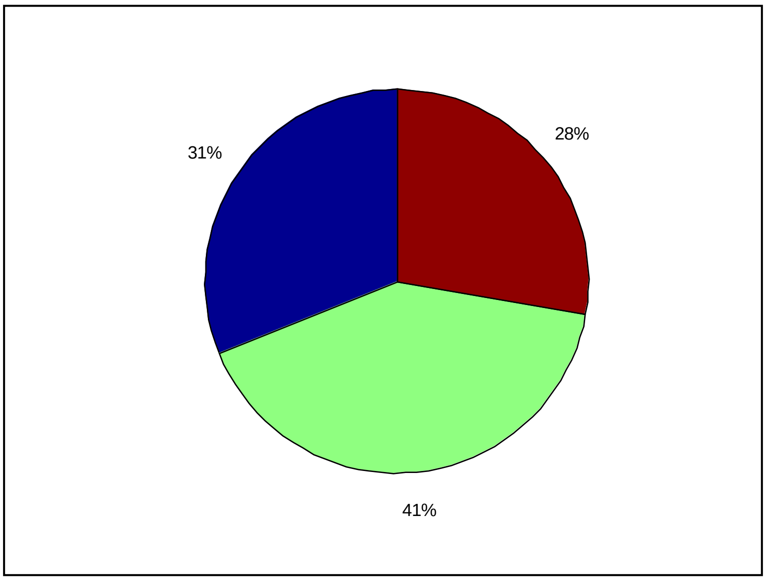
تبين مخططات الشرائح النسبة المئوية لمساهمة كل عنصر من عناصر الشعاع أو المصفوفة في القيمة الكلية لمجموع العناصر. يمكن توليد مخططات شرائح ثنائية الأبعاد **pie(x)** أو ثلاثية الأبعاد **pie3(x)**. حيث x شعاع.

باستخدام الأمر **h=pie(x,explode)** يتم اختيار الشرائح التي سيتم إبعادها عن بقية الشرائح ( تمييزها ) بواسطة الشعاع explode و الذي له نفس أبعاد الشعاع x عند وجود أي عنصر قيمته تساوي الواحد في الشعاع explode يتم إبعاد الشريحة المقابلة لهذا العنصر.

مثال:

```
» x=[3 5 8;1 7 4;10 15 2;7 9 10;7 1 1];
» x1=sum(x)
x1 =
   28    37    25
» pie(x1)
```

في المثال السابق استعملنا الأمر **sum** و الذي جمع عناصر كل عمود من أعمدة المصفوفة x و خزنها في عناصر الشعاع x1. و نلاحظ من الشكل أن الأمر **pie** أدى إلى رسم الشرائح متصلة مع كتابة النسب المئوية.




الشكل ( 3-22 )

## التوابع الأولية (الدوال الجاهزة ) في MATLAB ...

التابع **disp(A)** إذا كانت A مصفوفة فإن التابع يظهر المصفوفة بدون طباعة اسمها و إذا كانت A سلسلة حروف فإنه يظهرها، كمثال:

```
» a=[3 4;5 6]
a =
    3    4
    5    6
» disp(a)
```

```
    3    4
    5    6
» disp('Display array')
Display array
```

التابع **B=fix(A)** يدور عناصر المصفوفة A إلى اقرب عدد صحيح باتجاه الصفر و بالنسبة للعناصر العقدية فإنه يدور القسم الحقيقي و التخيلي بشكل منفصل.

```
» A=[-1.9 -0.2 3.4 5];
» B=fix(A)
B =
   -1    0    3    5
```

التابع **B=floor(A)** يدور عناصر المصفوفة A إلى أقرب عدد صحيح أصغر أو يساوي إلى A باتجاه ∞- و هذا يعني أنه يدور عناصر المصفوفة A إلى أقرب عدد صحيح أصغر أو يساوي إلى A و من أجل العناصر العقدية يتم تدوير القسم الحقيقي و القسم التخيلي بشكل منفصل.

```
» A=[-1.9 -0.2 3.4 5];
» B=floor(A)
B =
   -2   -1    3    5
```

