

CHAPTER 9

常微分方程式——邊界值問題

(Ordinary Differential Equation – Boundary Value Problem)

張榮興 博士

豐映科技股份有限公司

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

描述實際問題所得到的微分方程式
極少能利用解析方法求解，通常需利用數值方法求解

V
i
s
u
a
l
B
a
s
i
c



無限平板上的境界層流場

流體平行流經一無限平板時，在平板上會形成所謂的「境界層」(Boundary layer)；其質量及動量守衡方程式分別為 [1]：

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0$$

$$u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = \nu \frac{\partial^2 u}{\partial y^2}$$

邊界條件為

$$y=0 \text{ 時, } u=v=0 ;$$

$$y=\infty \text{ 時, } u=u_\infty$$

利用無因次座標轉換

$$\eta = y \sqrt{\frac{u_\infty}{\nu x}}, \quad \phi = \sqrt{\nu x u_\infty} f(\eta), \quad u = \frac{\partial \phi}{\partial y}, \quad v = -\frac{\partial \phi}{\partial x}$$

則質量及動量守恆微分方程式可簡化成

$$ff'' + 2f''' = 0$$

$$\text{BC1 } \eta=0 \text{ 時, } f=0$$

$$\text{BC2 } \eta=0 \text{ 時, } f'=0$$

$$\text{BC3 } \eta=\infty \text{ 時, } f'=1$$

定義無因次速度

$$U = \frac{u}{u_\infty} = f' ; \quad V = \frac{v}{\frac{1}{2} \sqrt{\frac{\nu u_\infty}{x}}} = \eta f' - f$$

試解此問題，並作 U 及 V 對 η 的函數關係圖。

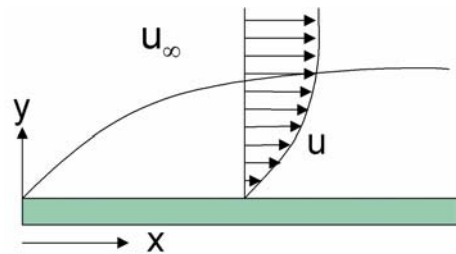


圖 9.1 境界層流場

常 微分方程式依邊界條件的定義位置分類，可分成初值問題 (I.V.P) 及邊界值問題 (B.V.P)，其中初值問題的數值解法我們已在第八章中詳加討論。本章中，我們將討論幾種解邊界值常微分方程式的解法。

邊界值常微分方程式的解法基本上可分成三大類：

1. 初值法：包括投射法 (Shooting) 及加成法 (Superposition)
2. 有限差分法 (Finite difference method)
3. 配重殘值法 (Method of Weighted Residuals)

所謂初值法基本上將邊界值常微分方程式問題視為一組聯立的初值常微分方程式問題，再利用迭代法求解。初值法可大致分成投射法、多重投射法及加成法三類，以下只對投射法作討論。加成法及多重投射法雖常見於商業化程式，但由於涉及較多數學處理，故此處不加以討論，有興趣讀者請參閱 [5, 6]。在第六章中，我們曾討論過利用差分表示導函數，若進一步將微分方程式中的導函數用差分表示，建立一組差分方程式，即稱為有限差分法。配重殘值法基本想法是利用一組近似函數加上配重函數（請回顧第六章所提高斯積分法）來表示微分方程式的近似解，使得函數積分殘值達到最小化，即為最佳近似解。我們將在本章中討論初值法及有限差分法，配重殘值法則單獨於第十章中詳加討論。

第一節 投射法—線性問題

Visual Basic

投射法 (Shooting Method) 屬於常微分方程式解法中的初值法的一種。首先考慮線性二次常微分方程式

$$Ly \equiv y'' + p(x)y' + q(x)y = f(x), a < x < b \quad (9-1.1)$$

$$\text{BC1} \quad x = a \text{ 時, } -k_1 \frac{dy}{dx} + h_1 y = \alpha$$

$$\text{BC2} \quad x = b \text{ 時, } k_2 \frac{dy}{dx} + h_2 y = \beta$$

其中 $k_1, k_2, h_1, h_2, \alpha$ 及 β 均為常數，且

$$h_1 k_1 \geq 0, \quad |h_1| + |k_1| \neq 0 \quad (9-1.2)$$

$$h_2 k_2 \geq 0, \quad |h_2| + |k_2| \neq 0$$

$$|h_1| + |h_2| \neq 0$$

由於線性常微分方程式具有加成性，首先將函數 $y(x)$ 寫成 y_1 及 y_2 兩部分的加成：

$$y(x) = y(x, s) = y_1(x) + sy_2(x) \quad (9-1.3)$$

其中 s 為任一常數， $y_1(x)$ 及 $y_2(x)$ 分別為下列初值常微分方程式的解：

$$y_1'' + p(x)y_1' + q(x)y_1 = f(x) \quad , \quad a < x < b \quad (9-1.4)$$

IC1 $x = a$ 時， $y_1 = \alpha C_1$

IC2 $x = a$ 時， $\frac{dy_1}{dx} = \alpha C_2$

$$y_2'' + p(x)y_2' + q(x)y_2 = 0 \quad , \quad a < x < b \quad (9-1.5)$$

IC3 $x = a$ 時， $y_1 = k_1$

IC4 $x = a$ 時， $\frac{dy_1}{dx} = h_1$

其中 C_1 及 C_2 均為常數，要特別注意方程式 (9-1.5) 的右側為 0。由於 $y = y_1 + sy_2$ ，因此，根據原微分方程式的第一個邊界條件，得到

$$-k_1 [\alpha C_2 + sh_1] + h_1 [\alpha C_1 + sk_1] = \alpha$$

或

$$C_1 h_1 - C_2 k_1 = 1 \quad (9-1.6)$$

而由原方程式的第二個邊界條件，則可以得到

$$k_2 [y_1'(b) + s y_2'(b)] + h_2 [y_1(b) + s y_2(b)] = \beta$$

解上式，得到 s 的表示式為

$$s = \frac{\beta - k_2 y_1'(b) - h_2 y_1(b)}{k_2 y_2'(b) + h_2 y_2(b)} \quad (9-1.7)$$

因此，解線性邊界值常微分方程式的執行策略可歸納如下：

1. 將原邊界值 BVP 常微分方程式 (9-1.1) 轉換成初值 IVP 常微分方程式 (9-1.4) 及 (9-1.5)。
2. 令 C_1 為一任意值 [或根據方程式 (9-1.6) 作適當猜測]。
3. 利用方程式 (9-1.6) 求出 C_2 。
4. 將 IVP 常微分方程式 (9-1.4) 積分，求得 $y_1(b)$ 及 $y_1'(b)$ 。

5. 將 IVP 常微分方程式 (9-1.5) 積分，求得 $y_2(b)$ 及 $y_2'(b)$ 。
6. 利用方程式 (9-1.7) 求得 s 。
7. 利用方程式 (9-1.3) 建立 y 與 x 之關係。

例題 9-1 投射法—線性 BVP 常微分方程式

試利用投射法解常微分方程式

$$y'' - 3y = 2x; \quad 0 < x < 1$$

BC1 $x=0$ 時， $y=0$

BC2 $x=1$ 時， $y=1$

解：

首先將原微分方程式轉換成 (9-1.4) 及 (9-1.5) 的型式：

$$y_1'' - 3y_1 = 2x; \quad 0 < x < 1$$

IC1* $x=0$ 時， $y_1=0$

IC2 $x=0$ 時， $y_1'=0$

$$y_2'' - 3y_2 = 0$$

IC3 $x=0$ 時， $y_2=0$

IC4 $x=0$ 時， $y_2'=0$

或寫成矩陣型式得到

$$\frac{d}{dx} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} u_2 \\ 3u_1 + 2x \\ u_4 \\ 3u_3 \end{bmatrix}$$

IC: $x=0$ 時； $[u_1, u_2, u_3, u_4]^T = [0, 0, 0, 1]^T$

其中 $u_1 = y_1; u_2 = y_1'; u_3 = y_2; u_4 = y_2'$

方程式 (9-1.7) 可以改寫成

$$s = \frac{1 - u_1(1)}{u_3(1)}$$

微分方程式的解為

$$y(x) = u_1(x) + su_3(x)$$

【註】此微分方程式的正確解為 $y(x) = \frac{5 \sinh(\sqrt{3}x)}{3 \sinh(\sqrt{3})} - \frac{2}{3}x$

TOP-DOWN 設計

@輸入基本參數
輸入 A, B, H0, ErrTol, StepCtrl\$, NP
@執行阮奇庫塔法
Call RungeKutta
@計算 s 值，然後計算 y 值
$s = (1 - u_1(1)) / u_3(1)$
$y(x) = u_1(x) + su_2(x)$
列印結果

