الفصل السادس

استكمال وتقريب الدوال على MATLAB

عند وجود بيانات معينة حصيلة تجربة علمية ، فإن تحليل هذه البيانات سيعطي معلومات عديدة، مثل : التنبؤ بالقيم المستقبلية، والحصول على قيم البيانات السابقة، أو قيم تقع بين نقاط البيانات المستخدمة. إن استخدام دالة تقريبية تمر بالبيانات لتسهيل تحليل البيانات يسمى بالاستكمال Interpolation. ولبرنامج MATLAB دورٌ باهرٌ في هذا المجال، فقد زوّد بأوامر جاهزة لاستكمال الدوال ورسمها بدقة فائقة في أي عدد من الأبعاد، كما يسهّل برمجة الدوال المتعلقة بالاستكمال، مما يجعله ينافس الكثير من البرامج المتخصصة في الرسم بسرعة ودقة فائقة. سنتطرق أيضاً في هذا الفصل لموضوع تقريب الدوال بدوال أبسط منها، مثل كثيرات الحدود والدوال المثلثية بنوعين من التقريب المتقطع والمتصل. وستتضمن الدراسة نظرية التقريب بدالة صريحة، والبحث عن الدوال الأكثر ملاءمة للتقريب باستخدام MATLAB.

(٦,١) استخدام كثيرة حدود للاستكمال Polynomial Interpolation

تستخدم كثيرات الحدود في التقريب، لأنها دوال متصلة، ولسهولة حساب كل من مشتقاتها وتكاملاتها. وتُعد كثيرات الحدود من أشهر الدوال وأكثرها استخداماً ،

خصوصاً في الاستكمال، وذلك لسهولة تطبيقها، وتكون على هذا الشكل:

$$p_n(x) = a_0x^n + a_1x^{n-1} + a_2x^{n-2} + ... + a_{n-1}x + a_n$$

حيث إن n عدد صحيح غير سالب و $a_0, a_1, ..., a_n$ أعداد حقيقية ثابتة، ومن أحد أسباب أهمية هذه الدوال هو أنها تقرّب الدوال المتصلة بانتظام. من كثيرات الحدود التي تستخدم في التقريب متسلسلة تايلور، وهي طريقة أساسية في التحليل العددي، ولكنها غير ملائمة للاستكمال لأن دقتها تتركز فقط في جوار نقطة التقريب. يتطلب الاستكمال كثيرة حدود تعطي تقريباً دقيقاً على فترة محددة وباستخدام معلومات من نقاط مختلفة.

سنعرض الاستكمال باستخدام كثيرات الحدود لاجرانج و نيوتن Newton, Lagrange polynomialsوكذلك تقريب كثيرات الحدود التجزيئية مثل دالة الشريحة التكعيبية للاستكمال Cubic Spline Interpolation مع تطبيقاتها على برنامج MATLAB.

(٦,١,١) استكمال كثيرة حدود لاجرانج

Polynomial Lagrange Interpolation

إذا كانت $x_0, x_1, ..., x_n$ أعداد مختلفة عددها n+1 وكانت $f(x)$ دالة قيمتها عند هذه الأعداد معروفة، فان طريقة Lagrange تعتمد على البحث عن كثيرة حدود P وحيدة، وبدرجة لا تزيد على n بحيث نحصل على:

$$p(x_i) = f(x_i) \qquad , i = 0,1,2,...,n$$

عندها تكون كثيرة الحدود من الدرجة n بالشكل :

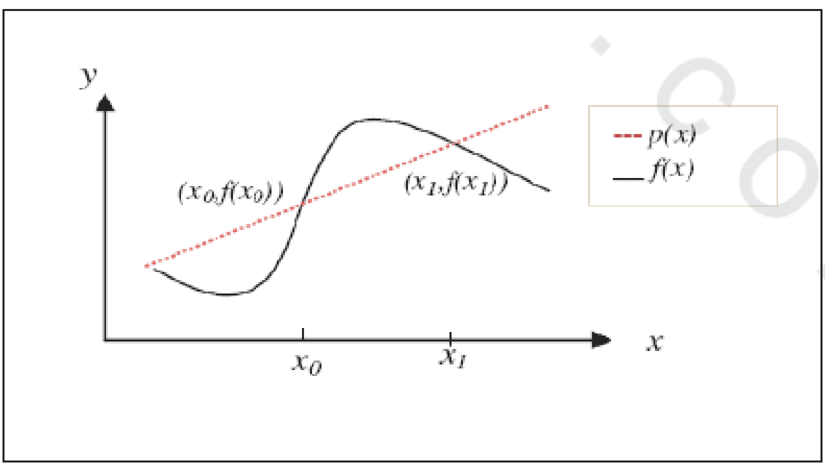
$$p_n(x) = L_0(x)f(x_0) + L_1(x)f(x_1) + \dots + L_n(x)f(x_n)$$

بحيث إن :

$$L_k(x) = \frac{(x-x_0)(x-x_1)..(x-x_{k-1})(x-x_{k+1})..(x-x_n)}{(x_k-x_0)(x_k-x_1)..(x_k-x_{k-1})(x_k-x_{k+1})..(x_k-x_n)} = \prod_{i=0,k\neq i}^{n} \frac{x-x_i}{x_k-x_i}$$

وتسمى هذه الصيغة كثيرة حدود لاجرانج للاستكمال Lagrange Interpolation polynomial .

نبدأ بأسهل حالات الاستكمال، وهي الاستكمال الخطي ، فلتكن $f(x)$ دالة معرفة عند النقطتين x_0, x_1 ونريد أن نكون كثيرة حدود $p_1(x)$ من الدرجة الأولى، و تمر بالنقطتين $(x_0, f(x_0))$ و $(x_1, f(x_1))$. ويتضح من الشكل رقم (٦،١) أن الخط المستقيم هو أقصر منحنى يمر بين النقطتين السابقتين.



الشكل رقم (٦،١). الاستكمال الخطي.

ويمكن تمثيل هذا المستقيم بكثيرة حدود خطية كما يلي :

$$p(x) = \frac{x - x_1}{x_0 - x_1} f(x_0) + \frac{x - x_0}{x_1 - x_0} f(x_1)$$

فإذا كانت :

$$L_0(x) = \frac{x - x_1}{x_0 - x_1} \quad \text{و} \quad L_1(x) = \frac{x - x_0}{x_1 - x_0}$$

فإن كثيرة الحدود الخطية تأخذ الشكل التالي :

$$p_1(x) = L_0(x)f(x_0) + L_1(x)f(x_1)$$

التقريب بدوال لاجرانج يلعب دوراً مهماً في إيجاد صيغ التكامل العددية المختلفة التي تم عرضها في الفصل الرابع. يساعد برنامج MATLAB في إيجاد كثيرة حدود لاجرانج وحساب قيمتها وقيم L_i عند أي نقطة وذلك بتعريف الدالة *lin* [7]

```
% Lagrange Interpolation Method
function fi = lin (x,y,xi);
 dxi = xi-x; m = length(x); zeros(size(y));
L(1) = prod (dxi(2:m))/prod(x(1)-x(2:m));
L(m) = prod (dxi(1:m -1))/prod(x(m)-x(1:m -1));
for j = 2:m -1
   n = prod (dxi(1:j -1))*prod(dxi(j+1:m));
   d = prod (x(j)-x(1:j-1))*prod(x(j)-x(j+1:m));
   L(j) = n/d ;
end;
fi = sum (y.*L);
```

خوارزمية (٦,١).

مثال رقم (١,٦)

لتكن قيم الدالة معطاة في الجدول التالي ، أوجد كثيرة حدود لاجرانج لاستكمال هذه الدالة ، وأحسب قيمة الدالة عند 1.5 ؟

i	x_i	$f(x_i)$
0	0	1
1	1	1/2
2	2	1/3

الحل :

بإدخال البيانات نحصل على كثيرة حدود لاجرنج كما هي معرفة في ملف الدالة *lin* وبالأمر *syms xi* :

```
>> x = [0 1 2];
>> y = [1 1/2 1/3];
>> syms xi ;
>> p2 = lin (x,y,xi)
   p2 =
        1/2*(xi-1)*(xi-2)-1/2*xi*(xi-2)+1/6*xi*(xi-1)
```

ويمكن ترتيبها كما يلي :

```
>> p2 = expand (p2)
p2 =
 1/6*xi^2-2/3*xi+1
```

وهذا يعطي كثيرة الحدود :

$$p_2(x) = 0.1667x^2 - 0.6667x + 1$$

ولحساب قيمة كثيرة الحدود عند (1.5) ندخل في دالة *lin* كما يلي :

```
>> p2 = lin(x, y, 1.5)
   p2 =
         0.375
```

مثال رقم (٢,٦)

أوجد كثيرة حدود لاجرانج بالنسبة للنقاط $x_0 = 0, x_1 = 1, x_2 = 2.5$

لاستكمال الدالة $f(x) = \dfrac{2}{x+2}$ عند $x = 2.3$.

