Lecture 9

第八章 內插法(Interpolation)

大綱

- 多項式內插 (Polynomial Interpolation)
- Hermite內插
- 分段多項式內插 (Piecewise polynomial interpolation)
- 進階問題

概述

- 在許多情形下希望用函數來表示在一組離散的點的行為。在一些情形下，可能只有某些點的數值。另外一些情形則是，希望用較簡單的形式來表示函數，同時維持某些點 (節點)的值與原函數一致。這兩種觀點可分別叫做「數據內插」及「函數內插」，它們之間的一點差異就是已知資訊有所不同。不論是那種情形，都希望能估計出其它點的函數值，否則就必須使用完整的函數形式以滿足其它數值方法，例如數值微分或數值積分。

概述

- 內插法能夠產生一個函數，可在所有已知點上完全符合原函數;要找的是，對中間的點也能提供良好近似值的函數。這些數據可能來自實驗的量測值，或其它數值方法的計算結果，例如微分方程的解。
- 考慮的第一種內插法，它用一個適當次數的多項式，以完全吻合一組已知數據。對兩種標準型式的多項式(Lagrange和牛頓)都會做說明。除了函數值之外，也可內插導數值。本章也納入Hermite內插，做為對函數及一階導數作「密切內插 (Osculatory interpolation)」的最簡單範例。

概述

- 分段多項式內插讓我們可以產生，通過相當多數據點，且沒有過多「折曲」的平滑曲線。
  - *分段線性及二次式內插。*
  - *分段多項式內插法之一，所謂的三次雲形線(Cubic splines)。*
- 有理函數(Rational function)內插。

應用問題8-A化學反應產物

- 將簡單化學反應之生成物濃度，視為時間的函數:
  - *時間:　　0.0　0.5　1.0　1.5　2.0*
  - *生成物:　0.0　0.19　0.26　0.29　0.31*
- 在本章要討論幾種不同的方法，以求出生成物濃度在其它時間的近似值。已知數據顯示如圖8.1。

應用問題8-A化學反應產物

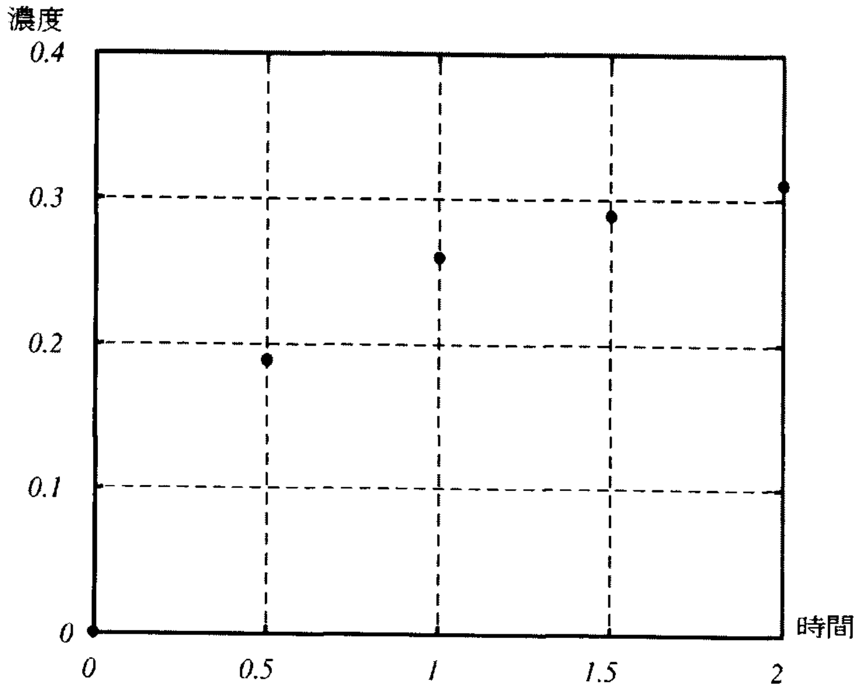


圖 8.1　化學反應生成物數據

應用問題8-A化學反應產物

- 考慮使用更多數據
- 時間:　0.0　0.1　0.4　0.5　0.6　0.9　1.0　1.1　1.4　1.5　1.6　1.9　2.0
- 生成物:0.0　0.06　0.17　0.19　0.21　0.25　0.26　0.27　0.29　0.29　0.30　0.31　0.31
-

應用問題8-B參數化函數的雲形線內插

- 在電腦繪圖中，常用雲形線內插以產生平滑曲線(以參數型式)。沿曲線選取數個點，用參數t來標示。分別對x和y座標進行內插(分別視為是t的函數)。所得的參數圖形(x(t),y(t))即為內插曲線。
- 例如：

```
t = [ 1    2    3    4    5    6    7    8    9    10    11    12 ]
x = [ 0    1    2    2    1    1    2    3    3     3     4     5 ]
y = [ 0    0    0    1    1    2    2    2    1     0     0     0 ]
```


應用問題8-B參數化函數的雲形線內插

- 雖然在本例中，似乎很適合使用等間隔參數，但這並非必要的。當圖形中兩相鄰點間的歐幾里得距離並不全都一樣時，有時須要調整參數值，以反應點與點間的距離。對於參數t的選取，可能要做些實驗以獲得最好的結果。

應用問題8-B參數化函數的雲形線內插

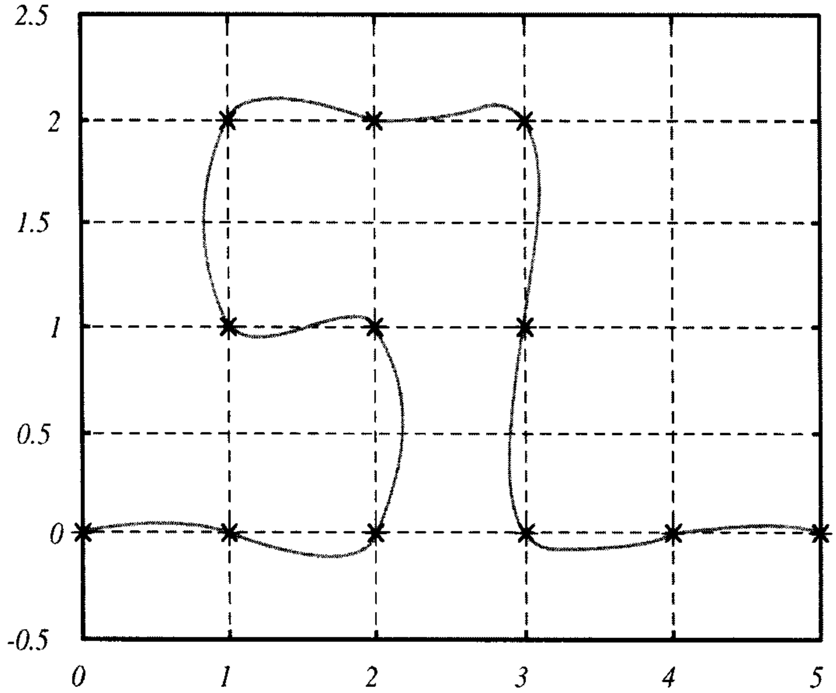


圖 8.2　參數曲線的雲形線內插

8.1 多項式內插

- 最常用的兩種多項式內插為
  - *Lagrangian內插法*
  - *牛頓內插法*
- 當然，由同一組數據所決定之多項式是唯一的，不同的形式只出現在求多項式的過程中(以及呈現的方式)，而不是最後所得的函數。不同情況下，每種方法各有優點。

Lagrange形式

- 通過(x_1,y_1)和(x_2,y_2)兩點之直線方程式的Lagrange形式為

$$p(x)=\frac{(x-x_2)}{(x_1-x_2)}y_1+\frac{(x-x_1)}{(x_2-x_1)}y_2$$

- 此方程式代表一條通過所給兩點的直線。
- 同理，通過(x_1,y_1)和(x_2,y_2)三點之拋物線方程式的Lagrange形式為

$$p(x)=\frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)}y_1+\frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)}y_2+\frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)}y_3$$

Lagrange形式

- 因此，通過(x_1,y_1), …,(x_n,y_n)等n點之多項式的Lagrange通式在等號右側有n項，每項對應一個點

$$p(x)=L_1y_1+L_2y_2+\cdots+L_ny_n$$

- 第k項是第k筆數據和以下(n-1)次多項式的乘積

$$L_k(x)=\frac{(x-x_1)\cdots(x-x_{k-1})(x-x_{k+1})\cdots(x-x_n)}{(x_k-x_1)\cdots(x_k-x_{k-1})(x_k-x_{k+1})\cdots(x_k-x_n)}$$

分子為

$$N_k(x)=(x-x_1)\cdots(x-x_{k-1})(x-x_{k+1})\cdots(x-x_n)$$

Lagrange形式

- 分母的形式一樣，只是將變數x換成已知值x_k:

$$D_x = N_k(x_k) = (x_k - x_1)\cdots(x_k - x_{k-1})(x_k - x_{k+1})\cdots(x_k - x_n)$$

- 因此$L_k(x_k)$是1，且當x等於其它已知點的值時，$L_k(x)$為0，亦即當$x=x_j$時$L_k(x_j)=0$，其中$j \neq k$。

例題8.1 Lagrange內插拋物線

- 可以用以下三個已知點找出一個二次多項式

$$(x_1, y_1) = (-2, 4)$$
$$(x_2, y_2) = (0, 2)$$
$$(x_3, y_3) = (2, 8)$$

代入通式：

$$p(x) = \frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)}y_1 + \frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)}y_2 + \frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)}y_3$$

由此可得

$$p(x) = \frac{(x-0)(x-2)}{(-2-0)(-2-2)}4 + \frac{(x-(-2))(x-2)}{(0-(-2))(0-2)}2 + \frac{(x-(-2))(x-0)}{(2-(-2))(2-0)}8$$

化簡爲

$$p(x) = \frac{x(x-2)}{8}4 + \frac{(x+2)(x-2)}{-4}2 + \frac{x(x+2)}{8}8$$
$$= x^2 + x + 2$$

例題8.1 Lagrange內插拋物線

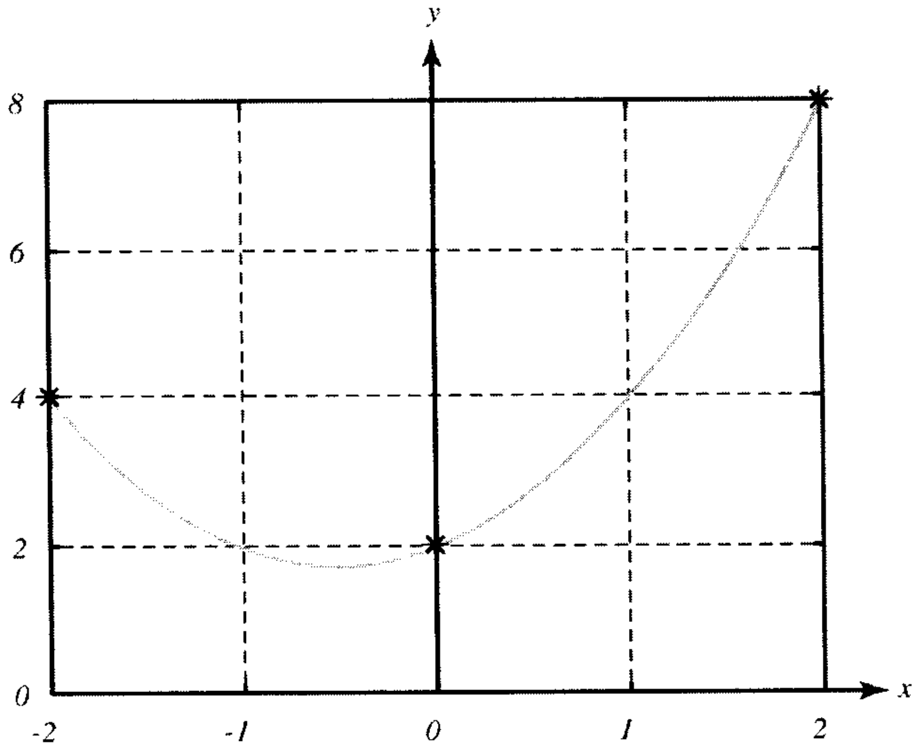


圖 8.3　拋物線內插多項式

例題8.1 Lagrange內插拋物線

- 以下的 MALAB 函數，可以求出表示為以下形式之Lagrange 內插多項式的係數 c_k

$$p(x) = c_1 N_1 + c_2 N_2 + \cdots + c_n N_n$$

其中

$$c_k = \frac{y_k}{D_k} = \frac{y_k}{(x_k - x_1)\cdots(x_k - x_{k-1})(x_k - x_{k+1})\cdots(x_k - x_n)}$$

且

$$N_k(x) = (x - x_1)\cdots(x - x_{k-1})(x - x_{k+1})\cdots(x - x_n)$$

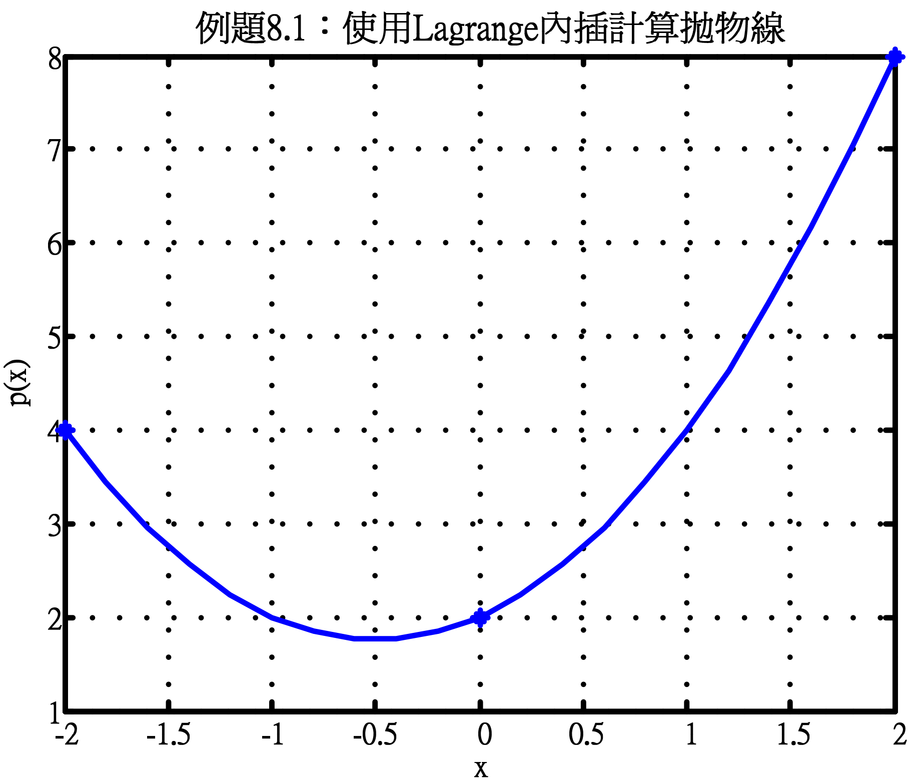
例題8.1Matlab程式

- 主程式：Main_Lagrange_Ex_81.m
- Lagrange內插係數程式：LagrangeCoef.m
- Lagrange內插點程式：LagrangeEval.m

```
clear all;
close all;

x=[-2 0 2];
y=[4 2 8];
t=-2:0.2:2;
c = LagrangeCoef(x, y);
p = LagrangeEval(t, x, c);
plot(t,p,'-b',x,y,'*');
xlabel('x');
ylabel('p(x)');
title('例題8.1：使用Lagrange內插計算拋物線');
grid on;
```


例題8.1Matlab程式

- 結果：



例題8.2　化學反應生成物數據

- 求應用問題8-A中所給的生成物濃度的內插多項式，其中
- x=[0.00 0.50 1.00 1.50 2.00]
- y=[0.00 0.19 0.26 0.29 0.31]

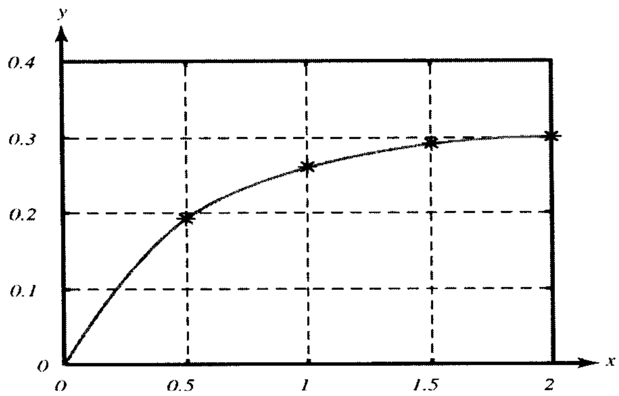


圖 8.4　化學反應生成物數據之二次內插多項式

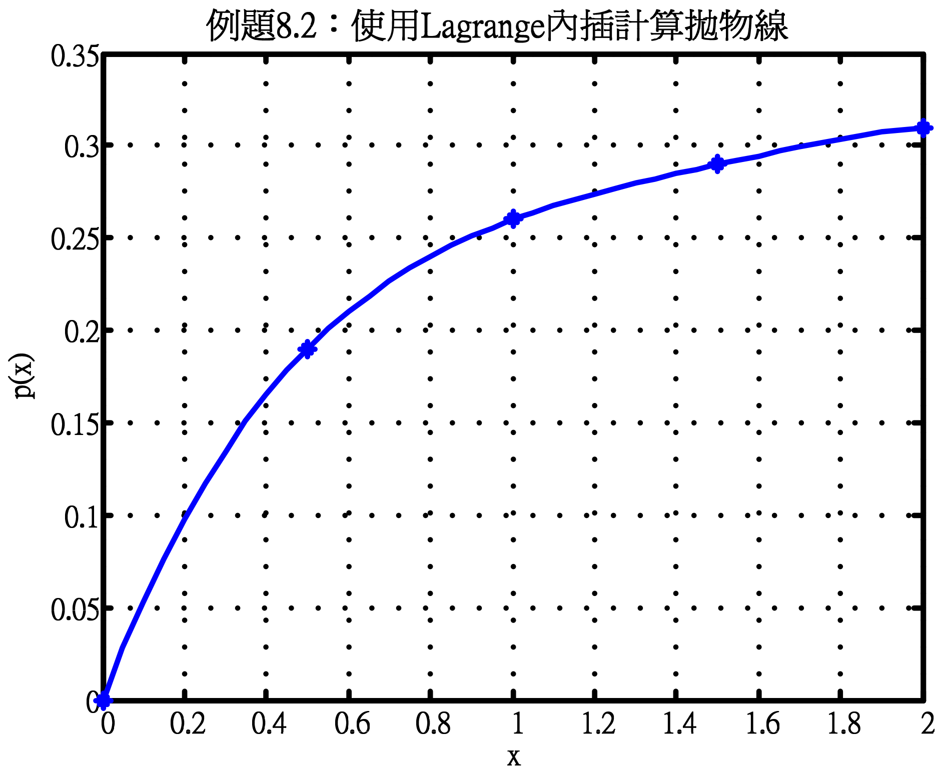
例題8.2Matlab程式

- 主程式：Main_Lagrange_Ex_82.m
- Lagrange內插係數程式：LagrangeCoef.m
- Lagrange內插點程式：LagrangeEval.m

```
clear all;
close all;

x=[0.00 0.50 1.00 1.50 2.00];
y=[0.00 0.19 0.26 0.29 0.31];
t=0:0.1:2;
c = LagrangeCoef(x, y);
p = LagrangeEval(t, x, c);
plot(t,p,'-b',x,y,'*');
xlabel('x');
ylabel('p(x)');
title('例題8.2：使用Lagrange內插計算拋物線');
grid on;
```

例題8.2Matlab程式

- 結果：



例題8.2　化學反應生成物數據

- 考慮使用更多數據
  - ***時間:*** ***0.0 0.1 0.4 0.5 0.6 0.9 1.0 1.1 1.4 1.5 1.6 1.9 2.0***
  - ***生成物:0.0 0.06 0.17 0.19 0.21 0.25 0.26 0.27 0.29 0.29 0.30 0.31 0.31***

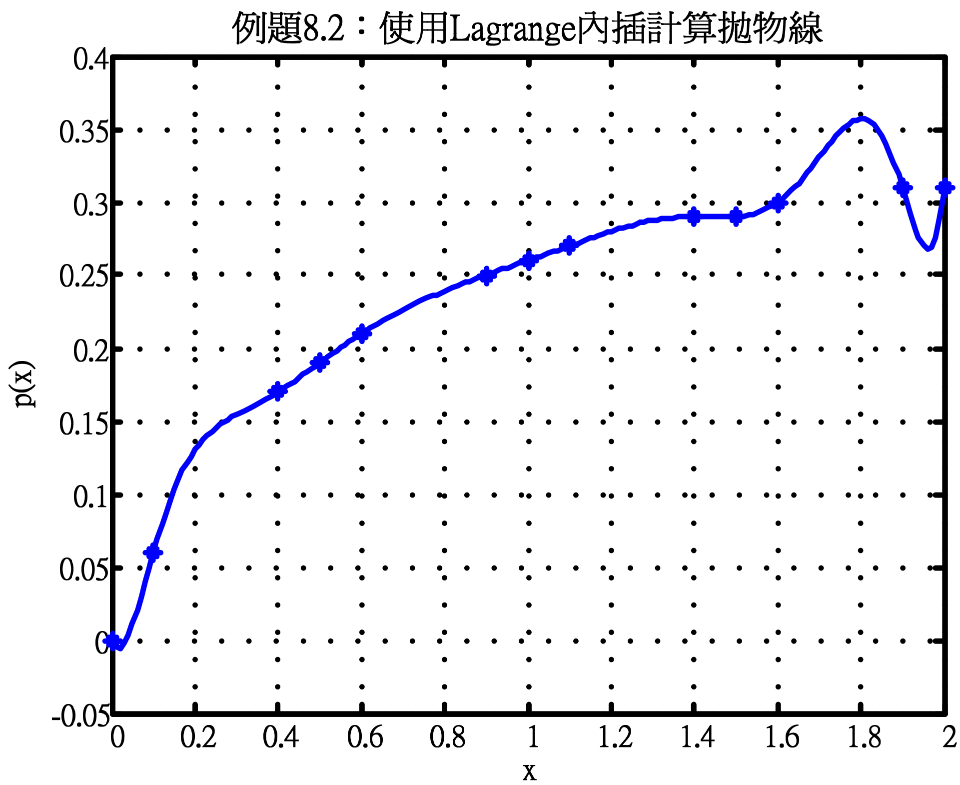
例題8.2Matlab程式

- 主程式：Main_Lagrange_Ex_82_2.m
- Lagrange內插係數程式：LagrangeCoef.m
- Lagrange內插點程式：LagrangeEval.m

```
clear all;
close all;

x=[0.0   0.1    0.4   0.5    0.6   0.9    1.0    1.1    1.4     1.5    1.6    1.9
2.0];
y=[0.0   0.06   0.17  0.19  0.21  0.25  0.26  0.27  0.29  0.29  0.30
0.31  0.31];
t=0:0.01:2; c = LagrangeCoef(x, y);
p = LagrangeEval(t, x, c);
plot(t,p,'-b',x,y,'*');
xlabel('x');
ylabel('p(x)');
title('例題8.2：使用Lagrange內插計算拋物線');
grid on;
```


例題8.2Matlab程式

- 結果：



Lagrange內插討論

- 在整個問題中，我們首先考慮如何寫出通過(x_1,y_1)和(x_2,y_2)兩點之直線方程式。很容易看出

$$p(x)=\frac{(x-x_2)}{(x_1-x_2)}y_1+\frac{(x-x_1)}{(x_2-x_1)}y_2$$

- 具有以下符合期望的性質:
  - *它是直線方程式。*
  - *若$x=x_1$，則y_1的係數為1，且y_2的係數為0，所以$y=y_1$。*
  - *若$x=x_2$，則y_1的係數為0，且y_2的係數為1，所以$y=y_2$。*