程式列印

```

' *****
' O.D.E.-B.V.P. Linear Shooting Method
' *****
'
Private Sub LinearShootingBVPODE(Xpos, Ypos)
'
' LINEAR SHOOTING
'
' USER'S SPACE
Cls
Print " SOLUTION OF LINEAR B.V.P.-O.D.E. BY THE SHOOTING METHOD"
Print
Print
Dim Y0(50), F(50), YWK(50, 101) As Double

Neqn% = 4
Lower = 0
Upper = 1
Y0(1) = 0
Y0(2) = 0
Y0(3) = 0
Y0(4) = 1
H0 = 0.01
    
```

```

TolErr = 0.001
StepCtrl$ = "Y"

Print ">ENTER LIMITS OF INTERGRATION"
Print "Lower Limit of X = ";
Lower = Val(InputBox(" Lower Limit, Lower = ", "Low Limit of X", Lower, Xpos, Ypos))
Print Lower

Print "Upper Limit of X = ";
Upper = Val(InputBox(" Upper Limit, Upper = ", "Upper Limit XS", Upper, Xpos, Ypos))
Print Upper

Print "> Initial Step Size = ";
H0 = Val(InputBox("> Initial Step Size = ", "Step Size", H0, Xpos, Ypos))
Hsav = H0
Print H0

Print "> Error Tolerance on Y = ";
TolErr = Val(InputBox("> ERROR TOLERANCE ON Y = ", "Error Tolerance", TolErr, Xpos, Ypos))
Print TolErr

Print "> Need Step Size Control <Y/N> = ";
StepCtrl$ = InputBox("> STEPSIZE CONTRAL ? (Y/N) ", "STEP CONTROL", "Y", Xpos, Ypos)
Print StepCtrl$

Print "> Need Detail Printout <Y/N> = ";
PNT$ = UCase(InputBox(">Need Detail Printout ? (Y/N) ", "PRINT CONTROL", "N", Xpos, Ypos))
Print PNT$

Print
NP = Val(InputBox("> NUMBER OF PRINTING INTERVALS = (<-50) ", "NP", NP, Xpos, Ypos))
XPT = (Upper - Lower) / NP
Print

Call RK4Shooting(Neqn%, Y0, YWK, TolErr, PNT$, H0, Upper, Lower, NP, StepCtrl$)

ND = NP + 1

S = (1 - YWK(1, ND)) / YWK(3, ND)

MsgBox "Ready To Calculate S"
Cls

Print "** LINEAR SHOOTING **"
Print
Print "*** S = "; S
Print
Print " X Y(exact) Y(Numerical) Rel Err"
Print "*****"
X = Lower
Y1 = Exp(Sqr(3)) - Exp(-Sqr(3))
For I = 1 To ND

```



```

    Y = Sqr(3) * X
    Y = 5 / 3 * (Exp(Y) - Exp(-Y)) / Y1 - 2 / 3 * X
    YC = YWK(1, I) + S * YWK(3, I)
    If Y = 0 Then
        ErY = 0
    Else
        ErY = (YC - Y) / Y
    End If
    Print Format(X, "0.00E+00 ");
    Print Format(Y, "0.00000E+00 ");
    Print Format(YC, "0.00000E+00 ");
    Print Format(ErY, "0.0E+00 ");
    X = X + XPT
Next I
End Sub

Private Sub DeriveFunction(F, X, Y)
' ===== DEFINE FUNCTIONS
F(1) = Y(2)
F(2) = 3 * Y(1) + 2 * X
F(3) = Y(4)
F(4) = 3 * Y(3)
End Sub

Public Sub RK4Shooting(Neqn%, Y0, YWK, TolErr, PNT$, H0, Upper, Lower, NP, StepCtrl$)
'
' Linera Shooting Using Runge-Kutta Integration Algorithm
'
' Neqn%      =No. of Eqns.
' Y0         =Initial Condition
' TolErr     =Local Error Tolerance.
' PNT$       = Detail Print Control, Y for Print Everything
' H0        =Initial Step Size.
' Upper     =Upper Bound
' Lower     =Lower Bound
' NP        = Total Number of Printing Intervals
' StepCtrl$ =Auto. Step Size Control ID.
'
' F=Function; Yi = F(I)

Call MachineErrorCheck(Eps, U20)
Cls
PrtEnt% = 1
Call ReportRK4(PrtEnt%, Neqn%, X0, Y0, TolErr)

GoBackPrt% = 1
PT$ = PNT$
XPT = (Upper - Lower) / NP
X0 = Lower
H = H0

```

```

XF = X0 + XPT
Pointer% = 0

Do
    If Pointer% = 0 Or IPchk = 1 Then
        Pointer% = Pointer% + 1
        For K = 1 To Neqn%
            YWK(K, Pointer%) = Y0(K)
        Next K
        IPchk = 0
        Print "IP = "; Pointer%; " ";
    End If
    If PT$ = "Y" Or GoBackPrt% = 1 Then
        PrtEnt% = 2
        Call ReportRK4(PrtEnt%, Neqn%, X0, Y0, TolErr)
        PT$ = PNT$
    End If
    If X0 >= Upper Then Exit Do
    Call RungeKutta(Neqn%, X0, Y0, TolErr, H, H1, StepCtrl$)

'==CHECK FOR END INTERVAL==

    If Abs(X0 - XF) <= Eps Then
        XF = X0 + XPT
        PT$ = "Y"
        IPchk = 1
    End If
    If X0 + H > XF + U20 Then
        H = XF - X0
    End If
    If (X0 - H1) <= XF And PT$ = "Y" Then
        GoBackPrt% = 1
    Else
        GoBackPrt% = 0
    End If
Loop While X0 <= Upper + U20
End Sub

Public Sub MachineErrorCheck(Eps, U20)
'
' Calculate Machine Round Off Error
'
Eps = 1#
Do
    Eps = Eps / 2
    EpsP1 = Eps + 1
Loop While EpsP1 > 1
U20 = 20 * Eps

End Sub

```

```

Public Sub ReportRK4(PrtEnt%, Neqn%, X0, Y0, TolErr)
Select Case PrtEnt%
Case 1
Print
Print "SOLUTION          with Error Tolerance =";
Print Format(TolErr, " 0.00E+00")
Print "*****"
Print "      X          ";
Print "      Y          "
Case 2
Print Format(X0, " 0.000000E+00");
For IK = 1 To Neqn%
Print Format(Y0(IK), " 0.000000E+00");
Next IK
Print
End Select
End Sub

Public Sub RungeKutta(Neqn%, X0, Y0, TolErr, H, H1, StepCtrl$)
'
' Runge-Kutta Controller
'
Dim Y(50), W(50), Z(50), S(50), Q1(50), Q3(50) As Double
T2 = H
T = T2
E5 = TolErr / 100
Ent% = 0
Call RungeKuttaModual(Ent%, Neqn%, S, X0, Y0, T, X, Y, D, W)
For K = 1 To Neqn%
Q1(K) = W(K)
Next K
Do
T = T2 / 2
Ent% = 0
Call RungeKuttaModual(Ent%, Neqn%, S, X0, Y0, T, X, Y, D, W)
For K = 1 To Neqn%
Z(K) = W(K)
S(K) = Z(K)
Next K
X = X0 + T
Ent% = 1
Call RungeKuttaModual(Ent%, Neqn%, S, X0, Y0, T, X, Y, D, W)
I8 = 0
For K = 1 To Neqn%
Q3(K) = (W(K) - Q1(K)) / 15
DetErr = Abs(Q3(K)) / T2
If DetErr >= TolErr Then
ConvergentFlag% = 0: ' Not Converged
Else
If DetErr >= E5 Then I8 = 1

```

```

        ConvergentFlag% = 1
    End If
    Next K
,
' CONVERGENCE ATTAINED==
,
    If ConvergentFlag% = 1 Then
        For K = 1 To Neqn%
            Y0(K) = W(K) + Q3(K)
        Next K
        H = T2
        X0 = X0 + H
        H1 = H
,
' CHECK FOR STEP-SIZE CONTROL==
,
        If StepCtrl$ = "N" Then Exit Sub
        If I8 = 1 Then Exit Sub
        H = 2 * H
        Exit Sub
    End If
,
' CONVERGENCE NOT ATTAINED==
,
    If StepCtrl$ = "N" Then
        Print
        Print "  **ERROR** "
        Print "    CONVERGENCE NOT ATTAINED"
        Print "    WITHIN REQUIRED TOLERANCE "
        Print "    FOR GIVEN STEP-SIZE"
        Print "    H="; H; "  AT X="; X0
    Else
        T2 = T2 / 2
        For K = 1 To Neqn%
            Q1(K) = Z(K)
        Next K
    End If
    Loop While ConvergentFlag% = 0
End Sub

Public Sub RungeKuttaModual(Ent%, Neqn%, S, X0, Y0, T, X, Y, D, W)
,
' RUNGE-KUTTA MODUAL
,
' ==INITIALIZE VARIABLES ==
,
    Dim F(50) As Double
    If Ent% = 0 Then
        X = X0
        For K = 1 To Neqn%
            S(K) = Y0(K)