الحل :

```
>> x = [0 1 2.5];
>> y = [1 0.6677 0.4444];
>> syms xi;
>> p2 = lin(x,y,xi)
p2 = 2/5*(xi-1)*(xi-5/2)-4/9*xi*(xi-5/2) +16/135*xi*(xi-1)
```

وبعد التبسيط نحصل على :

$$p_2(x) = 0.074x^2 - 2.63x + 1$$

ولحساب قيمة $P_2(2.3)$:

```
p2 = lin(x, y, 2.3)
p2 =
   0.4548
```

وللمقارنة نوجد القيمة الفعلية $f(2.3)$:

$$f(2.3) = \frac{2}{2.3+2} = 0.4651$$

(٦،١،٢) استكمال كثيرة حدود نيوتن

Newton Polynomial Interpolation

تعتمد هذه الطريقة على تقريب المشتقات ، وتسمى الفروق المقسومة Divided Difference وتعرف كما يلي :

الفرق المقسوم الصفري عند النقطة x_i :

$$f[x_i] = f(x_i)$$

والفرق المقسوم من الدرجة الأولى عند النقاط x_0 و x_1 :

$$f[x_0, x_1] = \frac{f[x_1] - f[x_0]}{(x_1 - x_0)} = \frac{f(x_1) - f(x_0)}{(x_1 - x_0)}$$

ومن الدرجة n-1 عند النقاط $x_0, x_1, \ldots, x_n$:

$$f[x_0, x_1, \ldots, x_n] = \frac{f[x_1, x_2, \ldots, x_n] - f[x_0, x_1, \ldots, x_{n-1}]}{(x_n - x_0)}$$

لنحصل على صيغة كثيرة حدود نيوتن لاستكمال الدالة $f(x)$ في النقاط $(x_i)_{i=0}^{n}$:

$$P_n(x) = f[x_0] + \sum_{k=1}^{n} f[x_0, x_1, \ldots x_k](x - x_0)\ldots(x - x_{k-1})$$

مثال رقم (٦,٣)

أ) احسب الفرق المقسوم للدالة $f(x)=x^3+7x^2+1$ عند النقاط $\{x_i\}_{i=0}^{4}$ المعطاة بالجدول التالي :

i	0	1	2	3	4
x_i	1	2	3	4	5

ب) احسب كثيرة حدود نيوتن لاستكمال الدالة عند (1.5).

الحل :

نطبق على MATLAB خوارزمية (٦,٢) ونقوم بإيجاد الفروق المقسومة و كثيرة الحدود وقيمتها عند أي نقطة بالدالة *divdif* [7] :

```
function D = divdif(x,y)
m = length (x);D = zeros(m,m);
D(:,1) = y(:);
for j = 2:m
   for i = j:m
      D(i,j) = (D(i,j-1)- D(i-1,j-1))/(x(i)-x(i-j+1));
   end ;
end
```

خوارزمية (٦,٢).

```
x = [1 2 3 4 5];
y = [9 37 91 177 301];
D = divdif(x,y)
D =
    9     0     0    0    0
   37    28     0    0    0
   91    54    13    0    0
  177    86    16    1    0
  301   124    19    1    0
```

نعوض القيم الموضحة بالجدول في صيغة نيوتن :

```
>> syms x;
>> p = 9+28*(x-1)+13*(x-1) *(x-2)+(x-1) *(x-2) *(x-3);
>> expand (p)
ans =
  1+7*x^2+x^3
```

أي أن كثيرة الحدود هي : $p(x) = x^3 + 7x^2 + 1$.

لإيجاد قيمة الدالة عند (1.5) :

```
>> subs (p, 1.5)
ans =
   20.125
```

كما يوجد لدى MATLAB دوال مخزنة وجاهزة للاستكمال الخطي ومنها *interp1* ولها الصيغة :

p = interp1(x, y, [xi],'linear')

مثال رقم (٦,٤)

لتكن الدالة $f(x) = x^{2.9}$ معرفة عند النقاط $x = 1,2,\ldots,5$ نريد إيجاد تقريب للدالة عند $x = 2.3, 3.8$:

```
>> x = 1:5;
>> y = x. ^(2.9);
>> p = interp1(x, y, [2.3, 3.8],'linear')
p =
  12.4822   49.4104
```

نجد أن الاستكمال يصبح أدق عندما نعتمد على نقاط أكثر ودرجة أعلى، فللحصول على كثيرة حدود من الدرجة الثالثة في MATLAB نستخدم دالة *interp1* ولكن باستخدام ('cubic') :

```
>> x = 1:5;
>> y = x.^(2.9);
>> p = interp1(x, y, [2.3, 3.8],'cubic')
p =
  11.0710   48.1599
```

مثال رقم (٦,٥)

استخدم صيغة الاستكمال المناسبة على الدالة $f(x)$ عند x = 1.23 وجدول القيم التالي، مع المقارنة بين النتائج ، علماً بأن القيمة الحقيقية هي 3.4212 .

x_i	1	1.2	1.4	1.6	1.8
$f(x_i)$	2.7183	3.3201	4.0552	4.9530	6.0496

الحل :

```
>> x = [1 1.2 1.4 1.6 1.8];
>> y = [2.7183 3.3201 4.0552 4.9530 6.0496];
```

تقريب الدالة بصيغة (Lagrange) :

```
>> p = lin (x, y, 1.23)
p =
   3.4212
```

تقريب الدالة بصيغة (Newton) :

```
>> D = divdif(x,y)

D =
   2.7183        0        0        0        0
   3.3201   3.0090        0        0        0
   4.0552   3.6755   1.6663        0        0
   4.9530   4.4890   2.0337   0.6125        0
   6.0496   5.4830   2.4850   0.7521   0.1745

>> x = 1.23;
```

```
>>pd=2.7183+3.0090*(x-1)+1.6663*(x-1)*(x-1.2)+0.6125*(x-1)*(x-1.2)*(x-1.4)
pd =
   3.4211
```

تقريب الدالة بصيغة (*interp1*) الخطية :

```
>> p1 = interp1(x, y, [1.23],'linear')
p1 =
   3.4304
```

تقريب الدالة بصيغة (*interp1*) المكعبة :

```
>> pc = interp1(x, y, [1.23],'cubic')
pc =
   3.4210
```

نلاحظ أن القيم كلها متقاربة ، والأفضل هو تقريب لاجرانج.

(٦,٢) الشريحة التكعيبية للاستكمال Cubic Spline Interpolation

يتضح مما سبق أننا نحتاج إلى زيادة درجة كثيرة الحدود للحصول على تقريب أفضل ، ويمكن تجزئة مجال النقاط المعطاة إلى أجزاء أصغر ، وتطبيق الاستكمال في كل جزء ، وبذلك نتفادى عدم الاستقرار الذي قد يظهر في بعض الفترات الصغيرة من المجال . إحدى الطرق المشهورة لتجزئة فترة الاستكمال إلى فترات صغيرة وإيجاد كثيرات حدود مختلفة تسمى دالة كثيرة حدود تجزيئية Piecewise Polynomial Function ، ومن ثم فإن الدالة التقريبية هي عبارة عن تجميع من هذه الدوال ، و تُعرف بما يلي :

نعرف الدالة $f(x)$ بدالة كثيرة حدود تجزيئية إذا تحقق :

$$f(x) = p_i(x) \text{ لكل } x \in [x_i, x_{i+1}] \text{ و } i = 0, 1, 2, \ldots, n-1$$

وتسمى الأعداد $x_0, x_1, \ldots, x_n$ عقد الدالة f(x) على الفترة [a,b] بحيث إن :

$$a = x_0 < x_1 < x_2 < \ldots < x_{n-1} < x_n = b$$

و $p_0(x), p_1(x), \ldots, p_{n-1}(x)$ كثيرات حدود من الدرجة m على الأكثر.