Lagrange內插討論

- 以幾何的觀點來說，這相當於決定兩條直線，其中一條 (叫做L_1)在x_1是1，在x_2為0;另一條線 (叫做L_2)在x_2是1，且在x_1為0。這些線取決於數據點的橫座標值。最終的內插多項式，是這兩條線的線性組合。因此，有

$$L_1: y=\frac{x-x_2}{x_1-x_2}$$

$$L_2: y=\frac{x-x_1}{x_2-x_1}$$

$$L(x)=L_1y_1+L_2y_2$$

Lagrange內插討論

對於 $x_1=0$ 及 $x_2=1$ 的基底直線，即 L_1：$y=-x+1$ 和 L_2：$y=x$ ，顯示於圖 8.6。

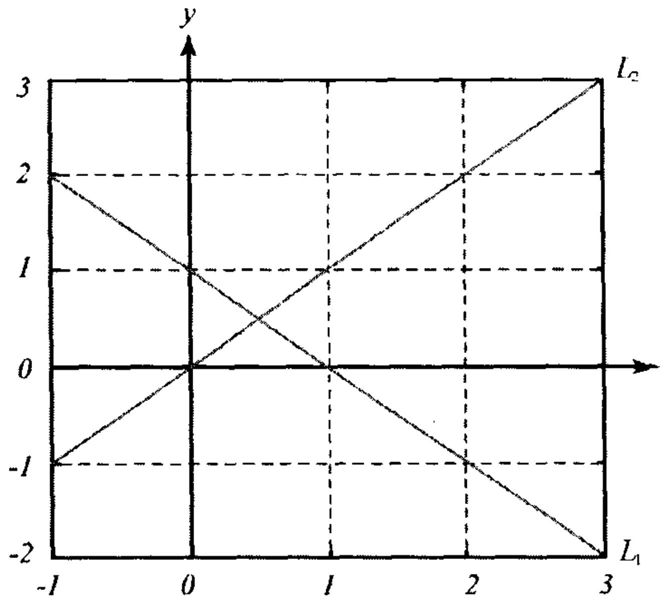


圖 8.6　線性內插多項式的 Lagrange 基底函數

Lagrange內插討論

現在我們用同樣的觀念來寫出多項式 $p(x)$，通過(x_1,y_1), (x_2,y_2)和(x_3,y_3)三點

$$p(x)=\frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)}y_1+\frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)}y_2+\frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)}y_3$$

很容易看出，$p(x)$具有以下理想的性質：

它是一條拋物線方程式。

若 $x=x_1$，則 $y=y_1$；

若 $x=x_2$，則 $y=y_2$；

若 $x=x_3$，則 $y=y_3$。

Lagrange內插討論

$x_1=0$、$x_2=1$ 及 $x_3=2$ 的基底多項式顯示於圖 8.7；分別爲

$P_1=(x-1)(x-2)/2$、$P_2=x(2-x)$及 $P_3=x(x-1)/2$。

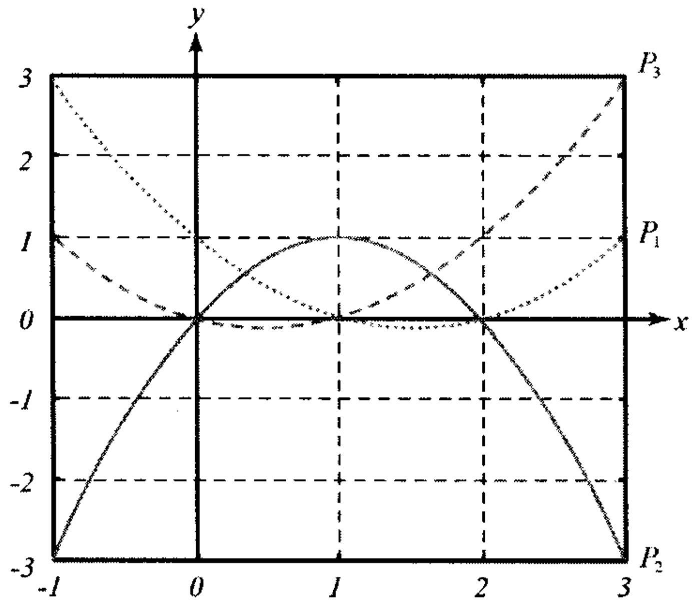


圖 8.7　二次內插多項式的 Lagrange 基底多項式

Lagrange內插討論

- 通常，有多少筆數據，等號右側就有多少項;第k項的係數是一個分數，其分子為$(x-x_1)\ldots(x-x_{k-1})(x-x_{k+1})\ldots(x-x_n)$，其分母和分子一樣，只是將x換成$x_k$。當$x=x_k$時此分數為1，當x等於任何其它給定的自變數值的時候，此分數為0。

Lagrange內插討論

- 優點與缺點
  - *當同樣的節點(自變數x的值)會用於不同的算例中時(只有對應之y值改變)，此種情形特別適合Lagrange形式的多項式內插。對某些問題，如果須要加入新的數據點，或不確定應使用幾次多項式(也就是無須用上所有數據時)，則此種形式不如牛頓形式方便。*

Lagrange內插討論

- 誤差界限
  - *令工為包含$x_1,\ldots,x_n$及t的最小區間，並假設f(x)有n階連續導數。則對某個 $\eta\in I$ ，利用在$x_1,\ldots,x_n$內插f(x)的多項式來計算f(x)在x=t的值時，其誤差為*

$$\frac{(t-x_1)\cdots(t-x_n)}{n!}f^{(n)}(\eta)$$

牛頓形式

- 現在考慮一種稱為牛頓形式的方法，它便於由低次多項式擴充為高次多項式。
- 通過(x_1,y_1)和(x_2,y_2)兩點之直線的牛頓形式方程式為

$$p(x) = a_1 + a_2(x - x_1)$$

- 通過(x_1,y_1)、(x_2,y_2)和(x_3,y_3)三點之拋物線的牛頓形式方程式為

$$p(x) = a_1 + a_2(x - x_1) + a_3(x - x_1)(x - x_2)$$

牛頓形式

- 通過$(x_1,y_1),\ldots(x_n,y_n)$等n點之多項式的牛頓形式通式為

$$p(x) = a_1 + a_2(x - x_1) + a_3(x - x_1)(x - x_2) + \cdots + a_n(x - x_1)\cdots(x - x_{n-1})$$

- 為說明求係數的方法，考慮求通過(x_1,y_1)、(x_2,y_2)和(x_3,y_3)三點之拋物線的係數a_1、a_2和a_3的問題。

牛頓形式

將 (x_1,y_1) 代入 $y=a_1+a_2(x-x_1)+a_3(x-x_1)(x-x_2)$ 得

$$a_1=y_1$$

將 (x_2,y_2) 代入 $y=a_1+a_2(x-x_1)+a_3(x-x_1)(x-x_2)$ 得 $y_2=a_1+a_2(x_2-x_1)$ ，或

$$a_2=\frac{y_2-y_1}{x_2-x_1}$$

將 (x_3,y_3) 代入 $y=a_1+a_2(x-x_1)+a_3(x-x_1)(x-x_2)$ 得 $y_3=a_1+a_2(x_3-x_1)$ $+a_3(x_3-x_1)(x_3-x_2)$ ，或 (經過一些代數運算)

$$a_3=\frac{\frac{y_3-y_2}{x_3-x_2}-\frac{y_2-y_1}{x_2-x_1}}{x_3-x_1}$$

牛頓形式

可以用系統的方式進行此計算，如例題 8.3 所示。此種表格化的方法通常叫做「除差法 (divided differences) 」。

在後續的討論中，各種會用到的量都用函數值 y_i 和定義如下的差分式 $Dy_i, D^2y_i, \cdots$ 來表示：

$$Dy_i=\frac{y_{i+1}-y_i}{x_{i+1}-x_i}$$

$$D^2y_i=\frac{Dy_{i+1}-Dy_i}{x_{i+2}-x_i}$$

$$D^3y_i=\frac{D^2y_{i+1}-D^2y_i}{x_{i+3}-x_i}$$

例題8.3 牛頓內插拋物線

- 再一次考慮例題8.1中的數據。我們可以找到一條通過點$(x_1,y_1)=(-2,4)$、$(x_2,y_2)=(0,2)$及$(x_3,y_3)=(2,8)$的拋物線。此方程式的牛頓形式為

$$p(x) = a_1 + a_2(x = (-2)) + a_3(x-(-2))(x-0)$$

其係數為

$$a_1 = y_1 = 4$$

$$a_2 = \frac{y_2 - y_1}{x_2 - x_1} = \frac{2-4}{0-(-2)} = -1$$

$$a_3 = \frac{\dfrac{y_3-y_2}{x_3-x_2} - \dfrac{y_2-y_1}{x_2-x_1}}{x_3 - x_1} = \frac{\dfrac{8-2}{2-0} - \dfrac{2-4}{0-(-2)}}{2-(-2)} = 1$$

因此

$$P(x) = 4-(x+2)+x(x+2) = x^2+x+2$$

與前面所得相同。

例題8.3 牛頓內插拋物線

- 計算過程顯示於「除差表(divided difference table)」，第三及第四行中元素的定義為:

$$Dy_i = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}$$

$$D^2 y_i = \frac{Dy_{i+1} - Dy_i}{x_{i+2} - x_i}$$

x_i	y_i	Dy_i	$D^2 y_i$
-2	4		
		$\frac{y_2-y_1}{x_2-x_1} = \frac{2-4}{0-(-2)} = -1$	
0	2		$\frac{Dy_2 - Dy_1}{x_3 - x_1} = \frac{3-(-1)}{2-(-2)} = 1$
		$\frac{y_3-y_2}{x_3-x_2} = \frac{8-2}{2-0} = 3$	
2	8		

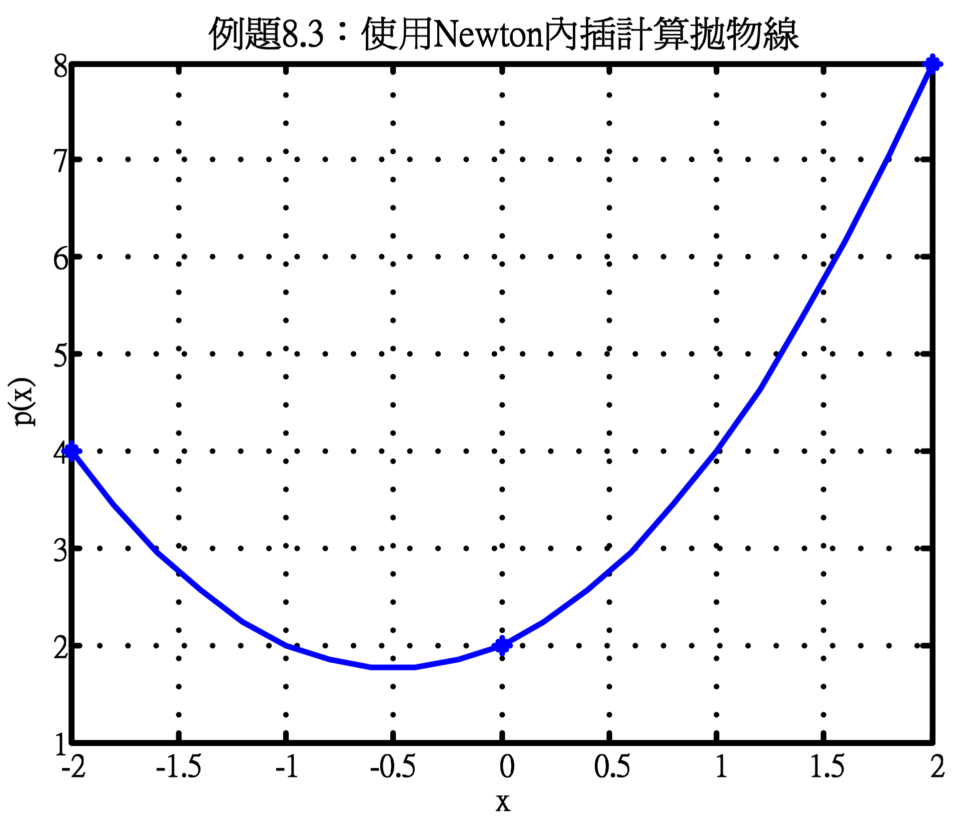
例題8.3 Matlab程式

- 主程式：Main_ Newton _Ex_83.m
- Newton內插係數程式： Newton Coef.m
- Newton內插點程式： Newton Eval.m

```
clear all;
close all;

x=[-2 0 2];
y=[4 2 8];
t=-2:0.2:2;
a = NewtonCoef(x, y);
p = NewtonEval(t, x, a);
plot(t,p,'-b',x,y,'*');
xlabel('x');
ylabel('p(x)');
title('例題8.3：使用Newton內插計算拋物線');
grid on;
```


例題8.3 牛頓內插拋物線

- 結果：



牛頓法討論

- 當每筆數據的x值都是等間隔分佈的，牛頓形式的內插多項式特別好用。在例題8.4中，如果依序取用數據，它就是等間隔分佈。不過在表的下方加入額外的數據點，用以強調牛頓內插的主要優點，也就是，它可以很容易的納入更多的數據點並產生更高階的多項式，而無須重覆求低階多項式所做的計算。這和Lagrange內插剛好成對比，Lagrange內插在產生高階多項式的時候，並不會用到低階函數的計算。

牛頓法討論

- 幾何上，牛頓形式的內插多項式由常數函數開始，它在$x=x_1$處有正確值。下一項為線性函數，它在x_1處為0，並在x_2有所期望的值-亦即此一次項和常數項的和為y_2。通過點(0,1)、(1,3)和(2,6)之牛頓形式二次式的三個項顯示於圖8.8。

$$P_1(x) = 1$$
$$P_2(x) = x$$

和

$$P_3(x) = x(x-1)$$

內插多項式爲

$$N(x) = 1 + 2(x-0) + 0.5x(x-1)$$

牛頓法討論

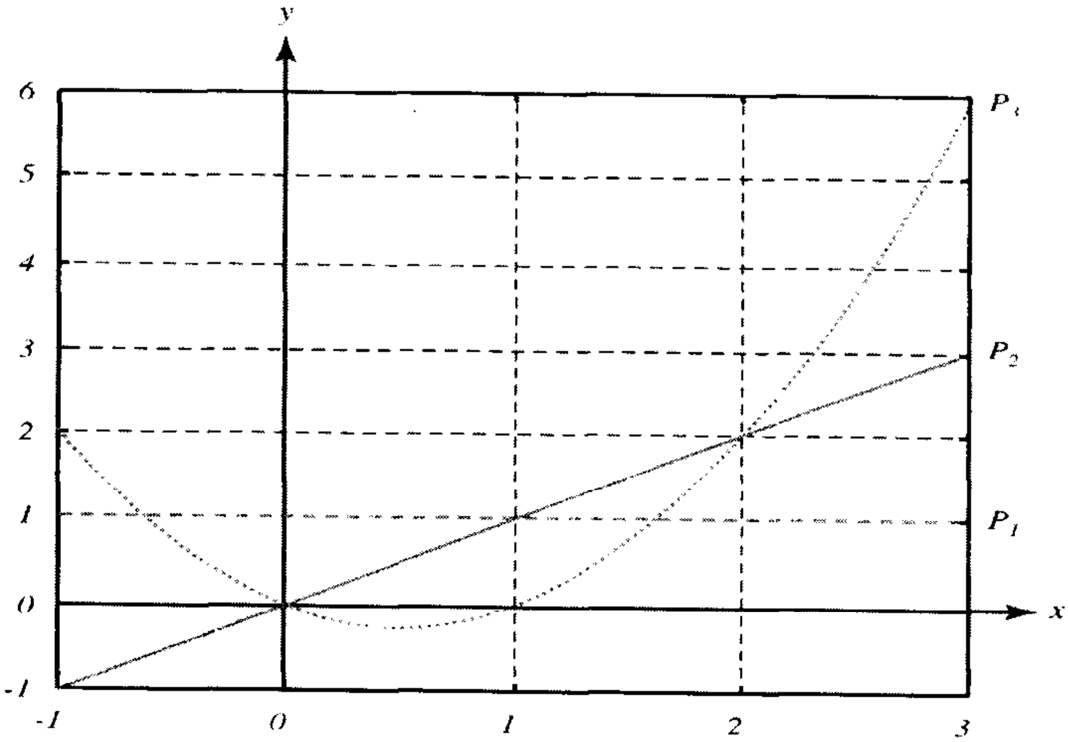


圖 8.8　內插(0,1)、(1,3) 和 (2,6)三點的牛頓多項式

瞭解除差表

- 在對除差表的價值再做深入一些的探討。如同在介紹牛頓形式內差多項式時所指出的，前幾個係數可直接求得。我們在此重覆計算過程，將一個數據點的內插多項式叫做 $\mathbf{N_1(x)}$，兩個數據點的多項式叫做 $\mathbf{N_2(x)}$，並依此類推。

對一個點 (x_1,y_1) 的內插多項式是常數函數

$$N_1(x) = a_1 = y_i$$

通過點 (x_1,y_1) 和 (x_2,y_2) 的內插多項式的形式爲

$$N_2(x) = a_1 + a_2(x - x_1) = N_1(x) + a_1(x - x_1)$$

我們要求 $N_2(x_2) = y_2 = a_1 + a_2(x_2 - x_1)$；解 a_2 可得

$$a_2 = \frac{y_2 - y_1}{x_2 - x_1}$$

瞭解除差表

通過點 (x_1,y_1)、(x_2,y_2) 和 (x_3,y_3) 的內插多項式的形式為

$$N_3(x) = a_1 + a_2(x-x_1) + a_3(x-x_1)(x-x_2) = N_2(x) + a_3(x-x_1)(x-x_2)$$

在此我們要找到一個 a_3 使得

$$N_3(x_3) = y_3 = a_1 + a_2(x_3-x_1) + a_3(x_3-x_1)(x_3-x_2)$$

得到

$$a_3 = \frac{\dfrac{y_3-y_2}{x_3-x_2} - \dfrac{y_2-y_1}{x_2-x_1}}{x_3-x_1}$$

瞭解除差表

與其繼續這種直接計算的方式，我們較希望能找到通用的迭代方法。將內插點 (x_1,y_1)、(x_2,y_2) 和 (x_3,y_3)的多項式寫成

$$N_3(x) = a_1 + a_2(x-x_1) + a_3(x-x_1)(x-x_2) = N_2(x) + a_3(x-x_1)(x-x_2)$$

由此可以看出，$N_3(x)$的組成是將點(x_1,y_1) 和 (x_2,y_2) 的內插多項式 (把它叫做 $N_2(x)$) 加以擴展，再加入點 (x_3,y_3) 的結果。在另一方面，我們也可以假設，我們已有一個對 (x_2,y_2) 和 (x_3,y_3) 兩點的內插多項式 (稱之為 $M_2(x)$)，我們要將其擴展以納入點 (x_1,y_1)。由此觀點，我們可以寫出

$$M_3(x) = b_1 + b_2(x-x_2) + b_3(x-x_2)(x-x_3) = M_2(x) + b_3(x-x_2)(x-x_3)$$

但是，這兩個多項式只是同一個函數的不同表示式。比較兩者，我們看到，x 的最高次方項係數必須相等，所以 $a_3 = b_3$。我們接著再找 a_3 的表示式。設 $N_3(x) = M_3(x)$，可得

$$N_2(x) + a_3(x-x_1)(x-x_2) = M_2(x) + a_3(x-x_2)(x-x_3)$$

瞭解除差表

經過移項重組並提出因式 a_3，得到

$$M_2(x) - N_2(x) = a_3[(x-x_1)(x-x_2)-(x-x_2)(x-x_3)] = a_3(x-x_2)(x_3-x_1)$$

等號左邊 x 的係數是 $b_2 - a_2$；等號右邊 x 的係數是 $a_3(x_3-x_1)$。由 $b_2 - a_2 = a_3(x_3-x_1)$ 解 a_3 得

$$a_3 = \frac{b_2 - a_2}{x_3 - x_1}$$

其中

$$b_2 = \frac{y_3 - y_2}{x_3 - x_2} \qquad a_2 = \frac{y_2 - y_1}{x_2 - x_1}$$

瞭解除差表

以通式來表示，設 $P_{k-1}(x)$ 在 $x_1,\cdots,x_{k-1}$ 內插 $f(x)$，且 $Q_{k-1}(x)$ 在 $x_2,\ldots,x_k$ 內插 $f(x)$。用 p 代表 $P_{k-1}(x)$ 的首項係數；此係數是由點 $(x_1,y_1),\cdots,(x_{k-1},y_{k-1})$ 所構成之第 $(k-1)$ 個除差值。同樣的，令 q 代表 $Q_{k-1}(x)$ 的首項係數；則 q 是由點 $(x_2,y_2),\cdots,(x_k,y_k)$ 所構成之第 $(k-1)$ 個除差值。我們可將 $f(x)$ 在 $x_1,\ldots,x_k$ 處的內插多項式寫成兩種形式：

由 $P_{k-1}(x)$ 再加上點 x_k，此多項式爲

$$N(x) = P_{k-1}(x) + a(x-x_1)\cdots(x-x_{k-1})$$

由 $Q_{k-1}(x)$ 再加上點 x_1，此多項式爲

$$N(x) = Q_{k-1}(x) + a(x-x_2)\cdots(x-x_k)$$

瞭解除差表

和前面一樣，我們要求得的係數 a，在兩個式子中必須一樣，因為它是 x 的最高次方項的係數，且內插多項式是唯一的。令 $N(x)$ 的兩個表示式相等，並移項重組，我們得到

$$Q_{k-1}(x)+a(x-x_2)\cdots(x-x_k)=P_{k-1}(x)+a(x-x_1)\cdots(x-x_{k-1})$$

$$\begin{aligned}Q_{k-1}(x)-P_{k-1}(x)&=a(x-x_1)\cdots(x-x_{k-1})-a(x-x_2)\cdots(x-x_k)\\&=a(x-x_2)\cdots(x-x_{k-1})[(x-x_1)-(x-x_k)]\\&=a(x-x_2)\cdots(x-x_{k-1})(x_k-x_1)\end{aligned}$$

令 x_{k-1} 的係數相等可得 $q-p=a(x_k-x_1)$，所以 $a=(q-p)/(x_k-x_1)$。

瞭解除差表

下表顯示出除差式的標準表示法，但是在實際計算時這並不方便 (在計算的時候，我們可不希望像表中所示的建立多維陣列)。

$x_1 \quad f[x_1]$

$$f[x_1;x_2]=\frac{f[x_2]-f[x_1]}{x_2-x_1}$$

$x_2 \quad f[x_2]$

$$f[x_1;x_2;x_3]=\frac{f[x_2;x_3]-f[x_1;x_2]}{x_3-x_1}$$

$$f[x_2;x_3]=\frac{f[x_3]-f[x_2]}{x_3-x_2}$$

$x_3 \quad f[x_3]$

缺點

- 在許多類型的問題中，通過柑當數量之數據點的內插多項式，效果柑當不好。在例題8.4-8.6，我們展示三個著名的例子。第一個例子是，一個函數在其定義域的某一部份有所變化，但在其餘部份則接近常數。第二個例子就是一條直線。第三個則是Runge所提出的古典例題。

例題8.4 拱起與平直數據

- x=[-2 -1.5 -1 -0.5 0 0.5 1 1.5 2]
- y=[0 0 0 0.87 1 0.87 0 0 0]
- 這些數據顯示出使用高次多項式內插中等數量的數據所會出現的困難，特別是在區間內曲線形狀有明顯的改變(在某部份平直在其餘部份則否)。