```

```

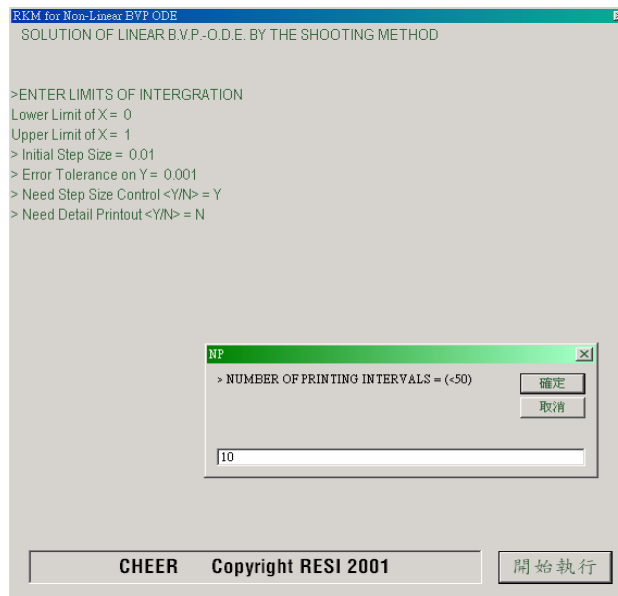
        Next K
    End If
    For K = 1 To Neqn%
        Y(K) = S(K)
        W(K) = S(K)
    Next K
    '==COMPUTE W VALUE==
    ' --1st PASS
    For K = 1 To Neqn%
        Call DeriveFunction(F, X, Y)
        D = T * F(K)
        W(K) = W(K) + D / 6
        Y(K) = S(K) + D / 2
    Next K
    X = X + T / 2
    ' --2nd PASS
    For K = 1 To Neqn%
        Call DeriveFunction(F, X, Y)
        D = T * F(K)
        W(K) = W(K) + D / 3
        Y(K) = S(K) + D / 2
    Next K
    ' --3rd PASS
    For K = 1 To Neqn%
        Call DeriveFunction(F, X, Y)
        D = T * F(K)
        Y(K) = S(K) + D
        W(K) = W(K) + D / 3
    Next K
    X = X + T / 2
    ' --4th PASS
    For K = 1 To Neqn%
        Call DeriveFunction(F, X, Y)
        D = T * F(K)
        W(K) = W(K) + D / 6
    Next K
End Sub

```

注 意：

1. 本程式所使用阮奇庫塔法副程式，爲了提高收斂性，與第八章所附程式比較，已經略作修正，增加機器運算誤差的處理；因此，可明顯改善運算性能。
2. 有關副程式使用方法，詳第八章說明。

程式執行結果與討論



第9章 常微分方程式 — 邊界值問題

RKM for Non-Linear BVP ODE

SOLUTION with Error Tolerance = 1.00E-03

	X	Y			
IP = 1	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	1.000000E+00
IP = 2	1.000000E-01	3.338646E-04	1.002622E-02	1.005186E-01	1.015391E+00
IP = 3	2.000000E-01	2.683197E-03	4.041116E-02	2.040960E-01	1.061331E+00
IP = 4	3.000000E-01	9.124736E-03	9.207532E-02	3.138490E-01	1.139213E+00
IP = 5	4.000000E-01	2.185905E-02	1.665821E-01	4.330812E-01	1.251390E+00
IP = 6	5.000000E-01	4.327675E-02	2.661788E-01	5.653823E-01	1.401248E+00
IP = 7	6.000000E-01	7.603051E-02	3.938708E-01	7.147368E-01	1.593309E+00
IP = 8	7.000000E-01	1.231149E-01	5.535089E-01	8.856432E-01	1.833365E+00
IP = 9	8.000000E-01	1.879567E-01	7.499063E-01	1.083250E+00	2.128654E+00
IP = 10	9.000000E-01	2.745178E-01	9.889836E-01	1.313511E+00	2.488076E+00
IP = 11	1.000000E+00	3.874151E-01	1.277947E+00	1.583363E+00	2.922468E+00

Linear Shooting BVP ODE

Ready To Calculate S

確定

CHEER Copyright RESI 2001

開始執行

```

RKM for Non-Linear BVP ODE
* LINEAR SHOOTING *

** S = 0.386888489138194

X      Y(exact)  Y(Numerical)  Rel Err
-----
0.00E+00  0.00000E+00  0.00000E+00  0.0E+00
1.00E-01  3.93074E-02  3.92234E-02  -2.1E-03
2.00E-01  8.18021E-02  8.16456E-02  -1.9E-03
3.00E-01  1.30767E-01  1.30549E-01  -1.7E-03
4.00E-01  1.89680E-01  1.89413E-01  -1.4E-03
5.00E-01  2.62317E-01  2.62017E-01  -1.1E-03
6.00E-01  3.52868E-01  3.52554E-01  -8.9E-04
7.00E-01  4.66063E-01  4.65760E-01  -6.5E-04
8.00E-01  6.07309E-01  6.07054E-01  -4.2E-04
9.00E-01  7.82859E-01  7.82700E-01  -2.0E-04
1.00E+00  1.00000E+00  1.00000E+00  0.0E+00

CHEER Copyright RESI 2001

```

1. 積分的相對誤差設為 1.00E-3 時，計算結果 y 值與理論解的相對誤差最大約為 2.0E-3。準確性相當高。
2. 當相對誤差取得過小時，將使得計算發生無法收斂的困擾。若將程式中的機器計算誤差 EPS 去除，將使得計算積分步伐變成 1.E-17，發生無法收斂的困擾。

第二節 投射法 — 非線性問題

Visual Basic

考慮二次非線性常微分方程式及其邊界條件為

$$y'' = f(x, y, y'), \quad a < x < b \quad (9-2.1)$$

$$\text{BC1} \quad x = a, \quad -k_1 y' + h_1 y = \alpha$$

$$\text{BC2} \quad x = b, \quad k_2 y' + h_2 y = \beta$$

其中 $k_1, k_2, h_1, h_2, \alpha$ 及 β 均為常數，且

$$k_1 h_1 \geq 0; \quad |k_1| + |h_1| \neq 0$$

$$k_2 h_2 \geq 0; \quad |k_2| + |h_2| \neq 0$$

$$h_1 + h_2 > 0$$

相對應的初值問題為

$$u'' = f(x, u, u'); \quad a < x < b \quad (9-2.2)$$

$$\text{IC1} \quad x = a \text{ 時, } u(a) = \alpha C_1 + sk_1$$

$$\text{IC2} \quad x = a \text{ 時, } u'(a) = \alpha C_2 + sh_1$$

其中 IC1 及 IC2 設為 $u(a) = \alpha C_1 + sk_1$ 及 $u'(a) = \alpha C_2 + sh_1$ ，目的是讓第二項 sk_1 及 sh_1 代入 BC1 後，會相減而消失。同理， αC_1 及 αC_2 也是要讓二式相加變成 α ，以簡化後續處理。將上式代回方程式 (9-2.1) 可以得到

$$C_1 h_1 - C_2 k_1 = 1$$

將 IVP 方程式 (9-2.2) 積分至 $x = b$ 時，必須滿足原微分方程式的第二個邊界條件，即殘值 $R(s)$ 必須為 0。

$$R(s) = k_2 u'(b; s) + h_2 u(b; s) - \beta = 0 \quad (9-2.3)$$

$R(s)$ 為一非線性方程式，可以利用牛頓拉福森法解上式求 s ，迭代式為

$$s_{k+1} = s_k - \frac{R(s_k)}{R'(s_k)} \quad (9-2.4)$$

由方程式 (9-2.3)，得 $\partial R / \partial s$ 為

$$\begin{aligned} R'(s) &= k_2 \frac{\partial}{\partial s} [u'(b, s)] + h_2 \frac{\partial}{\partial s} [u(b, s)] \\ &\equiv k_2 \eta(b) + h_2 \xi(b) \end{aligned} \quad (9-2.5)$$

其中

$$\xi(x) = \frac{\partial}{\partial s} [u(x; s)] \quad (9-2.6)$$

$$\eta(x) = \frac{\partial}{\partial s} [u'(x; s)] \quad (9-2.7)$$

將方程式 (9-2.2) 改寫成矩陣型式的一次聯立微分方程式，得

$$\frac{d}{dx} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} u_1 \\ f(x, u_1, u_2) \end{bmatrix} \quad (9-2.8)$$

$$\text{I.C.} \quad x = a \text{ 時, } [u_1, u_2]^T = [\alpha C_1 + sk_1, \alpha C_2 + sh_1]^T$$

