Microsoft Visual Basic

第 2 部份

基礎數值解析

張 榮 興 博士

豐映科技股份有限公司

E-mail: chang.ronhsin@msa.hinet.net

Http://resi.com.tw/vb.htm

CHAPTER 3

數據插值法

(Interpolation)

張 榮 興 博士

豐映科技股份有限公司

E-mail: chang.ronhsin@msa.hinet.net

http://resi.com.tw/vh.htm

工程科學是一種對經驗及資料依賴性相當高的科學，工程師或研究人員時常需面對大量的圖表，讀取資料以便進行設計計算

Visual Basic

設計問題 D-III

P-$\hat{V}$-T 數據內插

基本資料

下表係取自 Perry [1] 化學工程師手冊，爲過熱態甲烷的體積 $\hat{V}$ (ft^3/lb) 與溫度及壓力的關係。

表 3.1　甲烷的 P-$\hat{V}$-T 關係

溫　度	壓力 (psia)						
(°F)	10	20	30	40	60	80	100
−200	17.15	8.47	5.57	4.12	2.678	1.954	1.518
−100	23.97	11.94	7.91	5.91	3.91	2.903	2.301
0	30.72	15.32	10.19	7.63	5.06	3.78	3.014
100	37.44	18.70	12.44	9.33	6.21	4.65	3.71
200	44.13	22.07	14.7	11.03	7.37	5.5	4.40
300	50.83	25.42	16.94	12.71	8.46	6.35	5.07
400	57.51	28.76	19.17	14.38	9.58	7.19	5.75
500	64.20	32.10	21.40	16.05	10.70	8.03	6.42

問題敘述

李明哲是一名服務於化學品公司的年輕化學工程師，他站立在一座大型甲烷貯存槽前，發現當時槽體溫度爲 56.4°F，槽壓 82.3 pisa。他希望計算槽內現存甲烷氣體的存量，但翻開「化學工程師手冊 (Perry’s Chemical Engineer’s Handbook)」，找到表 3.1，他卻不知道該如何正確的進行計算。請爲他寫一個小程式，使他能快速的求得甲烷現在的單位質量體積 $\hat{V}$ 值，以便計算儲存槽內的甲烷存量。

工程科學是一種對經驗及資料依賴性相當高的科學，工程師或研究人員時常需面對大量的圖、表，讀取資料，以便進行設計計算。表 3.1 的 P-$\hat{V}$-T 關係、蒸汽表、黏度與溫度關係表、熱傳導係數與溫度關係表、溫溼度表等，都是以類似的方法表現。但在實際使用時，表列值並不一定恰好是我們所要讀取的數值，例如，要從表 3.1 讀取溫度 56.4°F，壓力爲 82.3 psia 時的體積 $\hat{V}$，就必須借助於適當的近似方法。這種在表列數據範圍內求出近似值的方法稱爲插值法 (Interpolation)。若需藉表列數據求出表列數據範圍以外的近似函數值，則使用外推 (Extrapolation)近似法。

插值法利用數學語言描述，可簡述如下：假設存在一函數 F，爲自變數 x、y、z、……之函數；

$$F = \phi(x, y, z, \cdots) \tag{3-1}$$

此函數在某些特定點爲 $P_i\,(x_i, y_i, z_i, \cdots\cdots), i = 0, 1, 2, \cdots\cdots, n$ 的函數值爲已知；現在希望由這些點的函數關係，求一未知點 $P\,(x, y, z, \cdots\cdots)$ 之函數值。

爲了說明近似法的原理，首先我們假設此函數只有一個自變數 x，即函數關係爲 $y \equiv f(x)$；且已經知道有 $n+1$ 個已知點的函數對應關係，分別爲 $y_i = f(x_i), i = 0, 1, 2, \cdots\cdots, n$。通常，函數 $f(x)$ 可利用多項式、三角函數或指數函數等來表示。若以多項式來表示函數 $f(x)$，即

$$y = a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \cdots + a_1 x + a_0 \tag{3-2}$$

如果利用 $n+1$ 個已知的數據 $y_i = f(x_i), i = 0, 1, 2, \cdots\cdots, n$，代入以上的多項式，可以得到 $n+1$ 個線性聯立方程式，原則上應該可以求得 $n+1$ 個變數 $a_n, a_{n-1}, a_{n-2}, \cdots, a_1, a_0$ 的解。然後，利用所求得的近似方程式，即可求得在 $Max\,(x)$ 至 $Min\,(x)$ 範圍內其他各點的函數值。但是，如果數據點數很多的時候，這種處理方式將會變得非常費時，且計算所引進的誤差也將變得甚爲可觀，因此，實際上並不是一種有效率的方法。

本章將引介幾種常見的插值法，說明程式設計方法，並以實際工程問題說明其應用。

第一節　線性插值法

Visual Basic

假設函數在 $(x_i\,,\,f(x_i))$ 與 $(x_{i+1}\,,\,f(x_{i+1}))$ 間近似於成線性關係，如圖 3.1 所示。當 $\Delta h=(x_{i+1}-x_i)$ 很小時，或函數 $f(x)$ 變化較緩和時，假設線性函數關係通常是相當合理且誤差相當小的，此時，若希望找出介於 x_i 及 x_{i+1} 間的一個 x 值之函數值，則可利用以下直線方程式計算之：

$$f(x)=f(x_i)+\frac{f(x_{i+1})-f(x_i)}{x_{i+1}-x_i}(x-x_i) \qquad \textbf{(3-1.1)}$$

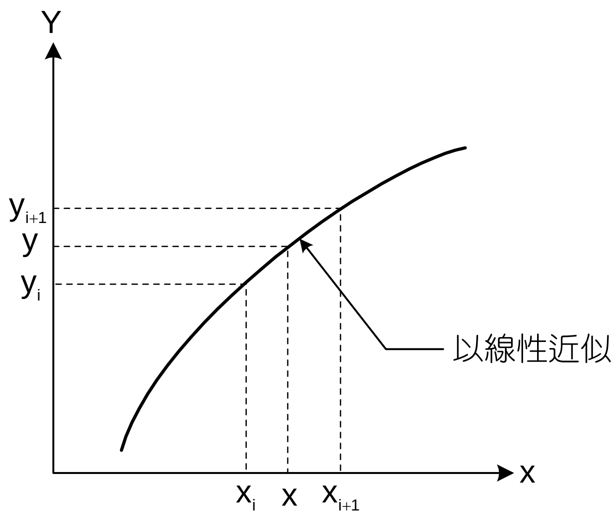


圖 3.1　線性插值法

例題 3-1　熱電偶溫度對照表之內插值

熱電偶是工程上一種相當常見的溫度測量元件。T 型熱電偶（銅及銅鎳合金）的溫度–電壓對照表，如表 3.2 所示 [2]。

實驗時，通常測得熱電偶的感應電動勢（或電壓，mV)，然後查表 3.2 以求得對應的溫度值。為了便於查對溫度，請製作一程式，可利用表 3.2 資料，查任何輸入 mV 值之對應溫度，但溫度顯示至小數以下一位即可（熱電偶之誤差通常約在 0.5℃ 左右）。

表 **3.2**　T 型熱電偶溫度 — 電壓對照表

溫度單位：°C　　　　　　電壓單位：mV　　　　　　參考點：0°C

°C	0	10	20	30	40	50	60	70	80	90	100
–200	–5.603	–5.753	–5.889	–6.007	–6.105	–6.181	–6.232	–6.258			
–100	–3.378	–3.656	–3.923	–4.177	–4.419	–4.648	–4.865	–5.069	–5.261	–5.439	–5.603
0	0	–0.383	–0.757	–1.121	–1.475	–1.819	–2.152	–2.475	–2.788	–3.089	–3.378
0	0	0.391	0.789	1.196	1.611	2.035	2.467	2.908	3.357	3.813	4.277
100	4.277	4.749	5.277	5.712	6.204	6.702	7.207	7.718	8.235	8.757	9.286
200	9.286	9.820	10.360	10.905	11.456	12.011	12.572	13.137	13.707	14.281	14.860
300	14.860	15.443	16.030	16.621	17.217	17.816	18.420	19.027	19.638	20.252	20.869
400	20.869										

TOP-DOWN 設計：

由於熱電偶的溫度 - 電壓函數關係相當接近線性關係，因此，利用線性插值法即可獲得滿意的結果。線性插值法的程式設計可用下示 TOP-DOWN 設計圖示表示之。

輸入 (x_1, y_1) 及 (x_2, y_2)