例題8.4 拱起與平直數據

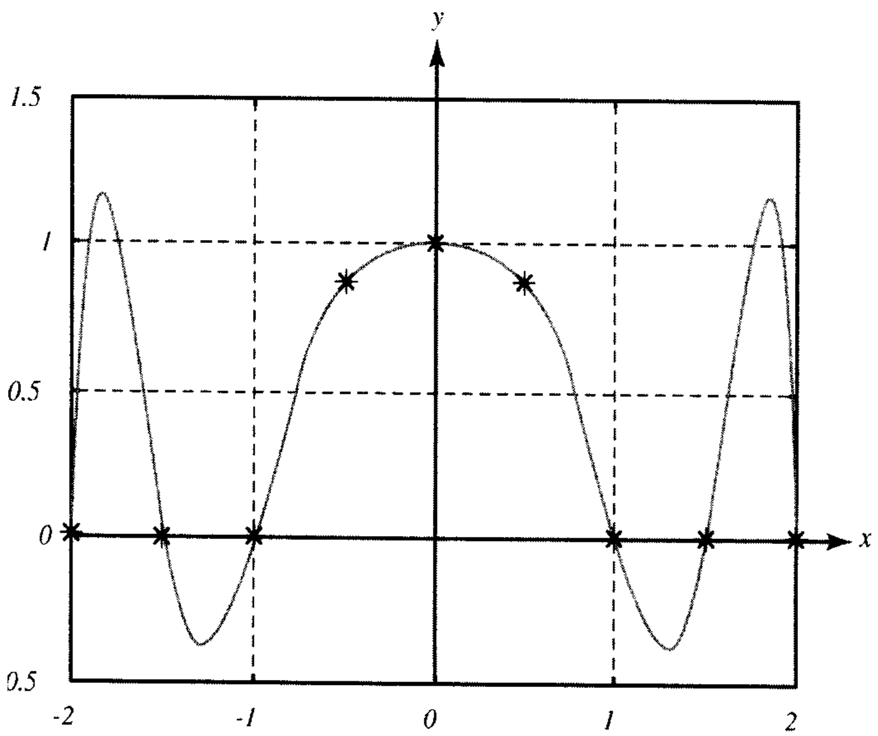


圖 8.9　對多項式內插構成困擾的數據

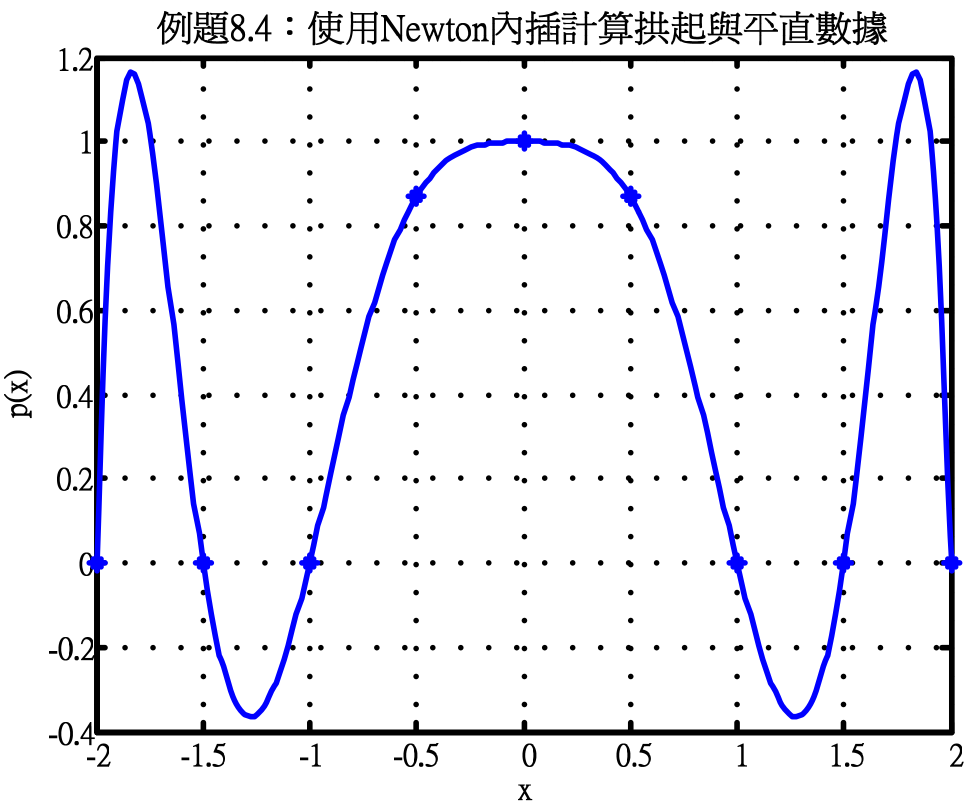
例題8.4Matlab程式

- 主程式：Main_ Newton _Ex_84.m
- Newton內插係數程式： Newton Coef.m
- Newton內插點程式： Newton Eval.m

```
clear all;
close all;

x=[-2 -1.5 -1 -0.5 0 0.5 1 1.5 2];
y=[0 0 0 0.87 1 0.87 0 0 0];
t=-2:0.02:2;
a = NewtonCoef(x, y);
p = NewtonEval(t, x, a);
plot(t,p,'-b',x,y,'*');
xlabel('x');
ylabel('p(x)');
title('例題8.4：使用Newton內插計算拱起與平直數據');
grid on;
```

例題8.4 牛頓內插拱起與平直數據

- 結果：



例題8.5　帶雜訊的直線

- 多項式內插不大適用的另一個例子是帶雜訊的直線，其各個y值對應到非等間隔分佈的x值，如以下數據
  - *x = [0.00 0.20 0.80 1.00 1.20 1.90 2.00 2.10 2.95 3.00]*
  - *y = [0.01 0.22 0.76 1.03 1.18 1.94 2.01 2.08 2.90 2.95]*
- 以下除差表是得自MATLAB求牛頓內插的函數;在此未顯示數據點。表中最上方的一列是內插多項式的係數，多項式圖形顯示於圖8.10。

例題8.5　帶雜訊的直線

1.050	−0.188	0.750	−2.344	2.399	−2.407	2.406	−1.333	0.725
0.900	0.562	−2.062	2.215	−2.415	2.646	−1.527	0.843	
1.350	−1.500	1.703	−2.132	2.612	−1.554	0.833		
0.750	0.373	−0.855	1.264	−0.728	0.279			
1.086	−0.482	0.536	−0.154	−0.169				
0.700	0.000	0.265	−0.459					
0.700	0.279	−0.239						
0.965	0.039							
1.000								

例題8.5　帶雜訊的直線

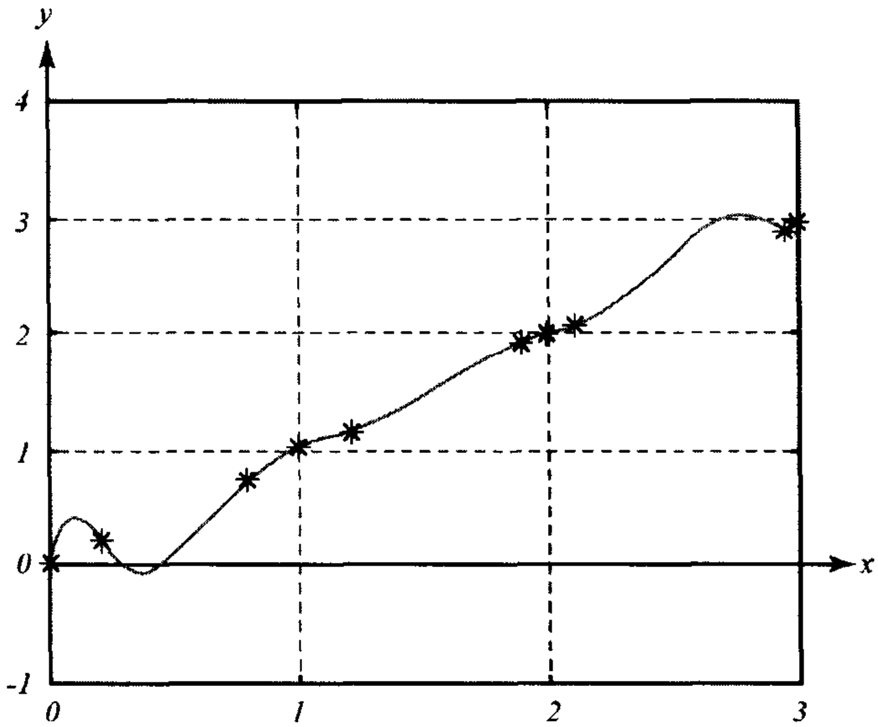


圖 8.10　帶雜訊直線的數據

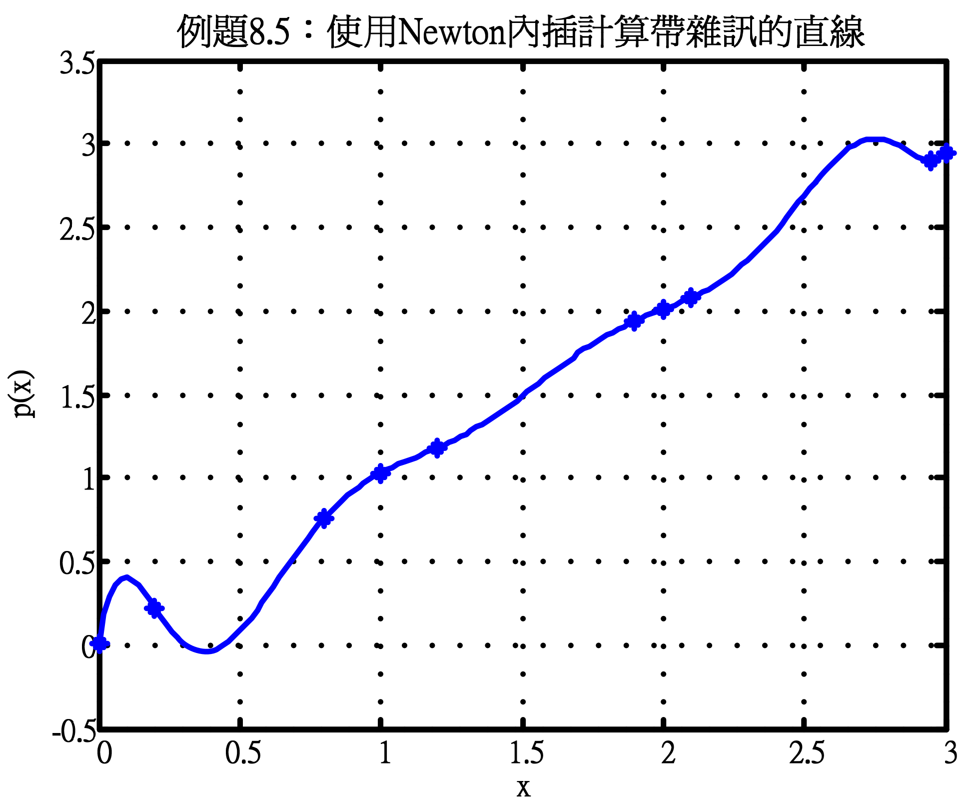
例題8.5 Matlab程式

- 主程式：Main_ Newton _Ex_85.m
- Newton內插係數程式： Newton Coef.m
- Newton內插點程式： Newton Eval.m

```
clear all;
close all;

x = [ 0.00 0.20 0.80 1.00 1.20 1.90 2.00 2.10 2.95
3.00 ];
y = [ 0.01 0.22 0.76 1.03 1.18 1.94 2.01 2.08 2.90
2.95 ];
t=0:0.02:3;
a = NewtonCoef(x, y);
p = NewtonEval(t, x, a);
plot(t,p,'-b',x,y,'*');
xlabel('x');
ylabel('p(x)');
title('例題8.5：使用Newton內插計算帶雜訊的直線');
grid on;
```


例題8.5 帶雜訊的直線

- 結果：



例題8.6 Runge函數

- 函數

$$f(x) = \frac{1}{1+25x^2}$$

- 一個很有名的例子，它說明了**對某些函數，多項式內插無法獲得良好的近似值，而且就算使用更多的函數值(在等間隔的x值)也不會改善情況**。在文獻中此一例子被稱做Runge範例，或是Runge函數。
- 首先，我們用區間 [-1,1] 中等距的五點做內插

例題8.6 Runge函數

- x=[-1.0 -0.5 0.0 0.5 1.0]
- y=[0.0385 0.1379 1.0000 0.1379 0.0385]
- 除差表及原數據點顯示於下：

x	y				
-1	0.0385				
		0.1989			
-0.5	0.1379		1.5252		
		1.7241		-3.3156	
0.0	1.0000		-3.4483		3.3156
		-1.7241		3.3156	
0.5	0.1379		1.5252		
		-0.1989			
1	0.0385				

圖 8.11 中顯示了此內插多項式 (實線) 和原函數 (虛線)。

例題8.6 Runge函數

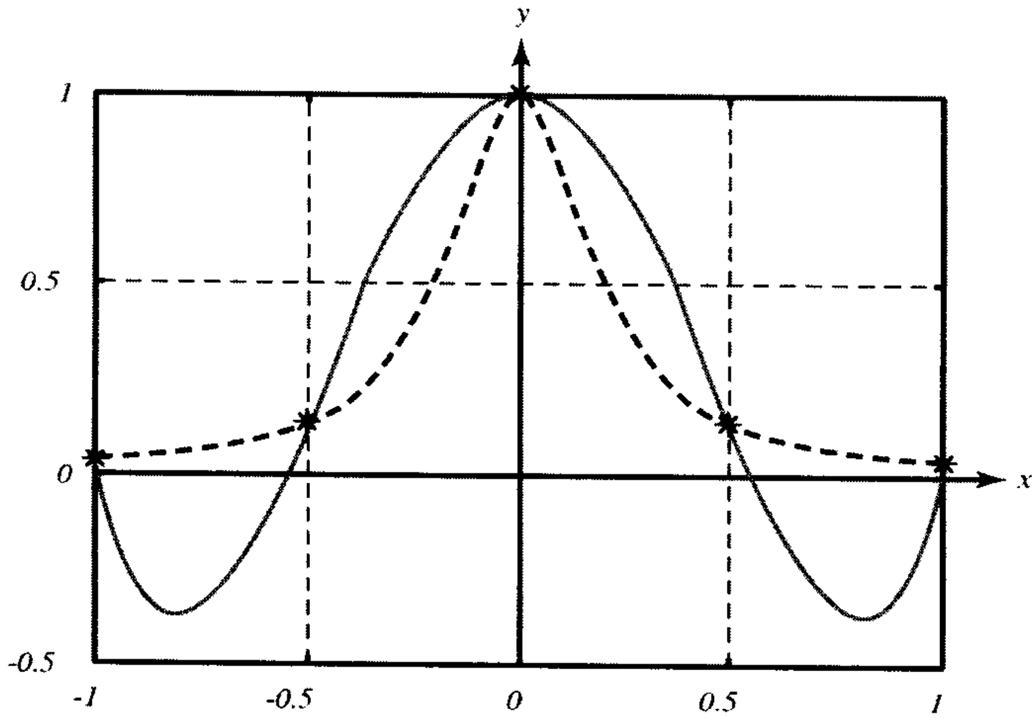


圖 8.11 Runge 函數及內插多項式

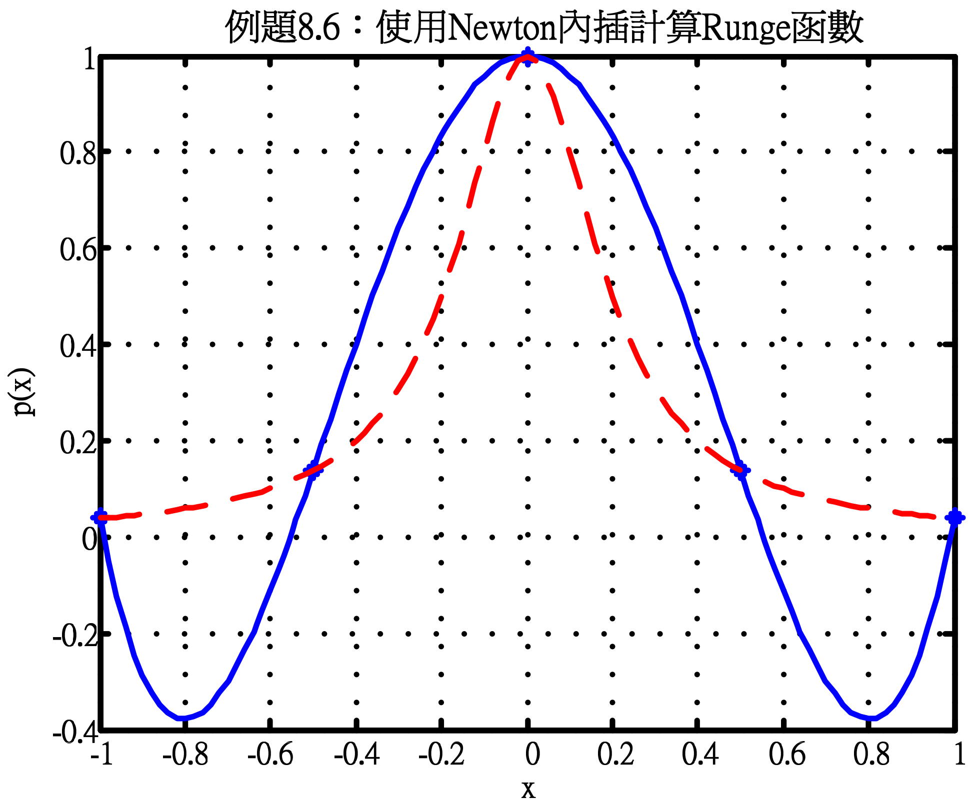
例題8.6 Matlab程式

- 主程式：Main_Newton_Ex_86.m
- Newton內插係數程式： Newton Coef.m
- Newton內插點程式： Newton Eval.m

```
clear all;
close all;

x=[-1.0   -0.5   0.0    0.5    1.0];
y=[0.0385 0.1379 1.0000 0.1379    0.0385];
t=-1:0.02:1;
for i=1:length(t)
   y1(i)=1/(1+25*t(i)^2);
end
a = NewtonCoef(x, y);
p = NewtonEval(t, x, a);
plot(t,p,'-b',x,y,'*',t,y1,'--r');
xlabel('x');
ylabel('p(x)');
title('例題8.6：使用Newton內插計算Runge函數');
grid on;
```

例題8.6 Runge函數

- 結果：



例題8.6 Runge函數

- 如果用九個等距分佈的點來做內插，內插多項式超越原函數的情形，會比只使用五個點時更嚴重。圖8.12顯示以下數據所得的結果
- x=[-1.0 -0.75 -0.5 -0.25 0 0.25 0.5 0.75 1.0];
- y=[0.039 0.066 0.138 0.39 1.0 0.39 0.138 0.066 0.039];

例題8.6 Runge函數

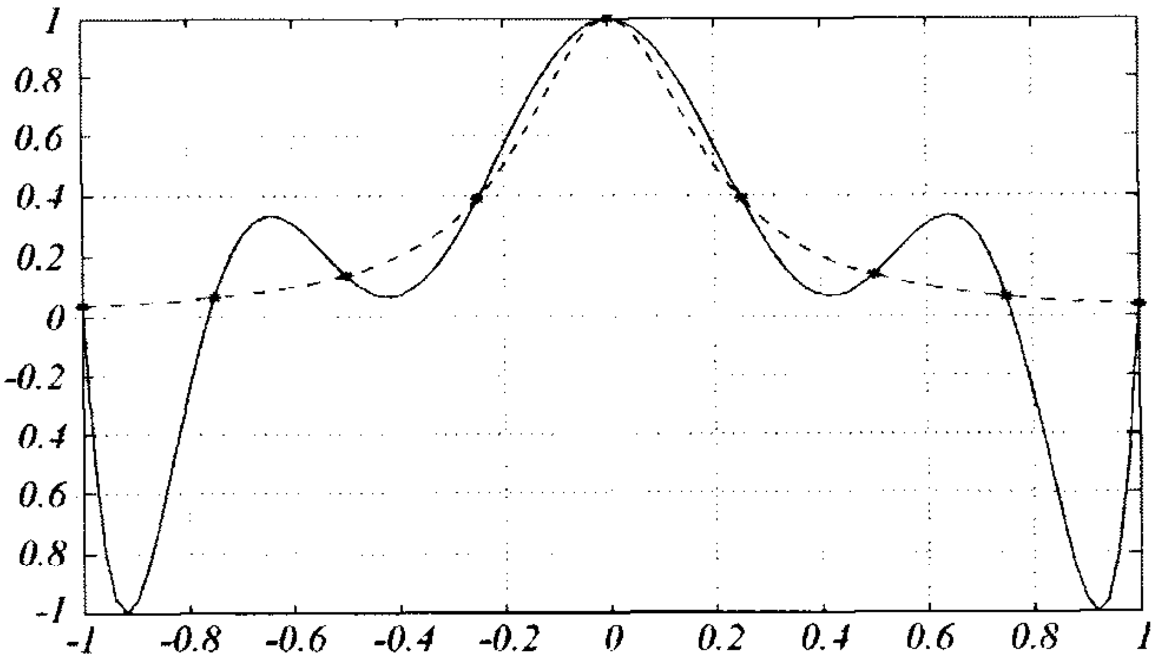


圖 8.12　用更多等間隔數據結果更差！

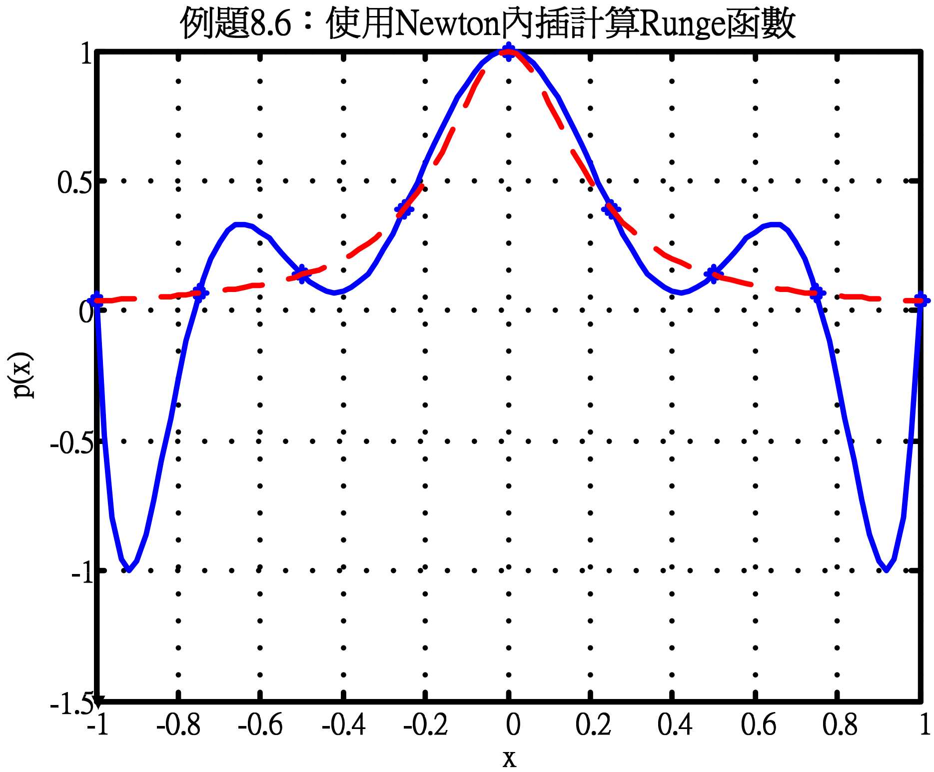
例題8.6 Matlab程式

- 主程式：Main_Newton_Ex_86_2.m
- Newton內插係數程式： Newton Coef.m
- Newton內插點程式： Newton Eval.m

```
clear all;
close all;

x=[-1.0 -0.75 -0.5 -0.25 0 0.25 0.5 0.75 1.0];
y=[0.039 0.066 0.138 0.39 1.0 0.39 0.138 0.066 0.039];
t=-1:0.02:1;
for i=1:length(t)
   y1(i)=1/(1+25*t(i)^2);
end
a = NewtonCoef(x, y);
p = NewtonEval(t, x, a);
plot(t,p,'-b',x,y,'*',t,y1,'--r');
xlabel('x');
ylabel('p(x)');
title('例題8.6：使用Newton內插計算Runge函數');
grid on;
```