بصورة عامة دالة الشريحة S(x) Spline functions من الدرجة m على الفترة [a,b]، حيث إن $a = x_0 < x_1 < x_2 < \dots < x_{n-1} < x_n = b$ هي دالة كثيرة حدود تجزيئية بحيث إن :

$$S(x) = p_i(x) \text{ على الفترة } [x_i, x_{i+1}]$$

حيث إن $p_i(x)$ كثيرة حدود من الدرجة m على الأكثر وتحقق :

$p_i^{(k)}(x_{i+1}) = p_{i+1}^{(k)}(x_{i+1})$ لكل و $i = 1, 2, \dots, n-2$ و $k = 1, 2, \dots, m-1$.

وتسمى $x_1, \dots, x_{n-1}$ بالعقد الداخلية. والأكثر شيوعاً من بين هذه الدوال هي الدالة الشريحة التكعيبية من الدرجة الثالثة.

(٦,٢,١) دالة الشريحة التكعيبية Cubic Spline function

لتكن f دالة معرفة على الفترة [a, b] حيث إن $a = x_0 < x_1 < \dots < x_{n-1} < x_n = b$ ، فنقول إن دالة الشريحة S(x) هي دالة شريحة تكعيبية و تستكمل f عند العقد $x_0, x_1, \dots, x_{n-1}$ إذا كانت تحقق :

- $S_i(x_i)$ وهي كثيرة حدود من الدرجة الثالثة على الفترة $[x_i, x_{i+1}]$ لكل $i = 0, 1, \dots, n-1$.
- $S(x_i) = f(x_i)$ لكل $i = 0, 1, \dots, n$.
- $S_{i+1}(x_{i+1}) = S_i(x_{i+1})$ لكل $i = 0, 1, 2, \dots, n-2$.
- $S'_{i+1}(x_{i+1}) = S'_i(x_{i+1})$ لكل $i = 0, 1, 2, \dots, n-2$.
- $S''_{i+1}(x_{i+1}) = S''_i(x_{i+1})$ لكل $i = 0, 1, 2, \dots, n-2$
- أحد الشروط الحدية التالية يكون متحققاً :

i) $S''(x_0) = S''(x_n) = 0$ (حد حر)

ii) $S'(x_0) = f'(x_0)$, $\qquad S'(x_n) = f'(x_n)$, (حد مقيد)

الشريحة التكعيبية تسمى الشريحة الطبيعية عندما يتحقق شرط الحد الحر، ولكن الشروط المقيدة الحدية تؤدي إلى تقريب أكثر دقة، لأنها تحتوي على معلومات أكثر حول الدالة. ويمكن كتابة الدالة التكعيبية على الشكل التالي:

$$S_i(x) = a_i(x-x_i)^3 + b_i(x-x_i)^2 + c_i(x-x_i) + d_i$$

حيث إن a_i, b_i, c_i, d_i لكل $i = 0, 1, 2, \ldots, n-1$ هي الثوابت التي يتم تعيينها.

أما على MATLAB فإنه يوجد دوال *interp1* و *spline* للتعامل مع الشريحة التكعيبية وفق المثال رقم (٦،٦).

مثال رقم (٦،٦)

لتكن قيم x و y التالية $x = \{0, 1, 2, 3, 4\}$ و $y = \{3, 1, 0, 2, 4\}$

أوجد قيمة الدالة y عند $x = 1.5$ باستخدام الشريحة التكعيبية (cubic spline) مع رسم الدالة .

الحل :

نوجد قيمة الدالة y عند $x = 1.5$ كما يلي :

```
>> x = [0 1 2 3 4 ];
>> y = [3 1 0 2 4 ];
>> xval = 1.5;
>> yval = spline(x,y,xval)
yval =
   0.1719
```

ويمكن الوصول لنفس النتيجة بالأمر التالي :

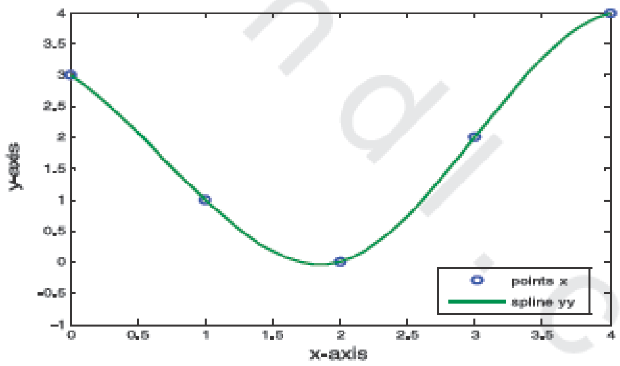
```
YY = ppval(spline(X,Y),XX)
```

أما استخدام *interp1* فيتم بالخطوات التالية :

```
>> p = interp1 (x,y ,[1.5], 'spline')
p =
   0.1719
```

ونرسم الشكل رقم (٦،٢) كما يلي :

```
>> xx = 0:.1:4;
>> yy = spline (x,y,xx);
>> plot (x,y,'o',xx,yy) ; axis ([0 4 -1 4 ]);
>> xlabel ( 'x-axis' ) ; ylabel ('y-axis')
```



الشكل رقم (٦،٢). رسم كثيرة الحدود الناتجة عن استكمال الشريحة المكعبة .

(٦،٣) تقريب بطريقة أصغر المربعات Least squares approximation

لدراسة إيجاد دالة تُمثل كمية كبيرة نسبياً من البيانات تحوي على أخطاءً حصلت في أثناء جمعها ، فإن عملية التقريب تتعلق بإيجاد دالة مطابقة للبيانات ،

واختيار أحسن دالة لتمثيل هذه البيانات. وينصح بطرائق أصغر المربعات المتقطعة عندما تكون الدالة معينة ببيانات نقطية قد لا تمثل قيماً حقيقيةً لدالة. في هذه الحال يمكن لأصغر المربعات أن يأخذ شكلاً خطياً أو بدرجة كثيرة حدود أخرى، أو دالة أُسية أو مثلثية. و ليس من المهم أن نحصل على كثيرة حدود ذات درجة عالية، ولكن الأهم أن تكون انسيابية وبسيطة، حتى لو لم تطابق تماماً البيانات بسبب الأخطاء في إجراء تجميع للبيانات، مثلاً إذا جمعت بيانات لتجربة ما، و كانت كما يلي :

X_i	1	2	3	4	5	6	7	8	9	10
Y_i	1.3	3.5	4.2	5.0	7.0	8.8	10.1	12.5	13.0	15.6

ونريد إيجاد دالة خطية تمر بهذه القيم (x_i , y_i) باستخدام نظرية تقريب أصغر مربعات، فإننا نحتاج لتعريف مجموع نسبة الخطأ ، أو ما يسمى بالانحراف المطلق Absolute deviation وهو :

$$\sum_{i=1}^{m} |y_i - (ax_i + b)|$$

حيث إن $(ax_i + b)$ هي القيمة على الخط المستقيم التقريبي و y_i هي قيمة y المعطاة عند i ، و m هي عدد النقاط المعطاة. إن طريقة أصغر مربعات تعتمد على إيجاد أقل قيمة للانحراف المطلق حيث إنه يُعد أحسن تقريب خطي ، لذلك يتطلب إيجاد قيم a و b حتى نتمكن من تقليل نسبة الخطأ إلى الحد الأدنى. ومن ثم حساب :

$$\varepsilon = \sum_{i=1}^{m} [y_i - (ax_i + b)]^2 \ldots\ldots\ldots\ldots(*)$$

طريقة أصغر مربعات تكون أكثر إجراء ملائم لتحديد أحسن تقريب خطي ، حيث إن هذا التقريب يعتمد على كل النقاط على الخط المستقيم التقريبي وخارجه، و لا يسمح للنقاط الخارجة بالسيطرة الكاملة على التقريب. نريد الآن تقليل المقدار إلى