輸入 x

方程式 (3-1.1)　$y = y_1 + \dfrac{y_2 - y_1}{x_2 - x_1}(x - x_1)$

列印 y

繼續執行此直線函數之插值？

執行新直線函數之插值？

程式列印：

```
' LINEAR INTERPOLATION WITH DATA ENTRY FROM FILE
'
' PROGRAM DEVELOPED BY ENYA CHANG
' COPYRIGHT 2001
'
' CHEER
'
Sub LinearInterpolation(Xpos, Ypos)
Dim dig As Integer, Entry As String, Flag As Integer
Dim X(100) As Single, Y(100) As Single, N As Integer

Cls
Print "LINEAR INTERPOLATION"
dig = Val(InputBox("Number of decimal pts. req'd DIG = ", "DIG", 3, Xpos, Ypos))
digit = 10 ^ dig
'
' DATA ENTRY MODE SELECTION
        Entry = InputBox("Data Entry From File <Y/N> ", "SELECT ENTRY MODE", "Y", Xpos, Ypos)
                If Entry = "Y" Then
                Call DataEntryFromFile(N, X, Y, Xpos, Ypos)
        Else
                Call DataKeyIn(N, X, Y, Xpos, Ypos)
        End If
'
Cls
Print "=======Linear Interpolation======="
Do
        Xstar = Val(InputBox("INTERPOLATE: X = ", "X", 0, xpt, ypt))
        Call LookUpTable(N, X, Y, Xstar, Ystar, Flag)
        If Flag = 0 Then
                Print "X ="; Xstar; "          Y = "; Int(Ystar * digit + 0.5) / digit
        Else
                Print "X out of bound"
        End If
        LYN$ = InputBox("MORE POINTS ON THIS TABLE [Y/N] ", "YesNo", "Y", xpt, ypt)
Loop While LYN$ <> "N" And LYN$ <> "n"
'    CHEER 2001
End Sub


Private Sub DataKeyIn(N, X, Y, Xpos, Ypos)
Dim FileNo As Long, FileName As String
N = 0

Do
        N = N + 1
        Do
```

```
                Debug.Print "X,Y OF THE #", N, " POINT";
                X(N) = Val(InputBox("Enter X value of the point", "X", , Xpos, Ypos))
                Debug.Print "     X ="; X(N);
                Y(N) = Val(InputBox("Enter Y value of the point", "Y", , Xpos, Ypos))
                Debug.Print "     Y ="; Y(N)
                YN$ = InputBox("Are the data input correct?", "YesNo", "Y", Xpos, Ypos)
        Loop While YN$ <> "Y" And YN$ <> "y"
        YN$ = InputBox("Input Next Data?", "YesNo", "Y", Xpos, Ypos)
Loop While YN$ = "Y" Or YN$ = "y"

Cls
YN$ = InputBox("Save Data to File?", "YesNo", "Y", Xpos, Ypos)
If YN$ = "Y" Or YN$ = "y" Then
        FileNo = FreeFile
        FileName = InputBox("Enter File Name ", "FILE NAME", "Example301.dat", Xpos, Ypos)
        Open FileName For Output As #FileNo
                For I = 1 To N
                        Print #FileNo, X(I), Y(I)
                Next I
        Close #FileNo
End If
End Sub


Private Sub DataEntryFromFile(N, X, Y, Xpos, Ypos)
Dim FileNo As Long, FileName As String
FileNo = FreeFile
FileName = InputBox("Enter File Name for Data Input", "FILE NAME", "Example301.dat", Xpos, Ypos)

Open FileName For Input As #FileNo
        N = 0
        Do While Not EOF(FileNo)
                N = N + 1
                Input #FileNo, X(N), Y(N)
                Debug.Print N, X(N), Y(N)
        Loop
Close #FileNo
End Sub


Sub LookUpTable(N, X, Y, Xstar, Ystar, Flag)
Flag = 0
'
' Check to see if Xstar lies within the scope
' of the tabulated values, X(1) ... X(N)
If Xstar < X(1) Or Xstar > X(N) Then
        Flag = 1
        Exit Sub
```

```
Else
        I = 1
        Do
'
'               Search to find two successive entries
                If (X(I) > Xstar) Or (I >= N) Then
'
'                       Linear Interpolation to find corresponding value of Y
                        Ystar = Y(I - 1) + (Xstar - X(I - 1)) * (Y(I) - Y(I - 1)) / (X(I) - X(I - 1))
                        Exit Sub
                Else
                        I = I + 1
                End If
        Loop
End If
End Sub


Private Sub Start_Click()
Xpos = 7500
Ypos = 6000
Call LinearInterpolation(Xpos, Ypos)
End Sub
```

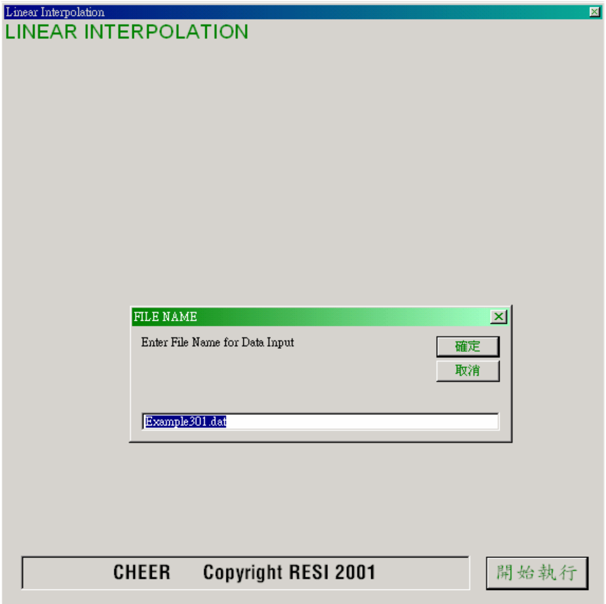
符號說明：

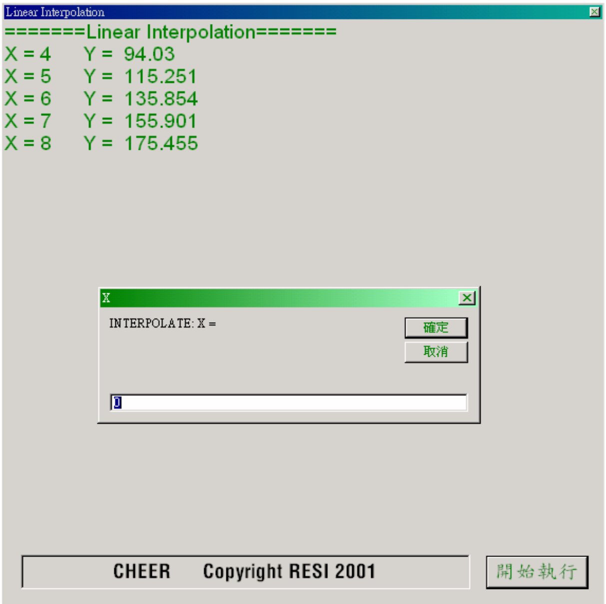
N：　數據點數
X：　熱電偶之 mV 值
Xstar：內插位置
Y：　溫度
DIG：小數點位數

副程式使用說明：

1. 主要副程式 **Sub LookUpTable (N, X, Y, Xstar, Ystar, Flag)** 的使用方法如下：
   (1) 宣告數據點 (X, Y)、數據點數 N 及內差點的 Xstar 值
   (2) 執行副程式 Call LookUpTable (N, X, Y, Xstar, Ystar, Flag)
   (3) 得到結果爲 Ystar 及錯誤旗幟 Flag
2. 副程式 **Private Sub DataKeyIn (N, X, Y, Xpos, Ypos)** 用於輸入數據及存檔。
3. 副程式 **Private Sub DataEntryFromFile (N, X, Y, Xpos, Ypos)** 由檔案讀取數據。

測試數據與執行結果：





線性內插法利用 Excel 撰寫程式更爲方便，只要將 (x_1, y_1) 及 (x_2, y_2) 分別定義於 Excel 表格中，利用方程式 (3-1.1) 及 (x_1, y_1) 及 (x_2, y_2) 對應之位置，可非常容易寫成程式。但是，讀者可以利用本程式了解數據輸入及建檔的方法，以利其他應用之需。

第二節　拉格蘭奇內插法

Visual Basic

任何一個 n 次多項式函數都可以利用拉格蘭奇多項式 (Lagrange Polynomial) 表示，因此，若有已知的 $n+1$ 個數據點，可利用拉式多項式作近似，以求得中間任何一點之數值，這種方法即稱爲拉格蘭奇內插法 (Lagrange Interpolation)。

根據前述，若有已知的 $n+1$ 個數據點 $(x_0\,,y_0)\,,(x_1\,,y_1)\,,\cdots,(x_n\,,y_n)$ ，則我們可找到一個 n 次多項式通過這些點，亦即函數 $y=f(x)$ 可以表示成以下的多項式：

$$y=\sum_{i=0}^{n} a_i\,x^i = a_n x^n + a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \ldots + a_1 x + a_0 \qquad \textbf{(3-2.1)}$$

由於 y 爲 x 之 n 次多項式，因此，可將 y 寫成 $n+1$ 個 x 的 n 次多項式 $L_i(x)$ 之線性組合：

$$y=\sum_{i=0}^{n} L_i(x)\,y_i \qquad \textbf{(3-2.2)}$$

其中假設多項式 $L_i(x)$ 均爲 x 之 n 次多項式。且假設正交函數 $L_i(x)$ 除了在第 i 點以外，其餘各點處其函數值均爲零；且 $L_i(x)$ 在第 i 點的函數值爲 1，亦即

$$L_i(x_j)=\begin{cases}0 & i\neq j\\ 1 & i=j\end{cases} \qquad \textbf{(3-2.3)}$$

由於 $y=f(x)$，考慮在第 k 點時，若將方程式 (3-2.3) 代入 (3-2.2)，得到

$$y=f(x_k)=\sum_{i=0}^{n} L_i(x_k)\,y_i = L_k(x_k)\,y_k = y_k \qquad \textbf{(3-2.4)}$$

其中 k 可以是 0 至 n 的任一自然數，故可知方程式 (3-2.2) 可通過所有的已知數據點。

由於 $L_i(x)$ 爲一 n 次多項式，且由方程式 (3-2.3)，可以知道由 $x_0\,,x_1\,,\cdots\cdots$ 至 x_n 的 $n+1$ 個點中，除了 x_i 以外，其餘的 x_j 值都是方程式 $L_i(x)=0$ 之根，故可假設 $L_i(x)$ 爲一常數係數 A_i 與 $(x-x_0)\,,(x-x_1)\,,\cdots\cdots,(x-x_{i-1})\,,(x-x_{i+1})\,,\cdots\cdots,(x-x_n)$ 之乘積。

$$L_i(x)=A_i(x-x_0)(x-x_1)\cdots(x-x_{i-1})(x-x_{i+1})\cdots(x-x_n) \qquad \textbf{(3-2.5)}$$

$$L_i(x)=A_i\prod_{\substack{j=0\\ j\neq i}}^{n}(x-x_j)$$

由於 $L_i(x_i)=1$，代入上式，經整理後可以得到

$$A_i=\frac{1}{\prod_{\substack{j=0\\j\neq i}}^{n}(x_i-x_j)}$$

$$=\frac{1}{(x_i-x_0)(x_i-x_1)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_n)} \qquad \textbf{(3-2.6)}$$

故由方程式 (3-2.5) 及 (3-2.6) 整理後，可將拉格蘭奇多項式 $L_i(x)$ 寫成

$$L_i(x)=\prod_{\substack{j=0\\j\neq i}}^{n}\frac{(x-x_j)}{(x_i-x_j)} \qquad \textbf{(3-2.7)}$$

將上式代入方程式 (3-2.2)，可得拉格蘭奇內插方程式為

$$y=\sum_{i=0}^{n}\left\{y_i\cdot\left[\prod_{\substack{j=0\\j\neq i}}^{n}\frac{(x-x_j)}{(x_i-x_j)}\right]\right\} \qquad \textbf{(3-2.8)}$$