例題8.6 Runge函數

- 結果：



例題8.6 Runge函數

- 但在另一方面，如果我們將數據點做較佳的分佈，在區間的兩端用較多點，中央較少，則內插的結果會較好。圖8.13顯示用以下數據所得的結果。如此可證明，在函數值與內插值的最大偏差為最小的條件下，最佳的內插是以Chebyshev多項式之零點做為節點所得的多項式。
- x=[-1.0 -0.9 -0.8 -0.5 0.0 0.5 0.8 0.9 1.0];
- y=[0.039 0.047 0.059 0.138 1.0 0.138 0.059 0.047 0.039];

例題8.6 Runge函數

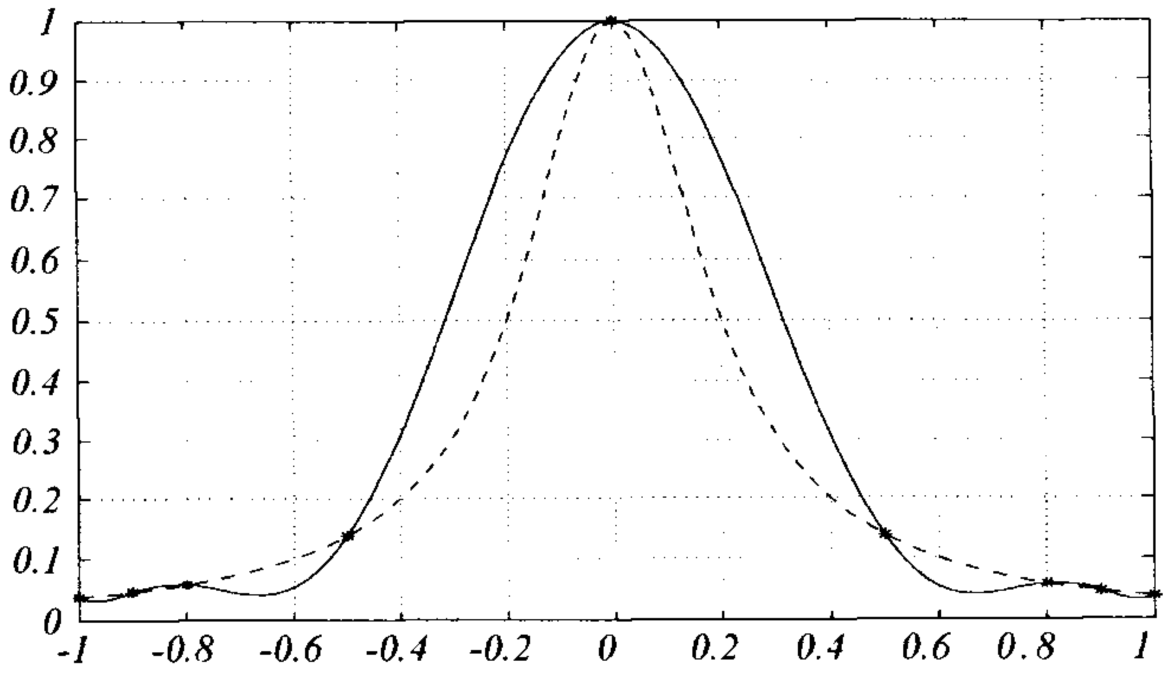


圖 8.13 以較佳分佈之數據內插 Runge 函數

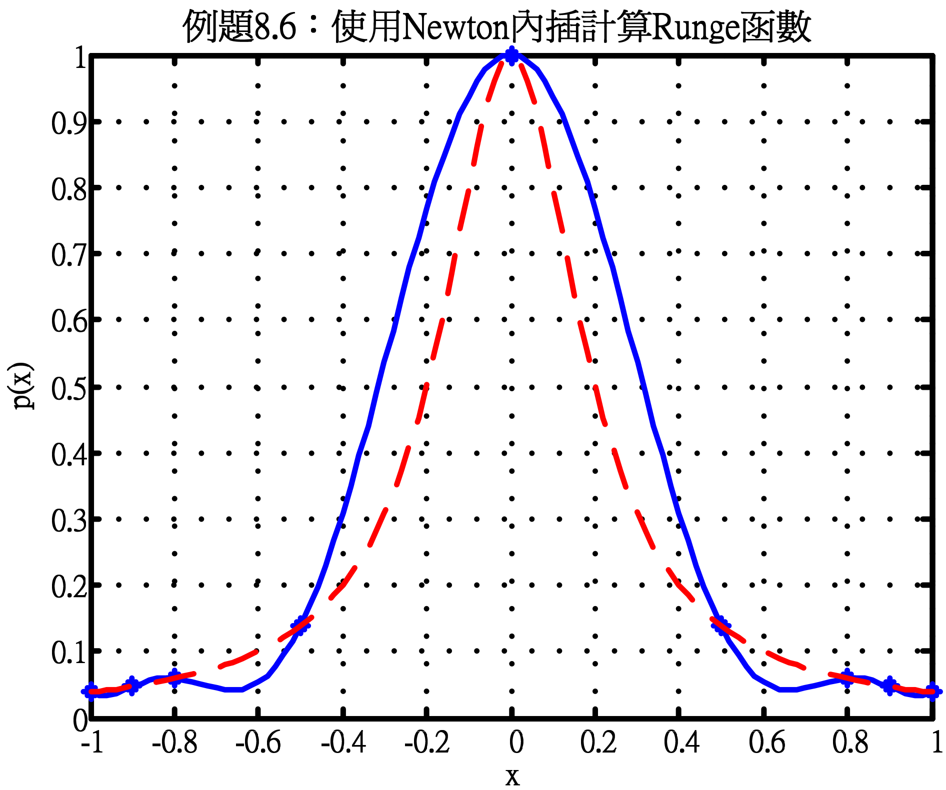
例題8.6 Matlab程式

- 主程式：Main_Newton_Ex_86_3.m
- Newton內插係數程式： Newton Coef.m
- Newton內插點程式： Newton Eval.m

```
clear all;
close all;

x=[-1.0 -0.9 -0.8 -0.5 0.0 0.5 0.8 0.9 1.0];
y=[0.039 0.047 0.059 0.138 1.0 0.138 0.059 0.047 0.039];
t=-1:0.02:1;
for i=1:length(t)
    y1(i)=1/(1+25*t(i)^2);
end
a = NewtonCoef(x, y);
p = NewtonEval(t, x, a);
plot(t,p,'-b',x,y,'*',t,y1,'--r');
xlabel('x');
ylabel('p(x)');
title('例題8.6：使用Newton內插計算Runge函數');
grid on;
```

例題8.6 Runge函數

- 結果：



Hermite內插

- Hermite內插讓能找到一個多項式，在指定的自變數點上，它的函數值和某些導數值都與原函數一致；Taylor多項式、Lagrange及牛頓多項式都是它的特例。在本節中考慮最簡單的Hermite內插，在此每一數據點的函數值與一階導數值為已知。
- 傳統上會使用一輛汽車在不同距離處的速度值做為範例數據，但在此使用前面兩個例題中的數據，以估計各數據點處的導數值。

Hermite內插

- 假設在數個不同自變數值的地方，有代表函數值與其一階導數的量測數據。
- 計算效率最好的Hermite內插是利用牛頓除差表。但是要內插多項式吻合每個點兩次。對兩個數據點可得到三次多項式。除差表的格式和牛頓法所用的一樣，只是每一數據點都輸入兩次，在除差式分母為零的點，亦即在兩重覆點之間，填入一階導數值:

Hermite內插

z_i	w_i	Dw_i	D^2w_i	D^3w_i
$z_1 = x_1$	$w_1 = y_1$			
		y'_1		
$z_2 = x_1$	$w_2 = y_1$		$\dfrac{Dw_2 - Dw_1}{z_3 - z_1}$	
		$\dfrac{w_3 - w_2}{z_3 - z_2}$		$\dfrac{D^2w_2 - D^2w_1}{z_4 - z_1}$
$z_3 = x_2$	$w_3 = y_2$		$\dfrac{Dw_3 - Dw_2}{z_4 - z_2}$	
		y'_2		
$z_4 = x_2$	$w_4 = y_2$			

Hermite內插

內插三次多項式爲

$$H(x)=a_1+a_2(x-x_1)+a_3(x-x_1)(x-x_1)+a_4(x-x_1)(x-x_1)(x-x_2)$$

和牛頓形式一樣，式中各係數可由表中獲得，即

$$a_1=y_1,\quad a_2=Dw_1=y'_1,\quad a_3=D^2w_1,\quad a_4=D^3w_1$$

使用

$$H'(x)=a_2+2a_3(x-x_1)+a_4[(x-x_1)^2+2(x-x_1)(x-x_2)]$$

很明顯可以看出來，在 x_1 處的導數值和給定的一樣；要驗證在 x_2 的結果，我們看到

$$\begin{aligned}H'(x_2)&=y'_1+2\frac{Dw_2-Dw_1}{x_2-x_1}(x_2-x_1)+\frac{D^2w_2-D^2w_1}{x_2-x_1}(x_2-x_1)^2\\&=y'_1+2(Dw_2-Dw_1)+(D^2w_2-D^2w_1)(x_2-x_1)\\&=y'_1+2\frac{y_2-y_1}{x_2-x_1}-2y'_1+Dw_3-2Dw_2+Dw_1=y'_2\end{aligned}$$

Hermite內插

- 為顯示將節點移近所造成的影響，我們考慮 y=sin(πx)的多項式內插，使用節點x=0、1/2、1、3/2(如圖8.14)及節點x=0、1/12、17/12、3/2(如圖8.15)。將節點移近，在區間的兩端得到的斜率較準確 (但在區間中段吻合度就變差)。這對於在呈現三次Hermite內插時使用重覆數據點，提供了一個圖形的動機。

Hermite內插

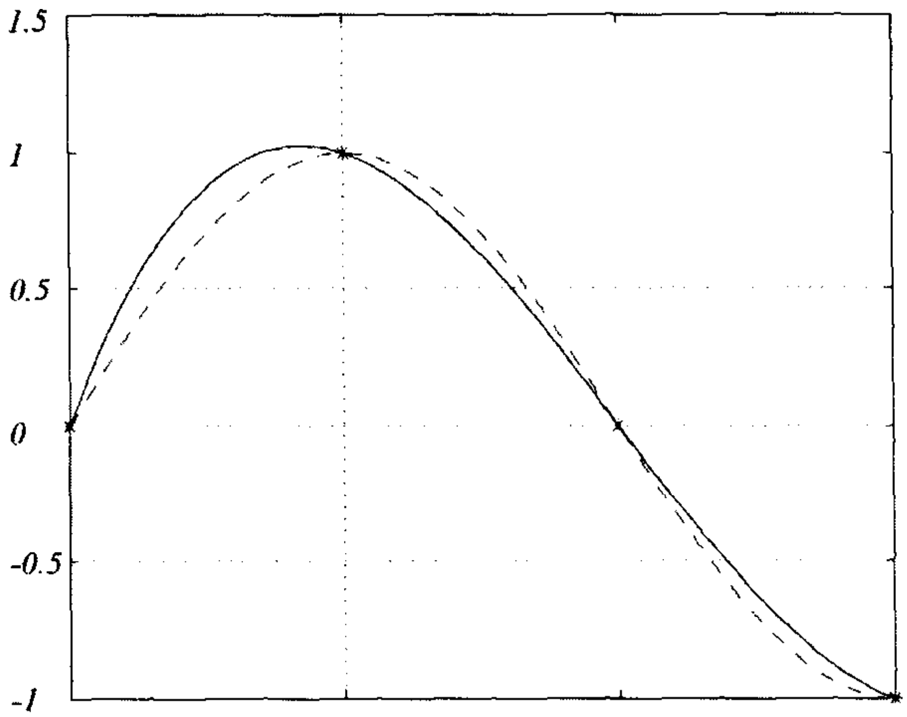


圖 8.14　多項式內插

Hermite內插

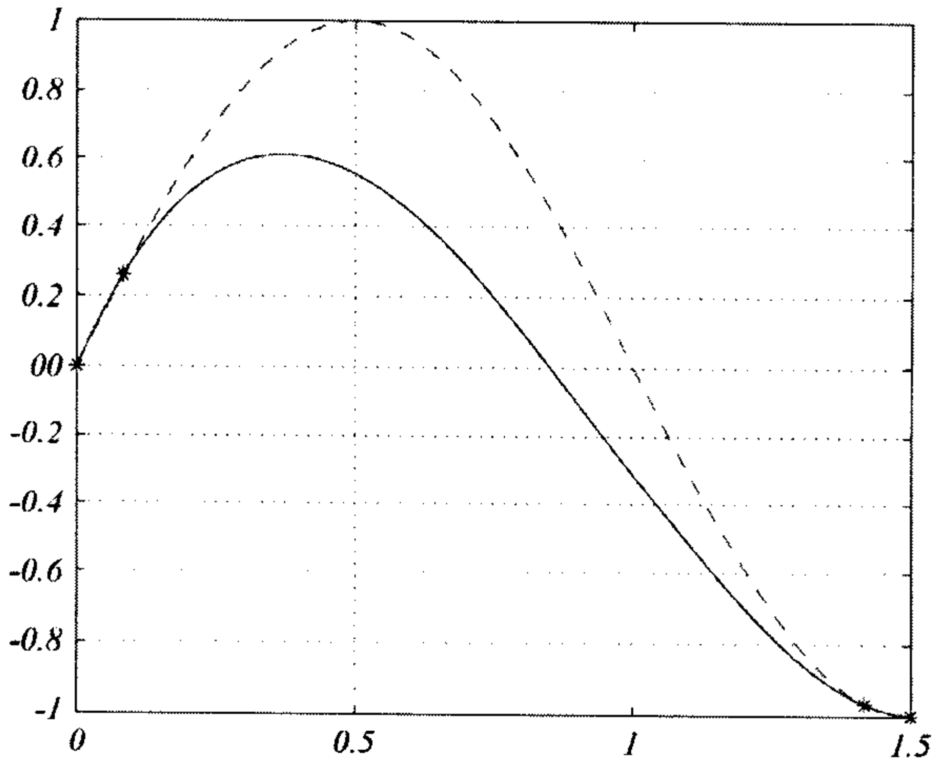


圖 8.15　接進 Hermite 內插

Hermite內插的基底函數

- Hermite內插的一個重要用途，就是下一節要介紹的分段內插。基本觀念是在每一個子區間建構符合理想的內插多項式，使得所得的分段函數不僅內插所有數據點，且在子區間的接點處也是平滑的。
- 利用四個三次基底函數，我們可以構建出三次Hermite函數，使其在0和1處有指定的函數值，且在0和1處有指定的一階導數值。

Hermite內插的基底函數

- 第一個基底函數滿足 $y_a(0)=0$、$y_a(1)=0$：

$$y_a(t) = -0.5t^3 + t^2 - 0.5t$$

- 第二個基底函數滿足 $y_b(0)=1$、$y_b(1)=0$：

$$y_b(t) = 1.5t^3 - 2.5t^2 + 1.0$$

- 第三個基底函數滿足 $y_c(0)=0$、$y_c(1)=1$：

$$y_c(t) = -1.5t^3 + 2.0t^2 + 0.5t$$

- 第四個基底函數滿足 $y_d(0)=0$、$y_d(1)=0$：

$$y_d(t) = 0.5t^3 - 0.5t^2$$

函數

$$y(t) = f_a y_a + f_b y_b + f_c y_c + f_d y_d$$

Hermite內插的基底函數

- 具有以下性質：$y(0)=f_b$、$y(1)=f_c$ 、$y'(0)=s_a=(f_c-f_a)/2$、$y'(1)=s_b=(f_d-f_b)/2$ 。因此，如果已知t=0和t=1的斜率，可以解f_a和f_d的值。或者，可將此公式用於一系列的區間，以獲得分段三次內插多項式，並且其一階導數為連續。

三次Hermite基底函數演算法

```
% 函數定義於 -1 < t < 2，用以內插 0 < t < 1。
   t = -1:0.1:2;
   ya = -0.5*t.^3 + t.^2 -0.5*t;        % 第一基底函數
   yb = 1.5*t.^3 -2.5* t.^2 +1.0;       % 第二基底函數
   yc = -1.5*t.^3 + 2.0*t.^2 +0.5*t;    % 第三基底函數
   yd = 0.5*t.^3 -0.5*t.^2;             % 第四基底函數
% 繪出基底函數
   figure(1),  plot(t,ya,t,yb,t,yc,t,yd),   grid on
% 給出在t=0和t=1的函數值fb和fc。
   fb = 3;   fc = 2;    fa = 0;   fd = 4;
   yy = fa*ya + fb*yb + fc*yc + fd*yd;
   figure(2), plot(t, yy, [-1, 0, 1, 2], [fa, fb, fc, fd],'*')
   grid on
```

Hermite內插的基底函數

- 在[0,1]上用於三次Hermite內插多項式的四個基底函數顯示於圖8.16。通過點(0,3)和(1,2)的三次Hermite內插多項式顯示於圖8.17。圖中同時顯示了點 (0,3)處的切線，及通過$(-1,f_a)=(-1,0)$和$(1,f_c)=(1,2)$兩點的割線。用來定義此區間端點斜率的點，同樣也用來定義區間(-1,0)的右側端點斜率，以確保一階導數連續。

Hermite內插的基底函數

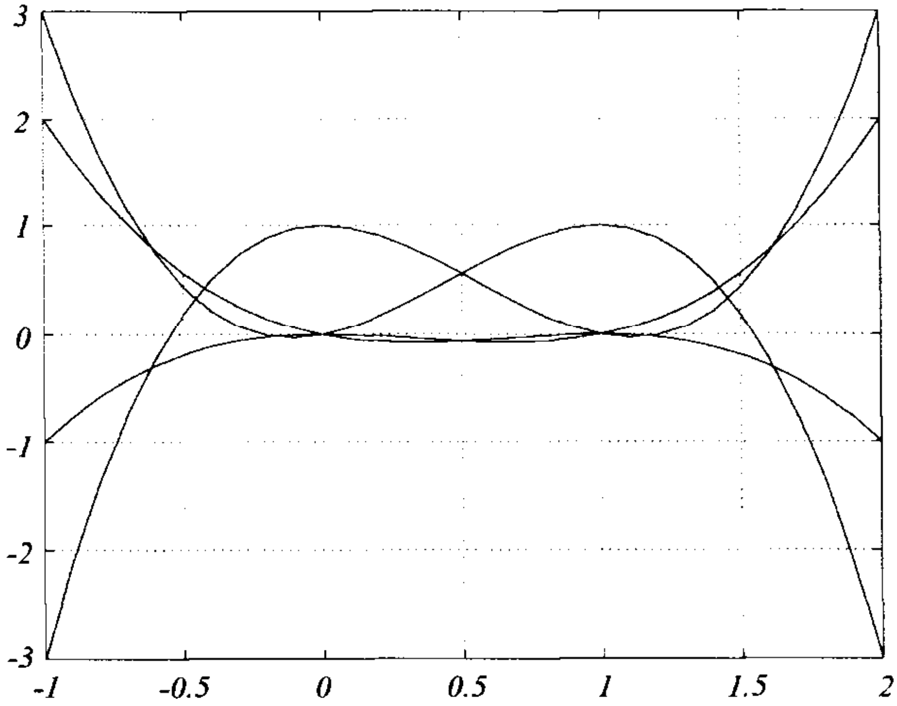


圖 8.16　Hermite 內插的基底函數

Hermite內插的基底函數

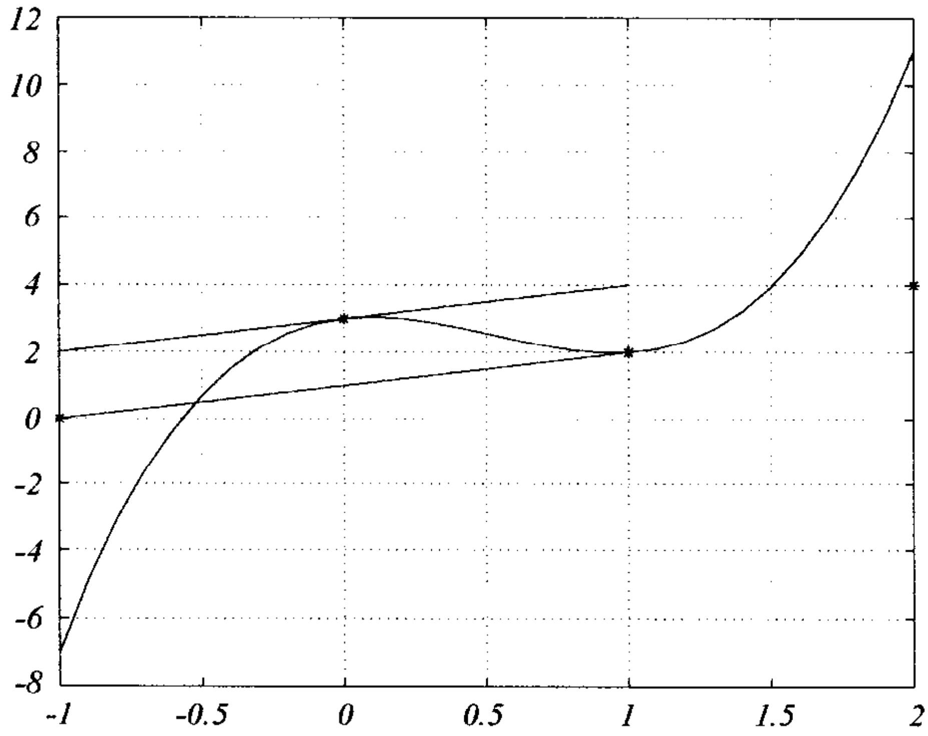


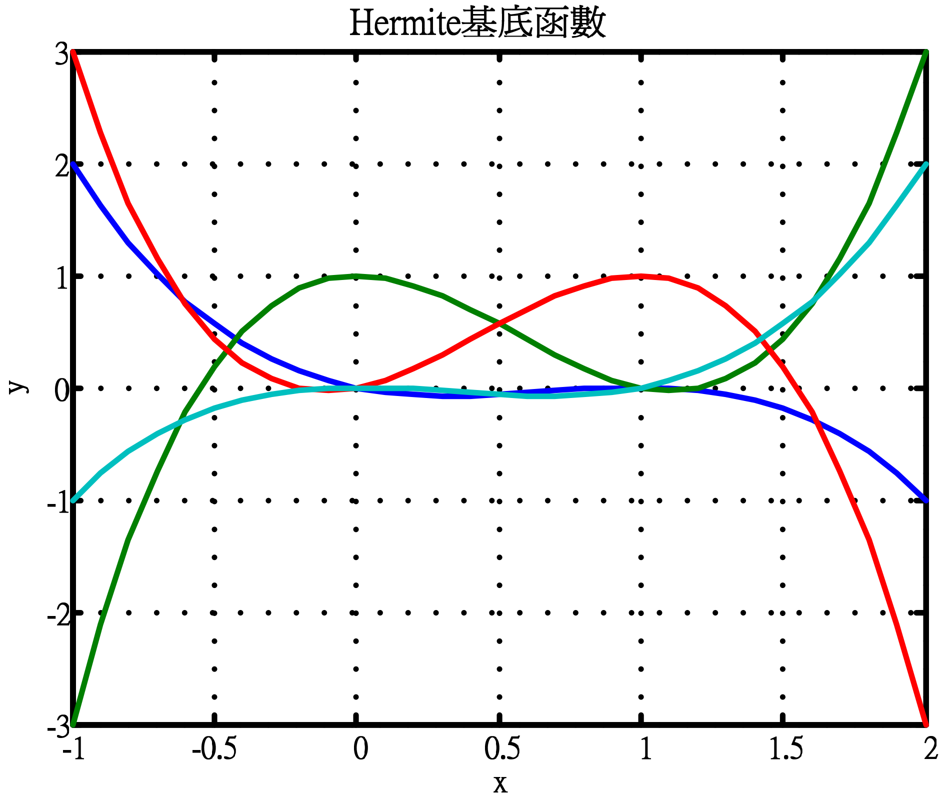
圖 8.17　$y(0)=3$、$y(1)=2$ 的 Hermite 內插

Matlab程式

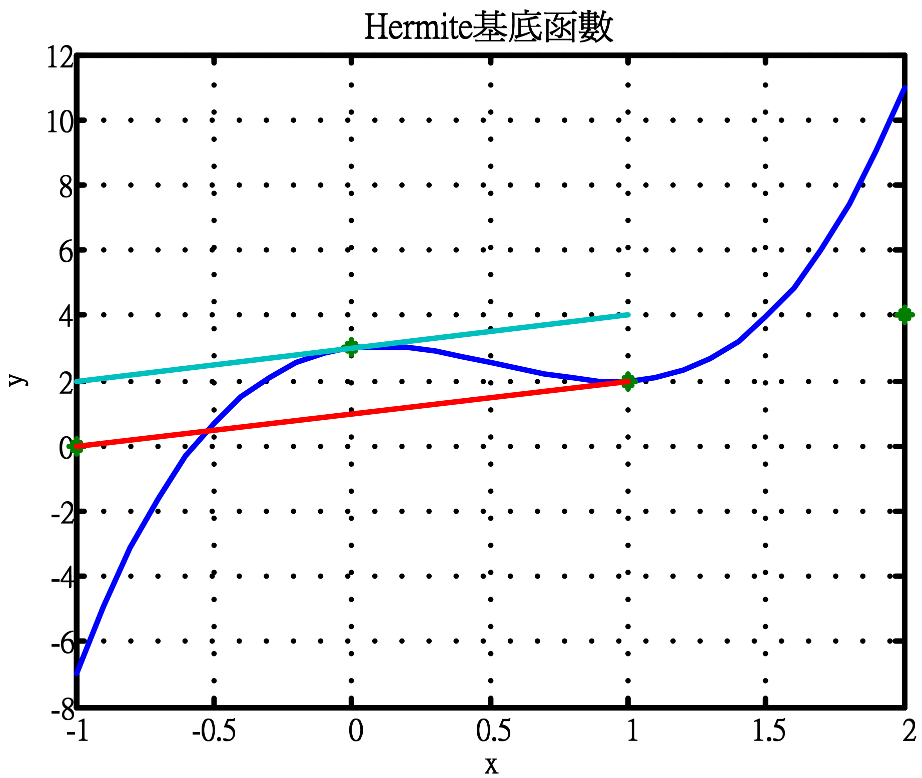
- 主程式：P8_24.m

```
clear all;
close all;

t = -1:0.1:2;
ya = -0.5*t.^3 + t.^2 -0.5*t;
yb = 1.5*t.^3 -2.5* t.^2 +1.0;
yc = -1.5*t.^3 + 2.0*t.^2 +0.5*t;
yd = 0.5*t.^3 -0.5*t.^2;
figure(1), plot(t,ya,t,yb,t,yc,t,yd);
xlabel('x');
ylabel('y');
title('Hermite基底函數');
grid on;
fb =3;   fc = 2;    fa = 0;   fd = 4;
yy = fa*ya + fb*yb + fc*yc + fd*yd;
figure(2), plot(t, yy, [-1, 0, 1, 2],[fa,fb,fc,fd],'*',[-1, 1],[fa,fc],'-',[-1,
1],[2,4],'-');
xlabel('x');
ylabel('y');
title('Hermite基底函數');
grid on;
```