其中 $u_1 = u(x; s)$; $u_2 = u'(x; s)$ 。將上式對 s 微分，由於

$$\frac{\partial}{\partial s} \left[\frac{d}{dx} u_1 \right] = \frac{d}{dx} \left[\frac{\partial u_1}{\partial s} \right] = \frac{d\xi}{dx}$$

$$\frac{\partial}{\partial s} \left[\frac{d}{dx} u_2 \right] = \frac{d}{dx} \left[\frac{\partial u_2}{\partial s} \right] = \frac{d\eta}{dx}$$

因此，利用方程式 (9-2.6) 及方程式 (9-2.7) 的 ξ 及 η 定義，可以得到以下結果

$$\frac{d\xi}{dx} = \eta \quad (9-2.9)$$

$$\frac{d\eta}{dx} = \frac{df}{ds} \equiv \frac{df}{\partial u_1} \frac{\partial u_1}{\partial s} + \frac{df}{\partial u_2} \frac{\partial u_2}{\partial s} = \frac{\partial f}{\partial u_1} \xi + \frac{\partial f}{\partial u_2} \eta \quad (9-2.10)$$

或寫成矩陣型式

$$\frac{d}{dx} \begin{bmatrix} \xi \\ \eta \end{bmatrix} = \begin{bmatrix} \eta \\ \frac{\partial f}{\partial u_1} \xi + \frac{\partial f}{\partial u_2} \eta \end{bmatrix} \quad (9-2.11)$$

I.C. $x = a$ 時， $[\xi, \eta]^T = [k_1, h_1]^T$

將上式積分，即可求得 $\xi(b)$ 及 $\eta(b)$ ，代回方程式 (9-2.5) 可求得 $R'(s)$ 。再利用方程式 (9-2.4) 即可求得新的 s 值。

非線性邊界值常微分方程式的投射法執行策略，可歸納如下：

1. 將原 BVP 常微分方程式 (9-2.1) 轉換成 IVP 方程式 (9-2.8)。
2. 任意猜測一個 C_1 值。
3. 利用 $C_1 h_1 - C_2 k_1 = 1$ 求出 C_2 。
4. 任意假設一個 s_0 值。
5. 將 IVP 常微分方程式 (9-2.8) 及 (9-2.11) 積分至 $x = b$ 。
6. 利用方程式 (9-2.3) 及 (9-2.5) 分別求 $R(s)$ 及 $R'(s)$ 。
7. 利用方程式 (9-2.4) 求 s_{k+1} 。
8. 重複步驟 5~7，直到 s 收斂為止。

例題 9-2 滲透與反應

多孔狀觸媒粒子的滲透及反應分析是反應工程中相當重要的問題。球形觸媒粒子的質量守恆方程式為

$$D \left[\frac{1}{r^2} \frac{d}{dr} \left(r^2 \frac{dC}{dr} \right) \right] = kR(C) \quad , \quad 0 < r < r_0$$

其中 r 為徑向座標， r_0 為觸媒粒子半徑， D 為滲透率， C 為反應物濃度， k 為反應常數， $R(C) = C^n$ ， n 為反應次數，其邊界條件為

$$\text{BC1} \quad r=0 \text{ 時, } \frac{dC}{dr} = 0 \quad (\text{對原點對稱})$$

$$\text{BC2} \quad r=r_0 \text{ 時, } C = C_0 \quad (\text{觸媒表面濃度})$$

定義無因次變數希笠模數 (Thiele Modulus) 及 x, y 分別為

$$\Phi = r_0 \sqrt{\frac{kC_0^{n-1}}{D}} \quad ; \quad x = r/r_0 \quad ; \quad y = C/C_0$$

則質量守恆方程式可被轉換成

$$\frac{d^2 y}{dx^2} + \frac{2}{x} \frac{dy}{dx} = \Phi^2 R(y)$$

$$\text{BC1} \quad x=0 \text{ 時, } \frac{dy}{dx} = 0$$

$$\text{BC2} \quad x=1 \text{ 時, } y=1$$

若觸媒粒子為直徑 5 mm 的 γ -氧化鋁擔體含浸鉑金屬而成，用於催化環己烷的脫氫反應。反應速率常數 k 為 4.21 s^{-1} ，滲透率為 $2.105 \times 10^{-5} \text{ m}^2/\text{s}$ 。若此反應為一次或二次反應，試分別求其濃度分布。

解：

令 $u = y; v = y'; f = \Phi^2 R(y) - \frac{2}{x} \frac{dy}{dx}$ ，則根據本節所介紹的方法，可以寫出對應的

IVP 常微分方程式為

$$\frac{d}{dx} \begin{bmatrix} u \\ v \\ \xi \\ \eta \end{bmatrix} = \begin{bmatrix} v \\ \Phi^2 R(u) - \frac{2}{x} v \\ \eta \\ \Phi^2 \frac{\partial}{\partial y} R(y) - \frac{2}{x} \end{bmatrix} \quad (9-2.12)$$

$$\text{IC1} \quad x=0 \text{ 時, } u = s$$

$$\text{IC2} \quad x=0 \text{ 時, } v = 0$$

IC3 $x = 0$ 時, $\xi = 1$

IC4 $x = 0$ 時, $\eta = 0$

根據殘餘值的定義式 (9-2.3) 得殘值 $\mathcal{G}(s)$ 及 $\mathcal{G}'(s)$ 分別為

$$\mathcal{G}(s) = u(1; s) - 1 \quad (9-2.13)$$

$$\mathcal{G}'(s) = \xi(1; s) \quad (9-2.14)$$

將 IVP 常微分方程式 (9-2.12) 由 $x = 0$ 積分到 $x = 1$, 再利用牛頓法解方程式 (9-2.13), 求出 s 值。然後重複此積分及求 s 的步驟, 直到 s 收斂為止。

程式列印 (請參考第八章 RKM 副程式說明)

```

'
'*****
' SOLUTION OF NONLINEAR B.V.P.-O.D.E.
' BY THE SHOOTING METHOD
'*****
'
' CHEER-2001   Ron-Hsin Chang
'
' Example 9-2
' First Order Reaction
'
' Private Sub RKM4NonlinearBVPODE(Xpos, Ypos)
'
' N=Number of dependent variables
' H=Initial step size
' Y0=Initial conditions of X and Y(l)
' Y0(1)=Initial condition of independent variabgle
' RelErr=Interval error allowed, suggested value max : Y0*1E-5
' XS=Value of independent variable at which answer is required
' DY=Values of derivatives at Xs
' YS=Solution of all the variables
' W=Working space
'
' Dim Y0(11), RelErr(10), DY(10), YS(11), W(44), Y(11)
' Dim YY(11, 101)
'
' MACHINE ERROR
'
' MErr = 1E-16
'
' ===== USER'S SPACE
'
' Cls
' N = 4: ' Number of eqns.
' NP = 50

```

```

IPrt$ = InputBox("Print Detail InformaTION <Y/N> ", "Require Detail Information", "N", Xpos, Ypos)
IPrt$ = UCase(IPrt$)

Print "> Enter Limits of Integration:"
Print "Lower Limit of X = ";
X0 = Val(InputBox(" Lower Limit X0 = ", "Low Limit of X", 0, Xpos, Ypos))
Print X0

Print "Upper Limit of X = ";
XS = Val(InputBox(" Upper Limit XS = ", "Upper Limit XS", XS, Xpos, Ypos))
Print XS

Print "> Initial Step Size = ";
H = Val(InputBox("> Initial Step Size = ", "Step Size", (XS - X0) / 100, Xpos, Ypos))
Hsav = H
Print H
'
'--ERROR TOLLERANCE
'
Print "> Error Tolerance on Y = ";
E1 = Val(InputBox("> ERROR TOLERANCE ON Y = ", "Error Tolerance", H * 0.00001, Xpos, Ypos))
Print E1

Print "> Error Tolerance on S = ";
E2 = Val(InputBox("> ERROR TOLERANCE ON S = ", "Error Tolerance", H * 0.00001, Xpos, Ypos))
Print E2

For I = 1 To N
    RelErr(I) = E1
Next I
'
'--PRINTING CONTROL
'
Print
NP = Val(InputBox("> NUMBER OF PRINTING INTERVALS = (<50) ", "NP", NP, Xpos, Ypos))
XPT = (XS - X0) / NP
Print
'
Print
ThieleModulus = Val(InputBox("> THIELE MODULUS = ", "Thiele Modulus", ThieleModulus, Xpos,
Ypos))
Print "Thiele Modulus = "; ThieleModulus
'
'--INITIAL CONDITIONS
'
S = Val(InputBox("> Enter Initial Guess : S= ", "S", S, Xpos, Ypos))
Iteration% = 0
Cls
'
'--START ITERATION