利用上式，由輸入的 $n+1$ 組數據點 $(x_0,y_0),(x_1,y_1),\cdots\cdots,(x_n,y_n)$，代入上式即自動產生一個 n 次多項式。將任何 x 值代入此多項式中，即可求得對應的函數 y 值。

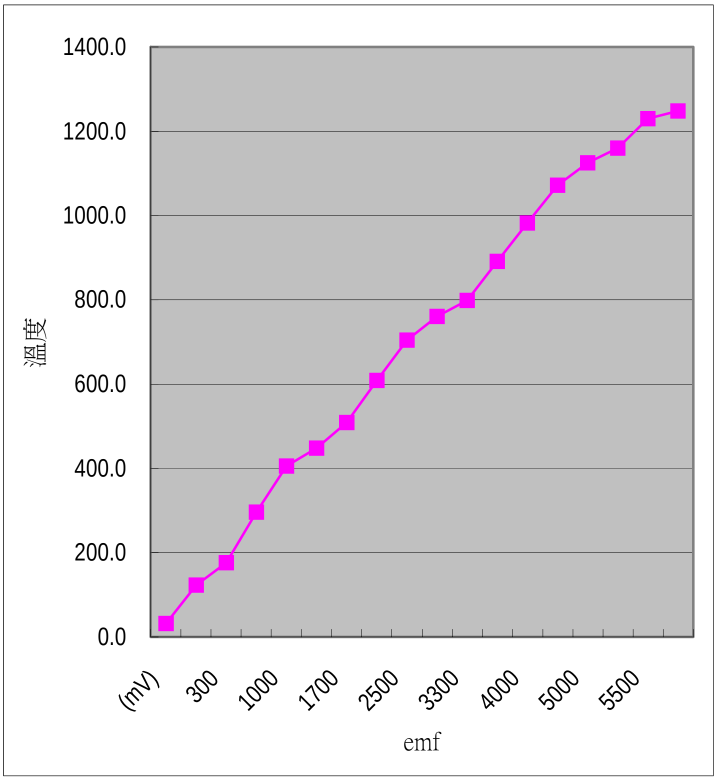
拉格蘭奇內插法 (3-2.8) 由於本身運算重複性高，相當容易編寫成計算機程式，但是其缺點是每次計算都需重複作相同項次之計算，所需時間較長。此外，如果設定多項式的階次過高，有時會使函數變成波動形式，而使內插結果產生嚴重誤差，因此，使用上應該要小心。

例題 3-2　S 型熱電偶溫度對照表之插值

例題 3-1 中所述之 T 型熱電偶溫度，在低溫用途上，準確度相當高，但是在高溫用途上，則以選用鉑 - 銠合金熱電偶為宜。工業上常使用的 S 型熱電偶，是利用鉑 - 銠 (10%) 合金及白金線所構成，適用溫度為 -50°C 至 1760°C。表 3.3 為 S 型熱電偶之溫度對照表，請利用拉格蘭奇內插法製作程式，以便由輸入電壓 (mV) 可內插求得對應的溫度。

表 **3.3** S 型熱電偶溫度對照表

emf	溫度
(mV)	(°F)
0	32.0
300	122.4
500	176.0
1000	296.4
1500	405.7
1700	447.6
2000	509.0
2500	608.4
3000	704.7
3300	761.4
3500	799.0
4000	891.9
4500	983.0
5000	1072.6
5300	1125.7
5500	1160.8
5900	1230.3
6000	1247.5



TOP-DOWN 設計：

主　程　式

INPUT AND PRINT DATA	
	INPUT X, DEG, MIN
	CALL LAGRANGE
	PRINT X, DEG, MIN, Y
REPEAT?	

Lagrange 副程式

FAC = 1.0
@calculate $\prod_{\substack{j=\min \\ j\neq i}}^{\max}(x-x_j)$
FOR J = MIN TO MAX

IF (X ≠ X(J))

THEN: FAC = FAC*(X-X(J))

ELSE: Y = Y(J)

RETURN

@Calculate $y=\sum_{i=0}^{n}\left\{y_i\cdot\left[\prod_{\substack{j=0\\ j\neq i}}^{n}\frac{(x-x_j)}{(x_i-x_j)}\right]\right\}$

FOR I = MIN TO MAX

TERM = Y(I) * FAC/(X-X(I))

FOR J = MIN TO MAX

IF (I≠J)

THEN: TERM = TERM/(X(I)-X(J))

ELSE

YEST = YEST + TERM

YLARG = YEST

RETURN

程式列印：

```
Sub LagrangeInterpolation(Xpos, Ypos)
Dim X(100), Y(100) As Single
Dim N As Integer, Min As Integer, Deg As Integer

Rem    *************READ N,X,Y
Cls
Print "LAGRANGE INTERPOLATION"
'
' DATA ENTRY MODE SELECTION
    Entry = InputBox("Data Entry From File <Y/N> ", "SELECT ENTRY MODE", "Y", Xpos, Ypos)
        If Entry = "Y" Then
        Call DataEntryFromFile(N, X, Y, Xpos, Ypos)
    Else
        Call DataKeyIn(N, X, Y, Xpos, Ypos)
    End If
'
Rem **************CALL LAGRANGE***************
Cls
Print "XI", "DEG", "MIN", "YINTER"

Do
    XI = Val(InputBox("INTERPOLATING ARGUMENT XI = ", "XI", 0, Xpos, Ypos))
    Deg = Val(InputBox("DEG OF INTERPOLATING POLYNOMIAL = ", "DEG", Deg, Xpos, Ypos))
    Min = Val(InputBox("SMALLEST SUBSCRIPT = ", "MIN", Min, Xpos, Ypos))
    Call LAGRANGE(N, X, Y, XI, Min, Deg, YLAG)
```

```
        Print XI, Deg, Min, YLAG
        YN$ = InputBox("Next Interpolation? <Y/N>", "YN$", "Y", Xpos, Ypos)
Loop While YN$ <> "n" And YN$ <> "N"

'   CHEER 2001
End Sub


Private Sub DataKeyIn(N, X, Y, Xpos, Ypos)
Dim FileNo As Long, FileName As String
N = 0
Do
        N = N + 1
        Do
                Debug.Print "X,Y OF THE #", N, " POINT";
                X(N) = Val(InputBox("Enter X value of the point", "X", , Xpos, Ypos))
                Debug.Print "    X ="; X(N);
                Y(N) = Val(InputBox("Enter Y value of the point", "Y", , Xpos, Ypos))
                Debug.Print "    Y ="; Y(N)
                YN$ = InputBox("Are the data input correct?", "YesNo", "Y", Xpos, Ypos)
        Loop While YN$ <> "Y" And YN$ <> "y"
        YN$ = InputBox("Input Next Data?", "YesNo", "Y", Xpos, Ypos)
Loop While YN$ = "Y" Or YN$ = "y"

Cls
YN$ = InputBox("Save Data to File?", "YesNo", "Y", Xpos, Ypos)
If YN$ = "Y" Or YN$ = "y" Then
        FileNo = FreeFile
        FileName = InputBox("Enter File Name ", "FILE NAME", "Example302.dat", Xpos, Ypos)
        Open FileName For Output As #FileNo
                For I = 1 To N
                        Print #FileNo, X(I), Y(I)
                Next I
        Close #FileNo
End If
End Sub


Private Sub DataEntryFromFile(N, X, Y, Xpos, Ypos)
Dim FileNo As Long, FileName As String
FileNo = FreeFile
FileName = InputBox("Enter File Name ", "FILE NAME", "Example302.dat", Xpos, Ypos)
Open FileName For Input As #FileNo
        N = 0
        Do While Not EOF(FileNo)
                N = N + 1
                Input #FileNo, X(N), Y(N)
                Debug.Print N, X(N), Y(N)
```

```
        Loop
Close #FileNo
End Sub


Sub LAGRANGE(N, X, Y, XI, Min, Deg, YLAG)
'
' Lagrange Interpolation Subroutine
'
'      N        = Number of data sets
'      X,Y              = Data set
'      XI               = X value for Interpolation
'      MIN      = Smallest subscript for interpolating polynomial
'      DEG      = Degree of the interpolating polynomial
'      YLAG     = Function value of Lagrange Interpolation
'
FAC = 1!
Max = Min + Deg
Iflag = 0

For J = (Min) To (Max)
        If (XI = X(J)) Then
                YLAG = Y(J)
                Iflag = 1
                Exit For
        Else
                FAC = FAC * (XI - X(J))
        End If
Next J

If Iflag = 0 Then
        YEST = 0
        For I = (Min) To (Max)
                TERM = Y(I) * FAC / (XI - X(I))
                For J = (Min) To (Max)
                        If (I <> J) Then TERM = TERM / (X(I) - X(J))
                Next J
                YEST = YEST + TERM
        Next I
        YLAG = YEST
End If
' CHEER 2001
End Sub
```

符號說明：

DEG： 多項式之次數
Min： 用於決定多項式基準點之最小註標
N： 原始數據點數
X： 原始數據點之 X 值
XI： 內插位置
Y： 原始數據點之 Y 值
YLAG： 內插所得函數值
TERM： 拉格蘭奇多項式之單項值
YEST： 內插函數值，運算時使用

副程式使用說明：

1. **副程式** Sub LAGRANGE (N, X, Y, XI, MIN, DEG, YLAG) **使用方法如下：**

(1) 宣告 N, X, Y, XI, MIN, DEG

(2) Call LAGRANGE (N, X, Y, XI, MIN, DEG, YLAG)

(3) 結果為 YLAG

測試結果：

將表 3.3 的數據輸入，然後選擇內插位置 XI，並指定多項式之次數 DEG，及用於決定多項式基準點之最小註標 MIN。所得測試結果如下：

Lagrange Interpolation

XI	DEG	MIN	YINTER
300	1	1	122.4
300	2	1	122.4
300	3	1	122.4
300	4	1	122.4
1700	1	4	449.42001953125
1700	2	3	446.312026367188
1700	2	4	447.6
1700	3	3	447.6
1700	4	2	447.6
1700	4	3	447.6
3300	1	7	767.440063476563
3300	3	6	761.263948974609
3300	4	5	760.777486083985
3300	4	6	761.4
3300	5	5	761.4
3300	6	4	761.4