التابع **B=ceil(A)** يدور عناصر A إلى أقرب عدد باتجاه اللانهاية أو بمعنى آخر يدور عناصر المصفوفة A إلى أقرب عدد صحيح أكبر أو يساوي إلى عناصر A و من أجل العناصر العقدية يتم تدوير القسم التخيلي و القسم الحقيقي كلاً على حدة.

```
» A=[-1.9 -0.2 3.4 5];
» B=ceil(A)
B =
   -1    0    4    5
```

التابع **B=round(A)** يدور عناصر المصفوفة A إلى أقرب عدد صحيح و من أجل العناصر العقدية في المصفوفة يدور القسم الحقيقي و القسم التخيلي كلاً على حدة.

```
» A=[-1.9 -0.2 3.4 5];
» B=round(A)
B =
```

```
   -2    0    3    5
```

التابع **y=sign(x)** يعطي مصفوفة y أبعادها من نفس أبعاد x و قيمة كل عنصر من y هي:

1 إذا كان العنصر المقابل أكبر من الصفر.

0 إذا كان العنصر المقابل تساوي الصفر.

-1 إذا كان العنصر المقابل أصغر من الصفر.

```
» A=[-1.9 -0.2 3.4 5];
» B=sign(A)
B =
   -1   -1    1    1
```

التابع **M=mod(x,y)** يعطي باقي قسمة x على y حيث x, y مصفوفتين بنفس الحجم أو عددين حقيقيين و لباقي القسمة نفس إشارة y و بالتعريف:

mod(x,0)=x

mod(x,y)=x-y.*floor(x./y)

y~=0

مثال...

```
» x=[3 3 -8 -1];
» y=[2 -2 -3 2];
» mod(x,y)
ans =
    1   -1   -2    1
```

التابع **M=rem(x,y)** يعطي باقي قسمة x على y حيث x, y مصفوفتين بنفس الحجم أو عددين حقيقيين و لباقي القسمة نفس إشارة x و بالتعريف:

rem(x,0)=nan

rem(x,y) =x-y.*fix(x./y)

y~=0

مثال:

```
» rem(x,y)
```

```
ans =
    1    1   -2   -1
```

التابع **y=abs(x)** يعطي القيمة المطلقة |x| لكل عنصر من عناصر المصفوفة x و من أجل العناصر العقدية يعطي التابع مطال العدد العقدي.

```
» abs(-5)
ans =
    5
» abs(3+4i)
ans =
    5
```

التابع **factor(x)** تحليل العدد إلى عوامله الأولية .

التابع **primes(x)** الحصول على جميع الأعداد الأولية الأصغر من x .

التابع **factorial(x)** الحصول على x! .

التابع **gcd(x,y)** إيجاد القاسم المشترك الأكبر . وفي حال المصفوفات فللعناصر المتقابلة .

التابع **lcm(x,y)** إيجاد المضاعف المشترك الأصغر.

التابع **conj(z)** تعطي مرافق الأعداد العقدية لعناصر المصفوفة z.

التابع **real(z)** يعطي القسم الحقيقي لعناصر المصفوفة z.

التابع **imag(z)** يعطي القسم التخيلي لعناصر المصفوفة z.

التابع **angle(z)** يعطي الطور أو المضمون لعناصر المصفوفة العقدية z.

## المعاملات ...

استعرضنا بشكل مفصل المعاملات الحسابية و فيما يلي سنستعرض معاملات المقارنة و المعاملات المنطقية.

معاملات المقارنة:

< أصغر من.

<= أصغر أو يساوي.

> أكبر من.

>= أكبر أو يساوي.

== يساوي.

~= لا تساوي.