Hermite內插的基底函數



Hermite內插的基底函數



例題8.6　Runge函數

- 函數 $$f(x) = \frac{1}{1+25x^2}$$
- x=[-1.0　　-0.5　　0.0　　0.5　　1.0]
- y=[0.0385 0.1379 1.0000 0.1379　0.0385]

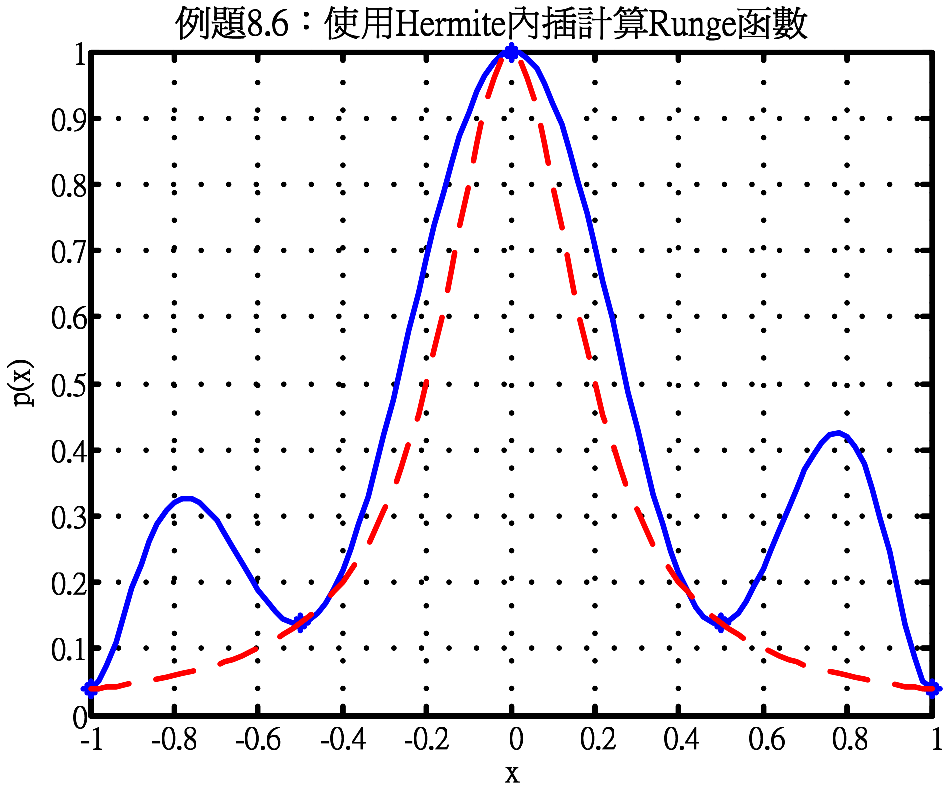
例題8.6 Matlab程式

- 主程式：Main_Hermite_Ex_86.m
- Hermite內插係數程式： HermiteCoef.m
- Hermite內插點程式： HermiteEval.m

```
clear all;
close all;

x=[-1.0   -0.5   0.0    0.5    1.0];
y=[0.0385 0.1379 1.0000 0.1379    0.0385];
t=-1:0.02:1;
for i=1:length(t)
   y1(i)=1/(1+25*t(i)^2);
   dy(i)=-(1+25*t(i)^2)^-2*50*t(i);
end
a = HermiteCoef(x, y, dy);
p = HermiteEval(t, x, a);
plot(t,p,'-b',x,y,'*',t,y1,'--r');
xlabel('x');
ylabel('p(x)');
title('例題8.6：使用Hermite內插計算Runge函數');
grid on;
```

例題8.6 Runge函數

- 結果：



分段多項式內插

- 用單獨一個多項式(高次)來內插大量的數據點，其缺點已顯示於例題8.6。為避免這些困擾，可以使用分段多項式。傳統上，飛機或船隻的設計都需要用到全尺寸模型。要畫出一條通過幾個指定點的平滑曲線，要用一條很長且有彈性的金屬或木質長條，將它彎曲成所要形狀，並以夾釘固定在所指定的點上。然後描出該曲線，以用於設計中。以此種方式所產生的曲線叫做雲形線 (splines)(用來產生雲形線的細長條叫做雲形規)。雲形線代表最低應變(strain)能量的曲線;它們看起來也很美觀。此種雲形線具有分段三次多項式的形狀。

分段多項式內插

- 在數學上，一條雲形線(次數為m)是一個分段多項式(次數m)，並在每個多項式的接點處具有最大的平滑度。因此，一條線性雲形線是連續的，二次雲形線則有連續的一階導數，並依此類推。分段三次Hermite內插具有連續的一階導數 (二階導數沒有)，因為它有「保形(shape-preserving)」內插的特性，它可避免雲形線內插可能會有的「波折」。MATTLAB的函數pchip即使用此方法。

分段多項式內插

- 在本節中，我們考慮內插一組數據的問題，並求出每個子區間的分段函數。在對分段線性與二次內插做簡短說明後，我們將注意力集中在三次雲形線內插。三次雲形線函數，在其它的數學問題中也可做為表示解的基底函數，包括以共位法(collocation)解常微分方程的邊界值問題。

分段多項式內插

- 在此方法中x的值常被稱做節點(nodes);它們是已知數據 (y值)所在的位置。
- 假設$x_1<\ldots<x_n$。對分段內插，區間$[x_1,x_n]$會劃分為多個子區間;區隔子區間的x值叫做結點(knots)。結點可以是和節點一樣的點，但並不一定需如此。

分段多項式內插

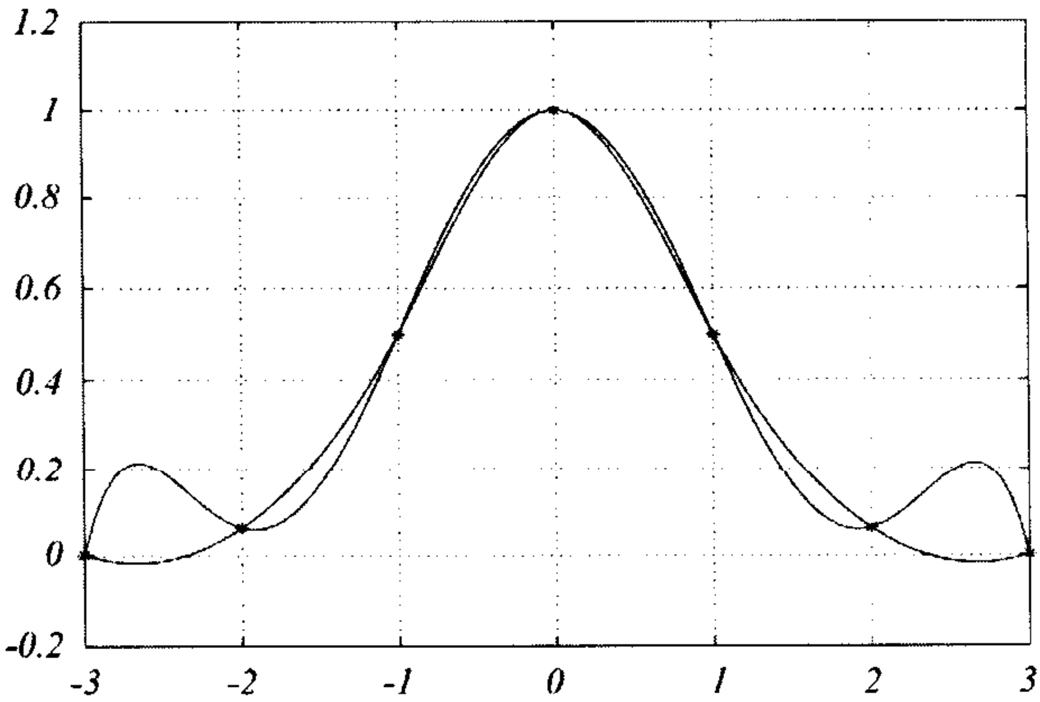


圖 818　多項式與雲形線內插

分段線性內插 (Piecewise Linear Interpolation)

- 為說明最簡單的分段多項式內插，即分段線性內插，考慮一組四個數據點

$$(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$$

- 其中 $x_1<x_2<x_3<x_4$，這些點定義出x軸上的三個子區間

$$I_1 = [x_1, x_2], I_2 = [x_2, x_3], I_3 = [x_3, x_4]$$

- 子區間交界處的「結點」和已知數據的「節點」一樣。

分段線性內插

- 如果在每個子區間用一條直線，可用以下分段線性函數內插這組數據。

$$P(x) = \begin{cases} \dfrac{x - x_2}{x_1 - x_2} y_1 + \dfrac{x - x_1}{x_2 - x_1} y_2, & x_1 \le x \le x_2 \\ \dfrac{x - x_3}{x_2 - x_3} y_2 + \dfrac{x - x_2}{x_3 - x_2} y_3, & x_2 \le x \le x_3 \\ \dfrac{x - x_4}{x_3 - x_4} y_3 + \dfrac{x - x_3}{x_4 - x_3} y_4, & x_3 \le x \le x_4 \end{cases}$$

- 分段線性內插函數，在結點處是連續的，但並不平滑。換句話說，它無法要求結點處的任一階導數為連續。

例題8.7 分段線性內插

- 利用x=[0,1,2,3]和y=[0,1,4,3]，求得分段線性內插函數如圖8.19所示。此函數為連續，但並不平滑。

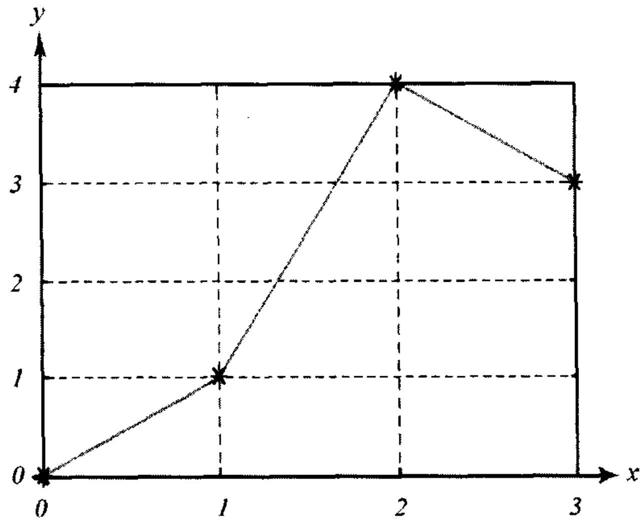


圖 8.19 分段線性內插

分段二次內插 (Piecewise Quadratic Interpolation)

- 想在每一個子區間使用二次函數，並希望能使得節點處的函數值與一階導數都吻合。對n+1筆數據，計有n個區間。同時，每個二次多項式有三個未知數，所以我們共有3n個未知數。再者，每個子區間有兩個方程式，分別對應於此二次函數在兩個端點的指定值。(亦即，在第一個區間中(x_1,y_1)和(x_2,y_2)必須滿足該二次式，餘此類推。)
- 此外，所有的子區間計有n-1個交點，在這些點上，相鄰兩條拋物線的一階導數要連續。這樣就有3n個未知數的2n+n-1個方程式。有幾種可能的方式可定義出這個額外條件。

分段二次內插

- 一個簡單的方法是，定義區間$[x_i,x_{i+1}]$內的二次函數$S_i(x)$之形式為

$$S_i(x)=y_i+z_i(x-x_i)+\frac{z_{i+1}-z_i}{2(x_{i+1}-x_i)}(x-x_i)^2$$

- 其中z_i是函數在x_i處的斜率。

用這種方式定義二次多項式，可自動使得節點處的一階導數爲連續；因爲 $S'_i(x)=z_i+(z_{i+1}-z_i)(x-x_i)/((x_{i+1}-x_i))$，我們有 $S'_i(x_i)=z_i$ 及 $S'_i(x_{i+1})=z_{i+1}$ 。

現在我們要讓函數值在節點處爲連續。特別是在 $x=x_{i+1}$，我們需要

$$S_i(x_{i+1})=y_i+z_i(x_{i+1}-x_i)+\frac{z_{i+1}-z_i}{2(x_{i+1}-x_i)}(x_{i+1}-x_i)^2=y_{i+1}$$

分段二次內插

化簡之後我們得到 $y_i+z_i(x_{i+1}-x_i)+(z_{i+1}-z_i)(x_{i+1}-x_i)/2=y_{i+1}\Rightarrow$

$$z_{i+1}=2\frac{y_{i+1}-y_i}{x_{i+1}-x_i}-z_i$$

因此，如果我們指定 z_1，我們可以循序求得其它的斜率。

爲說明此方法，取數據 $\mathbf{x}=[0,1,2,3]$ 及 $\mathbf{y}=[0,1,4,3]$。設 $z_1=0$。則

$$z_2=2(y_2-y_1)/(x_2-x_1)-z_1=2(1-0)/(1-0)-0\ =2$$
$$z_3=2(y_3-y_2)/(x_3-x_2)-z_2=2(4-1)/(2-1)-2\ =4$$
$$z_4=2(y_4-y_3)/(x_4-x_3)-z_3=2(3-4)/(3-2)-4=-6$$

分段二次內插

所以

$$S_1(x) = 0 + 0(x-0) + \frac{2-0}{2(1-0)}(x-0)^2 = x^2, \qquad 0 \le x \le 1$$

$$S_2(x) = 1 + 2(x-1) + \frac{4-2}{2(2-1)}(x-1)^2 = 1 + 2(x-1) + (x-1)^2, \qquad 1 \le x \le 2$$

$$S_3(x) = 4 + 5(x-2) + \frac{-6-4}{2(3-2)}(x-2)^2 = 4 + 5(x-2) - 5(x-2)^2, \quad 2 \le x \le 3$$

分段二次內插

- 很不幸的，用我們剛討論的「結點，節點」的方法，我們選擇的狗處的斜率(或任何一個節點處)對曲線的整體形狀有很大的影響。一種替代的分段二次內插法，則是用兩個節點的中點做為結點，節點處的函數值為已知。

分段二次內插

- 對於(x_1,y_1)、(x_2,y_2)、(x_3,y_3)及(x_4,y_4)四組數據，我們定義結點$z_1=x_1$, $z_2=(x_1+x_2)/2$, $z_3=(x_2+x_3)/2$, $z_4=(x_3+x_4)/2$, $z_5=x_4$，且相鄰兩點的間隔為$h_1=x_2-x_1$, $h_2=x_3-x_2$, $h_3=x_3-x_2$。則$z_2-x_1=h_1/2$, $z_3-x_2=h_2/2$, $z_4-x_3=h_3/2$, $z_4-x_3=h_3/2$,且$z_2-x_2=-h_1/2$, $z_3-x_3=-h_2/2$, $z_4-x_4=-h_3/2$, 。
- 現在定義

$$P_1(x)=a_1(x-x_1)^2+b_1(x-x_1)+c_1,\quad x\in[z_1,z_2]$$

$$P_2(x)=a_2(x-x_2)^2+b_2(x-x_2)+c_2,\quad x\in[z_2,z_3]$$

$$P_3(x)=a_3(x-x_3)^2+b_3(x-x_3)+c_3,\quad x\in[z_3,z_4]$$

$$P_4(x)=a_4(x-x_4)^2+b_4(x-x_4)+c_4,\quad x\in[z_4,z_5]$$

分段二次內插

- 因為$P_k(x_k)=c_k$，加入內插條件$P_k(x_k)=y_k$得到$c_k=y_k$，其中$k=1,2,3,4$。如果現在將連續條件加在內部節點，可得到以下三個方程式：

$$P_1(z_2)=P_2(z_2):\quad h_1^2a_1-h_1^2a_2+2h_1b_1+2h_1b_2=4(y_2-y_1)$$

$$P_2(z_3)=P_3(z_3):\quad h_2^2a_2-h_2^2a_3+2h_2b_2+2h_2b_3=4(y_3-y_2)$$

$$P_3(z_4)=P_4(z_4):\quad h_3^2a_3-h_3^2a_4+2h_3b_3+2h_3b_4=4(y_4-y_3)$$

- 類似的，如果將一階導數連續的條件加在內部節點，可得到另外三個方程式：

$$P'_1(z_2)=P'_2(z_2):\quad h_1a_1+h_1a_2+b_1-b_2=0$$

$$P'_2(z_3)=P'_3(z_3):\quad h_2a_2+h_2a_3+b_2-b_3=0$$

$$P'_3(z_4)=P'_4(z_4):\quad h_3a_3+h_3a_4+b_3-b_4=0$$

分段二次內插

- 到此階段，有六個方程式和八個未知係數($a_1,a_2,a_3,a_4,b_1,b_2,b_3,b_4$)。因為 $P'_k(x)=2a_k(x-x_k)+b_k$，可加入區間端點 x_1 和 x_4 處之導數值的條件以求得 b_1 和 b_4。設 $P'_1(x_1)=0$ 可得 $b_1=0$，設 $P'_4(x_4)=0$ 可得 $b_4=0$。利用在兩端點處斜率為零的條件，係數的方程式成為

$$
\begin{aligned}
a_1h_1^2 - a_2h_1^2 \qquad\qquad &+ 2b_2h_1 &= 4(y_2 - y_1)\\
+a_2h_2^2 - a_3h_2^2 \qquad &+ 2b_2h_2 + 2b_3h_2 &= 4(y_3 - y_2)\\
+a_3h_3^2 - a_4h_3^2 &+ 2b_3h_3 &= 4(y_4 - y_3)\\
a_1h_1 + a_2h_1 \qquad\qquad &- b_2 &= 0\\
+a_2h_2 + a_3h_2 \qquad &+ b_2 - b_3 &= 0\\
+a_3h_3 + a_4h_3 &+ b_3 &= 0
\end{aligned}
$$

例題8.8 二次雲形線內插

考慮數據點 (0,0)、(1,1)、(2,4) 和 (3,3)。我們可以用第 3 章中介紹的 MATLAB 中高斯消去法的函數，求解係數的線性方程組。矩陣 **A** 及右側項 **r** 為

$$
\mathbf{A} = \begin{bmatrix} 1 & -1 & 0 & 0 & 2 & 0\\ 0 & 1 & -1 & 0 & 2 & 2\\ 0 & 0 & 1 & -1 & 0 & 2\\ 1 & 1 & 0 & 0 & -1 & 0\\ 0 & 1 & 1 & 0 & 1 & -1\\ 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}, \quad \mathbf{r} = \begin{bmatrix} 4\\ 12\\ -4\\ 0\\ 0\\ 0 \end{bmatrix}
$$

解向量即為未知係數 a_1、a_2、a_3、a_4、b_2 和 b_3：

$$
\mathbf{c} = [0.7429 \quad 1.7714 \quad -3.3714 \quad 2.4571 \quad 2.5143 \quad 0.9143]'
$$

例題8.8 二次雲形線內插

此分段內插多項式顯示於圖 8.20 並列出如下

$$P_1(x) = 0.7429(x-0)^2, \qquad x \in [0.0, 0.5]$$

$$P_2(x) = 1.7714(x-1)^2 + 2.5143(x-1) + 1, \qquad x \in [0.5, 1.5]$$

$$P_3(x) = -3.3714(x-2)^2 + 0.9143(x-2) + 4, \qquad x \in [1.5, 2.5]$$

$$P_4(x) = 2.4571(x-3)^2 + 3, \qquad x \in [2.5, 3.0]$$

例題8.8 二次雲形線內插

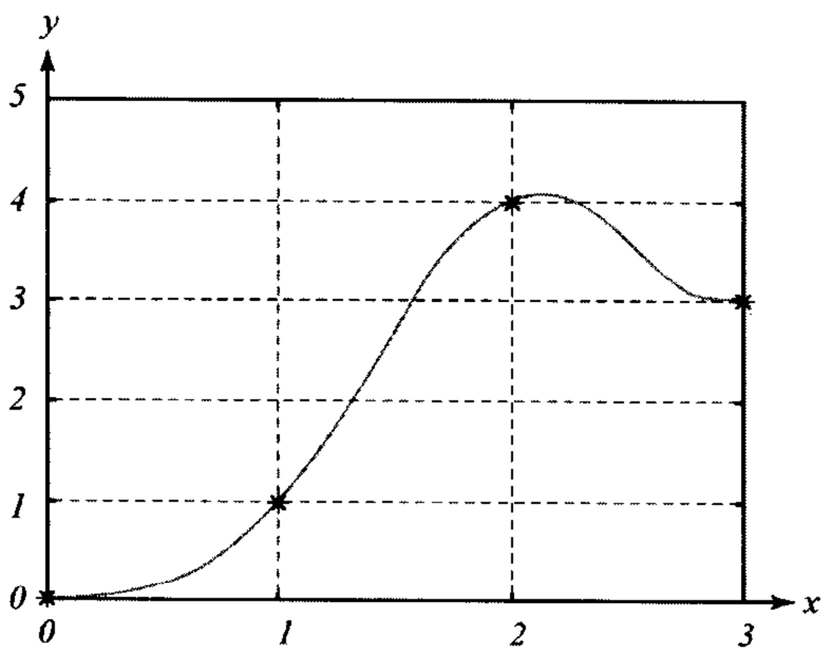


圖 8.20 四筆數據的分段二次內插

分段二次內插

- 此一方法的平衡性遠比「節點=結點」的做法好，但隨著數據點的增加，解線性方程組所須的計算量會變得非常大。係數矩陣並不具有能減少計算量的特別結構(它不是三對角線，也不是帶狀矩陣)。接下來，我們將注意力移到分段三次內插，先介紹三次Hermite內插的使用，然後再介紹三次雲形線內插，它可得到更平滑的結果;附帶的好處則是，它要解的線性方程組是三對角線的。