CHEER Copyright RESI 2001 開始執行

結果討論：

由於真正的函數關係 $y = f(x)$ 並不知道，因此，無法估計執行時誤差之上限。但是，由測試數據與原表列值比較，顯示利用 2 次以上的多項式，且內插位置位於中央者，所得結果與實驗值相當接近。在末端附近所得結果最差，原因是函數在末端處曲率較大，而解無法利用高階多項式並使 x 位於中央位置。

雖然上列測試數據 x 值取等間距，但事實上本程式之使用並不受此限制。使用本程式時，輸入數據採漸增或漸減排列，可得較高準確度。

第三節　三次弧線內插法

Visual Basic

線性內插法執行速度雖然快速，但只適用於曲率非常小的函數或極小的區域範圍作內插。拉格蘭奇內插法雖然可以自在地調節多項式次數，以得到適當的準確度，但這種內插法最大的缺點是每次作內插計算時，均需一再的重複計算 $L_i(x)$ 值，因此，執行效率較差。

對於一個連續而且可微分的函數 $f(x)$，假設我們希望在封閉區間 [a, b] 內，利用低次多項式，以階段方式來作爲它的近似函數，理論上應該也可以得到相當好的結果。

假設已知數據基準點 x_i 的範圍爲 $a = x_0 < x_1 < \cdots\cdots < x_{n-1} < x_n = b$，其對應的每一個點 x_i 的函數值分別爲 $y_i = f(x_i), i = 0, 1, \cdots, n$。在 [a, b] 區間內，假設存在 n 個低次多項式近似函數爲 $S_X(Y, x)$，其中 $X = [x_0, x_1, \cdots, x_n]^T$，且 $Y = [y_0, y_1, \cdots, y_n]^T$。則我們希望在 [a, b] 內之任何 x 值，多項式近似函數 $S_X(Y, x)$ 的一次及二次導函數（微分）均爲連續函數，並滿足以下 $n+1$ 個條件。

$$S_X(Y, x) = f(x_i) = y_i \qquad i = 0, 1, \cdots, n \tag{3-3.1}$$

在 [a, b] 所分割成的每一個小區間 $[x_i, x_{i+1}], i = 0, 1, \cdots, n-1$ 中，假設 $S_X(Y, x)$ 都等於三次多項式 $P_{3,i}(x)$，其中下註標 3 表示三次多項式，因此，其二次導函數 $P''_{3,i}(x)$ 即爲一線性方程式，可以寫成 $P''_{3,i}(x_i)$ 及 $P''_{3,i}(x_{i+1})$ 的線性組合：

$$P''_{3,i}(x) = \frac{(x_{i+1} - x)}{h_i} P''_{3,i}(x_i) + \frac{(x - x_i)}{h_i} P''_{3,i}(x_{i+1}) \qquad i = 0, 1, 2, \cdots, n-1 \tag{3-3.2}$$

其中 $h_i \equiv x_{i+1} - x_i$；且 $P''_{3,i}(x_i)$ 及 $P''_{3,i}(x_{i+1})$ 均為常數。將方程式 (3-3.2) 積分兩次，得到

$$P_{3,i}(x) = \frac{P''_{3,i}(x_i)}{6h_i}(x_{i+1} - x)^3 + \frac{P''_{3,i}(x_{i+1})}{6h_i}(x - x_i)^3 + C_1 x + C_2 \qquad i = 0,1,2,\cdots\cdots,n-1 \tag{3-3.3}$$

其中 C_1, C_2 為積分常數。由於 $P_{3,i}(x)$ 通過 (x_i, y_i) 及 (x_{i+1}, y_{i+1}) 兩點，故

$$\left.\begin{aligned} P_{3,i}(x_i) &= y_i \\ P_{3,i}(x_{i+1}) &= y_{i+1} \end{aligned}\right\} \qquad i = 0,1,2,\cdots,n-1 \tag{3-3.4}$$

將上式代入方程式 (3-3.3)，求解 C_1, C_2，經整理後，得到 $P_{3,i}(x)$ 為

$$\begin{aligned} P_{3,i}(x) = {} & \frac{P''_{3,i}(x_i)}{6h_i}(x_{i+1} - x)^3 + \frac{P''_{3,i}(x_{i+1})}{6h_i}(x - x_i)^3 \\ & + \left[\frac{y_{i+1}}{h_i} - \frac{h_i\,P''_{3,i}(x_{i+1})}{6}\right](x - x_i) + \left[\frac{y_i}{h_i} - \frac{h_i\,P''_{3,i}(x_i)}{6}\right](x_{i+1} - x) \\ & i = 0,1,2,3,\cdots\cdots,n-1 \end{aligned} \tag{3-3.5}$$

由於前述 $S_X(Y, x)$ 之一次導函數為一連續函數，亦即

$$P'_{3,i}(x_i) = P'_{3,i-1}(x_i) \qquad i = 1,2,\cdots,n-1 \tag{3-3.6}$$

將方程式 (3-3.5) 對 x 作一次微分，並利用上式條件加以整理以後，可以得到以下 $n-1$ 個差分方程式：

$$\begin{aligned} & \frac{h_{i-1}}{h_i} P''_{3,i-1}(x_{i-1}) + \frac{2(h_1 + h_{i-1})}{h_i} P''_{3,i}(x_i) + P''_{3,i+1}(x_{i+1}) \\ & \quad \equiv \frac{6}{h_i}\left[\frac{(y_{i+1} - y_i)}{h_i} - \frac{(y_i - y_{i-1})}{h_{i-1}}\right] \\ & \quad i = 1,2,\cdots,n-1 \end{aligned} \tag{3-3.7}$$

將方程式 (3-3.7) 改寫成矩陣形式，可以得到

$$\begin{bmatrix} h_0 & 2(h_0+h_1) & h_1 & & & & \\ & h_1 & 2(h_1+h_2) & h_2 & & & \\ & & h_2 & 2(h_2+h_3) & h_3 & & \\ & & & \cdots\cdots & & & \\ & & & & \cdots\cdots & & \\ & & & & h_{n-2} & 2(h_{n-2}+h_{n-1}) & h_{n-1} \end{bmatrix} \begin{bmatrix} P_0'' \\ P_1'' \\ P_2'' \\ \vdots \\ \vdots \\ P_n'' \end{bmatrix}$$

$$= 6 \begin{bmatrix} \frac{y_2-y_1}{h_1} - \frac{y_1-y_0}{h_0} \\ \frac{y_3-y_2}{h_2} - \frac{y_2-y_1}{h_1} \\ \frac{y_4-y_3}{h_3} - \frac{y_3-y_2}{h_2} \\ \vdots \\ \vdots \\ \frac{y_n-y_{n-1}}{h_{n-1}} - \frac{y_{n-1}-y_{n-2}}{h_{n-2}} \end{bmatrix} \tag{3-3.8}$$

或可以寫成

$$\underline{\underline{A}}_{(n-1)x(n+1)} \quad \underline{\underline{S}}_{(n+1)x1} = \underline{\underline{Y}}_{(n-1)x1} \tag{3-3.9}$$

方程式 (3-3.9) 中， $P_0'', P_1'', \cdots\cdots, P_n''$ 共有 $n+1$ 個未知數，但總共只能寫出 $n-1$ 個方程式。因此，必須再加上兩個已知條件，方程式 (3-3.9) 才能解出所有的未知數。假設所加入的兩個已知條件爲端點條件 P_0'' 及 P_n''； 通常使用的端點條件可分爲三類：

1. 假設三次弧線近似方程式於接近兩端點時，趨近於直線 (即二次微分等於零)，即

$$P_0'' = 0 \ , \ P_n'' = 0$$

2. 假設三次弧線近似方程式於接近兩端點時，趨近於拋物線 (即二次微分等於定值)，即

$$P_0'' = P_1'' \ , \ P_n'' = P_{n-1}''$$

3. 假設三次弧線近似方程式於接近兩端點時，其二次微分可用往內二點的二次微分作線性外插。即 P_0'' 爲 P_1'' 及 P_2'' 之線性外插， P_n'' 爲 P_{n-1}'' 及 P_{n-2}'' 之線性外插。即

$$P_0'' = P_1'' + (h_0/h_1)(P_1'' - P_2'') \ , \ P_n'' = P_{n-1}'' + (h_{n-1}/h_{n-2})(P_{n-1}'' - P_{n-2}'')$$

以第一種端點條件為例，$P_0''=0, P_n''=0$，代入方程式 (3-3.8)，則第一個方程式及最後一個方程式分別變成：

$$2(h_0+h_1)P_1''+h_1P_2''=6\left[\frac{(y_2-y_1)}{h_1}-\frac{(y_1-y_0)}{h_0}\right] \quad \textbf{(3-3.10)}$$

$$h_{n-2}P_{n-2}''+2(h_{n-2}+h_{n-1})P_{n-1}''=6\left[\frac{(y_n-y_{n-1})}{h_{n-1}}-\frac{(y_{n-1}-y_{n-2})}{h_{n-2}}\right] \quad \textbf{(3-3.11)}$$

由於 $P_0''=0, P_n''=0$，在上式中已自然消去，方程式 (3-3.8) 中 $\underline{\underline{S}}_{(n+1)x1}$ 可去掉第一項及第 $n+1$ 項，成為 $\underline{\underline{S}}_{(n-1)x1}$；因此，係數矩陣 $\underline{\underline{A}}_{(n-1)X(n+1)}$ 可改寫成方陣 $\underline{\underline{A}}_{(n-1)X(n-1)}$：

$$\underline{\underline{A}}=\begin{bmatrix} 2(h_0+h_1) & h_1 & & & \\ h_1 & 2(h_1+h_2) & h_2 & & \\ & h_2 & 2(h_2+h_3) & h_3 & \\ & & \vdots & \vdots & \\ & & & h_{n-2} & 2(h_{n-2}+h_{n-1}) \end{bmatrix} \quad \textbf{(3-3.12)}$$