نتيجة العلاقات التي تحتوي هذه المعاملات إما ( 1 ) أي صحيح أو ( 0 ) أي خطأ فمثلاً :

```
» 1==5
ans =
    0
» 3<1
ans =
    0
» 10==1
ans =
    0
» 10==10
ans =
    1
» 10~=10
ans =
    0
```

عند تطبيق هذه المعاملات على المصفوفات ذات الأبعاد المتساوية فإن معاملات المقارنة تقارن كل عنصر من عناصر المصفوفات ذات الأبعاد المتساوية. تعامل معاملات المقارنة دائماً المصفوفات عنصر بعد عنصر و كمثال يبين كيف يقوم معامل المساواة بمقارنة جميع عناصر المصفوفات و إعطاء قيمة ( 1 ) للعناصر المتساوية و القيمة ( 0 ) للعناصر الغير متساوية.

```
» A=[3 5 2;7 2 1;6 2 6];
» B=[3 7 4;5 2 3;6 7 5];
» A==B
ans =
    1    0    0
    0    1    0
    1    0    0
» A>B
```

```
ans =
    0    0    0
    1    0    0
    0    0    1
» A~=B

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

المعاملات المنطقية: تستخدم MATLAB المعاملات المنطقية التالية.

| | |
|---|---|
| & | AND |
| \| | OR |
| ~ | NOT |

لكل معامل من هذه المعاملات وظيفة معينة.

تكون العلاقة التي تستعمل المعامل AND صحيحة إذا كان المتحولين ( العلاقتين ) على طرفي المعامل صحيحتين أو يمكن التعبير بشكل رقمي كما يلي: تكون العلاقة صحيحة إذا كان المتحولين ( العلاقتين ) لا يساويان الصفر.

```
» u=[1 0 2 3 0 5];
» v=[5 6 1 2 0 7];
» u&v
ans =
    1    0    1    1    0    1
```

تكون العلاقة التي تستعمل المعامل OR صحيحة إذا كان أحد المتحولين ( العلاقتين ) على جانبي المعامل صحيحين أو يمكن التعبير بشكل رقمي: تكون العلاقة خطأ ( 0 ) إذا كان كلا المتحولين مساويان الصفر، مثلاً:

```
» u|v
ans =
    1    1    1    1    0    1
```

العلاقة التي تستعمل النفي NOT تكون صحيحة إذا كان المتحول على يمينها خطأ و بشكل رقمي: أي عدد سالب لا يساوي الصفر يصبح بعد النفي مساوياً للصفر و أي عدد يساوي الصفر قبل النفي يصبح مساوياً الواحد، مثلاً:

```
» ~u
```

```
ans =
   0   1   0   0   1   0
» ~~u
ans =
   1   0   1   1   0   1
```

يطبق MATLAB المعاملات المنطقية على المصفوفات متساوية الأبعاد عنصر بعد عنصر.

التوابع المنطقية ...

بالإضافة إلى المعاملات المنطقية يوجد في MATLAB بعض التوابع المنطقية مثل XOR يجري عملية الجمع التناظري على العناصر.

**xor** يعطي القيمة 1 إذا كان أحد العناصر صحيح بينما العناصر الأخرى خطأ و بشكل رقم: يعطي التابع القيمة 1 إذا كان أحد العناصر فقط يساوي الواحد و قيمة بقية العناصر الأخرى مساوية للصفر.

```
» a=1;
» b=1
» xor(a,b)
ans =
   0
```

**all** يعطي القيمة 1 إذا كانت جميع العناصر في الشعاع لا تساوي الصفر و في المصفوفات يطبق التابع all على كل عمود من أعمدة المصفوفة على حده.

```
» u=[0 1 2 0];
» all(u)
ans =
   0
» A=[0 1 2;3 5 0];
» all(A)
ans =
   0   1   0
```

**any** يعطي هذا التابع القيمة 1 إذا كان أي عنصر من عناصر الشعاع صحيح أو لا يساوي الصفر و يطبق مثل all على أعمدة المصفوفة

```
» v=[3 0 8];
» any(v)
ans =
    1
```

**find** يحدد التابع find أدلة عناصر المصفوفة التي تحقق شرط منطقي معين و نحصل بنتيجة تطبيق هذا التابع على شعاع يحوي على أدلة عناصر المصفوفة التي تحقق الشرط المعطى.