الحد الأدنى بالنسبة للمتغيرين a و b ، ولكي نحصل على ذلك لا بد أن يكون :

$$0 = \frac{\delta}{\delta a}\sum_{i=1}^{m}[y_i - (ax_i + b)]^2 = 2\sum_{i=1}^{m}(y_i - ax_i - b)(-x_i)$$
$$0 = \frac{\delta}{\delta b}\sum_{i=1}^{m}[y_i - (ax_i + b)]^2 = 2\sum_{i=1}^{m}(y_i - ax_i - b)(-1)$$

نبسط هاتين المعادلتين إلى المعادلتين العاديتين normal equations :

$$a\sum_{i=1}^{m} x_i^2 + b\sum_{i=1}^{m} x_i = \sum_{i=1}^{m} x_i y_i$$
$$a\sum_{i=1}^{m} x_i + b\,.m = \sum_{i=1}^{m} y_i$$

الحل لهذا النظام يكون :

$$a = \frac{m\left(\sum_{i=1}^{m} x_i y_i\right) - \sum_{i=1}^{m} x_i \sum_{i=1}^{m} y_i}{m\left(\sum_{i=1}^{m} x_i^2\right) - \left(\sum_{i=1}^{m} x_i\right)^2}$$

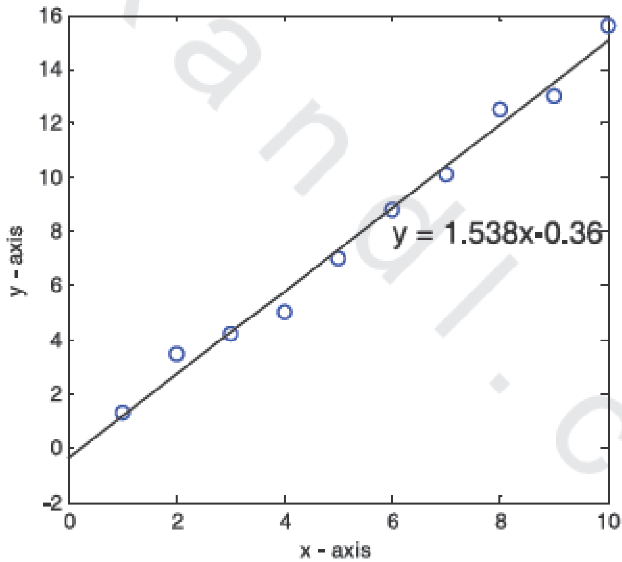
$$b = \frac{\left(\sum_{i=1}^{m} x_i^2 \sum_{i=1}^{m} y_i\right) - \sum_{i=1}^{m} x_i \sum_{i=1}^{m} x_i y_i}{m\left(\sum_{i=1}^{m} x_i^2\right) - \left(\sum_{i=1}^{m} x_i\right)^2}$$

ونطبق ذلك على البيانات لإيجاد أصغر مربعات لتقريب البيانات إلى خط مستقيم. حيث إن قيمتي a و b يمكن حسابهما كما يلي :

$$a = \frac{10(572.4) - 55(81)}{10(385) - (55)^2} = 1.538$$

$$b = \frac{385(81) - 55(572.4)}{10(385) - (55)^2} = -0.360$$

يمكننا الرسم بعد أن حصلنا على القيم التقريبية بوساطة أصغر مربعات لنقط البيانات، كما في الشكل رقم (٦,٣):



الشكل رقم (٦,٣). تمثيل لكثيرة الحدود الخطية الناتجة عن أصغر مربعات.

وبشكل عام لتقريب مجموعة من البيانات $\{(x_i, y_i) : i = 1, 2, \ldots, m\}$ إلى كثيرة حدود باستخدام أصغر

المربعات تقوم بنفس الأسلوب، أي يتطلب منا اختيار الثوابت $a_n , \ldots , a_1 , a_0$ حتى تخفض خطأ أصغر المربعات إلى الحد الأدنى. طريقة إيجاد كثيرة الحدود من الدرجة n لنقاط بيانات x_i بطريقة أصغر مربعات على MATLAB تتم باستخدام دالة تسمى *polfit* ، حيث تعطينا معاملات كثيرة الحدود $p_n(x)$. كذلك يمكننا إيجاد قيم لكثيرة الحدود تساعد في رسمها بدقة بدالة تسمى *polyval*. وهناك خوارزمية (٦،٣) للشكل العام لأصغر مربعات General Least Squares وتدعى *fgenfit1* [15] حيث يمكن استعمالها في تقريب أي مجموعة من البيانات إلى كثيرة حدود مناسبة للبيانات.

```
function c = fgenfit1(func,x,y)
n = length(y);
[p,jj] = feval (func,x(1));
A = zeros(p,p); b = zeros (p,1);
for i = 1:n
  [jj,f] = feval (func,x(i));
   for j = 1:p
      for k = 1:p
          A(j,k) = A(j,k)+f(j)*f(k);
end
        b(j) = b(j)+y(i)*f(j);
   end;
end
c=A\b;
```

خوارزمية (٦،٣).

مثال رقم (٦،٧)

أوجد تقريباً مناسباً للدالة $y = \sin\{1/(x + 0.2)\} + 0.2x$ عند البيانات التالية :

$$xs = [0:0.05:0.25 \qquad 0.25:0.2:4.85]$$

وبنسبة خطأ 0.06 ، مع مقارنة أمر *fgenfit1* بكثيرة الحدود من الدرجتين الثالثة والخامسة *polfit* .

الحل :

نوجد كثيرة الحدود بأمر *fgenfit1* وذلك بتعريف الدالة *f3* في m-file :

$$z_1 = 1 \quad , \quad z_2 = \sin\{1/(x + 0.2)\} \quad , \quad z_3 = x$$

```
function [df,z] = f3(xs)
[j,n] = size(xs);df = 3;  z = zeros (df,n);
z(1,:) = ones (1,n) ;  z (2,:) = sin (ones(1,n)./(xs+0.2));
z (3,:) = xs;
```

نعرف البيانات مع نسبة الخطأ كما يلي :

```
>> xs = [0:.05:.25  .25:.2:4.85];
>> us = sin (ones(size(xs))./(xs+.2))+.2*xs +0.06*randn(size(xs));
```

ننفذ أمر *fgenfit1* لنحصل على معاملات كثيرة الحدود كما يلي :

```
>>  xx=0:.05:5;
>>  c = fgenfit1('f3',xs,us)
c =
-0.0049
1.0716
0.1943
```

فيمكن كتابة كثيرة الحدود كما يلي :

$$p_2(x) = -0.0049 + 1.0716x + 0.1943x^2$$

نوجد قيم كثيرة الحدود الناتجة، و رسمها كما يلي :

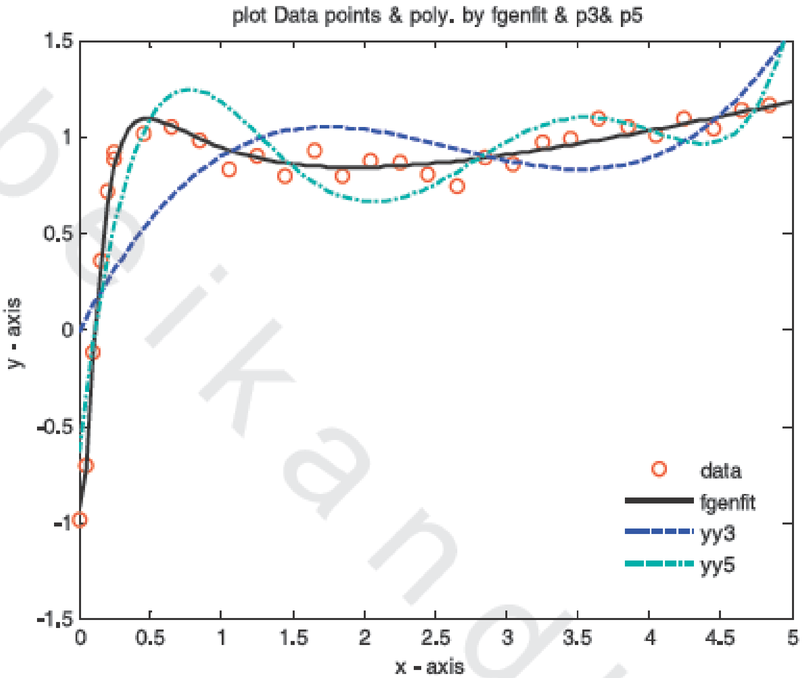
```
[j,p] = feval ('f3',xx);
 yy = c'*p;
 plot (xs,us,'o',xx,yy)
```