同理，利用第二種端點條件，可以整理得到方陣 $\underline{\underline{A}}_{(n-1)X(n-1)}$：

$$\underline{\underline{A}}=\begin{bmatrix} (3h_0+2h_1) & h_1 & & & \\ h_1 & 2(h_1+h_2) & h_2 & & \\ & h_2 & 2(h_2+h_3) & h_3 & \\ & & \vdots & \vdots & \\ & & & h_{n-2} & (2h_{n-2}+3h_{n-1}) \end{bmatrix} \quad \textbf{(3-3.13)}$$

利用第三種端點條件，可以整理得到方陣 $\underline{\underline{A}}_{(n-1)X(n-1)}$：

$$\underline{\underline{A}}=\begin{bmatrix} \frac{(h_0+h_1)(h_0+2h_1)}{h_1} & \frac{h_1^2-h_0^2}{h_1} & & & \\ h_1 & 2(h_1+h_2) & h_2 & & \\ & h_2 & 2(h_2+h_3) & h_3 & \\ & & \vdots & \vdots & \\ & & & \frac{h_{n-2}^2-h_{n-1}^2}{h_{n-2}} & \frac{(h_{n-1}+h_{n-2})(h_{n-1}+2h_{n-2})}{h_{n-2}} \end{bmatrix}$$

(3-3.14)

由於係數矩陣 $\underline{\underline{A}}_{(n-1)X(n-1)}$ 均成為三對角線矩陣 (Tridiagonal Matrix)，因此，求解及儲存都相當方便，其程式設計方法可以參考本書第四章有詳細討論。三次弧線內插法的程式設計請參考本書例題 3.4，執行方法歸納如下：

1. 先定義端點條件。

2. 求解方程式 (3-3.9) 得到 $S_X(Y, x)$。

3. 再利用方程式 (3-3.5) 可進行內插計算。

例題 3-3 三次弧線近似法

試利用三次弧線法表示下列數據。表中數據真正關係式為 $y = 2x^3 - 5$

表 3.4

I	X (I)	Y (I)
0	0	−5
1	1	−3
2	2	11
3	3	49
4	4	123

解：

利用端點條件 1：$P_0'' = 0, P_n'' = 0$，代入方程式 (3-3.9) 得到

$$\begin{bmatrix} 4 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 4 \end{bmatrix}\begin{bmatrix} S_1 \\ S_2 \\ S_3 \end{bmatrix} = \begin{bmatrix} 72 \\ 144 \\ 216 \end{bmatrix}$$

故得到 $S_0 = 0$, $S_1 = \dfrac{90}{7}$, $S_2 = \dfrac{144}{7}$, $S_3 = \dfrac{342}{7}$, $S_4 = 0$。

利用端點條件 2：$P_0'' = P_1'', P_n'' = P_{n-1}''$，則聯立方程式為

$$\begin{bmatrix} 5 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 5 \end{bmatrix}\begin{bmatrix} S_1 \\ S_2 \\ S_3 \end{bmatrix} = \begin{bmatrix} 72 \\ 144 \\ 216 \end{bmatrix}$$

故得到 $S_0 = \dfrac{48}{5}$, $S_1 = \dfrac{48}{5}$, $S_2 = 24$, $S_3 = \dfrac{192}{5}$, $S_4 = \dfrac{192}{5}$。

利用端點條件 3：P_0'' 為 P_1'' 及 P_2'' 之線性外插，P_n'' 為 P_{n-1}'' 及 P_{n-2}'' 之線性外插，則聯立方程式為

$$\begin{bmatrix} 6 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 6 \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \\ S_3 \end{bmatrix} = \begin{bmatrix} 72 \\ 144 \\ 216 \end{bmatrix}$$

故得到 $S_0 = 0$, $S_1 = 12$, $S_2 = 24$, $S_3 = 36$, $S_4 = 48$ 。

將原方程式 $y = 2x^3 - 5$ 微分兩次，得 $y'' = 12x$ ，將 x 值代入，恰與利用端點條件 3 所得結果一致。

例題 3-4 石墨之熱容量

表 3.5 爲石墨之熱容量與溫度之關係，其中溫度單位爲 °K ，熱容量單位爲 Cal/°K - mole。試利用三次弧線內插法寫一程式，以便求得 500°C 時的熱容量。

符號說明：

X (I), Y (I)： 原始數據
N： 數據點數
A： 方程式(3-3.9)中的係數矩陣
S： 二次導函數向量
IE： 端點條件 (= 1、2 或 3)

表 3.5 石墨之熱容量

(T (°K)	Cp (Cal/°K - mole)
300	2.08
400	2.85
500	3.50
600	4.03
700	4.43
800	4.75
900	4.98
1000	5.14
1100	5.27
1200	5.42

TOP-DOWN 設計：

主　程　式

<table>
<tr><td colspan="5">輸入及列印數據</td></tr>
<tr><td colspan="5">CALL　CubicSpline　[計算 $S_X(Y, x)$]</td></tr>
<tr><td rowspan="5"></td><td>@ MOD #1 計算單一內插值</td><td rowspan="5"></td><td colspan="2">@ MOD#2 計算並建立內插表格</td></tr>
<tr><td>輸入 X 值</td><td rowspan="3"></td><td>輸入 ΔX</td></tr>
<tr><td rowspan="3">計算內插值
CALL POLY</td><td>計算內插值</td></tr>
<tr><td>$X = X + \Delta X$</td></tr>
<tr><td colspan="2">重複計算至 $X = X_{max}$</td></tr>
<tr><td colspan="5">重複執行</td></tr>
</table>

副程式 **CubicSpline**

<table>
<tr><td colspan="2">@建立係數矩陣</td></tr>
<tr><td colspan="2">DO FOR I = 1 TO N-2</td></tr>
<tr><td rowspan="4"></td><td>$A(I, 1) = h_i$</td></tr>
<tr><td>$A(I, 2) = 2(h_i + h_{i+1})$</td></tr>
<tr><td>$A(I, 3) = h_{i+1}$</td></tr>
<tr><td>$A(I, 4) = 6[(y_{i+1} - y_i)/h_i - (y_i - y_{i-1})/h_{i-1}]$</td></tr>
<tr><td colspan="2">@調節第一及最後一項</td></tr>
<tr><td colspan="2">CALL　ChangeConditions</td></tr>
<tr><td colspan="2">@解 $\underline{\underline{A}}\ \ \underline{\underline{S}} = \underline{\underline{Y}}$</td></tr>
<tr><td colspan="2">CALL　TRID</td></tr>
<tr><td colspan="2">DO FOR I = 1 TO N-2</td></tr>
<tr><td></td><td>S (I+1) = A (I, 4)</td></tr>
<tr><td colspan="2">CALL　S1SN　[取得 S1 及 SN]</td></tr>
</table>

副程式 **POLY**（ 計算內插值 ）

<table>
<tr><td>@計算多項式值，求得內插值</td></tr>
<tr><td>$$y = \frac{S(k)}{6h_k}(x_{k+1} - x)^3 + \frac{S(k+1)}{6h_k}(x - x_k)^3 + \left[\frac{y_{k+1}}{h_k} - \frac{h_k S(k+1)}{6}\right](x - x_k) + \left[\frac{y_k}{h_k} - \frac{h_k S(k)}{6}\right](x_{k+1} - x)$$</td></tr>
</table>

副程式 **ChangeConditions (** 調節第一及最後一項 **)**

@將 $\underline{\underline{A}}_{(N-1)X(N+1)}$ 轉變成 $\underline{\underline{A}}_{(N-1)X(N-1)}$			
SELECT CASE IE			
	CASE 1	CASE 2	CASE 3
		$A(1,2)=A(1,2)+h_1$ $A(n-2,2)=A(n-2,2)+h_{n-1}$	$A(1,2)=\dfrac{(h_1+h_2)(h_1+2h_2)}{h_2}$ $A(1,3)=(h_2^2-h_1^2)/h_2$ $A(n-2,1)=\dfrac{h_{n-2}^2-h_{n-1}^2}{h_{n-2}}$ $A(n-2,2)=\dfrac{(h_{n-1}+h_{n-2})(h_{n-1}+2h_{n-2})}{h_{n-2}}$

副程式 **S1SN (** 取回 **S (1)** 及 **S (N))**

@利用端點條件，取得 $S(1)$, $S(N)$			
SELECT CASE IE			
	CASE 1	CASE 2	CASE 3
	$S(1)=0$ $S(n)=0$	$S(1)=S(2)$ $S(n)=S(n-1)$	$S(1)=\dfrac{(h_1+h_2)\,S(2)-h_1S(3)}{h_2}$ $S(n)=\dfrac{(h_{n-1}+h_{n-2})\,S(n-1)-h_{n-1}S(n-2)}{h_{n-2}}$

副程式 **TRID (** 解三項方程式，詳第四章說明 **)**

DO FOR I = 2 TO N-2	
	$A(I,2)=A(I,2)-\dfrac{A(I,1)}{A(I-1,2)}\cdot A(I-1,3)$
	$A(I,4)=A(I,4)-\dfrac{A(I,1)}{A(I-1,2)}\cdot A(I-1,4)$
$A(N-2,4)=A(N-2,4)/A(N-2,2)$	
DO FOR I = 2 TO N-2	
	J = (N-1)-1
	$A(J,4)=\dfrac{[A(J,4)-A(J,3)\cdot A(J+1,4)]}{A(J,2)}$
RETURN	