```
» A=[5 6 8 9;4 3 8 5;6 7 4 11;5 4 6 3]
A =
     5     6     8     9
     4     3     8     5
     6     7     4    11
     5     4     6     3
» i=find(A>=8)
i =
     9
    10
    13
    15
» A(i)=100
A =
     5     6   100   100
     4     3   100     5
     6     7     4   100
     5     4     6     3
```

أسبقية تنفيذ المعاملات: يتم تنفيذ المعاملات حسب الترتيب التالي:

( )        الأقواس

~ النفي

.' .^ ^ +(Number) –(number)

.* ./ .\ * / \

+ -

< <= > >= == ~=

& |

السلاسل الحرفية String Data :

لتوليد سلسلة حرفية يكفي أن نكتب اسم تحوّل يليه إشارة المساواة والسلسلة الحرفية بين إشارتي تنصيص على سبيل المثال :

```
name = 'ahmad'
```

للتحويل إلى **Ascii** :

a = **double**(name)

فتكون النتيجة :

```
a =
   97   104   109    97   100
```

لإرجاع **a** إلى سلسلة حرفية مرّة أخرى، نستخدم تابع **char**

b=**char**(a)

تكون النتيجة :

```
b =
Ahmad
```

المقارنة بين السلاسل :

يمكن إجراء جميع عمليات المقارنة على السلاسل، على سبيل المثال:

```
'ahmad' = = 'Ahmad'
```

يقارن كل حرف مع الحرف الذي يقابله :

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

يوجد تابعان لتحديد إذا كانت الرموز في السلسلة حروف أو فراغات :

a = 'ahmad  400'

**isletter** (a)

فتكون النتيجة :

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

حيث يضع مكان الأحرف القيمة (1) ومكان الفراغات والأرقام القيمة (0) ، (يوجد في السلسلة السابقة فراغين)

**isspace**(a)

تكون النتيجة

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

كما يوجد تابعان upper لتحويل جميع أحرف السلسلة إلى أحرف كبيرة و lower لتحويل جميع الأحرف إلى أحرف صغيرة .

## معالجة الإشارة

لا يمكن التحدث بشكل منفصل عن معالجة الإشارة ، فحتى الصور تعتبر معالجة للإشارة بشكل ما .

حيث يتعرف الماتلاب على المصفوفات ، سواء كانت المصفوفة معبرة عن صورة أو عن إشارة صوتية أو عن إشارة قياس مأخوذة من حساس خارجي فليس من فرق لديه، وكل ما يعرفه هو قيم مخزنة معاً يمكن إجراء ما نشاء من عمليات عليها .

تعد تحويلات فورييه من أكثر التوابع استخداماً والتي تساعدنا على النظر للإشارة من وجهة النظر الترددية وليس الزمنية و هذا مفيد لمعرفة المحتوى الترددي للإشارة .

**...fft**

حساب تحويل فورييه المقطع السريع