نرسم كثيرتي الحدود p_3 و p_5 على نفس الرسم بأمر *polyfit* الشكل رقم (٦،٤) كما يلي :

```
 p3 = polyfit (xs,us,3)
p3 =
   0.0787   -0.6263    1.4811   -0.0368
 yy3 = polyval (p3,xx);
 plot (xx,yy3,'--')
hold on
 p5 = polyfit (xs, us, 5)
p5 =
   0.0505   -0.6821    3.3577   -7.2294    6.3426   -0.6581
 yy5 = polyval (p5, xx,':');
 plot (xx,yy5)
 axis ([0 5 -1.5 1.5]); xlabel ('x - axis ');  ylabel ('y - axis');
title (' plot Data points & poly. by fgenfit & p3& p5'); legend
('data','fgenfit67','yy3','yy5');
```

نجد مما سبق أنه يمكن رسم منحنى تقريبي لأي دالة بأمر *polfit* حيث إنه من الممكن التحكم بدرجة كثيرة الحدود، ولكن لا يعطينا نتائج مثالية. وأما في أمر *fgenfit1* فإنه مبرمج على أن يختار أحسن درجة لكثيرة الحدود تناسب البيانات المعطاة، وبذلك نحصل على منحنى تقريبي مثالي، كما لاحظنا أن نسبة الخطأ بين كثيرات الحدود الناتجة والدالة المعطاة y هي الأقل بأمر *fgenfit1*.

من المناسب أحياناً أن نفترض أن المعطيات مرتبطة فيما بينها أُسّيا، وهذا يعني أن الدالة المقربة هي من الشكل $y=be^{ax}$ ، $y=bx^a$ أو دالة كسرية ، فيمكن استعمال أوامر MATLAB الجاهزة للتعامل معها ، كما في المثال رقم (٦،٨).



الشكل رقم (٦,٤). تمثيل لكثيرات الحدود P5 & P3 & P2 الناتجة عن أصغر مربعات

مثال رقم (٦,٨)

لدينا البيانات التالية :

x= [0:0.25:3]
y= [6.3806 ,7.1338 ,9.1662 ,11.5545 ,15.6414 ,22.7371 ,32.0696 ,47.0756 ,73.1596, 111.4684, 175.9895 ,278.5550 ,446.4441]

١- أوجد كثيرات الحدود على النمط الاتي :

1.$f_1(x) = a + be^x + ce^{2x}$ (استعمل أمر fgenfit)

2.$f_2(x) = a + b/(1+x) + c/(1+x)^2$ (استعمل أمر fgenfit)

3.$f_3(x) = a + bx + cx^2 + dx^3$ (استعمل أمر polyfit)

٢- ارسم الدوال الثلاث والنقاط (y). أيٌّ منها يمثل أقرب منحنى للنقاط المعطاة؟ علماً بأن القيم أخذت من $f(x) = 3 + 2e^x + e^{2x}$ مع إضافة تغييرٍ صغير عشوائي عليها.

الحل :

نعرف الدوال *f11(x)* و *f22(x)* على شكل m-file :

```
function [d,z]=f11(x)
[j,n]=size(x);d=3;z=zeros(d,n);
z(1,:)=ones(1,n);z(2,:)=exp(x);z(3,:)=exp(2.*x);
```

```
function [d,z]=f22(x)
[j,n]=size(x);d=3;z=zeros(d,n);
z(1,:)=ones(1,n);z(2,:)=(ones(1,n)./(1+x));z(3,:)=(ones(1,n)./((x+1).^2))
```

أما طريقة إيجاد الثوابت ورسم كثيرات الحدود *f1* و *f2* و *f3* مع النقاط y ،

فهي :

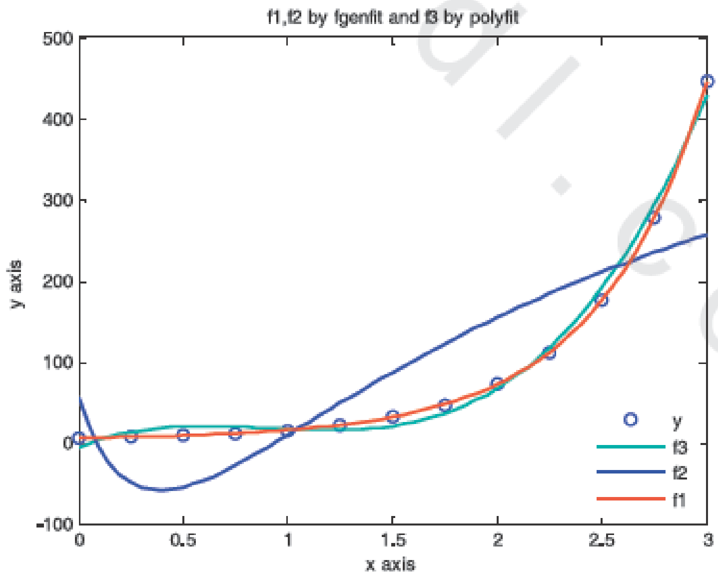
```
x = [0:0.25:3];
y = [6.3806 7.1338 9.1662 11.5545 15.6414 22.7371 32.0696 47.0756
73.1596   111.4684 175.9895 278.5550 446.4441];
xx = [0:.05:3];
f1 = fgenfit('f11',x,y)
[j,p] = feval('f11',xx); yy1 = (f1)'*p;
f2 = fgenfit1 ('f22',x,y)
[j,p] = feval('f22',xx); yy2 = (f2)'*p;
xx = [0:.05:3];
p = polyfit(x,y,3)
yy3 = polyval(p,xx);
```

```
plot (x,y,'o',xx,yy1,xx,yy2,xx,yy3) ;   xlabel ('x axis'); ylabel ('y axis');
title ('figure(1) f1,f2 by fgenfit and f3 by polyfit'); legend ('y','yy1','yy2','yy3')
```

f1 =
3.1276
1.9811
1.0000
f2 =
*1.0e + 003 **
0.6851
-2.0722
1.4438
f3 =
47.3747 -128.3479 103.4153   -5.2803

نجد في الشكل رقم (٦،٥) أن كثيرات الحدود الثلاث هي :

i) $f_1(x) = 3.1276 + 1.9811e^x + e^{2x}$
ii) $f_2(x) = 685.1 - 2072.2/(1+x) + 1443.8/(1+x)^2$
iii) $f_3(x) = 47.3747\, x^3 - 128.3479\, x^2 + 103.4153\, x - 5.2803$



الشكل رقم (٦،٥). رسم الدوال f3 & f2 & f1 مع نقاط الدالة الأصلية.

(٦,٤) تحليل فورير Fourier Analysis

(٦,٤,١) متسلسلات فورير Fourier Series

يمكن أن يُنتج تقريب عدد كبير من النقاط الموزعة بانتظام بواسطة كثيرات حدود مثلثية نتائج دقيقةً جداً ، وهي طريقة التقريب المناسبة للاستخدام في مجالات كثيرة، كالأجهزة التي تحتوي على مصفيات عددية، أو في الموجات التي يتم استقبالها في الهوائيات، وكذلك في الميكانيكا وغيرها من المجالات. إلا أن هذه الطريقة لم تطّبق كثيراً حتى منتصف عام ١٩٦٠م، وذلك بسبب الحسابات العديدة اللازمة لتعيين الثوابت في عملية التقريب، إذ إن استكمال عدد 2m من النقاط بالطرائق الحسابية المباشرة يتطلب حوالي $(2m)^2$ عملية ضرب، ومثلها عملية جمع. وفي عام ١٩٦٥ م نُشرت طريقة جديدة لحساب ثوابت كثيرة الحدود الاستكمالية المثلثية، و بعدد حسابات أقل ، بشرط أن نختار m بطريقة مناسبة. وأحدثت هذه الطريقة ثورة في استخدامات كثيرات حدود الاستكمال المثلثية وعرفت بخوارزمية تحويل فورير السريع .