程式列印：

```
'
'    * CUBIC SPLINE INTERPOLATION *
'
Sub CubicSplineInterpolation(xpos, ypos)

Dim X(100), Y(100), S(100), A(100, 4)
Print "*** CUBIC SPLINE INTERPOLATION ***"
IDFlag = 1
Do
        Cls
        If IDFlag = 1 Then
                Call OriginalDataEntry(N, X, Y, xpos, ypos)
        End If
        If IDFlag <> 3 Then
                Call EndConditions(IdEnd, xpos, ypos)
                Call CubicSpline(IdEnd, N, X, Y, A, S)
        End If
        Call InterpolationMode(IntMode, xpos, ypos)
        Select Case IntMode
                Case 1
                        Call FromX2Y(N, S, X, Y, xpos, ypos)
                        IDFlag = 3
                Case 2
                        Call IncrementX2Y(N, S, X, Y, xpos, ypos)
                        IDFlag = 3
                Case 3
                        IDFlag = 1
                Case 4
                        IDFlag = 2
                Case 5
                        IDFlag = 0
              End Select
Loop While IDFlag <> 0
End Sub


Sub OriginalDataEntry(N, X, Y, xpos, ypos)

Cls
Print "ENTER EACH DATA POINT AS X,Y"
Print
Do
N = 0
        Do
                N = N + 1
                        Do
                                Print "POINT #"; N; "     X= ";
```

```
                    X(N) = Val(InputBox("Enter X =", "X", X(N), xpos, ypos))
                    Print X(N); "        Y= ";
                    Y(N) = Val(InputBox("Enter Y =", "Y", Y(N), xpos, ypos))
                    Print Y(N)
                    If IR > 1 Then
                        If X(N) <= X(N - 1) Then
                            Print "NOT ALLOWED"
                            IFlag = 1
                        Else
                            IFlag = 0
                        End If
                    Else
                        IFlag = 0
                    End If
                Loop While IFlag = 1
            MoreDataYN$ = InputBox("More Data <Y/N>?", "", "Y", xpos, ypos)
        Loop While MoreDataYN$ = "Y" Or MoreDataYN$ = "y"
        Cls
        Print "*I*********X(I)******************Y(I)********"
        For I = 1 To N
            Print Format(I, " 0000    ");
            Print Format(X(I), "    0.00000E+        ");
            Print Format(Y(I), "    0.00000E+")
        Next I
        CorrectYN$ = InputBox("Data Entry Correct <Y/N>?", "", "Y", xpos, ypos)
    Loop While CorrectYN$ = "N" Or CorrectYN$ = "n"
    End Sub


    Sub EndConditions(IE, xpos, ypos)

    Do
        Cls
        Print "SELECT END CONDITIONS:"
        Print "[1] LINEAR ENDS"
        Print "[2] PARABOLIC ENDS"
        Print "[3] CUBIC ENDS"
        IE = Val(InputBox("Enter End Condition, IE =", "IE", 1, xpos, ypos))
    Loop While IE < 1 Or IE > 3
    End Sub


    Sub InterpolationMode(M, xpos, ypos)

    Do
        Cls
        Print
        Print "ENTER INTERPOLATION MODE #: "
        Print "      1=SINGLE POINT "
        Print "      2=INCREMENT    X "
```

```
        Print "      3=ENTER NEW DATA "
        Print "      4=CHANGE END CONDITIONS"
        Print "      5=END"
        Print "WHICH? ";
        M = Val(InputBox("Enter Interpolation Mode Q =", "IQ", M, xpos, ypos))
    Loop While M < 1 Or M > 5
    End Sub


    Sub FromX2Y(N, S, X, Y, xpos, ypos)

    IFlag = 0
    Cls
    Print "ENTER X VALUES ONE BY ONE."
    Print "RETURN TO MENU IF X IS OUTSIDE OF RANGE X(1) TO X(N)."
    Print
    Do
        XZ = Val(InputBox("Enter X Value = ", "X", , xpos, ypos))

        If XZ < X(1) Then
            IFlag = 2
        Else
            For K = 2 To N
                If XZ <= X(K) Then
                    IFlag = 0
                    Exit For
                Else
                    IFlag = 1
                End If
             Next K
        End If

        If IFlag = 1 Then
            Print "X IS TOO LARGE"
        ElseIf IFlag = 2 Then
            Print "X IS TOO SMALL"
        Else
            K = K - 1
            Call Poly(XZ, YZ, X, Y, S, K)
            Print "@ X ="; XZ; " Y ="; YZ
        End If

    Loop While IFlag = 0
    End Sub


    Sub IncrementX2Y(N, S, X, Y, xpos, ypos)

    USize = (X(N) - X(1)) / (N - 1)
    Cls
```

```
Print "ENTER X INCREMENT ";
U = Val(InputBox("Enter X Increment", "Step Size ", USize, xpos, ypos))
Print
Print "******* X **************** Y ***"
XZ = X(1)
K = 1

Do
        Do
                Call Poly(XZ, YZ, X, Y, S, K)
                Print Format(XZ, "       0.0000E+");
                Print "              ";
                Print Format(YZ, "       0.0000E+")
                XZ = XZ + U
        Loop While XZ <= X(K + 1)

        K = K + 1
Loop While K < N

Continue$ = MsgBox("Hit Any Key to Continue", , "CONTINUE")
End Sub


Sub CubicSpline(IE, N, X, Y, A, S)
'
' Cubic Spline Subroutine
'
'IE              = Mode of end condition
'N       = Number of original data sets
'X,Y   = Original data sets
'A       = Matrix A
'S       = Vector S
'
'-- COMPUT N-2 ROWS
N1 = N - 1
N2 = N1 - 1
DX1 = X(2) - X(1)
DY1 = (Y(2) - Y(1)) / DX1 * 6!

For I = 1 To N2
        DNX = X(I + 2) - X(I + 1)
        D2Y = (Y(I + 2) - Y(I + 1)) / DNX * 6!
        A(I, 1) = DX1
        A(I, 2) = 2! * (DX1 + DNX)
        A(I, 3) = DNX
        A(I, 4) = D2Y - DY1
        DX1 = DNX
        DY1 = D2Y
Next I
```

```
'--Adjust first and last rows appropriated to end conditions.
Call ChangeEndConditions(IE, A, X, N1, N2)

'--Solve tridiagonal matrix eqns.
Call TridiagonalSystemSolver(N1, N2, A)

'--Put A to S vector
For I = 1 To N2
        S(I + 1) = A(I, 4)
Next I

'--Get S(1) & S(N)
Call S1SN(S, N)
'-- CHEER by Ron Hsin Chang, Copyright 2001
End Sub


Sub ChangeEndConditions(IE, A, X, N1, N2)

'--IE=1,No change is needed
'--IE=2,S(1)=S(2),S(N)=S(N-1),PARABOLIC ENDX.
'--IE=3, CUBIC ENDS, S(1),S(N) ARE EXTRAPOLATED.

Select Case IE
        Case 1

        Case 2
                A(1, 2) = A(1, 2) + X(2) - X(1)
                A(N2, 2) = A(N2, 2) + X(N) - X(N1)
        Case 3
                DX1 = X(2) - X(1)
                DNX = X(3) - X(2)
                A(1, 2) = (DX1 + DNX) * (DX1 + 2! * DNX) / DNX
                A(1, 3) = (DNX * DNX - DX1 * DX1) / DNX
                EX1 = X(N) - X(N1)
                ENX = X(N1) - X(N2)
                A(N2, 1) = (ENX * ENX - EX1 * EX1) / ENX
                A(N2, 2) = (EX1 + EX1) * (EX1 + 2! * ENX) / ENX
End Select
End Sub


Sub TridiagonalSystemSolver(N1, N2, A)
'
'    TRIS    SUBROUTINE
'
' -TRIDIAGONAL SYSTEM SOLVER
'-- REDUCE
For I = 2 To N2
        A(I, 2) = A(I, 2) - A(I, 1) / A(I - 1, 2) * A(I - 1, 3)
```

```
        A(I, 4) = A(I, 4) - A(I, 1) / A(I - 1, 2) * A(I - 1, 4)
Next I
'-- BACK SUBSTITUTE
A(N2, 4) = A(N2, 4) / A(N2, 2)
For I = 2 To N2
        J = N1 - I
        A(J, 4) = (A(J, 4) - A(J, 3) * A(J + 1, 4)) / A(J, 2)
Next I
End Sub


Sub S1SN(S, N)
'
'          S1SN SUBROUTINE
'
'            GET S(1) & S(N)
'                IE = 1, LINEAR ENDS
'                IE = 2, PARABOLIC ENDS
'                IE = 3, CUBIC ENDS
'
Select Case IE
        Case 1
                S(1) = 0
                S(N) = 0
        Case 2
                S(1) = S(2)
                S(N) = S(N - 1)
        Case 3
                S(1) = ((DX1 + DXN) * S(2) - DX1 * S(3)) / DNX
                S(N) = ((ENX + EX1) * S(N1) - EX1 * S(N3)) / ENX
End Select
End Sub


Sub Poly(XZ, YZ, X, Y, S, K)
'
'          POLY SUBROUTINE
'
D1 = X(K + 1) - XZ
D2 = XZ - X(K)
H = X(K + 1) - X(K)
YZ = (S(K) * D1 ^ 3 + S(K + 1) * D2 ^ 3) / 6 / H
YZ = YZ + (Y(K + 1) / H - H * S(K + 1) / 6) * D2
YZ = YZ + (Y(K) / H - H * S(K) / 6) * D1
End Sub
```

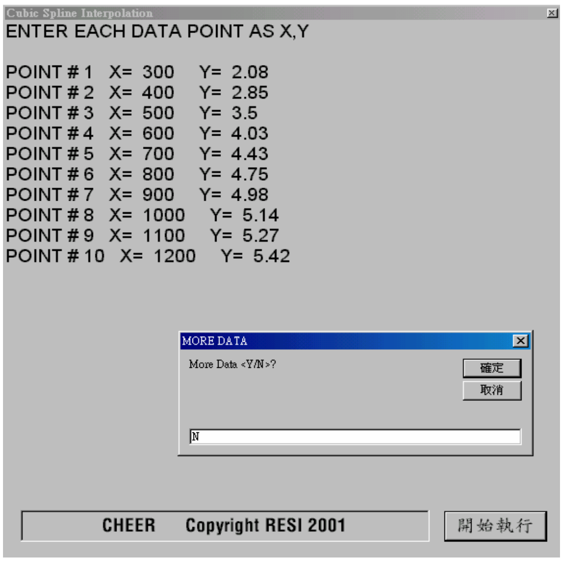
副程式使用說明：

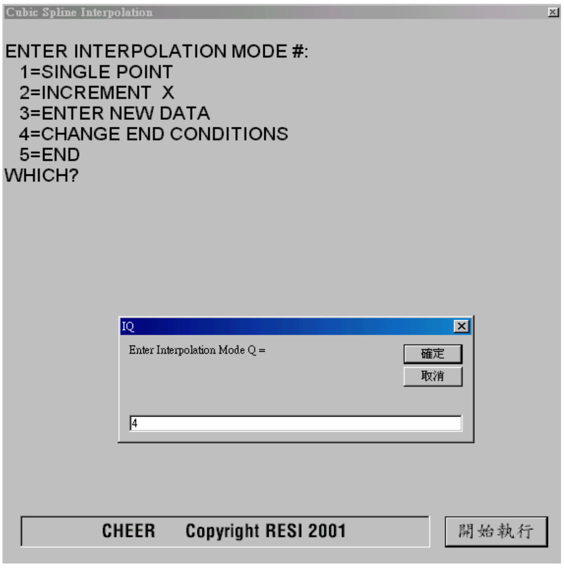
1. 副程式 Sub OriginalDataEntry (N, X, Y, xpos, ypos) 使用方法如下：
   (1) Call OriginalDataEntry (N, X, Y, xpos, ypos)
   (2) 逐筆輸入 X, Y 值
   (3) 程式回覆數據列表及 N 值