```

```

Do
    Y0(1) = 0
    Y0(2) = S
    Y0(3) = 0
    Y0(4) = 1
    Y0(5) = 0
    ENT% = 0
    Iteration% = Iteration% + 1
    For I = 1 To N + 1
        YY(I, 1) = Y0(I)
    Next I
    IPT = 1
    Do
        XS = Y0(1) + XPT
    '--CALL KUTTA-MERSON
        Call RungeKuttaMerson(ENT%, N, Y0, Y, DY, XS, YS, W, RelErr, IPrt$, H, HOLD,
            Hsav, MErr, ThieleModulus)
        For LP = 1 To N + 1
            Y0(LP) = YS(LP)
            YY(LP, IPT + 1) = YS(LP)
        Next LP
        If HOLD > XPT Then HOLD = XPT
        IPT = IPT + 1
    Loop While IPT <= NP
    '--CALCULATE R(S)
    R = YS(2) - 1
    '--CALCULATE R'(S)
    RP = YS(4)
    '--CALCULATE S
    SOLD = S
    S = S - R / RP
    Print "Iteration #"; Iteration%; " S = ";
    Print Format(S, " 0.000000E+00")
    '--CHECK CONVERGENCE ON S
    If Abs((S - SOLD) / S) < E2 Or Abs(S - SOLD) < E2 Then
        ConvergentFlag% = 1
    Else
        H = Hsav
        ConvergentFlag% = 0
    End If
    Loop While ConvergentFlag% = 0
    MsgBox ("Integration Converged, Hit Button to Continue")
    '--FINAL FUNCTION VALUE
    Cls
    Print " ***** X ***** Y *****"
    For IP = 1 To NP + 1
        Print Format(YY(1, IP), " 0.00000E+00 ");
        Print Format(YY(2, IP), "0.00000E+00 ");
        Print Format(YY(3, IP), "0.00000E+00 ");

```

```

        Print Format(YY(4, IP), "0.00000E+00 ");
        Print Format(YY(5, IP), "0.00000E+00 ")
    Next IP
    Print " *****"
    Print
    ' --CALCULATE EFFECTIVENESS FACTOR
    EFF = 3 * YY(3, NP + 1) / ThieleModulus / ThieleModulus
    Print ">> EFFECTIVENESS FACTOR =";
    Print Format(EFF, " 0.000000E+00")
    End Sub

Private Function RC(C)
'
' User Defined Function
'
RC = C
End Function

Private Function DRDW(C)
'
' User Defined Derivative Function
'
DRDW = 1
End Function

Private Sub DeriveFunction(Y, DY, MErr, ThieleModulus)
' =====
' User DEFINED FUNCTIONS
' =====
RX = Y(1)
DY(1) = Y(3)

If RX < MErr Then
    DY(2) = ThieleModulus ^ 2 * RC(Y(2))
Else
    DY(2) = ThieleModulus ^ 2 * RC(Y(2)) - 2 / RX * Y(3)
End If

DY(3) = Y(4)

If RX < MErr Then
    DY(4) = ThieleModulus ^ 2 * DRDW(Y(2))
Else
    DY(4) = ThieleModulus ^ 2 * DRDW(Y(2)) - 2 / RX * Y(5)
End If
End Sub

Public Sub RungeKuttaMerson(ENT%, N, Y0, Y, DY, XS, YS, W, RelErr, IPrt$, H, HOLD, Hsav,

```

```

MErr, ThieleModulus)
'
'=====
'  RUNGE KUTTA MERSON SUBROUTINE
'=====
'
' ENT%   = Entry Identification
'         ENT% = 0 for Initial Echo
'         System will adjust ENT% by itself
' N      = Number of Equations
' Y0     = Initial conditions of X and Y(I)
'         Y0(1) = Initial condition of independent variable
' Y      = Dependent Variable
' DY     = Value of Derivative Function at Xs
' XS     = Value of independent variable at which answer is required
' YS     = Solution of all the variables
' W      = Working space
' RelErr = Interval error allowed, suggested valuemax : Y0*1E-5
' IPrt$  = Control of Printing Intermediate Results
'         IPrt$ = "Y" for Printing everything
'         IPrt$ = "N" for Final Results Only
' H      = Initial step size
' HOLD   = Work Variable for Transfer Step Size Between Subroutines
' Hsav   = Initial Step Size
'
'-- Internal Variable
'
NN = N + 1
ND = NN * 2
NX = NN * 3
NY = NN * 4
'
'--Initial echo
'
If ENT% = 0 Then
    ENT% = 1
    If IPrt$ = "Y" Then
        Cls
        Print
        Print
        Print "***** X *****";
        For I = 1 To N
            Print "**** Y( "; I; ") ****";
        Next I
        Print
        For I = 1 To N + 1
            Print Format(Y0(I), " 0.000000E+");
        Next I
        Print
    End If
Else
    H = HOLD

```



```

End If
' ***** GET START
'
Do
  For I = 1 To NN
    YS(I) = Y0(I)
    Y(I) = YS(I)
  Next I
  Call DeriveFunction(Y, DY, MErr, ThieleModulus)
' --STORE Y0, CREATE Y1, DY1
'
  For I = 1 To N
    W(NN + I) = DY(I)
    W(I + 1) = YS(I + 1) + H * DY(I) / 3!
  Next I
  W(1) = YS(1) + H / 3
  For I = 1 To NN
    Y(I) = W(I)
  Next I
  Call DeriveFunction(Y, DY, MErr, ThieleModulus)
' --CREATE Y2, DY2
'
  For I = 1 To N
    W(I + 1) = YS(I + 1) + H * W(NN + I) / 6! + H * DY(I) / 6!
  Next I
  For I = 1 To NN
    Y(I) = W(I)
  Next I
  Call DeriveFunction(Y, DY, MErr, ThieleModulus)
' -- STORE DY2, CREATE Y3, DY3
'
  For I = 1 To N
    W(ND + I) = DY(I)
    W(I + 1) = YS(I + 1) + H * W(NN + I) / 8! + H * DY(I) * 3! / 8!
  Next I
  W(1) = YS(1) + H * 0.5
  For I = 1 To NN
    Y(I) = W(I)
  Next I
  Call DeriveFunction(Y, DY, MErr, ThieleModulus)
' --CREATE Y4, STORE Y4, DY3
'
  For I = 1 To N
    W(I + 1) = YS(I + 1) + H * W(NN + I) * 0.5 - W(ND + I) * H * 1.5 + H * DY(I) * 2!
    W(NX + I + 1) = W(I + 1)
    W(ND + I) = DY(I)
  Next I

```

```

W(1) = YS(1) + H
For I = 1 To NN
    Y(I) = W(I)
Next I
Call DeriveFunction(Y, DY, MErr, ThieleModulus)
,
'--CREATE Y5
,
For I = 1 To N
    W(I + 1) = YS(I + 1) + H * W(NN + 1) / 6! + H * W(ND + 1) * 2 / 3 + H * DY(I) / 6
Next I
,
'***** TEST ERRO ON EACH Y
,
TestH = H
For J = 1 To N
    EE = (Abs(W(NX + J + 1) - W(J + 1))) / 5!
    If EE > RelErr(J) Then
        H = H * (RelErr(J) / EE) ^ 2
        If IPrt$ <> "N" Then
            Print "Step size reduced to ";
            Print Format(H, "0.000000E+")
        End If
        ConvergentFlag% = 0
        Exit For
    ElseIf EE < 0.95 * RelErr(J) Then

        If (EE - 100000000000#) <= 0 Then
            TestH = 5! * H
        Else
            TestH = H * TestH ^ 2
            If (TestH - 5 * H) > 0 Then TestH = 5 * H
        End If
        If IPrt$ <> "N" Then
            Print "Step size provisional increase to ";
            Print Format(TestH, "0.000000E+")
        End If
        ConvergentFlag% = 2
    Else
        ConvergentFlag% = 2
    End If
Next J
,
' ERROR ON THAT INTERVAL WAS NICELY WITHIN BOUNDS,
' TEST WHETHER UPPER BOUND WITHIN NEXT INTERVAL OR NOT
,
If ConvergentFlag% <> 0 Then
    H = TestH
    For I = 1 To NN
        Y0(I) = W(I)
    Next I
    If (W(1) + H - XS) <= 0 Then