分段三次 Hermite內插

- Hermite內插的一項重要用途是分段內插;分段Hermite內插可用以保留單調性(monotonicity)。例題8.9說明MATLAB內建函數pchip的用法。上一節所介紹的基底函數也可用來組成分段三次Hermite內插多項式，並且在內部節點上有連續的一階導數 (但不指定節點的導數值)。

例題8.9　分段三次Hermite內插多項式

- 考慮以下數據：
- x=[-3, -2, -1, 0, 1, 2, 3];
- y=[-1, -1.1, -1, 0, 1, 1.1, 1];

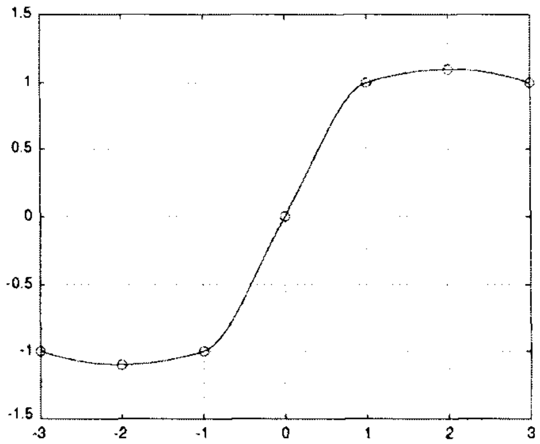


圖 8.21　分段 Hermite 內插

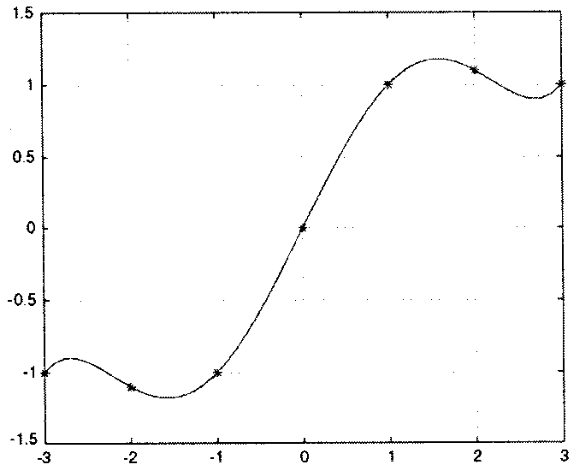


圖 8.22　多項式內插

三次雲形線內插(Cubic Spline Interpolation)

- 如果使用三次雲形線(亦即分段三次多項式)，經由簡單的計算就可瞭解，所有的資訊足夠要求在每一個「結點」處，即子區間的交接處的一階與二階導數均連續。
- 藉由適當的選取每一子區間之三次多項式的代數表示方式，我們可以簡化係數的計算。n個結點石$x_1< x_2< \ldots < x_i< \ldots < x_n$，定義出n-1個子區間
- $I_1=[x_1,x_2], \ldots, I_i=[x_i,x_{i+1}], \ldots, I_{n-1}=[x_{n-1},x_n]$

三次雲形線內插(Cubic Spline Interpolation)

- 因為並未要求工是等距分佈的，令
- $H_i = x_{i+1} - x_i$
- 在$I_i=[x_i,x_{i+1}]$區間

$$P_i(x) = a_i \frac{(x_{i+1}-x)^3}{h_i} + a_{i+1}\frac{(x-x_i)^3}{h_i} + b_i(x_{i+1}-x) + c_i(x-x_i)$$

- 此一形式下的二階導數為分段線性函數，它在結點處為連續。因此，藉由選取P_i的形式，確保二階導數為連續。

三次雲形線內插(Cubic Spline Interpolation)

- 利用$P_i(x_i)=y_i$及$P_i(x_{i+1})=y_{i+1}$的事實，可以用a_i來表示b_i及c_i如下：

$$b_i = \frac{y_i}{h_i} - a_i h_i, \quad c_i = \frac{y_{i+1}}{h_i} - a_{i+1} h_i$$

- 最後，令結點處的一階導數連續。利用此條件我們可得到n個未知數$a_1, \ldots, a_n$，的n-2個方程式。對i=1, …, n-2，

$$h_i a_i + 2(h_i + h_{i+1}) a_{i+1} + h_{i+1} a_{i+2} = \frac{y_{i+2} - y_{i+1}}{h_{i+1}} - \frac{y_{i+1} - y_i}{h_i}$$

三次雲形線內插(Cubic Spline Interpolation)

- 有幾種方式可決定端點處的二階導數，它們補足了解出所有未知數所須的條件。最簡單的選擇是自然三次雲形線，就是指定在x_1和x_n的二階導數為零。在此情形下$a_1=a_n=0$。

當 $n=6$，我們有

$$
\begin{aligned}
2(h_1+h_2)a_2+h_2a_3 &= \frac{y_3-y_2}{h_2}-\frac{y_2-y_1}{h_1}\\
h_2a_2+2(h_2+h_3)a_3+h_3a_4 &= \frac{y_4-y_3}{h_3}-\frac{y_3-y_2}{h_2}\\
+h_3a_3+2(h_3+h_4)a_4+h_5a_5 &= \frac{y_5-y_4}{h_4}-\frac{y_4-y_3}{h_3}\\
+h_4a_4+2(h_4+h_5)a_5 &= \frac{y_6-y_5}{h_5}-\frac{y_5-y_4}{h_4}
\end{aligned}
$$

例題8.10 自然三次雲形線內插

- 考慮數據點(-2,4)、(-1,-1)、(0,2)、(1,1)和(2,8)。若所有區間$h_i=1$，對自然雲形線則有$a_1=a_5=0$ 。係數a_2、a_3和a_4的方程式為：

$$
\begin{aligned}
a_1+4a_2+a_3 &= (y_3-y_2)-(y_2-y_1)\\
+a_2+4a_3+a_4 &= (y_4-y_3)-(y_3-y_2)\\
+a_3+4a_4+a_5 &= (y_5-y_4)-(y_4-y_3)
\end{aligned}
$$

代入 a_1、a_5 及 $y_i(i=1,\dots,5)$ 的值並化簡，我們得到

$$
\begin{aligned}
4a_2+a_3 &= 8\\
a_2+a_3+a_4 &= -4\\
+a_3+4a_4 &= 8
\end{aligned}
$$

這是一個三對角線方程組，可以用第 3 章介紹的方法求解。我們求得

例題8.10 自然三次雲形線內插

$$a_2 = 2.5714, \quad a_3 = -2.2857, \quad a_4 = 2.5714$$

解 b_i 得

$$b_1 = y_1 - a_1 = 4, \qquad b_2 = y_2 - a_2 = -3.5714$$
$$b_3 = y_3 - a_3 = 4.2857, \quad b_4 = y_4 - a_4 = -1.5714$$

再求 c_i 爲

$$c_1 = y_2 - a_2 = -3.5714, \quad c_2 = y_3 - a_3 = 4.2857$$
$$c_3 = y_4 - a_4 = -1.5714, \quad c_4 = y_5 - a_5 = 8$$

此三次雲形線顯示於圖 8.23；它可化簡爲

$$S(x) = \begin{cases} 2.57(x+2)^3 - 4(x+1) - 3.57(x+2), & -2 \le x \le -1 \\ -2.57x^3 - 2.29(x+1)^3 + 3.57x + 4.29(x+1), & -1 \le x \le 0 \\ -2.29(1-x)^3 + 2.57x^3 + 4.29(1-x) - 1.57x, & 0 \le x \le 1 \\ 2.57(2-x)^3 - 1.57(2-x) + 8(x-1), & 1 \le x \le 2 \end{cases}$$

例題8.10 自然三次雲形線內插

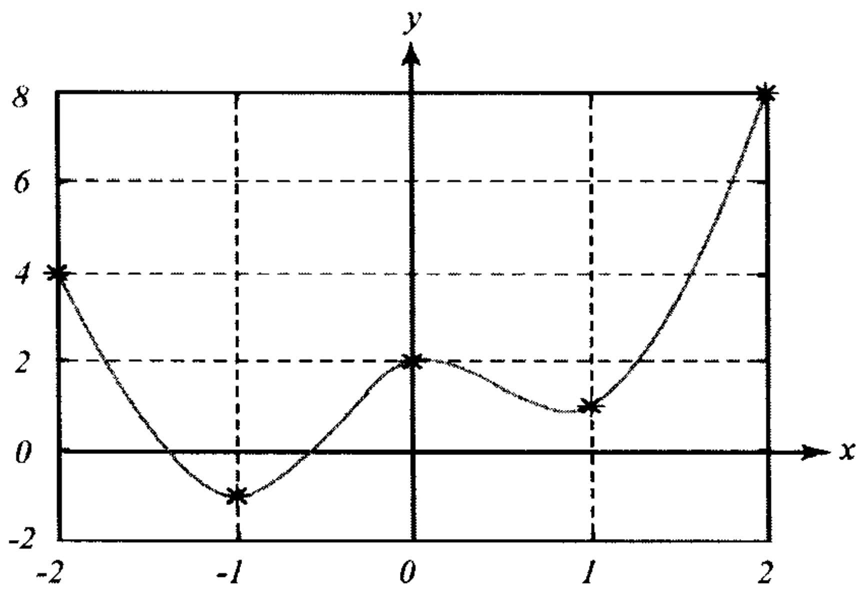


圖 8.23　三次雲形線內插

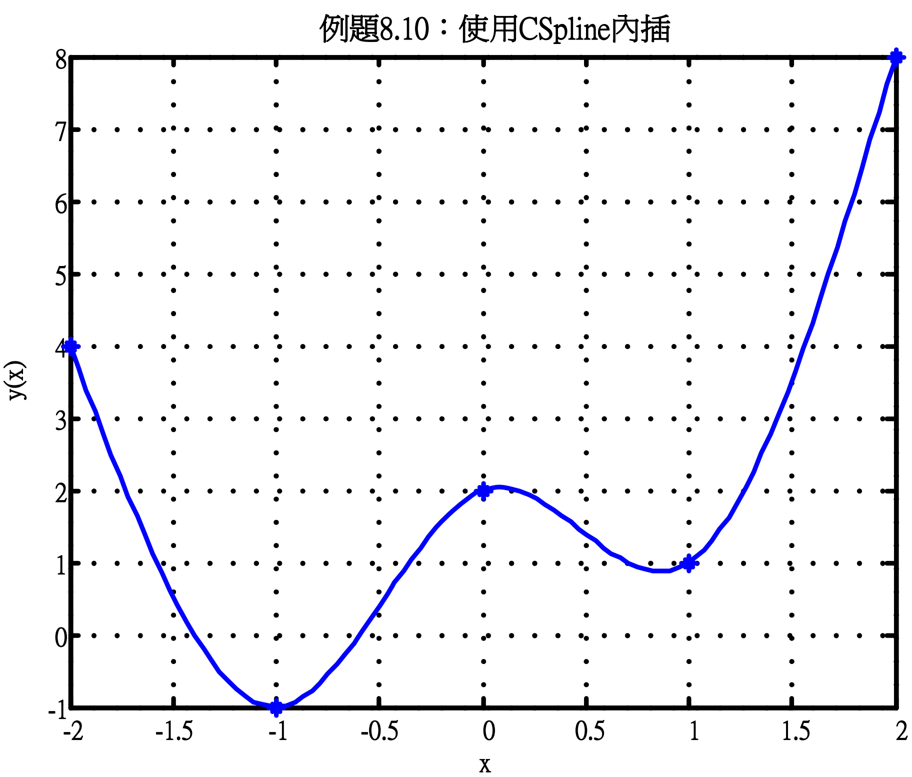
例題8.10 Matlab程式

- 主程式：Main_CSpline_Ex_810.m
- CSpline內插係數程式： CSplineCoef.m
- CSpline內插點程式： CSplineEval.m

```
clear all;
close all;

x=[-2 -1 0 1 2];
y=[4 -1 2 1 8];
t=linspace(-2,2);

c=CSplineCoef(x,y);
s=CSplineEval(x,t,c);

plot(t,s,'-b',x,y,'*');
xlabel('x');
ylabel('y(x)');
title('例題8.10：使用CSpline內插');
grid on;
```


例題8.10 自然三次雲形線內插

- 結果：



例題8.11 Runge函數

- 函數 $$f(x)=\frac{1}{1+25x^2}$$
- 三次雲形線內插多項式，使用區間 [-1,1]中五個等間隔點的數據如下：
  - *x=[-1.0 -0.5 0.0 0.5 1.0]*
  - *y=[0.0385 0.1379 1.0000 0.1379 0.0385]*

例題8.11 Runge函數

- 數據顯示於圖8.24，圖中同時顯示雲形線函數(實線)及實際的Runge函數(虛線)。其相符程度遠優於之前例題8.6的多項式內插，兩者使用同樣的數據點。

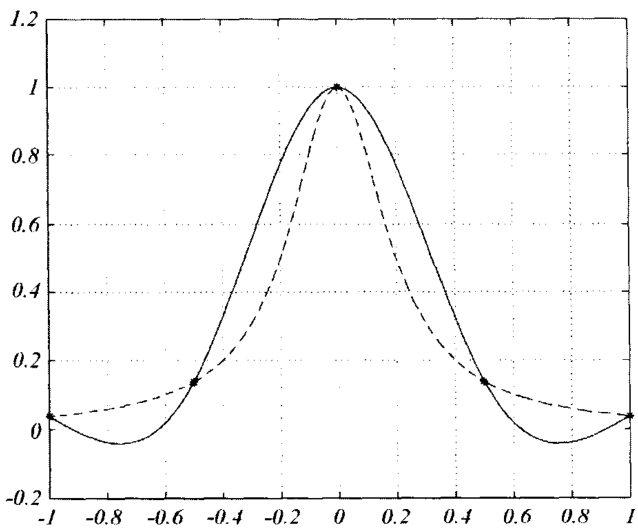


圖 8.24 Runge 函數的三次雲形線內插

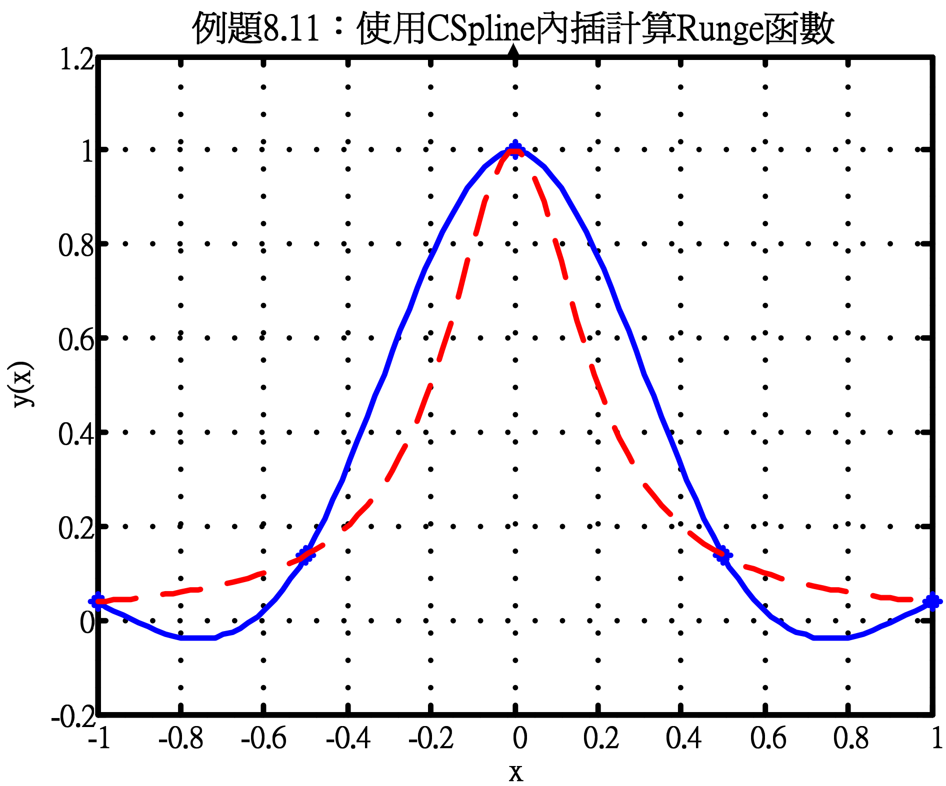
例題8.11 Matlab程式

- 主程式：Main_CSpline_Ex_811.m
- CSpline內插係數程式： CSplineCoef.m
- CSpline內插點程式： CSplineEval.m

```
clear all;
close all;

x=[-1.0   -0.5   0.0    0.5    1.0];
y=[0.0385 0.1379 1.0000 0.1379    0.0385];
t=linspace(-1,1);
y1=(1+25*t.^2).^(-1);

c=CSplineCoef(x,y);
s=CSplineEval(x,t,c);

plot(t,s,'-b',x,y,'*',t,y1,'--r');
xlabel('x');
ylabel('y(x)');
title('例題8.11：使用CSpline內插計算Runge函數');
grid on;
```


例題8.11 Runge函數

- 結果：



自然三次雲形線內插的討論

- 自然三次雲形線內插的值及其一、二階導數在節點處為連續，假設有n個數據點(x_1,y_1), …, (x_n,y_n) 。換句話說，此時節點就是數據點。有n-1個區間，每個區間中的三次多項式有四個未知數;因此我們共有4(n-1)個未知數。現在，每一個三次多項式上必定有兩個數據點，所以對已知數據計有2(n-1)個方程式。
- 此外，計有n-2個x的值，必須保有連續的一階和二階導數。這樣得到另外2(n-2)個方程式，所以總共有4n-6個方程式，包含4n-4個未知數。為獲得另外兩個方程式，我們可以指定兩個端點 (亦即x_1和x_n)處的一階或二階導數。

自然三次雲形線內插的討論

- 藉由適當的選取方程式的代數形式，可以簡化雲形線內插公式的推導。因為x之間距並不需要相等，所以令$h_i=x_{i+1}-x_i$。因此，所要求的雲形線函數的形式為

$$S(x)=\begin{cases} P_1(x), & x_1 \le x < x_2 \\ \quad\vdots & \\ P_i(x), & x_i \le x < x_{i+1} \\ \quad\vdots & \\ P_{n-1}(x), & x_{n-1} \le x \le x_n \end{cases}$$

自然三次雲形線內插的討論

- 為找到一個方便的P_i形式，我們先令它的二階導數在節點處為連續，它的二階導數是線性的。如果我們將二階導數寫成

$$P_i''(x)=6a_i\frac{(x_{i+1}-x)}{h_i}+6a_{i+1}\frac{(x-x_i)}{h_i}$$

得$P_i''(x_{i+1})=6a_{i+1}$。同理

$$P_{i+1}''(x)=6a_{i+1}\frac{(x_{i+2}-x)}{h_{i+1}}+6a_{i+2}\frac{(x-x_{i+1})}{h_{i+1}}$$

可得$P_{i+1}''(x_{i+1})=6a_{i+1}$。

自然三次雲形線內插的討論

將$P_i''(x)=6a_i\frac{(x_{i+1}-x)}{h_i}+6a_{i+1}\frac{(x-x_i)}{h_i}$積分兩次

得

$$P_i(x)=a_i\frac{(x_{i+1}-x)^3}{h_i}+a_{i+1}\frac{(x-x_i)^3}{h_i}+t_ix+s_i$$

若將線性部分t_ix+s_i表示成$b_i(x_{i+1}-x)+c_i(x-x_i)$

的對稱形式會更方便。

$$P_i(x)=a_i\frac{(x_{i+1}-x)^3}{h_i}+a_{i+1}\frac{(x-x_i)^3}{h_i}+b_i(x_{i+1}-x)+c_i(x-x_i)$$

自然三次雲形線內插的討論

加入$P_i(x_i)=y_i$與$P_i(x_{i+1})=y_{i+1}$，將b_i和c_i用$a_i, a_{i+1}, y_i, y_{i+1}$來表示可得

$$b_i=\frac{y_i}{h_i}-a_ih_i, c_i=\frac{y_{i+1}}{h_i}-a_{i+1}h_i$$

所以

$$P_i(x)=a_i\frac{(x_{i+1}-x)^3}{h_i}+a_{i+1}\frac{(x-x_i)^3}{h_i}+[y_i-a_ih_i^2]\frac{(x_{i+1}-x)}{h_i}+[y_{i+1}-a_{i+1}h_i^2]\frac{(x-x_i)}{h_i}$$

其一階導數為

$$P_i^{'}(x)=-3a_i\frac{(x_{i+1}-x)^2}{h_i}+3a_{i+1}\frac{(x-x_i)^2}{h_i}+\frac{-(y_i-a_ih_i^2)}{h_i}+\frac{(y_{i+1}-a_{i+1}h_i^2)}{h_i}$$

自然三次雲形線內插的討論

令P_i和P_{i+1}在x_{i+1}處有相同的一階導數，即

$$P_i^{'}(x_{i+1})=3a_{i+1}\frac{(x_{i+1}-x_i)^2}{h_i}+\frac{-(y_i-a_ih_i^2)}{h_i}+\frac{(y_{i+1}-a_{i+1}h_i^2)}{h_i}$$

$$=2a_{i+1}h_i+\frac{y_{i+1}-y_i}{h_i}+a_ih_i$$

和

$$P_{i+1}^{'}(x_i)=-2a_{i+1}h_{i+1}+\frac{y_{i+2}-y_{i+1}}{h_{i+1}}+a_{i+2}h_{i+1}$$

因此

$$2a_{i+1}h_i+\frac{y_{i+1}-y_i}{h_i}+a_ih_i=-2a_{i+1}h_{i+1}+\frac{y_{i+2}-y_{i+1}}{h_{i+1}}+a_{i+2}h_{i+1}$$

對$i=1,2,\cdots,n-2$，上式可得三對角線方程組

$$h_ia_i+2(h_i+h_{i+1})a_{i+1}+h_{i+1}a_{i+2}=\frac{y_{i+2}-y_{i+1}}{h_{i+1}}-\frac{y_{i+1}-y_i}{h_i}$$

此式中有$n-2$個方程式，包括n個未知數$(a_1,\cdots,a_n)$

自然三次雲形線內插的討論

- 有幾種方式可指定額外的兩個條件。自然三次雲形線指定a_1和a_n都是零；因此在端點處的二階導數為零。固定雲形線則指定端點處的一階導數值。
- 使用三次雲形線內插的誤差為$|S(x)-g(x)|$，其中S(x)是雲形線內插函數，g(x)則是產生數據點的函數。令h代表最大的節點間距，且$G=\max|g^{(4)}(x)|$。則只要所選的兩個額外條件是合理的，有

$$\left|S(x)-g(x)\right|<kh^4G=O(h^4)$$

三次雲線內插
(Cubic Spline Interpolation)

- 三次雲形線內插，它須要解一組三對角線方程組。
- 內插的數據是以點的形式$(x_1,y_1),\cdots,x_n,y_n)$出現，其係數矩陣只取決於橫座標值(x值)。方程組的右側項則取決於數據的縱座標值(y值)。
- 先求出係數矩陣的LU因式分解，再用因式來解線性方程組，會比用Gauss-Thomas演算法分別求解每一個方程組更有效率。