2. 副程式 Sub CubicSpline (IE, N, X, Y, A, S) 使用方法如下：
   (1) 指定 IE、N、X、Y 值
   (2) 程式將依指定之端點條件及數據計算矩陣 A，然後求解 S
   (3) 在此副程式中需使用另三個副程式
       Sub ChangeEndConditions (IE, A, X, N1, N2)
       Sub TridiagonalSystemSolver (N1, N2, A)
       Sub S1SN (S, N)

3. 副程式 Sub Poly (XZ, YZ, X, Y, S, K) 使用方法如下：
   (1) 指定 X、Y、S 及數據區間 K 值
   (2) 指定內差點 XZ 值
   (3) Call Poly (XZ, YZ, X, Y, S, K)
   (4) 程式計算內插結果 YZ

測試數據：



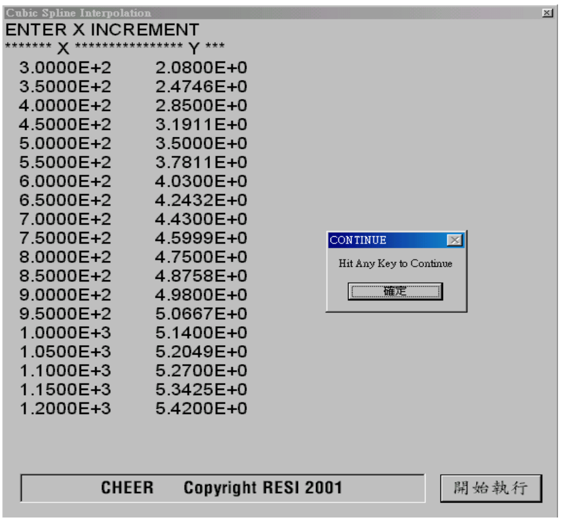


對話方塊與使用者介面：

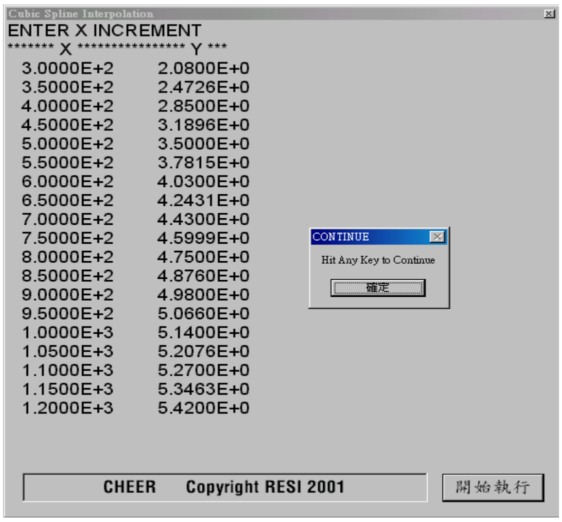
1. 單點內插：使用者輸入 X 值，程式計算 Y 值。
2. 增量列表：使用者輸入 X 增量，程式自動計算並製表。
3. 輸入新數據。
4. 變更端點條件。
5. 結束程式。

輸出結果：

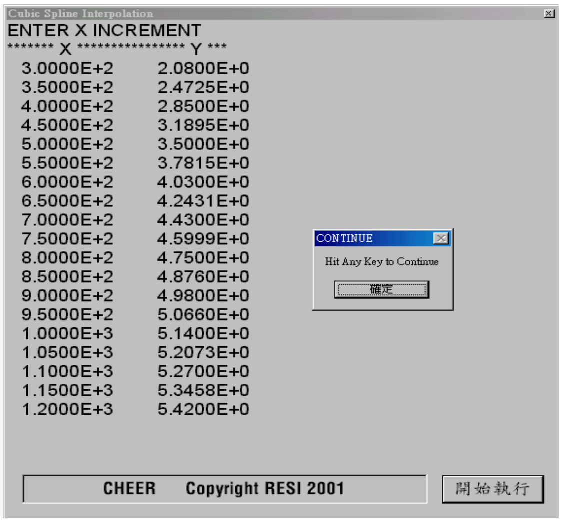
第一種端點條件內插法所得結果：



第二種端點條件內插法所得結果：



第三種端點條件內插法所得結果：



結果與討論：

三次弧線內插法內插溫度為 500°C （即 773.15°K）時之熱容量。利用三種不同端點條件，所得結果到小數以下第四位都是同樣為 4.6723。一般而言，利用不同端點條件，只對首尾第一個區間的內插值有顯著的影響，內部區間則三種端點條件所得結果相當一致。

第四節　二度空間線性內插

Visual Basic

考慮含有兩個自變數 x, y 的函數 f ：

$$f = f(x, y) \tag{3-4.1}$$

假設我們已有 $m \times n$ 個函數值 $f(x_i, y_j)$，對應 m 個 x_i $(i=1, 2, \cdots\cdots, m)$ 及 n 個 y_j $(j=1, 2, \cdots\cdots, n)$ 值的所有組合。爲了方便起見，將 $f_{ij} \equiv f(x_i, y_j)$ 建成一個二次元的數據表（如表 3.1)。本節所要探討的是如何由表中數據作內插，以求得任意 x、y 值的函數 $f(x, y)$ 近似值。

考慮線性內插，令 $x_i \le x \le x_{i+1}$，且 $y_i \le y \le y_{i+1}$，如圖 3.2 所示。圖中〇符號表示已知的數據點。首先由 f_{ij} 至 $f_{i+1,j}$ 和 $f_{i,j+1}$ 至 $f_{i+1,j+1}$ 分別作線性內插，得到 A 及 B 兩點的函數 $f(x, y)$ 的近似值 f_A 及 f_B；然後，再作線性內插，即可得最後的近似值 $f(x, y)$。

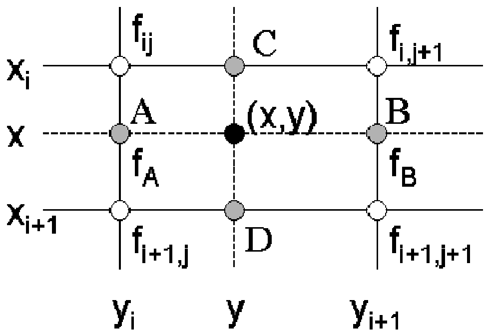


圖 3.2　二度空間線性內插

根據方程式 (3-1.1)，由 f_{ij} 至 $f_{i+1,j}$ 作線性內插，可以得到

$$f_A = f_{ij} + \frac{f_{i+1,j} - f_{ij}}{x_{i+1} - x_i}(x - x_i) \tag{3-4.2}$$

同理

$$f_B = f_{i,j+1} + \frac{f_{i+1,j+1} - f_{i,j+1}}{x_{i+1} - x_i}(x - x_i) \tag{3-4.3}$$

再由 A 點至 B 點作線性內插，得

$$f\,(x\,,y)=f_A+\frac{f_B-f_A}{y_{j+1}-y_j}(y-y_i) \qquad \textbf{(3-4.4)}$$

將 (3-4.2) 及 (3-4.4)，整理後，得到

$$f\,(x\,,y)=(1-\alpha)(1-\beta)\ f_{ij}+\beta(1-\alpha)\ f_{i\,,\,j+1}+\alpha\,(1-\beta)\ f_{i+1\,,\,j}+\alpha\beta\ f_{i+1\,,\,j+1} \qquad \textbf{(3-4.5)}$$

其中 $\alpha=(x-x_i)/(x_{i+1}-x_i)$；$\beta=(y-y_j)/(y_{j+1}-y_j)$。

例題 3-5　P-$\hat{V}$-T 數據內插

李明哲是一名服務於化學品公司的年輕化學工程師，他站立在一座大型甲烷貯存槽前，發現當時槽體溫度爲56.4°F，槽壓 82.3 pisa。他希望計算槽內現存甲烷氣體的存量，他卻不知道該如何進行計算。請爲他寫一個小程式，使他能快速的求得甲烷現在的單位質量體積 $\hat{V}$ 值，以便計算儲存槽內的甲烷存量。

設計問題 D-III 爲典型的二度空間內插問題。通常二度空間內插問題均需先輸入大量的數據，因此，宜建立數據檔案以便使用時隨時擷取。程式設計時，第一步驟應先確定 $(x\,,y)$ 在表中位置，再利用方程式 (3-4.5) 進行內插計算。

由於各種機器的磁碟操作指令不盡相同，因此，建立檔案及擷取檔案的程式本書不予列出，讀者可參照所使用機器的使用手冊自行設計程式。但若不將資料儲存，則下列程式即可直接使用。

甲烷的 P-$\hat{V}$-T 關係

溫　度	壓力 (psia)						
(°F)	10	20	30	40	60	80	100
−200	17.15	8.47	5.57	4.12	2.678	1.954	1.518
−100	23.97	11.94	7.91	5.91	3.91	2.903	2.301
0	30.72	15.32	10.19	7.63	5.06	3.78	3.014
100	37.44	18.70	12.44	9.33	6.21	4.65	3.71
200	44.13	22.07	14.7	11.03	7.37	5.5	4.40
300	50.83	25.42	16.94	12.71	8.46	6.35	5.07
400	57.51	28.76	19.17	14.38	9.58	7.19	5.75
500	64.20	32.10	21.40	16.05	10.70	8.03	6.42