```

```

        If IPrt$ <> "N" Then
            For I = 1 To N + 1
                Print Format(W(I), " 0.000000E+");
            Next I
            Print
            Print "Step size = ";
            Print Format(H, " 0.000000E+")
            Print
        End If
    Else
        If H > 0 Then
            HOLD = H
        Else
            HOLD = Hsav
        End If
        H = XS - W(1)
    End If
End If
If H <= 0 Then
    ConvergentFlag% = 1
    H = Hsav
    HOLD = Hsav
End If
Loop While ConvergentFlag% <> 1
Call RKMFinalResults(N, NN, W, YS, IPrt$)
End Sub

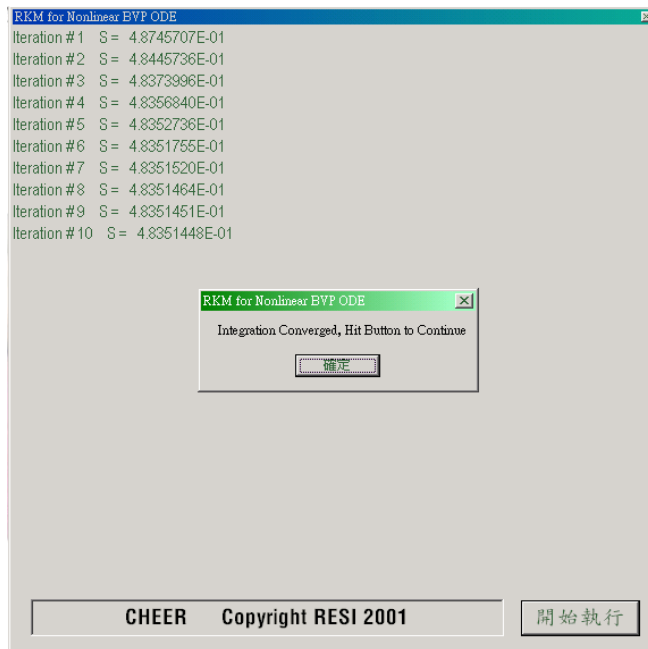
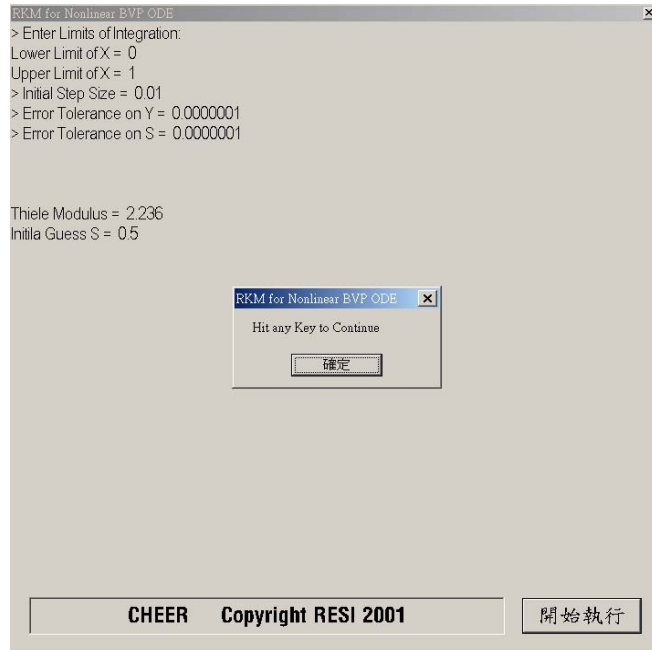
Public Sub RKMFinalResults(N, NN, W, YS, IPrt$)
' ***** FINAL VALUES
If IPrt$ <> "N" Then
    For I = 1 To N + 1
        Print Format(W(I), " 0.000000E+");
        If (Int(I / 5) - I / 5) = 0 Then Print
    Next I
    Print
End If
For I = 1 To NN
    YS(I) = W(I)
Next I
'
' CHEER-2001    by Ron-Hsin Chang
'
End Sub

```

副程式使用方法詳第八章阮奇庫塔摩森法說明。

執行結果：

1. 一次反應 $R(C) = C$



```

RKM for Nonlinear BVP ODE
***** X ***** Y *****
0.00000E+00  4.83515E-01  0.00000E+00  1.00000E+00  0.00000E+00
5.00000E-02  4.84522E-01  4.03408E-02  1.05127E+00  8.33283E-02
1.00000E-01  4.87554E-01  8.09845E-02  1.10517E+00  1.68657E-01
1.50000E-01  4.92631E-01  1.22236E-01  1.16183E+00  2.49985E-01
2.00000E-01  4.99793E-01  1.64408E-01  1.22140E+00  3.33313E-01
2.50000E-01  5.09092E-01  2.07818E-01  1.28403E+00  4.16641E-01
3.00000E-01  5.20601E-01  2.52797E-01  1.34986E+00  4.99970E-01
3.50000E-01  5.34404E-01  2.99689E-01  1.41907E+00  5.83298E-01
4.00000E-01  5.50607E-01  3.48855E-01  1.49182E+00  6.66626E-01
4.50000E-01  5.69334E-01  4.00679E-01  1.56831E+00  7.49954E-01
5.00000E-01  5.90726E-01  4.55565E-01  1.64872E+00  8.33283E-01
5.50000E-01  6.14949E-01  5.13948E-01  1.73325E+00  9.16611E-01
6.00000E-01  6.42187E-01  5.76291E-01  1.82212E+00  9.99939E-01
6.50000E-01  6.72652E-01  6.43098E-01  1.91554E+00  1.08327E+00
7.00000E-01  7.06580E-01  7.14909E-01  2.01375E+00  1.16660E+00
7.50000E-01  7.44236E-01  7.92311E-01  2.11700E+00  1.24992E+00
8.00000E-01  7.85915E-01  8.75943E-01  2.22554E+00  1.33325E+00
8.50000E-01  8.31946E-01  9.66501E-01  2.33965E+00  1.41658E+00
9.00000E-01  8.82693E-01  1.06474E+00  2.45960E+00  1.49991E+00
9.50000E-01  9.38562E-01  1.17150E+00  2.58571E+00  1.58324E+00
1.00000E+00  1.00000E+00  1.28768E+00  2.71828E+00  1.66657E+00
*****
>> EFFECTIVENESS FACTOR = 7.72654E-01

CHEER  Copyright RESI 2001  開始執行
    
```

理論上，線性問題應該不需要使用迭代即可直接求得 s 及解答，但由於解 IVP 常微分方程式時，通常都會引入計算誤差，因此，以本例為例，即需迭代十次才獲得相對誤差為 $1.0E-7$ 的收斂結果。

由於一次反應的濃度分布理論解為

$$C = \frac{\sinh(\Phi x)}{x \sinh(\Phi)}$$

在本例中 $\Phi = 2.2360$ 。當 $x = 0$ 時，可以利用 L'Hospital 定理，得到

$$C_0 = \frac{\Phi}{\sinh(\Phi)} = 0.48352$$

與數值解得 $C_0 = s = 0.483514$ ，二者相當接近。

有效度係數 (Effectiveness factor) 的理論解為 [4]：

$$\eta = \frac{3}{\Phi} \left[\frac{1}{\tanh(\Phi)} - \frac{1}{\Phi} \right] = 0.77266$$

根據有效度係數 η 的定義，數值解可利用下式計算之：

$$\eta = \frac{4 \pi r_0^2 D \left. \frac{dC}{dr} \right|_{r_0}}{\frac{4}{3} \pi r_0^3 k R(C_0)} = \frac{3v(1)}{\Phi^2}$$

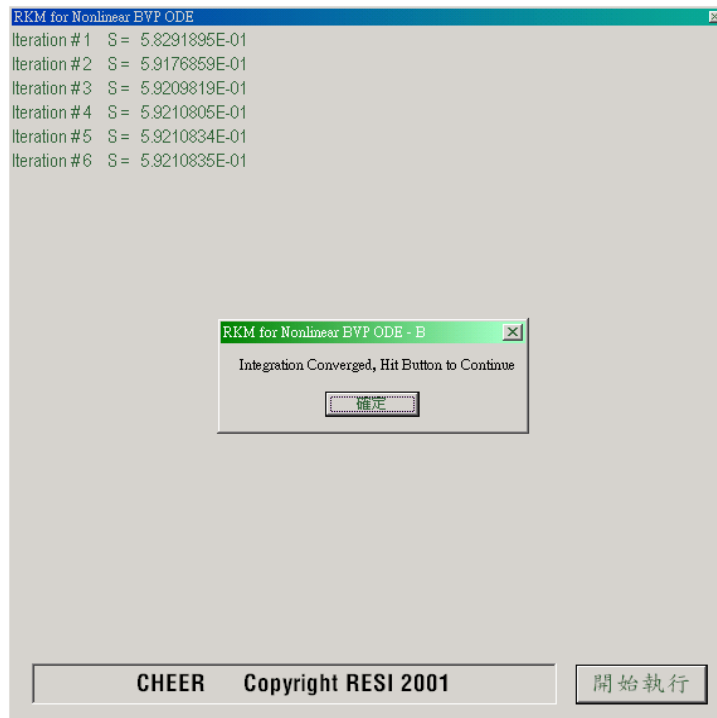
數值解法所得結果亦為 $\eta = 0.77266$ ，結果相當正確。

2. 二次反應： $R(C) = C^2$

沿用一次反應的程式，但需要先將程式中的函數及導函數定義修改如下：

```
Function RC(C)
RC = C * C
End Function
```

```
Function DRDW(C)
DRDW = 2 * C
End Function
```