三次雲線內插

- 如果有n組數據點，就須要求n-2個參數a(1),...,a(n-1)的值。如果每組數據的間隔相等，也就是對j=1,...,n-1都有x(j+1)-x(j)=h，則此三對線方程組(n=6)的形式為

$$\begin{bmatrix} 4 & 1 & 0 & 0 \\ 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ 0 & 0 & 1 & 4 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \frac{1}{h^2} \begin{bmatrix} y_3 - 2y_2 + y_1 \\ y_4 - 2y_3 + y_2 \\ y_5 - 2y_3 + y_2 \\ y_6 - 2y_5 + y_4 \end{bmatrix}$$

進階問題

- 先考慮有理函數內插。然後再介紹一些用於一或二維內插的MATLAB內建函數。

有理函數內插

- 要代表一個函數或一組數據，多項式並不一定是最有效的形式。有時有理函數 (兩個多項式的比值)會是較佳的選擇，特別是在我們所要的區域中，我們要近似的函數存在有極點 (分母為零)時。即使極點出現在複平面上，多項式內插也會有困難，除非它離我們內插的區域很遠。例題8.8中的Runge函數，若將其視為複變數z=x+yi的函數，它成為$f(z)=1/(1+25z^2)$;它的極點位於z=0.2i，太接近感興趣的區域，實數區間 [-1,1]。

有理函數內插

- 在本節中，簡短介紹有理函數內插。Bulirsch-Stoer算則會產生一個「對角線(diagonal)」有理函數，亦即，函數分子的次數等於分母的次數或比分母次數少一。此方法是遞迴性的，它依靠表列式的數據(類似於牛頓形式的多項式內插所用的方式)。
- 已知一組k筆數據點 $(x_1,y_1),\ldots,(x_k,y_k)$，要找一個如下形式的內插函數

$$r(x) = \frac{p_m(x)}{q_n(x)} = \frac{a_m x^m + \cdots + a_0}{b_n x^n + \cdots + b_0}$$

有理函數內插

- 一般來說，我們應該要指定分子與分母的次數。但是對 Bulirsch-Stoer法，r(x)次數必為m=n或m=n-1(取決於數據點數k是奇數或偶數)。
- 此算則可整理如下:

有理函數內插

第一階段；要內插 (x_i,y_i)

$$R_i = y_i$$

第二階段：要內插(x_1,y_1) 和 (x_2,y_2)

$$R_{12} = R_2 + \frac{R_2 - R_1}{\frac{x - x_1}{x - x_2}[1 - \frac{R_2 - R_1}{R_2}] - 1}$$

類似的，要內插(x_2,y_2) 和 (x_3,y_3)

$$R_{23} = R_3 + \frac{R_3 - R_2}{\frac{x - x_2}{x - x_3}[1 - \frac{R_3 - R_2}{R_3}] - 1}$$

有理函數內插

第三階段：要內插(x_1,y_1)、(x_2,y_2) 和 (x_3,y_3)，結合

R_{12} (內插 (x_1,y_1) 和 (x_2,y_2))、

R_{23} (內插 (x_2,y_2) 和 (x_3,y_3))、以及

R_2 (內插 (x_2,y_2)，如下：

$$R_{123} = R_{23} + \frac{R_{23} - R_{12}}{\frac{x - x_1}{x - x_3}\left[1 - \frac{R_{23} - R_{12}}{R_{23} - R_2}\right] - 1}$$

有理函數內插

第 k 階段：要內插 $(x_1,y_1), \ldots, (x_k,y_k)$，結合

$R_{23\ldots k}$ (內插點 $(x_2,y_2), \ldots, (x_k,y_k)$)、

$R_{12\ldots(k-1)}$ (內插點 $(x_1,y_1), \ldots, (x_{k-1},y_{k-1})$) 及

$R_{23\ldots(k-1)}$ (內插點 $(x_2,y_2), \ldots, (x_{k-1},y_{k-1})$)，如下：

$$R_{123\ldots k} = R_{23\ldots(k-1)} + \frac{R_{23\ldots k} - R_{12\ldots(k-1)}}{\frac{x - x_1}{x - x_k}\left[1 - \frac{R_{23\ldots k} - R_{12\ldots(k-1)}}{R_{23\ldots k} - R_{23\ldots(k-1)}}\right] - 1}$$

有理函數內插

- 此通用形式來自第三階段。第二階段的計算也是依照此種形式，但要瞭解，在此階段內插(k-2)個點的有理函數是零。

第一階段	第二階段	第三階段
$R_1 = y_1$		
	$R_{12} = R_2 + \dfrac{R_2 - R_1}{\dfrac{x-x_1}{x-x_2}\left[1-\dfrac{R_2-R_1}{R_2}\right]-1}$	
$R_2 = y_2$		$R_{23} + \dfrac{R_{23} - R_{12}}{\dfrac{x-x_1}{x-x_3}\left[1-\dfrac{R_{23}-R_{12}}{R_{23}-R_2}\right]-1}$
	$R_{23} = R_3 + \dfrac{R_3 - R_2}{\dfrac{x-x_2}{x-x_3}\left[1-\dfrac{R_3-R_2}{R_3}\right]-1}$	
$R_3 = y_3$		

有理函數內插的討論

- 很容易就能驗證，第二階段的表示式-亦即內插兩點的結果-具有預期的形式。不過，此算則並不會提供一個內插數點的有理函數的代數式；它是以一種系統化的方式計算指定點上的內插值。
- 下面的MATLAB函數即為Bulirsch-Stoer程序的應用。此函數會單獨處理在數據點的內插，以減少計算量並避免出現除以零的情形。對第j行元素的表示式(在第j階段求R)經過改寫，以減少可能出現無定義除法0/0的情況，這種情況MATLAB會用NaN來表示。如果在計算某一個y值的時候真的出現 NaN，在繪製內插數據時MATLAB會跳過該點。

例題8.12 有理函數內插

- 利用例題8.6中Runge函數的數據，以說明BuIirsch-Stoer有理函數內插

$$Runge\text{函數：}\quad f(x)=\frac{1}{1+25x^2}$$

- 使用區間[-1,1]中五個等間隔點的數據如下：
  - *x=[-1.0　-0.5　0.0　0.5　1.0]*
  - *y=[0.0385　0.1379　1.0000　0.1379　0.0385]*

例題8.12 有理函數內插

- 圖8.25顯示使用21筆數據點的結果。內插的結果與原函數值幾乎無法區別。用五個數據點的結果基本上一樣。很明顯的，不像多項式內插法般用更多的數據點也不會造成問題。

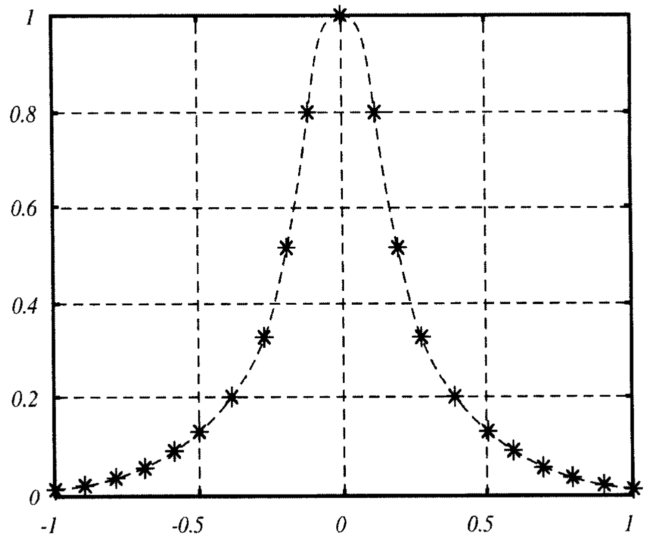


圖 8.25　Runge 函數的函數值與內插值

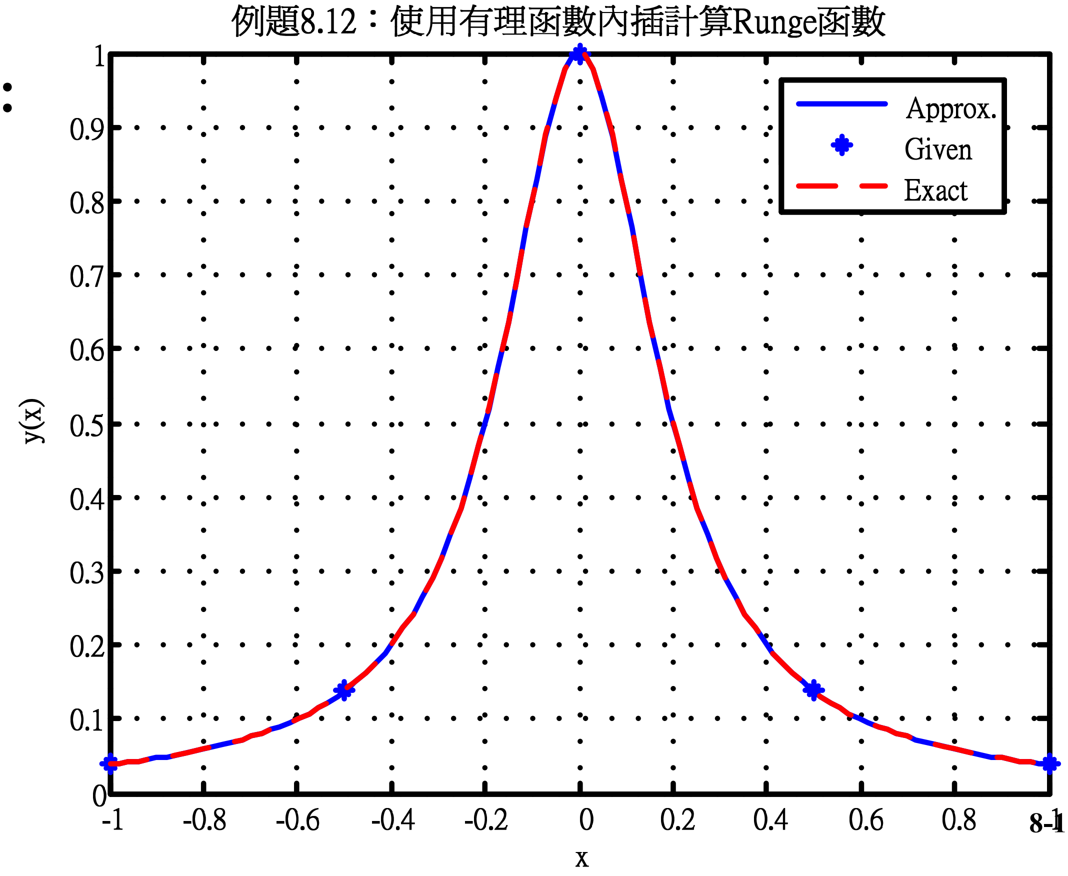
例題8.12 Matlab程式

- 主程式：Main_RatInterp_Ex_812.m
- 有理函數內插點程式： RatInterp.m

```
clear all;
close all;

x=[-1.0   -0.5   0.0    0.5    1.0];
y=[0.0385 0.1379 1.0000 0.1379    0.0385];
t=linspace(-1,1);
y1=(1+25*t.^2).^(-1);

yy=RatInterp(x,y,t);
plot(t,yy,'-b',x,y,'*',t,y1,'--r');
xlabel('x');
ylabel('y(x)');
title('例題8.12：使用有理函數內插計算Runge函數');
legend('Approx.','Given','Exact','Location','best');
grid on;
```


例題8.12 Runge函數

- 結果：



使用 MATLAB的函數

- MATLAB有數種內建函數可進行內插。

一維內插

對一維的數據，由自變數所組成的向量 **x** 必須是單調的遞增或遞減。MATLAB 函數 `interp1` 會依據所給的數據 **x** 和 **y**，求出在指定的 s 值處的內插值 t (s 和 t 也可以是向量)。內定的方法是線性內插；呼叫語法爲

```
t = interp1(x, y, s)
```

其它還包括三次、雲形線 (三次雲形線) 及最近値 (nearest-value) 法等；各個函數呼叫方式爲

```
t = interp1(x, y, s, 'cubic' ),
t = interp1(x, y, s, 'spline' ),
t = interp1(x, y, s, 'nearest' ).
```

在進行內插之前，向量 **x**、**y** 和 **s** 會先被映射到一個等間隔區間。如果所給數據原本就是等間隔分佈 (亦即 **x** 爲單調且等間隔)，可以使用快速法`*linear`、`*cubic`、`*nearest` 和`*spline`。對不是等間隔分佈的數據做快速線性內插，可以使用 `interp1q` 函數。

一維內插

MATLAB 的函數 pchip 可執行分段三次 Hermite 內插。它可用來內插一組給定點，或是傳回函數 ppval 所須的資料。

- yi=pchip (x,y,xi)

  向量 yi 是在 xi 處的內插值。如果輸入的 y 是一個矩陣，會針對 y 的每一行做內插。

- pp=pchip (x,y)

  pp 是供 ppval 使用的分段多項式的結構。

  x 是列或行向量均可。y 是長度和 x 一樣的列或行向量，或是一個有 length(x) 行的矩陣。

  除了 interp1 函數中的 spline 選項之外，MATLAB 還有一個 spline 函數可做三次雲形線內插。

一維內插

- pp=spline (x,Y) 會傳回三次雲形線內插的分段多項式結構，以於之後供 ppval 和 unmkpp 使用。x 必須是向量。Y 可以是純量、向量或任何維數的陣列。如果 Y 是一個純量或向量，它的長度必須和 x 一樣。
- yy=spline (x,Y,xx) 和 yy=ppval (spline (x,Y),xx) 是一樣的，它傳回的 yy 是在 xx 處的內插值。xx 可以是純量、向量或多維陣列。

由 spline 所做的內插和 pchip 的類似。通常，spline 會得到較平滑的結果——亦即 y'' 爲連續，而 pchip 只有 y' 是連續的。再者，如果所用數據來自一個平滑函數，則 spline 所得結果準確度較高。但在另一方面，如果數據並不平滑，pchip 所得結果較不會振盪也不會有超越 (overshoots) 的問題，且 pchip 的設定較容易。計算插值時，兩個函數所須計算量相當。

一維內插

- **圖8.26顯示了用spline和pchip以內插例題8.9之數據的差異;圖8.27則是對Runge函數的結果。**

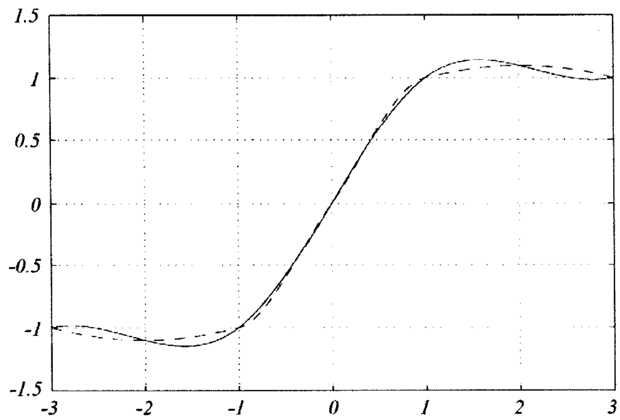


圖 8.26　Spline 和 pchip 內插

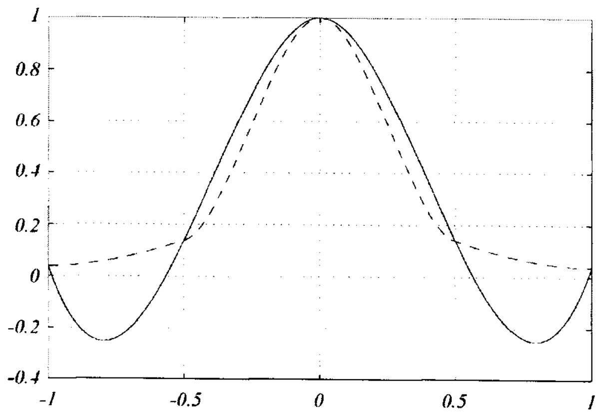


圖 8.27　Spline 和 pchip 對 Runge 函數的內插

二維內插

一般而言，對兩個 (或更多) 自變數的內插，是比單一變數時困難很多。其中一個原因是，除非已知函數值是位於矩形網格點上，否則要排列各點，或決定在某一位置該用哪些點做插值，都很困難。

讓我們考慮 MATLAB 的內建函數 `interp2` 以對定義於矩形格點的數據做內插:

$$\begin{array}{ccccc} (x_1,y_1) & (x_1,y_2) & (x_1,y_3) & \cdots & (x_1,y_m) \\ (x_2,y_1) & (x_2,y_2) & (x_2,y_3) & \cdots & (x_2,y_m) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (x_n,y_1) & (x_n,y_2) & (x_n,y_3) & \cdots & (x_n,y_m) \end{array}$$

換句話說，數據的形式爲 $z(i,j) = f(x(i), y(j))$。

MATLAB 的 `interp2` 函數所用的線性內插，更適當的名稱是雙線性 (bilinear) 內插。在每一個矩形的子區間中，它的插值是來自雙線性函數 (x 和 y 均爲線性)，其形式爲

$$z = a + bx + cy + dxy,$$

二維內插

使用此區域四個角上的數據。因此，對區間 R_{ij}，四組數據 (x_i, y_j)、(x_i, y_{j+1})、(x_{i+1}, y_j)、(x_{i+1}, y_{j+1})可得四個方程式，包含四個未知數 a_{ii}、b_{ii}、c_{ii} 及 d_{ii}：

$$
\begin{aligned}
z_{i,j} &= a_{ij} + b_{ij}x_i + c_{ij}y_j + d_{ij}x_i y_j \\
z_{i,j+1} &= a_{ij} + b_{ij}x_i + c_{ij}y_{j+1} + d_{ij}x_i y_{j+1} \\
z_{i+1,j} &= a_{ij} + b_{ij}x_{i+1} + c_{ij}y_j + d_{ij}x_{i+1} y_j \\
z_{i+1,j+1} &= a_{ij} + b_{ij}x_{i+1} + c_{ij}y_{j+1} + d_{ij}x_{i+1} y_{j+1}
\end{aligned}
$$

二維內插的基本語法為

```
zz=interp2 (x,y,Z,xx,yy)
```

其中 x (列向量)、y (行向量) 和 Z 為已知數據，而 xx (列向量) 和 yy (行向量) 則給定我們要求插值的位置。Z 必須有 length(x) 行及 length(y) 列。對於這樣的輸入格式，MATLAB 會自動將向量 x、y、xx 和 yy 轉換成 meshgrid 所用的「方格狀」矩陣的形式。如果想要的話，也可用方格狀形式做輸入。

二維內插

interp2 還有額外選項可讓使用者指定內插的形式，包括 cubic (雙三次內插) 和 spline；內定選項為 linear (雙線性內插)。

函數 mesh (x,y,Z) 可繪出簡單的表面圖形，其中向量 x 的元素對應於矩陣 Z 的行，而向量 y 的元素對應於矩陣 Z 的列。例如，要畫出下個例題中的數據，其中 x 由左到右 y 由前到後，我們可以用

```
x = [ 0  0.5  1];  y = [ 0  0.2  0.8  1];  z = (x.^2)'*(1+(y-1).^3),
figure(1), mesh(x, y, z')
```

例題8.13 二維矩形格點的內插

- 為說明Interp2的用法，考慮向量
- x=[0 0.5 1], y=[0 0.2 0.8 1]
- 並定義曲面

$$Z = (x^2)^T * (1 + (y-1)^3)$$

- 函數Interp2內插由向量xx和yy所給定的數據。

例題8.13 二維矩形格點的內插

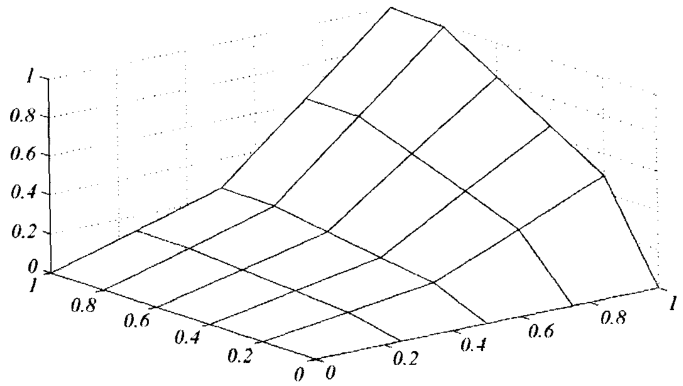


圖 8.28　用雙線性內插所得曲面

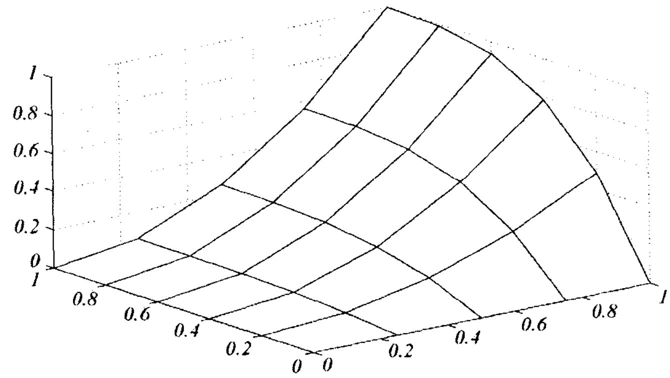


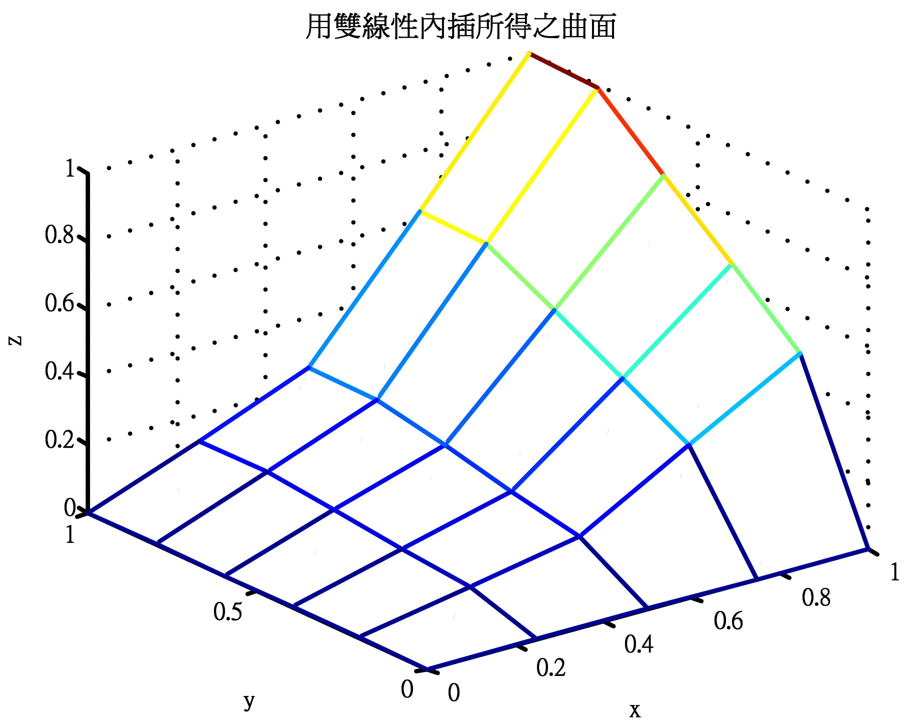
圖 8.29　用雙三次內插所得曲面

例題8.13 Matlab程式

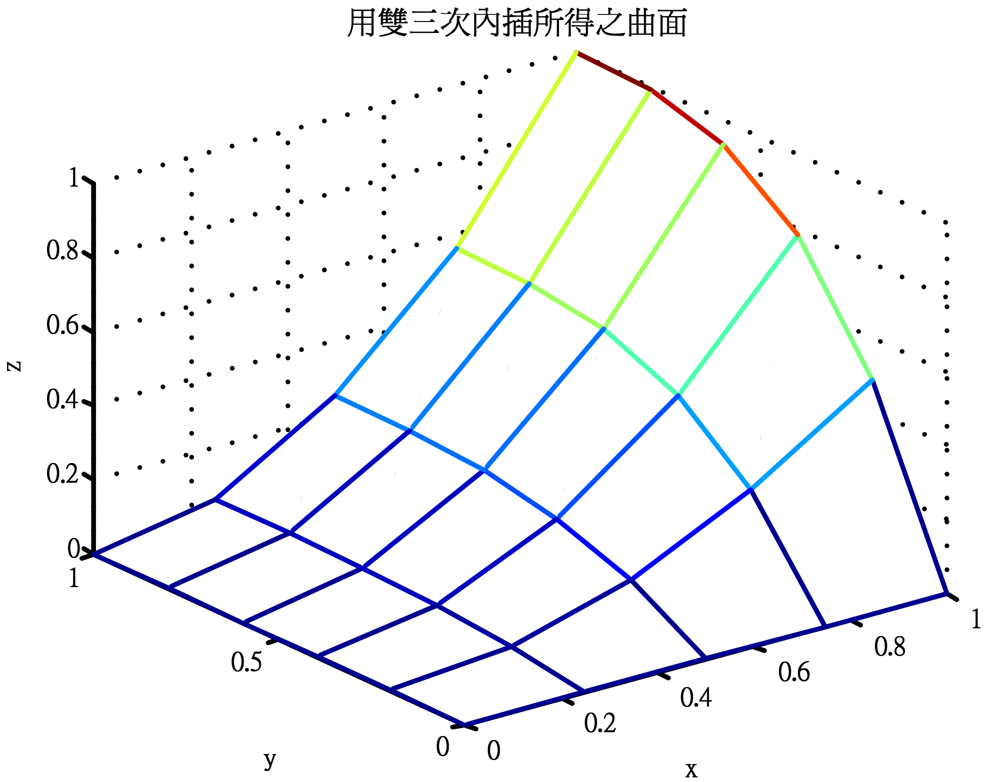
- 主程式：Main_Interp2_Ex_813.m

```
clear all;
close all;

x=[0 0.5 1]; y=[0 0.2 0.8 1]; z=(x.^2)'*(1+(y-1).^3);
xx = 0:0.25:1; yy = 0:0.2:1;
zz = interp2(x,y', z' , xx, yy');
zc = interp2(x,y', z', xx, yy', 'cubic');
figure(1), mesh(xx,yy,zz);
xlabel('x');
ylabel('y');
zlabel('z');
title('用雙線性內插所得之曲面');
grid on;
figure(2),mesh(xx,yy,zc);
xlabel('x');
ylabel('y');
zlabel('z');
title('用雙三次內插所得之曲面');
grid on;
```