وتعرف متسلسلات فورير للدالة $f(x)$ الدورية ودورتها 2π كما يلي :

$$f(t)=\frac{c_0}{2}+\sum_{n=1}^{\infty}c_n\cos(nt)+\sum_{n=1}^{\infty}s_n\sin(nt)$$

حيث إن الثوابت s_n و c_n تعرف كما يلي :

$$c_n=\frac{1}{\pi}\int_{-\pi}^{\pi}f(t)\cos(nt)dt$$

$$s_n=\frac{1}{\pi}\int_{-\pi}^{\pi}f(t)\sin(nt)dt$$

وفي MATLAB فإنـه مـن الممكـن إيجـاد ثوابـت متسلسـلة فـورير ورسـمها بدقـة ووضوح.

مثال رقم (٦,٩)

أوجد متسلسلة فورير للدالة $f(t)= sin(t)\ cos(t)^2$ على الفترة $[-\pi,\pi]$ والثوابت S_n و C_n بالرسم.

الحل :

نعرف الدالة بوساطة m-file كما يلي :

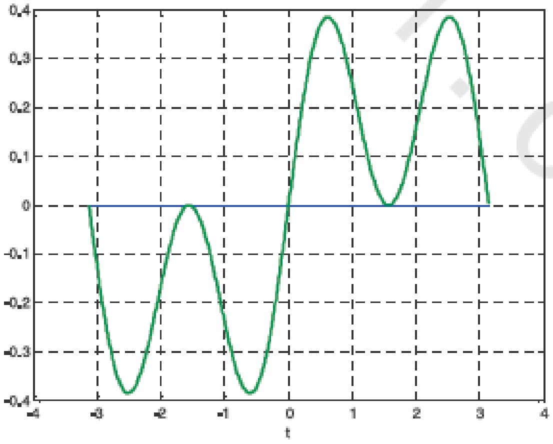
```
function F = fun (T);
F = sin (T).*(cos (T).^2);
```

ونعرف الدالتين *fc* و *fs* على شكل m-file كما يلي :

```
% c
function Y = fc (T);
global n
Y = fun(T).*cos (n.*T)
% S
function Y= fs (T);
global n
Y= fun (T).*sin (n.*T)
```

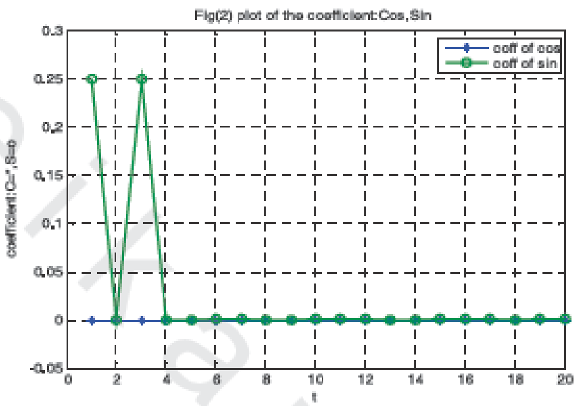
نستخدم للتكامل دالة *quad8* على الفترة [π- , π] ونجمع الحدود لتكوين كثيرة حدود فورير النهائية y. نرسم الدالة الأصلية f و الدالة الجديدة y ، وكذلك نرسم المعاملات بالأوامر التالية (الشكل رقم ٦,٦) :

```
global n;T = -pi:.01:pi ; Y = 0*T;
for n1= 0:20
   n = n1;
   c = quad8 ('fc',-pi,pi)/pi;
   if n == 0,c = c/2 ;end
   s = quad8 ('fs',-pi,pi)/pi;
     Y = Y + c*cos (n*T) + s*sin (n*T);
   figure (1),plot (T,Y,T, fun (T) ), grid on
         if n > 0
     N (n) = n ; C (n) = c ; S (n) = s;
                         end
end
figure (2), plot (N,C,'*-',N,C,N,S, 'o-',N,S), grid on
```



الشكل رقم (٦،٦). رسم متسلسلة فورير للدالة sin(t)^2* cos(t)^2 .

من خلال الشكل رقم (٦،٧) نجد أن S_1 , S_3 تساويان 0.25 وأما باقي الثوابت تكاد تكون صفرية.



الشكل رقم (٦،٧). رسم ثوابت متسلسلة فورير Sn و Cn .

(٦،٤،٢) تحويلات فورير Fourier Transforms

يُعد تحويل فورير من أهم وأكثر التوابع استخداماً ، حيث يساعدنا على تحليل البيانات لمعرفة التردد، وهذا مفيد جداً لمعرفة المحتوى الترددي للبيانات وتستخدم المصفوفات للتعبيرعن صورة أو إشارة صوتية أو غيرها. نعرف تكامل فورير للدالة *f(t)* الدورية، والتي طول دورتها L ، على الشكل :

$$f(t) = \int_0^{\infty} c(v)\cos(2\pi v t)dv + \int_0^{\infty} s(v)\sin(2\pi v t)dv$$

حيث إن $\frac{n}{L} = v$ التردد لدالة متغيرة مع الزمن ويعبر عنه أيضاً بالصيغة المركبة الأسّية :

$$f(t) = \int_{-\infty}^{\infty} g(v)\exp(i\,2\pi v t)\,dv$$

وبشكل مماثل يمكن التعبير عن g(v) كما يلي :

$$g(v) = \int_{-\infty}^{\infty} f(t)\exp(-i\,2\pi v t)\,dt$$

أي أن :

$$\begin{aligned} g(v) &= \int_{-\infty}^{\infty} f(t)\{\cos(-2\pi v t) + i\,\sin(-2\pi v t)\}dt \\ &= \int_{-\infty}^{\infty} f(t)\cos(2\pi v t)dt - i\int_{-\infty}^{\infty} f(t)\sin(2\pi v t)dt \,........(*) \end{aligned}$$

فعندما تكون الدالة $f(t)$ زوجية ، فإن التكامل الثاني يكون صفراً ، وتتحول الدالة $g(v)$ إلى دالة حقيقية. أما إذا كانت الدالة $f(t)$ فردية ، فإن التكامل الأول يتلاشى وتتحول الدالة $g(v)$ إلى دالة تخيلية.

(٦,٤,٣) تحويلات فورير المتقطعة (DFT) Discrete Fourier Transforms

لتحليل موجة نبضية من حيث التذبذب البسيط على سلسلة من الفترات الزمنية، لتكن $t = t + k\,\delta t$ حيث إن $k = 1, 2, ...$ و δt هي الفترة الزمنية بين قياسين متتاليين للزمن، نستخدم تحويلات فورير المتقطعة. ويمكن التعبير

عن متسلسلة فورير بالشكل التالي :

$$g(v_m) = \int_{-\infty}^{\infty} f(t)\exp(-i\,2\pi v_m)dt \cong \delta t \sum_{k=1}^{n} f(t_k)\exp(-i\,2\pi v_m t_k)$$

وفي كل مجموع يعطينا قيمة لـ $g(v)$ عند التردد v_m . تحويل فورير البسيط Simple Fourier Transform يعرف بالدالة *sft* (خوارزمية ٦,٤) التي تنفذ تحويل فورير البسيط على MATLAB [6] كما يلي ، لاستخدامها في العديد من الأمثلة.

```
% Simple Fourier Transform
function G = sft(T,F,N);
                    dt= T(2)-T(1);
n = length(N);
for k =1:n
   G(k)= dt*sum (F.*exp (-i*2*pi*N(k)*T));
end
```

خوارزمية (٦,٤).

والمثال رقم (٦,١٠) يوضح استخدام MATLAB في رسم كثيرة الحدود الناتجة عن تحويل فورير بسيط لدالة الجا Transform of a Sine Function .

مثال رقم (٦,١٠)

أخذت عينة تحتوي على 1001 نقطة من البيانات ، ليكن لدينا الثوابت التالية :

. a = (2/n) و n = 2

ارسم دالة تحويل فورير للدالة (sine) وكذلك المعاملات (b_n).