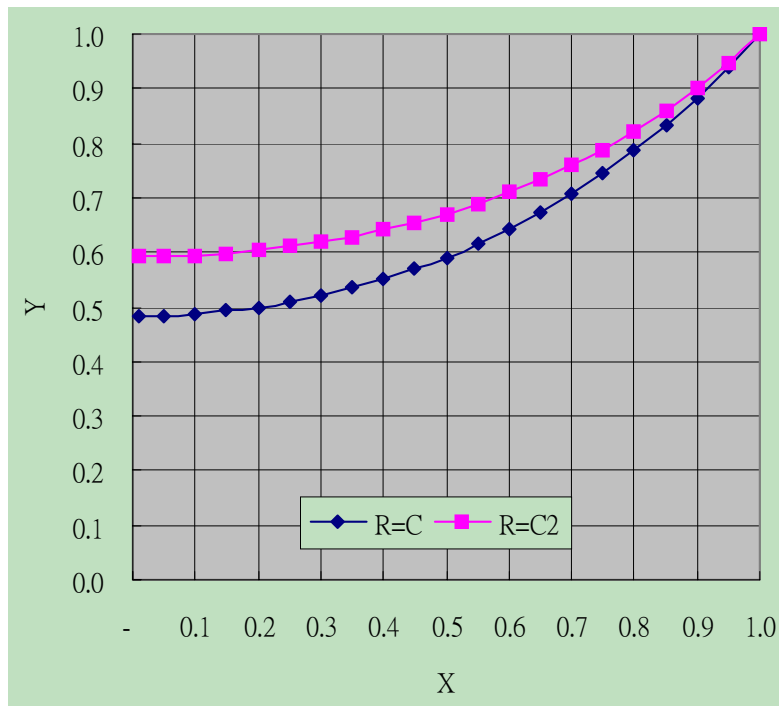
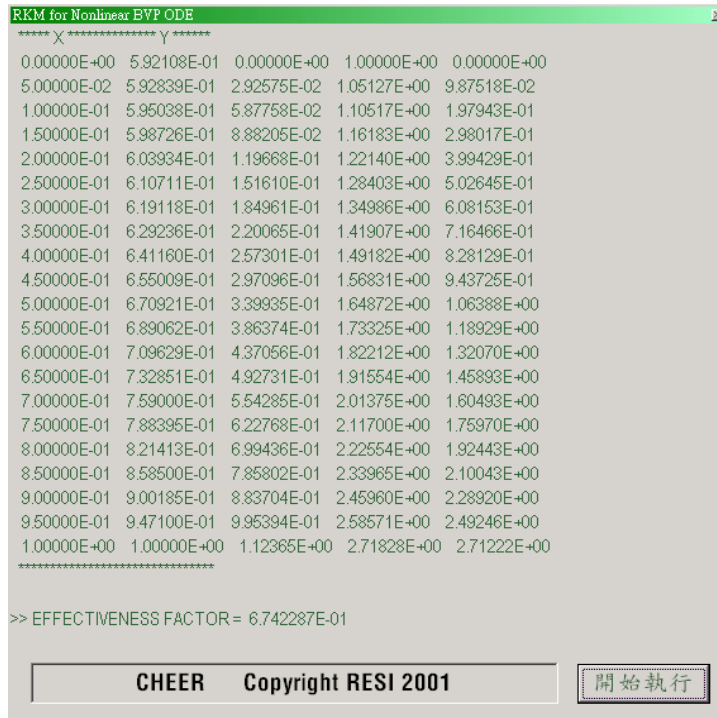


圖 9.2 一次反應及二次反應濃度分布

- (1) 一次反應及二次反應所得結果比較，如上圖所示。在觸媒中心位置 ($x=0$)， y 爲對稱，即 y 對 x 的一次導函數爲 0。
- (2) 二次反應只要迭代 6 次即可收斂。

例題 9-3 設計問題 D-IX

描述無限平板上流束函數 $f(\eta)$ 的常微分方程式爲

$$f f'' + 2 f''' = 0 \quad (9-2.15)$$

BC1 $\eta=0$ 時， $f=0$

BC2 $\eta=0$ 時， $f'=0$

BC3 $\eta=\infty$ 時， $f'=1$

此微分方程式稱爲 Blasius 方程式，試利用投射法解此問題，並求出無因次速度 U 及 V 。

解：

令 $g_1 = f, g_2 = f', g_3 = f''$ ，則可以將 Blasius 方程式改寫成以下的 IVP：

$$\frac{d}{d\eta} \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix} = \begin{bmatrix} g_2 \\ g_3 \\ -\frac{1}{2} g_1 g_3 \end{bmatrix} \quad (9-2.16)$$

IC1 $\eta=0$ ； $g_1=0$

IC2 $\eta=0$ ； $g_2=0$

IC3 $\eta=0$ ； $g_3=s$

將上式積分至 $\eta \rightarrow \infty$ ，則 g_2 需等於 1。實際執行時，此條件可以利用 $\eta = \eta_{\max}$ ： $g_2 = 1$ 取代之。積分至 $\eta = \eta_{\max}$ 時，定義殘值爲

$$R(s) = g_2(s; \eta_{\max}) - 1 \quad (9-2.17)$$

利用牛頓迭代法解此方程式，則 s 的迭代方程式爲

$$s_{k+1} = s_k - \frac{R(s)}{R'(s)} \quad (9-2.18)$$

$R(s)$ 的導函數 $R'(s)$ 可仿前例的方式續導之。將方程式 (9-2.16) 對 s 微分，得 IVP 微分方程式

$$\frac{d}{d\eta} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{bmatrix} = \begin{bmatrix} \xi_2 \\ \xi_3 \\ -\frac{1}{2}\xi_1 g_3 - \frac{1}{2}g_1 \xi_3 \end{bmatrix} \quad (9-2.19)$$

IC1 $\eta=0 ; \xi_1 = 0$

IC2 $\eta=0 ; \xi_2 = 0$

IC3 $\eta=0 ; \xi_3 = 1$

其中 ξ_i 的定義為

$$\xi_i = \frac{\partial}{\partial s} [g_i(s; \eta)] \quad (9-2.20)$$

殘值 $R(s)$ 的導函數為

$$R'(s) = \frac{\partial}{\partial s} [g_2(s; \eta_{\max}) - 1] = \xi_2(s, \eta_{\max}) \quad (9-2.21)$$

故方程式 (9-2.18) 可以簡化成

$$s_{k+1} = s_k - \frac{g_2(s, \eta_{\max}) - 1}{\xi_2(s, \eta_{\max})} \quad (9-2.22)$$

根據以上的說明，可以擬定程式設計邏輯如下：

1. 先猜測一個 s_0 值，並設定積分上限 η_{\max} 值。
2. 使用 RK4、RKM 或 RKF 等程式，將方程式 (9-2.16) 及 (9-2.19) 聯立積分至 $\eta = \eta_{\max}$ 。
3. 利用方程式 (9-2.22) 求出 s_{k+1} 。
4. 重複步驟 2 及 3，直到 s 值收斂為止。
5. 根據 U 及 V 的定義式，求出速度分布。

$$U \equiv f' = g_2$$

$$V \equiv \eta f' - f = \eta g_2 - g_1$$

程式列印

```

'
*****
'SOLUTION OF NONLINEAR B.V.P.-O.D.E.
'BY THE SHOOTING METHOD
    
```

```

*****
'
' CHEER-2001   Ron-Hsin Chang
'
' Example 9-3
' Boundary Layer Problem
'
Private Sub RK4NonlinearBVPODE(Xpos, Ypos)
'
' N=Number of dependent variables
' H=Initial step size
' Y0=Initial conditions of X and Y(I)
' Y0(1)=Initial condition of independent variable
' RelErr=Interval error allowed, suggested value max : Y0*1E-5
' XS=Value of independent variable at which answer is required
' DY=Values of derivatives at Xs
' YS=Solution of all the variables
' W=Working space
'
Dim Y0(11), RelErr(10), DY(10), YS(11), W(44), Y(11)
Dim YY(11, 101)
'
' MACHINE ERROR
'
MErr = 1E-16
'
' ===== USER'S SPACE
'
Cls
N = 6: ' Number of eqns.
NP = 20

IPrt$ = InputBox("Print Detail InformaTION <Y/N> ", "Require Detail Information", "N", Xpos, Ypos)
IPrt$ = UCase(IPrt$)

Print "> Enter Limits of Integration:"
Print "Lower Limit of X = ";
X0 = Val(InputBox(" Lower Limit X0 = ", "Low Limit of X", 0, Xpos, Ypos))
Print X0

Print "Upper Limit of X = ";
XS = Val(InputBox(" Upper Limit XS = ", "Upper Limit XS", XS, Xpos, Ypos))
Print XS

Print "> Initial Step Size = ";
H = Val(InputBox("> Initial Step Size = ", "Step Size", (XS - X0) / 100, Xpos, Ypos))
Hsav = H
Print H
'