```
>>Y=fft(x)
>>Y=fft(x,n)
>>Y=fft(x,[],dim)
```

```
>>Y=fft(x,n,dim)
```

حيث n عدد النقاط المحسوبة.

وعند كون x مصفوفة ثنائية فإنه بالإمكان تحديد فيما إذا كنا نود حساب تحويل فورييه للأعمدة dim=1 أو للأسطر dim=2

تعطى العلاقة بالشكل :

$$X(k)=\sum_{n=1}^{N} x(n)*\exp(-j*2*pi*(k-1)*(n-1)/ N) \quad : 1<=k<=N$$

مثال..

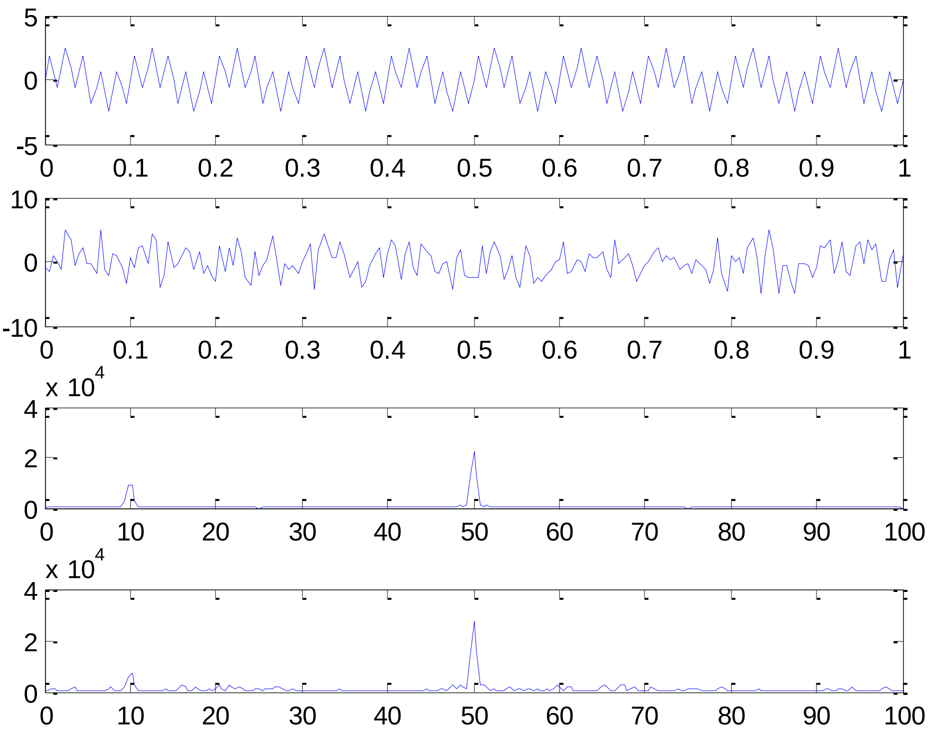
يبين المثال التالي إشارة تحوي على مركبتين ترددتين هما 10,50 .

نقوم بإضافة إشارة عشوائية تمثل الضجيج المتراكب معها و بمطال صغير نسبياً .

نقوم برسم الإشارة الزمنية الأصلية والإشارة المشوشة بالإضافة إلى طيف الإشارة الأصلية وطيف الإشارة المشوشة .

علماً أنه يمكن تغيير تأثير الضجيج بتغيير مطاله .

```
>> t=0:0.005:1;
>> x=sin(2*pi*10*t)+1.5*sin(2*pi*50*t);
>> y=x+2*randn(size(t));
>> subplot(4 ,1 ,1);
>> plot(t,x)
>> subplot(4 ,1 ,2)
>> plot(t,y)
>> xf=fft(x,512);
>> pxf=abs(xf).^2;
>> f=200*(0:256)/512;
>> subplot(4,1,3)
>> plot(f,pxf(1:257))
>> yf=fft(y,512);
>> pyf=abs(yf).^2;
>> f=200*(0:256)/512;
>> subplot(4,1,4)
>> plot(f,pyf(1:257))
```




**... ifft**

حساب عكس تحويل فورييه السريع