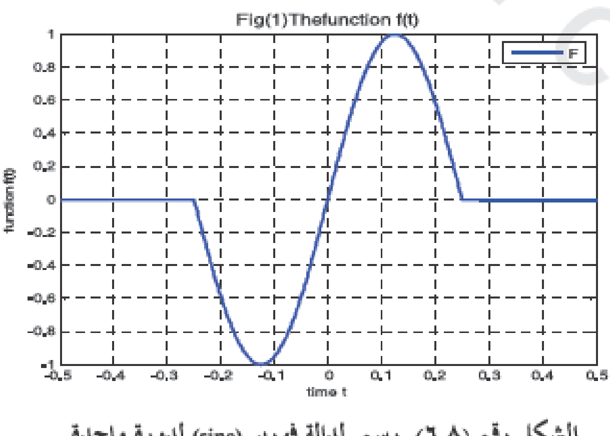
الحل :

في البداية نعرف الدالة التالية على شكل m-file وهي تساعد في تعريف الدالة F الوحدة النبضية [6] :

```
% Rectangular Pulse
function up = unitpulse (x,y,z);
up = unitstep (x-y) - unitstep(x-z);
```

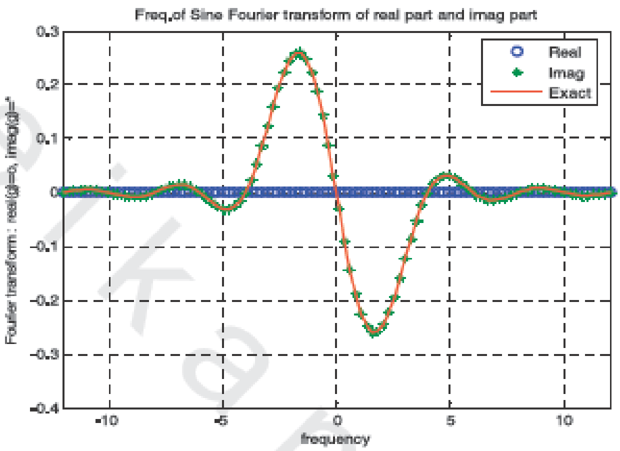
والآن لإيجاد تحويل فورير للدالة (sine) نقوم بتنفيذ الأوامر وينتج الشكل رقم (٦,٨):

```
nn = 1001;
 n = 2;  a =2/n; w = 2*pi*n;
t = 2*a;
dt = 2*t/(nn-1);
T = -t :dt :t;
F = sin (w*T).*unitpulse (T,-a,a);
figure (1), plot (T,F);grid on ;
 N = linspace (-12,12,100);
G = sft (T,F,N);
W = 2*pi*N;
Gm =(sin (a*(W+w))./(W+w) - sin(a*(W-w))./(W-w));
figure (2), plot(N, real (G), 'o', N, imag(G), '*', N, Gm), grid on
```



الشكل رقم (٦,٨). رسم لدالة فورير (sine) لدورة واحدة .

نلاحظ من الشكل رقم (٦،٩) أن المعاملات (a_n) الحقيقية كلها صفرية ، و ذلك لأن دالة (sine) عبارة عن دالة فردية.



الشكل رقم (٦،٩). رسم الثوابت الحقيقية (O) و التخيلية (*) فورير sine .

(٦،٤،٤) تحويلات فورير العكسية Inverse Transform of Fourier

قد نحتاج إلى الطريقة العكسية ، وذلك الحصول على الدالة *f(t)* من *g(v)* معلومة ، وهذا ما يسمى بتحويلات فورير العكسية **Inverse Transform of Fourier**، ويتم ذلك في MATLAB بالدالة (خوارزمية ٦،٥) *isft* ونلاحظ أن الفرق بينها وبين *sft* هو التبديل بين متغيرات المدخلات والمخرجات [6] .

```
Function f=isft(n,g,t);
D=n(2)-n(1);
nt=length(t);
for k=1:nt
F(k)=D*sum(g.*exp(i*2*pi*n*t(k)));
end
```

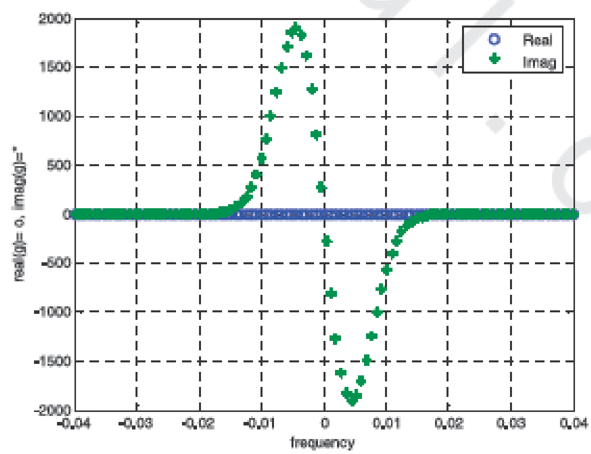
خوارزمية (٦،٥).

مثال رقم (٦,١١)

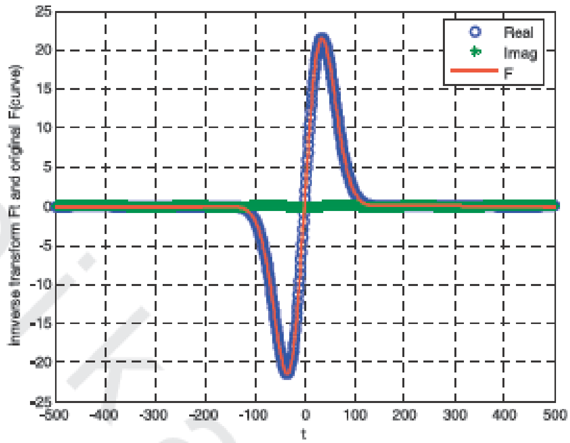
ليكن لدينا الدالة $f(t) = te^{-t^2/a^2}$ طبّق عليها تحويل فورير لتحصل على الدالة G ، استخدم الدالة G لإيجاد تحويل فورير العكسي (ft) وحيث إن a=50.

الحل :

نطبق دالتي isft و sft ونلاحظ من الرسم الأول (الشكل رقم ٦,١٠) المعاملات التخيلية للدالة (G) وهي التي نريد إيجاد فورير العكسي لها، والتي تشبه تماماً الدوال الفردية. كما نجد من الرسم في الشكل رقم (٦,١١) انطباق (ft) دالة التحويل العكسي على الدالة الأصلية f ويدل الرسم أيضاً على أن (ft) عبارة عن دالة ذات قيم حقيقية.



الشكل رقم (٦,١٠). رسم الثوابت الحقيقية (O) و التخيلية (*) لدالة فورير G(t) .



الشكل رقم (٦،١١). رسم الثوابت الحقيقية (O) لدالة فورير العكسي (ft) والدالة (f) .

(٦،٤،٥) دوال رمزية لتحويل فورير

توجد دوال جاهزة لتحويل فورير بالبيئة الرمزية Symbolic Fourier Transform .

فعلى سبيل المثال ، إذا كانت لدينا الدالة المعرفة في المثال السابق $f(t) = te^{-t^2/a^2}$

فتحويل فورير لهذه الدالة باستخدام syms يتم بالأمر *fourier* :

```
syms a t real
a=50
g=fourier(t*exp(- t^2/a^2))
ans
g=-62500*i*pi^(1/2)*w*exp(-625*w^2)
```

وكذلك يمكن إيجاد تحويل فورير العكسي للنتيجة بالأمر *ifourie* :

```
>> syms  w  t
>>f= ifourier(g,w,t)
 ans =
 t*exp(-1/2500*t^2)
```

(٦,٤,٦) تحويلات فورير السريع Fast Fourier Transforms

تُعد الدالة *fft* من الدوال الجاهزة في **MATLAB** وتحسب تحويل فورير المتقطع لمتجه باستخدام خوارزمية تحويل فورير السريع ، ولها عدة صيغ منها :

y = fft(x,n)
y = fft(x,n,dim)

هذه الدالة مبرمجة على أن تعمل على خوارزميتين لحساب تحويل فورير ، حيث تستخدم الخوارزمية الأسرع عند وجود عدد b بحيث يحقق هذا العدد العلاقة $n = 2^b$ (أي عندما يكون عدد النقاط n من قوى 2) وعند عدم تحقق ذلك يتم تنفيذ الخوارزمية ببطء ، وهناك من يضيف أصفاراً ليصبح عدد النقاط من قوى ٢ لغرض السرعة. يمكن إيجاد الثوابت الحقيقية (C_n) والثوابت التخيلية (S_n) باستخدام *fft*، لتعبر عن تحويل فورير المتقطع (DFT) ، ونبين في الجدول رقم (٦,١) دوال أخرى تحسب تحليل فورير والتحليل العكسي في أبعاد مختلفة.