TOP-DOWN 設計：

@將位置指標歸零

I1 = 0

J1 = 0

@確定 (x, y) 位置

DO FOR I = 1 TO N

> IF X1 > X (I)
>
> THEN: I1 = 1
>
> ELSE:

DO FOR J = 1 TO M

> IF Z1 > Z(J)
>
> THEN: J1 = J
>
> ELSE:

@依方程式 (3-4.5) 計算

IF (I1 = 0 .OR. J1 = 0 .OR. I1 = N .OR. J1 = M)

THEN:

> OUT OF RANGE ERROR

ELSE:

> $\alpha = (x - x_i)/(x_{i+1} - x_i)$
>
> $\beta = (y - y_j)/(y_{j+1} - y_j)$
>
> $Y1 = (1-\alpha)(1-\beta)\,Y_{ij} + \beta\,(1-\alpha)\,Y_{i,\,j+1} + \alpha\,(1-\beta)\,Y_{i+1,\,j} + \alpha\beta\,Y_{i+1,\,j+1}$

RETURN

程式列印：

```
Sub TwoDimensionalInterpolation(xpos, ypos)
'
'   TYPICAL DATA TABLE
'
'         Z1   Z2   Z3   Z4   ...
'   X1 Y11 Y12 Y13 Y14   ...
'   X2 Y21 Y22 Y23 Y24   ...
'   X3 Y31 Y32 Y33 Y34   ...
'    .    .    .    .    .   ...
'    .    .    .    .    .   ...
' X(I) : I = 1,N
' Z(J) : J = 1,M
' Y(I,J) : I = 1, N; J = 1, M
' XS : LOCATE X
' ZS : LOCATE Z
' YS : RETURN Y VALUE AFTER INTERPOLATION

Dim X(50), Z(50), Y(50, 50)
```

```
Cls
Print "Two-dimensional Linear Interpolation "
Print
Call EnterX(N, X, xpos, ypos)
Call EnterZ(M, Z, xpos, ypos)
Call EnterYIJ(X, Z, Y, M, N, xpos, ypos)
Cls
Print "**X************Z************Y**"
Do
        Call LocateXZ(XS, ZS, X, Z, xpos, ypos)
        Call Interpolation(N, M, X, Z, Y, XS, ZS, YS, ER$)
        If ER$ <> "" Then
                Print ER$
        Else
                Print XS, ZS, YS
        End If
        Call AnyMoreData(YN$, xpos, ypos)
Loop While YN$ <> "N" And YN$ <> "n"
End Sub


Sub EnterX(N, X, xpos, ypos)
Print "Enter X's: "
N = 0
Do
        N = N + 1
        Print "X("; N; ") = ";
        X(N) = Val(InputBox("Enter X Data = ", "X", X(N - 1), xpos, ypos))
        Print X(N)
        Call AnyMoreData(YN$, xpos, ypos)
Loop While YN$ <> "N" And YN$ <> "n"
End Sub


Sub EnterZ(M, Z, xpos, ypos)
Cls
Print "Enter Z's : "
M = 0
Do
        M = M + 1
        Print "Z("; M; ") = ";
        Z(M) = Val(InputBox("Enter Z Data = ", "Z", Z(M - 1), xpos, ypos))
        Print Z(M)
        Call AnyMoreData(YN$, xpos, ypos)
Loop While YN$ <> "N" And YN$ <> "n"
End Sub

Sub EnterYIJ(X, Z, Y, M, N, xpos, ypos)
Cls
```

```
Print "Enter Y(I,J):"
For I = 1 To N
        For J = 1 To M
                Print "X("; I; ") ="; X(I); "     Z("; J; ") = "; Z(J); " ==> Y("; I; ","; J; ") = ";
                Y(I, J) = Val(InputBox("Enter Y(i,j) Data", "Y(I,J)", , xpos, ypos))
                Print Y(I, J)
        Next J
        Print
Next I
End Sub


Sub LocateXZ(XS, ZS, X, Z, xpos, ypos)
'        **************
'            LOCATE X,Z
'        **************
        XS = Val(InputBox("Locate X=", "X", X(1), xpos, ypos))
        ZS = Val(InputBox("Locate Z=", "Z", Z(1), xpos, ypos))
End Sub


Sub AnyMoreData(YN$, xpos, ypos)
YN$ = InputBox("ANY MORE DATAS <Y/N> ? ", " MORE DATA", "Y", xpos, ypos)
End Sub


Sub Interpolation(N, M, X, Z, Y, X1, Z1, Y1, ER$)
'== LOCATE I,J
I1 = 0
J1 = 0
ER$ = ""
For I = 1 To N
        If X1 >= X(I) Then I1 = I
Next I
For J = 1 To M
        If Z1 >= Z(J) Then J1 = J
Next J
If X1 = X(N) Then I1 = N - 1
If Z1 = Z(M) Then J1 = M - 1
If (I1 = 0 Or J1 = 0) Or (I1 = N Or J1 = M) Then
        ER$ = "ERROR: (X,Z) BEYOND THE SCOPE OF THE TABLE !!"
Else
' == DO IT
        ALPHA = (X1 - X(I1)) / (X(I1 + 1) - X(I1))
        BETA = (Z1 - Z(J1)) / (Z(J1 + 1) - Z(J1))
        Y1 = (1 - ALPHA) * ((1 - BETA) * Y(I1, J1) + BETA * Y(I1, J1 + 1))
        Y1 = Y1 + ALPHA * ((1 - BETA) * Y(I1 + 1, J1) + BETA * Y(I1 + 1, J1 + 1))
End If
End Sub
```

符號說明：

M, N：　Z 軸及 X 軸之已知點數
X, Z：　座標值
Y：　函數值 Y = f(x，z)
X1, Z1：內插位置
Y1：　內插所得函數值

測試數據：

輸入數據：X = 溫度；Z = 壓力，Y = 體積 (見表 3.1)
內插：

X (°C)	Z (psia)
56.4	82.3
0.0	100.0
50.0	50.0
50.0	75.0

輸出結果：

T	P	$\hat{V}$
5.6400E+01	8.2300E+01	4.1713E+00
0.0000E+00	1.0000E+02	3.0140E+00
5.0000E+01	5.0000E+01	7.0575E+00
5.0000E+01	7.5000E+01	4.5700E+00

參考文獻

1. Perry, R. H. and C. H. Chilton,“Chemical Engineers’ Handbook”, McGraw-Hill, (1973).
2. Eckert, E. R. G. and R. J. Goldstein,“Measurement in Heat Transfer”Hemisphere, (1976).
3. Carnahan. B., H. A. Luther, and J. O. Wilkes,“Applied Numerical Methods”John Wiley, (1969).
4. Dorn, W.S., and D. D. McCracken,“Numerical Methods with FORTRAN IV Case Studies”, John Wiley, (1972).
5. “Encyclopedia of Chemical Technology”, Vol.2, 2nd ed., New York, Wiley, (1963).

習　題

1. 試改寫例題 3-1 的程式，使它能讀入表 3.2 的所有數據，並自動判斷輸入的 X 值介於哪兩個數據之間，再執行線性內插。
2. 將表 3.2 的 mV 值利用 DATA 敘述存放在問題 1 的程式中，溫度值則用 FOR/NEXT 迴路產生，重新設計問題 1 之程式。
3. 將本章所述之二度空間內插法擴展成二度空間三次弧線內插法，並設計程式，重作設計問題 D-III。二度空間三次弧線內插法，基本上與線性內插法的策略相似。如圖 3.3，○符號表示已知的數據點，首先由 x 軸作三次弧線內插，求得 A、B、C 及 D 四點，再利用 A、B、C 及 D 四點內插求得 $f\,(x\,,y)$ 之近似函數值。

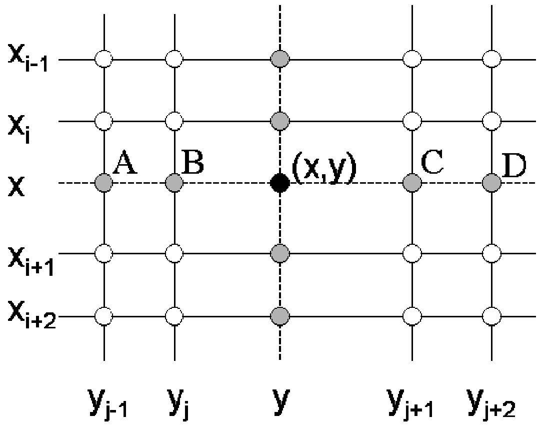


圖 3.3　二度空間三次弧線內插法

4. 上題中，若內插順序先作 y 軸，結果是否相同？

5. 表 3.5 爲氨之水溶液在各種不同溫度及莫耳濃度時的總蒸氣壓 [1]。試估計在 (66°F, 21.5 mole%)、(212°F, 12 mole%)、(115°F, 33.25 mole%) 時之總蒸氣壓。

表 3.5　氨水溶液的蒸氣壓

溫　度	氨之百分比莫耳濃度					
°F	0	10	20	25	30	35
60	0.26	1.42	3.51	5.55	8.65	13.22
80	0.51	2.43	5.85	9.06	13.86	20.61
100	0.95	4.05	9.34	14.22	21.32	31.16
140	2.89	9.98	21.49	31.54	45.73	64.78
180	7.51	21.65	44.02	62.68	88.17	121.68
220	17.19	42.47	81.91	113.81	156.41	211.24
250	29.83	66.67	124.08	169.48	229.62	305.60

6. 氫氣與氮氣反應生成氨的化學平衡常數 Kp，爲氫氣與氮氣的莫耳數比、壓力及溫度的函數。當氫氣-氮氣的莫耳數比爲 3 比 1 時，化學平衡常數與溫度及壓力的關係如表 3.6 所示 [5]。試估計在 473°C 及 217 atm 時，Kp 等於多少？

表 3.6　氨之化學反應平衡常數

溫　度	壓力 (atm)				
°C	100	200	300	400	500
400	0.014145	0.015897	0.018060	0.020742	0.024065
450	0.007222	0.008023	0.008985	0.010134	0.001149
500	0.004013	0.004409	0.004873	0.005408	0.006013
550	0.002389	0.002598	0.002836	0.003102	0.003392
600	0.001506	0.001622	0.001751	0.001890	0.002036

7. 試利用三次弧線內插法，重作例題 3-1。