```
>>Y=ifft(x)
>>Y=ifft(x,n)
>>Y=ifft(x,[],dim)
>>Y=ifft(x,n,dim)
```

تعطى العلاقة بالشكل :

$$x(n)= (1/N) \sum_{k=1}^{N} X(k)*\exp( j*2*pi*(k-1)*(n-1)/ N) \quad : 1<=n<=N$$

كما يحوي الماتلاب توابع إضافية من اجل حساب تحويلات فورييه نوردها كما يلي ....

**.... fft2**

تحويل فورييه السريع ثنائي البعد وهنا تكون أبعاد الخرج والدخل متساوية .

**...ifft2**

عكس تحويل فورييه السريع ثنائي البعد.

**... fftshift**

إزاحة الطيف الترددي بحيث يتوضع في المنتصف .

**.... conv(u,v)**

طي الشعاعين u و v .

طول الشعاع الناتج يساوي إلى مجموع طولي الشعاعين منقوصاً بقيمة واحد .

تعطى علاقة الطي كالتالي ..

$$w(k)=\sum_{j}u(j)v(k+1-j)$$

وكتطبيق سريع للعلاقة :

```
>>w(1)=u(1)*v(1)
>>w(2)=u(1)*v(2)+u(2)*v(1)
>>w(3)=u(1)*v(3)+u(2)*v(2)+u(3)*v(1)
...
>>w(n)=u(1)*v(n)+u(2)*v(n-1)+...+u(n)*v(1)
...
>>w(2*n-1)=u(n)*v(n)
```

**...deconv**

إيجاد مقلوب الطي ، أو الطي العكسي لشعاعين ،كما يمكننا إيجاد مقلوب الطي لشعاعين u,v كما يلي :

```
>>u=[1 2 3];
>>v=[6 7 8 9];
>>w=conv(u,v)
w=
 6 19 40 46 42 27
>>[dw,rw]=deconv(w,u)
dw=
  6 7 8 9
rw=
  0 0 0 0 0 0
```

**.. filter**

أي نظام موجود في الطبيعة يمكن القول بأنه مرشح ، وبشكل عام يوجد أربعة أنواع للمرشحات هي :

- مرشح تمرير ترددات منخفضة .
- مرشح تمرير ترددات عالية .
- مرشح تمرير مجال ترددي محدد .
- مرشح حذف مجال ترددي محدد .

فعلى سبيل المثال ، يتم استخدام مرشح تمرير الترددات المنخفضة كمرحلة أولية بعد الحصول على إشارة القياس ذات التغيرات البطيئة أو حتى الإشارة الصوتية ، وذلك بغية حذف المركبات الترددية العالية المتراكبة مع الإشارة والتي تعتبر مركبات ضجيجية .

وبالتالي سنحتاج كثيراً إلى تمرير الإشارة على مرشح مناسب للحصول على إشارة مناسبة .

تساعدنا التعليمة filter على تمرير إشارة محددة على مرشح معطى بشكل كسري و تابع تحويله .

```
>>y=filter(b,a,x)
>>[y,zf]=filter(b,a,x)
>>[…]=filter(b,a,x,zi)
>>[...]=filter(b,a,x,zi,dim)
```

zi:تمثل شعاع الشروط البدئية .

يعطى الشكل العام للمرشح وفق تحويل z بالشكل :

$$Y(z)=\frac{b(1)+b(2)\,z^{-1}+\ldots+b(nb+1)\,z^{-nb}}{a(1)+a(2)\,z^{-1}+\ldots+a(an+1)\,z^{-na}}\,X(z)$$

**... freqz**

تستخدم لحساب استجابة الترددية لمرشح رقمي معطى كبسط b ومقام a و تابع تحويله

```
>>[h,w]=freqz(b,a,n)
>>h=freqz(b,a,w)
>>[h,w]=freqz(b,a,n,'whole')
>>[h,w,units]=freqz(b,a,..)
>>[h,f]=freqz(b,a,n,fs)
>>h=freqz(b,a,n,fs)
>>[h,f]=freqz(b,a,n,'whole',fs)
>>[h,f,units]freqz(b,a,n,'whole',fs)
>>freqz(b,a,..)
```

**... freqs**

تستخدم لحساب الاستجابة الترددية لمرشح تمثيلي معطى كبسط b ومقام a و تابع تحويله.

```
>>h=freqs(b,a,w)
>>[h,w]=freqs(b,a)
```