例題8.13 Matlab程式



例題8.13 Matlab程式



散漫數據(Scattered Data)

- 現在說明如何以MATLAB的內建函數描繪及內插「散漫」的數據，亦即，不是排列於矩形網格的數據。

例題8.14 Delaunay三角法

對一組已知點 (x_i,y_i)，Delaunay 三角法是一組線段，可將整個區域劃分爲許多三角形，且任一三角形之外接圓的內部不含其它的數據點。

對於由向量 **x** 和 **y** 所定義的數據點，`TRI=delaunay(x,y)` 傳回一組 Delaunay 三角形。*m* 乘 3 矩陣 `TRI` 的每一列，定義出這樣一個三角形，包含的是 **x** 和 **y** 的指標。

函數 `delaunay` 的結果會被用於函數 `griddata` (內插散漫數據) 及 `voronoi` (計算 Voronoi 圖形)。函數 `delaunay` 本身的功用是爲散漫數據點建立出三角格點。

以下腳本說明這些函數的用法，它使用矩形區間 $0 \le x \le 1, 0 \le y \le 2$ 中的十個數據點。

考慮一組由向量 **x** 和 **y** 所決定的點 (x,y)，每一個點都有一個指定值 (由向量 **z** 給定)。利用 MATLAB 的函數以組成這些點的 Delaunay 三角形，然後繪出此曲面，我們可看到內插這些數據的分段曲面 (如圖 8.30 所示)。

例題8.14 Delaunay三角法

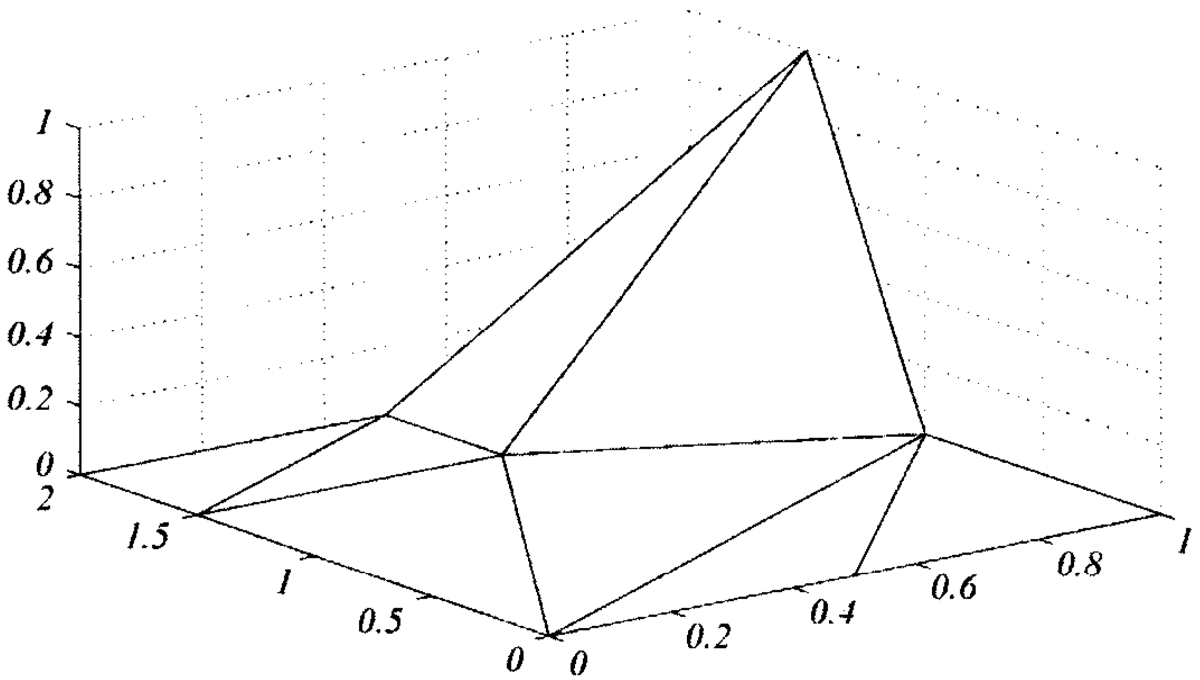


圖 8.30 使用 trimesh 所得的 Delaunay 三角面

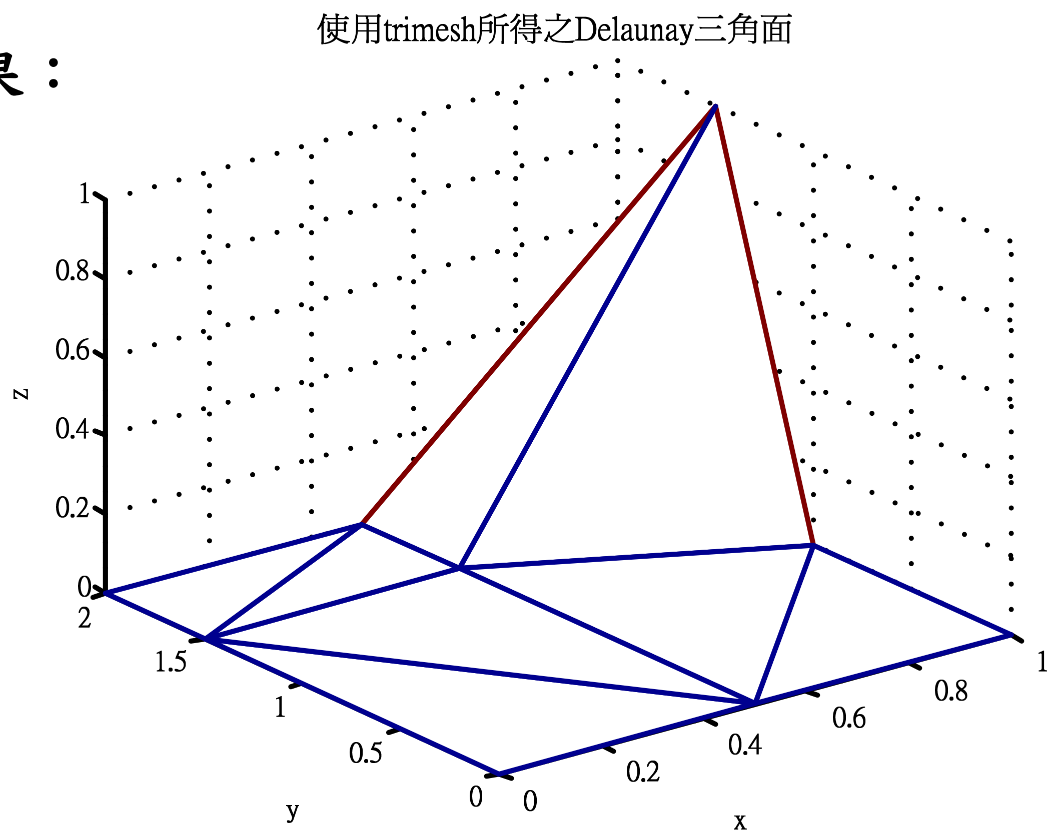
例題8.14 Matlab程式

- 主程式：Main_Delaunay_Ex_814.m

```
clear all;
close all;

x = [0.0  0.0 0.0 0.5 0.5 0.5 1.0 1.0 1.0 1.0];
y = [ 0.0 1.5 2.0 0.0 1.5 2.0 0.0 1.0 1.5 2.0];
z = [ 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0];
figure(1), plot(x,y,'*'), grid on;
tri = delaunay(x,y);
figure(2), trimesh(tri,x,y,z);
xlabel('x');
ylabel('y');
zlabel('z');
title('使用trimesh所得之Delaunay三角面');
grid on;
```


例題8.14 Matlab程式

- 結果：



散漫數據(Scattered Data)

• 現在考慮對散漫數據做實際內插的問題，而不僅是以分段線性曲面描繪出這些點的Delaunay三角形。觀念是構成一組矩形網格點，然後求包含指定點之曲面的凸包(convexhull)；MATLAB的函數griddata使用quickhull算則。我們用前一個例題中的數據來說明此一程序。內插表面顯示於圖8.31。

散漫數據(Scattered Data)

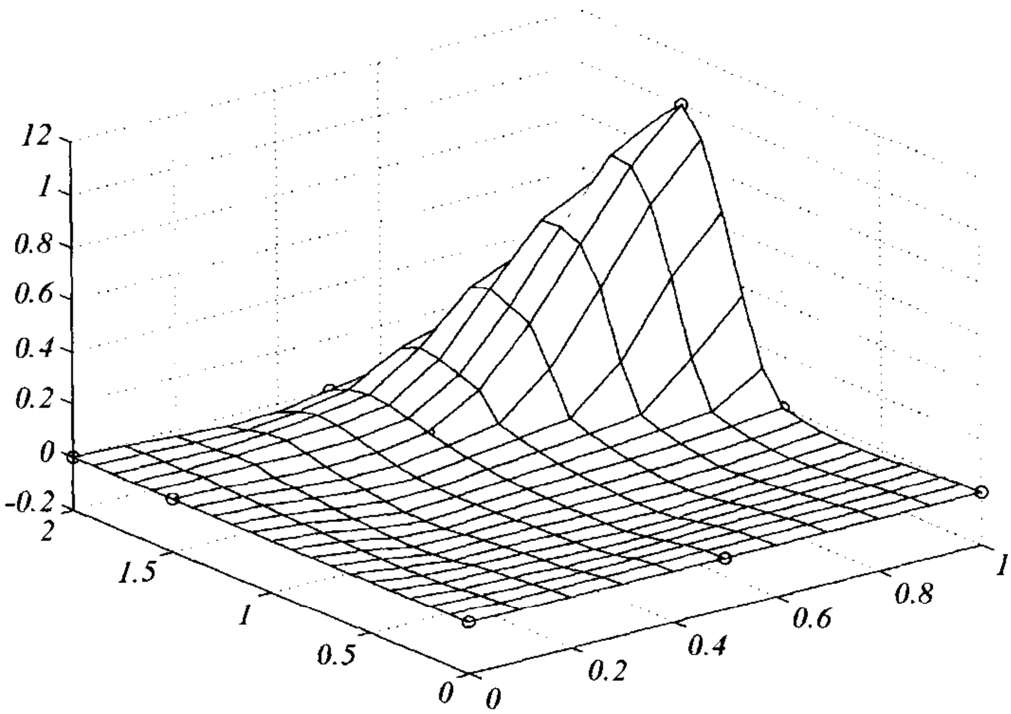


圖 8.31 利用 MATLAB 函數 griddata 所產生的曲面

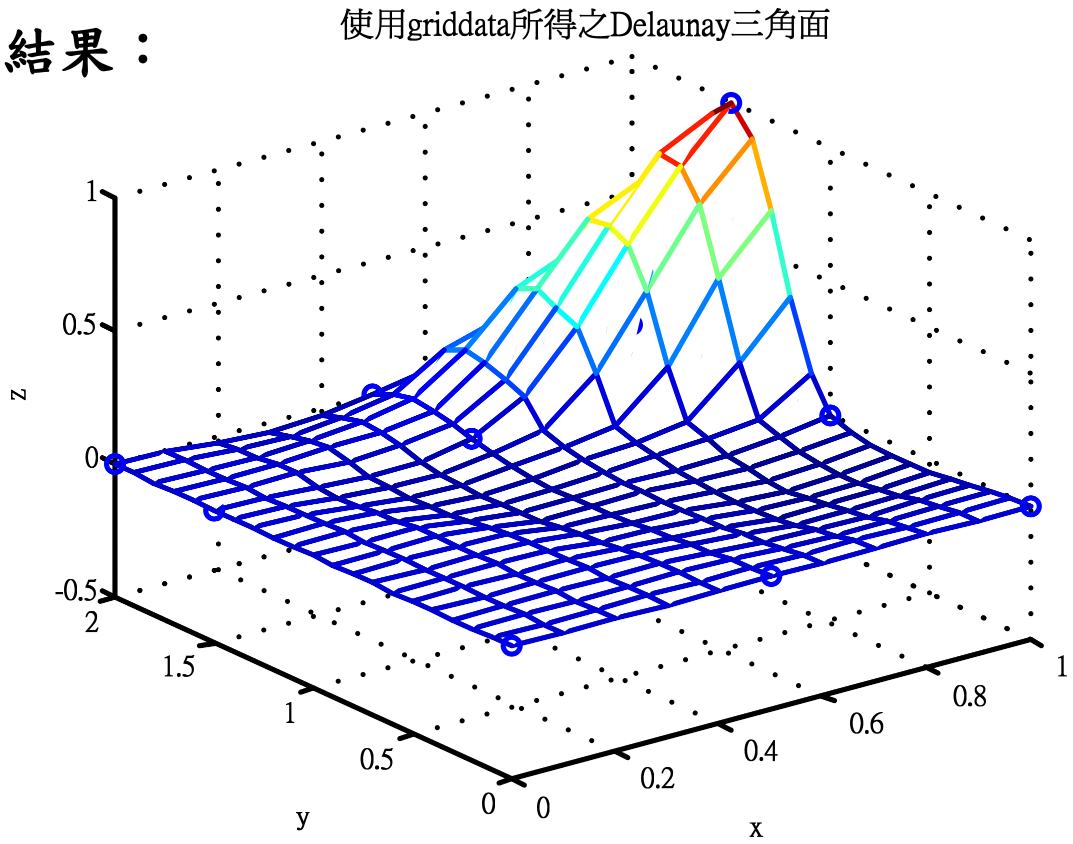
例題8.15 使用凸包

• 主程式：Main_Conveshull_Ex_815.m

```
clear all;
close all;

x = [0.0  0.0 0.0 0.5 0.5 0.5 1.0 1.0 1.0 1.0];
y = [ 0.0 1.5 2.0 0.0 1.5 2.0 0.0 1.0 1.5 2.0];
z = [ 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0];
ti = 0.0 : 0.1 : 1.0;
si = 0.0 : 0.1 : 2.0;
[XI, YI] = meshgrid(ti, si);
ZI = griddata(x,y,z, XI, YI, 'cubic');
mesh(XI, YI, ZI)
hold on, plot3(x, y, z, 'o'), hold off
xlabel('x');
ylabel('y');
zlabel('z');
title('使用griddata所得之Delaunay三角面');
grid on;
```


例題8.15 使用凸包

• 結果：



Matlab函數-meshgrid

MATLAB 函數 `meshgrid` 會產生出可用以計算雙變數函數之函數值的陣列，並用以繪出曲面圖。

- `[X,Y]=meshgrid (x,y)`
  會將向量 x 和 y 所指定之區域轉換成陣列 X 和 Y，它們可用以計算雙變數函數的函數值並產生三維格點/曲面圖形。輸出陣列 X 的各列是向量 x 的複本；輸出陣列 Y 的各行是向量 y 的複本。
- `[X,Y,Z]=meshgrid (x,y,z)`
  可產生三維陣列，可用以計算三變數函數的函數值並產生三維容體 (volumetric) 圖形。

  MATLAB 函數 `griddata` 可求得以非等間隔方式給定之數據的擬合曲面。它利用 quickhull 算則。

Matlab函數-meshgrid

- `ZI=griddata (x,y,z,XI,YI)`
  將形式為 `z=f(x,y)`的曲面，與非等間隔 (通常) 向量`(x,y,z)`做擬合。`griddata` 則在由`(XI,YI)`所指定的點內插此曲面，以產生 `ZI`。產生之曲面一定通過所給數據點。`XI` 和 `YI` 通常構成一組均勻格點 (和 `meshgrid` 所產生的一樣)。
  `XI` 可以是列向量，此時它代表一個行是常數的矩陣。類似的，`YI` 可以是列向量，此時它代表一個列為常數的矩陣。
- `[XI,YI,ZI]=griddata (x,y,z,XI,YI)`
  傳回插值矩陣 `ZI` 如上，同時傳回由列向量 `XI` 及行向量 `YI` 所組成的矩陣 `XI` 和 `YI`。後者和 `meshgrid` 所傳回的矩陣一樣。使用者也可指定方法：內定值為 `'linear'`，它使用三角形線性內插；`'cubic'`則使用三角形三次內插，以產生平滑曲面。

Matlab函數-trisurf

MATLAB 函數 `trisurf` 會產生一個三角曲面圖，對於以 「散漫」 方式給定之數據 (相對於矩形網格) 的可視化，它是非常有用的。MATLAB 函數 `trimesh` 可繪製三角網格圖形。

- `trisurf (Tri,X,Y,Z)`
  可繪出定義在 m 乘 3 面矩陣 `Tri` 中的三角形。`Tri` 的每一列定義一個三角形面，用的是指到向量或矩陣 `X`、`Y` 和 `Z` 中頂點的指標。
- `trimesh (Tri,X,Y,Z)`
  會將定義在 m 乘 3 面矩陣 `Tri` 的三角形以網格方式畫出。

綜合整理

多項式內插

通過(x_1,y_1)、(x_2,y_2) 和 (x_3,y_3)三點的 Lagrange 形式拋物線方程式爲

$$p(x)=\frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)}y_1+\frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)}y_2+\frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)}y_3$$

通過這三點的牛頓形式拋物線方程式爲

$$p(x)=a_1+a_2(x-x_1)+a_3(x-x_1)(x-x_2)$$

$$a_1=y_1,\quad a_2=\frac{y_2-y_1}{x_2-x_1},\quad a_3=\frac{\dfrac{y_3-y_2}{x_3-x_2}-\dfrac{y_2-y_1}{x_2-x_1}}{x_3-x_1}$$

綜合整理

Hermite 內插

Hermite 內插可求得在節點處，函數值與函數的一階導數值都相符的多項式。使用牛頓形式表格，數據點要重覆一次，導數值置於表中適當的位置。

或是以另一種方式，對 [0,1] 間的三次 Hermite 內插，其基底函數爲

$$
\begin{aligned}
y_a &= -0.5t^3 + t^2 - 0.5t \\
y_b &= 1.5t^3 - 2.5t^2 + 1.0 \\
y_c &= -1.5t^3 + 2.0t^2 + 0.5t \\
y_d &= 0.5t^3 - 0.5t^2
\end{aligned}
$$

若 $f(0)=f_b$ 、 $f(1)=f_c$ 、 $f'(0)=s_a$ 、 $f'(1)=s_b$ 爲已知，定義 $f_a = f_c - 2s_a$ 和 $f_d = 2s_b + f_b$ 。則所要的內插函數 (在 [0,1] 間) 爲

$$y = f_a y_a + f_b y_b + f_c y_c + f_d y_d$$

綜合整理

三次雲形線內插

令 $h_i = x_{i+1} - x_i$ 。對於 $x_i \le x \le x_{i+1}$ ， $S(x) = P_i(x)$ ，其中

$$P_i(x) = a_i \frac{(x_{i+1}-x)^3}{h_i} + a_{i+1}\frac{(x-x_i)^3}{h_i} + b_i(x_{i+1}-x) + c_i(x-x_i)$$

我們可將 b_i 和 c_i 用 a_i 來表示如下：

$$b_i = \frac{y_i}{h_i} - a_i h_i, \quad c_i = \frac{y_{i+1}}{h_i} - a_{i+1} h_i$$

解包含 $n-2$ 個未知數 $a_2, \ldots, a_{n-1}$ 的 $n-2$ 個方程式，其中 $a_1 = a_n = 0$ 。對 $i = 1, \cdots, n-2$

$$h_i a_i + 2(h_i + h_{i+1})a_{i+1} + h_{i+1} a_{i+2} = \frac{y_{i+2} - y_{i+1}}{h_{i+1}} - \frac{y_{i+1} - y_i}{h_i}$$

MATLAB示範函數

- `c=LagrangeCoef (x,y)`

  求 Lagrange 內插多項式之係數的函數，其內插數據給定於向量 x、y。

- `p=LagrangeEval (t,x,c)`

  計算向量 t 所指定點之 Lagrange 內插多項式值的函數。

- `a=NewtonCoef (x,y)`

  求牛頓內插多項式之係數的函數，其內插數據給定於向量 x、y。

- `N=NewtonEval (t,x,a)`

  計算向量 t 所指定點之牛頓內插多項式值的函數。

- 畫出三次 Hermite 基底在[-1,2]間圖形的腳本。

- `a=HermiteCoef (x,y,dy)`

  求 Hermite 內插多項式之係數的函數，其內插數據給定於向量 x、y、dy。

MATLAB示範函數

- `H=HermiteEval (t,x,a)`

  計算向量 t 所指定點之 Hermite 內插多項式值的函數。

- `coef=CSplineC (x,y)`

  求自然三次雲形內插多項式係數之函數，其內插數據給定於向量 x、y。

- `s=CSplineE (x,t,coef)`

  計算向量 t 所指定點之自然三次雲形內插多項式值的函數。

- `yy=RatInterp (x,y,xx)`

  求有理函數內插值的函數，內插數據給定於向量 x、y，內插點爲向量 xx。

MATLAB內建函數

- `t=interp1 (x,y,s)`

  內插給定於向量 x,y 之數據的內建函數，內插點給定於向量 s。內定值爲線性內插，可指定爲三次及雲形線內插。

- `zz=interp2 (x,y,Z,xx,yy)`

  做內插的內建函數，給定數據的點是由列向量 x 和行向量 y 所決定的網格。

- `yi=pchip (x,y,xi)`

  分段三次 Hermite 內插的內建函數，內插數據給定於向量 x、y 中，插值點給定於向量 xi。

- `Z=griddata (x,y,z,X,Y,'cubic')`

  對「散漫數據」，亦即數據不在網格點上，做內插的內建函數。矩陣 X、Y 由 meshgrid 所組成，以提供要插值的網格點。可以用 mesh (X,Y,Z) 畫出內插曲面。