الجدول رقم (٦,١). دوال التحليل الفورير.

fft	تحويل فورير المتقطع (Discrete Fourier Transform)
ifft	تحويل فورير العكسي المتقطع (Inverse Discrete Fourier Transform)
fft2	تحويل فورير المتقطع البعد الثاني (Discrete Fourier Transform 2-D)
fftn	تحويل فورير المتقطع البعد النوني (Discrete Fourier Transform n-D)
Ifft2	تحويل فورير المتقطع العكسي البعد الثاني (Inverse Discrete Fourier Transform 2-D)
ifftn	تحويل فورير المتقطع العكسي البعد النوني (Inverse Discrete Fourier Transform n-D)

مثال رقم (٦,١٢)

إذا كانت y كما يلي :

y = [2, -0.404, 0.2346, 2.6687, -1.4142, -1.0973, 0.8478, -2.37, 0, 2.37, -0.8478,... 1.0973, 1.4142,-2.6687, -0.2346, 0.404, -2 ,1.8182, 1.7654, -1.2545, 1.4142, -0.3169, -2.8478, 0.9558, 0, -0.9558, 2.8478, 0.3169, -1.4142, 1.2545, -1.7654, -1.8182].

وهي عبارة عن قيم دورية عددها 32 نقطة ، أُخذت هذه العينة بفرق 0.1 من الثانية بين كل نقطتين متتاليتين.

١– أوجد الجزء الحقيقي والجزء التخيلي من تحويل فورير المتقطع DFT باستخدام MATLAB بأمر *fft* .

٢– ارسم الجزء التخيلي والحقيقي الناتج عن تحويل فورير DFT .

الحل :

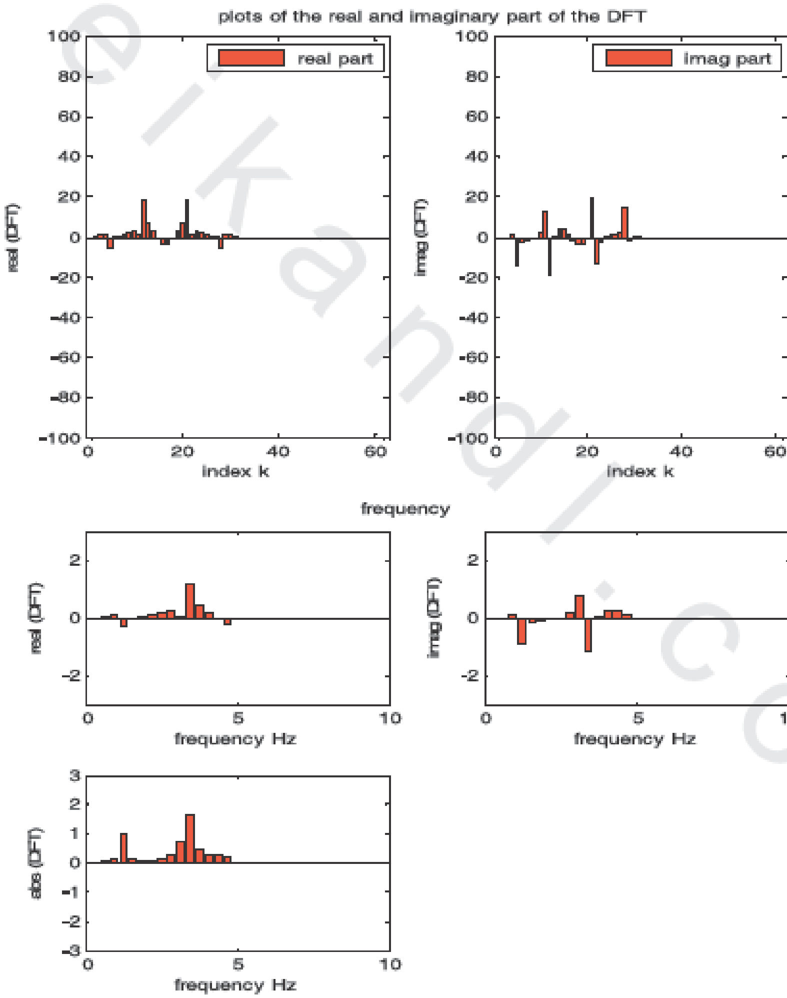
طريقة إيجاد الجزء الحقيقي والجزء التخيلي من تحويل فورير المتقطع DFT :

```
y = [2 -0.404 0.2346 2.6687 -1.4142 -1.0973 0.8478 -2.37 0 2.37 -0.8478
1.097 1.4142 -2.6687 -0.2346 0.404 -2 1.8182 1.7654 -1.2545 1.4142 -0.3169 -
2.8478 0.9558 0 -0.9558 2.8478 0.3169 -1.4142 1.2545 -1.7654 -1.8182];
v = 0:31; s = sum (y);
y1 = fft (y);
s = 4.4409e-016;
real(y1);imag(y1);
```

طريقة رسم الجزء التخيلي والحقيقي للقيم الناتجة عن تحويل فورير المتقطع DFT تتم بتنفيذ الخطوات التالية وينتج الشكل رقم (٦,٢) :

```
nt = 32; T = 3.2 ;  dt = T/nt
df = 1/T
fmax = (nt/2)*df
t = 0:dt:(nt-1)*dt;
f = 0:df:(nt-1)*df;
 figure (1);
subplot (121); bar(real (y1),'r'); axis ( [0 63 -100 100] )
```

```
subplot (122);  bar (imag (y1),'r');  axis([0 63 -100 100])
fss = 0:df: (nt/2-1)*df;
yss = zeros(1,nt/2); yss (1:nt/2) = (2/nt)*y1(1:nt/2);
figure(2);
subplot (221); bar (fss,real (yss),'r'); axis ( [0 10 -3 3] )
subplot (222); bar (fss,imag (yss),'r'); axis ( [0 10 -3 3] )
subplot (223);  bar (fss,abs (yss),'r'); axis ( [0 10 -3 3] )
```



الشكل رقم (٦،١٢). رسم للجزء الحقيقي والتخيلي لمتسلسلة فورير المتقطعة وطيف التردد .

(٦,٥) تمارين

١- استخدم كل دوال الاستكمال المذكورة في الفصل و قارن النتائج بالرسم :

x=[-2 0 2 3 4 5]
f(x)= [4 0 -4 -30 -40 -50]

٢- أوجد الدالة التي تمر بالقيم (x,y) باستخدام طريقة أصغر المربعات :

x=[0 .005 0.0075 0.0125 0.025 0.05 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1]
y=[0 0.0102 0.0134 0.017 0.025 0.0376 0.0563 0.0812 0.0962 0.1035 0.1033 0.0950.0802 0.0597 0.0340 0]

٣- أوجد تحويل فورير للدوال التالية :

أ) $f = A * \sin(w_0 T) * unitpulse(T, -a, a)$ باستخدام fft.

ب) $|\sin(w_0 t)|$ على الفترة [-π, π]

جـ) $f(t) = \begin{cases} -t & for \quad t \le 0 \\ t & for \quad t > 0 \end{cases}$ على الفترة [-10,10]

٤- استخدم قانون نيوتن للفرق التقدمي لإنشاء كثيرة حدود من الدرجة الأولى، الثانية و الثالثة للنقاط غير المتساوية المسافة المعطاة بالجدول التالي :

x	f(x)
0	-6.0
0.1	-5.89483
0.3	-5.65014
0.6	-5.17788
1.0	-4.28172

٥- قرب $f(0.5)$ باستخدام قانون نيوتن للفرق من الجدول التالي :

x	f(x)
0	1.0
0.2	1.22140
0.4	1.49182
0.6	1.82212
0.8	2.22554

٦- أوجد كثيرة حدود استكمال لاغرانج من الدرجة الثانية والتي تمر بالقيم في الجدول السابق.

٧- قدر $f(0.5)$ عن طريق كثيرة حدود استكمال لاغرانج من الدرجة الثالثة للقيم في الجدول السابق.