```

```

'--ERROR TOLLERANCE
'
Print "> Error Tolerance on Y = ";
E1 = Val(InputBox("> ERROR TOLERANCE ON Y = ", "Error Tolerance", H * 0.00001, Xpos, Ypos))
Print E1

Print "> Error Tolerance on S = ";
E2 = Val(InputBox("> ERROR TOLERANCE ON S = ", "Error Tolerance", H * 0.00001, Xpos, Ypos))
Print E2

For I = 1 To N
    RelErr(I) = E1
Next I
'

'--PRINTING CONTROL
'
Print
NP = Val(InputBox("> NUMBER OF PRINTING INTERVALS = (<50) ", "NP", NP, Xpos, Ypos))
XPT = (XS - X0) / NP
Print
'

'--INITIAL CONDITIONS
'
S = Val(InputBox("> Enter Initial Guess : S= ", "S", S, Xpos, Ypos))
Print "Initial Guess S = "; S
Iteration% = 0
MsgBox "Hit any Key to Continue"

Cls
'

'--START ITERATION
'
Do
    Y0(1) = 0
    Y0(2) = 0
    Y0(3) = 0
    Y0(4) = S
    Y0(5) = 0
    Y0(6) = 0
    Y0(7) = 1
    ENT% = 0
    Iteration% = Iteration% + 1
    For I = 1 To N + 1
        YY(I, 1) = Y0(I)
    Next I

    IPT = 1
    Do
        XS = Y0(1) + XPT

```

```

'--CALL KUTTA-MERSON
    Call RungeKuttaMerson(ENT%, N, Y0, Y, DY, XS, YS, W, RelErr, IPrt$, H, HOLD, Hsav, MErr)
    For LP = 1 To N + 1
        Y0(LP) = YS(LP)
        YY(LP, IPT + 1) = YS(LP)
    Next LP
    If HOLD > XPT Then HOLD = XPT
    IPT = IPT + 1
    Loop While IPT <= NP

'--CALCULATE R(S)
    R = YS(3) - 1
'--CALCULATE R'(S)
    RP = YS(6)
'--CALCULATE S
    SOLD = S
    S = S - R / RP
    Print "Iteration #", Iteration%; " S = ";
    Print Format(S, " 0.0000000E+00")
'--CHECK CONVERGENCE ON S
    If Abs((S - SOLD) / S) < E2 Or Abs(S - SOLD) < E2 Then
        ConvergentFlag% = 1
    Else
        H = Hsav
        ConvergentFlag% = 0
    End If
    Loop While ConvergentFlag% = 0
    MsgBox ("Integration Converged, Hit Button to Continue")

,

'--FINAL FUNCTION VALUE
,

Cls
Print " ***** X ***** Y *****"
For IP = 1 To NP + 1
    For JP = 1 To 4
        Print Format(YY(JP, IP), " 0.00000E+00 ");
    Next JP
    Print
Next IP
MsgBox "Ready to Print U & V"

Cls
Print
Print " ***** ETA ***** U ***** V *****"
For IP = 1 To NP + 1
    Print Format(YY(1, IP), " 0.00000E+00 ");
    Print Format(YY(3, IP), " 0.00000E+00 ");
    Z = YY(1, IP) * YY(3, IP) - YY(2, IP)

```

```

        Print Format(Z, " 0.00000E+00 ")
Next IP

End Sub

Private Sub DeriveFunction(Y, DY)
' =====
'   User DEFINED FUNCTIONS
' =====
DY(1) = Y(3)
DY(2) = Y(4)
DY(3) = -0.5 * Y(2) * Y(4)
DY(4) = Y(6)
DY(5) = Y(7)
DY(6) = -0.5 * (Y(5) * Y(4) + Y(2) * Y(7))
End Sub

Public Sub RungeKuttaMerson(ENT%, N, Y0, Y, DY, XS, YS, W, RelErr, IPrt$, H, HOLD, Hsav, MErr)
'
' =====
'   RUNGE KUTTA MERSON SUBROUTINE
' =====
'
' ENT%   = Entry Identification
'         ENT% = 0 for Initial Echo
'         System will adjust ENT% by itself
' N      = Number of Equations
' Y0     = Initial conditions of X and Y(l)
'         Y0(1) = Initial condition of independent variable
' Y      = Dependent Variable
' DY     = Value of Derivative Function at Xs
' XS     = Value of independent variable at which answer is required
' YS     = Solution of all the variables
' W      = Working space
' RelErr = Interval error allowed, suggested valuemax : Y0*1E-5
' IPrt$  = Control of Printing Intermediate Results
'         IPrt$ = "Y" for Printing everything
'         IPrt$ = "N" for Final Results Only
' H      = Initial step size
' HOLD   = Work Variable for Transfer Step Size Between Subroutines
' Hsav   = Initial Step Size
'
' -- Internal Variable
'
NN = N + 1
ND = NN * 2
NX = NN * 3

```

```

NY = NN * 4

'
'--Initial echo
'
If ENT% = 0 Then
    ENT% = 1
    If IPrt$ = "Y" Then
        Cls
        Print
        Print
        Print "***** X *****";
        For I = 1 To N
            Print "**** Y(;" I; ") ****";
        Next I
        Print
        For I = 1 To N + 1
            Print Format(Y0(I), " 0.000000E+");
        Next I
        Print
    End If
Else
    H = HOLD
End If

'
' ***** GET START
'
Do
    For I = 1 To NN
        YS(I) = Y0(I)
        Y(I) = YS(I)
    Next I
    Call DeriveFunction(Y, DY)

'
'--STORE Y0, CREATE Y1, DY1
'

    For I = 1 To N
        W(NN + I) = DY(I)
        W(I + 1) = YS(I + 1) + H * DY(I) / 3!
    Next I

    W(1) = YS(1) + H / 3

    For I = 1 To NN
        Y(I) = W(I)
    Next I
    Call DeriveFunction(Y, DY)

```

```

'--CREATE Y2, DY2
'
  For I = 1 To N
    W(I + 1) = YS(I + 1) + H * W(NN + I) / 6! + H * DY(I) / 6!
  Next I
  For I = 1 To NN
    Y(I) = W(I)
  Next I
  Call DeriveFunction(Y, DY)
'
'-- STORE DY2, CREATE Y3, DY3
'
  For I = 1 To N
    W(ND + I) = DY(I)
    W(I + 1) = YS(I + 1) + H * W(NN + I) / 8! + H * DY(I) * 3! / 8!
  Next I
  W(1) = YS(1) + H * 0.5
  For I = 1 To NN
    Y(I) = W(I)
  Next I
  Call DeriveFunction(Y, DY)
'
'--CREATE Y4, STORE Y4, DY3
'
  For I = 1 To N
    W(I + 1) = YS(I + 1) + H * W(NN + I) * 0.5 - W(ND + I) * H * 1.5 + H * DY(I) * 2!
    W(NX + I + 1) = W(I + 1)
    W(ND + I) = DY(I)
  Next I
  W(1) = YS(1) + H
  For I = 1 To NN
    Y(I) = W(I)
  Next I
  Call DeriveFunction(Y, DY)
'
'--CREATE Y5
'
  For I = 1 To N
    W(I + 1) = YS(I + 1) + H * W(NN + I) / 6! + H * W(ND + I) * 2 / 3 + H * DY(I) / 6
  Next I
'
*****TEST ERRO ON EACH Y
'
  TestH = H
  For J = 1 To N
    EE = (Abs(W(NX + J + 1) - W(J + 1))) / 5!
    If EE > RelErr(J) Then
      H = H * (RelErr(J) / EE) ^ 2
    End If
  Next J

```

```

        If IPrt$ <> "N" Then
            Print "Step size reduced to ";
            Print Format(H, " 0.000000E+")
        End If
        ConvergentFlag% = 0
        Exit For
    ElseIf EE < 0.95 * RelErr(J) Then

        If (EE - 1000000000000#) <= 0 Then
            TestH = 5! * H
        Else
            TestH = H * TestH ^ 2
            If (TestH - 5 * H) > 0 Then TestH = 5 * H
        End If
        If IPrt$ <> "N" Then
            Print "Step size provisional increase to ";
            Print Format(TestH, " 0.000000E+")
        End If
        ConvergentFlag% = 2
    Else
        ConvergentFlag% = 2
    End If
Next J
,
' ERROR ON THAT INTERVAL WAS NICELY WITHIN BOUNDS,
' TEST WHETHER UPPER BOUND WITHIN NEXT INTERVAL OR NOT
,

    If ConvergentFlag% <> 0 Then
        H = TestH
        For I = 1 To NN
            Y0(I) = W(I)
        Next I
        If (W(1) + H - XS) <= 0 Then
            If IPrt$ <> "N" Then
                For I = 1 To N + 1
                    Print Format(W(I), " 0.000000E+");
                Next I
                Print
                Print "Step size = ";
                Print Format(H, " 0.000000E+")
                Print
            End If
        Else
            If H > 0 Then
                HOLD = H
            Else
                HOLD = Hsav
            End If
        End If
    End If

```



```

        H = XS - W(1)
    End If
End If

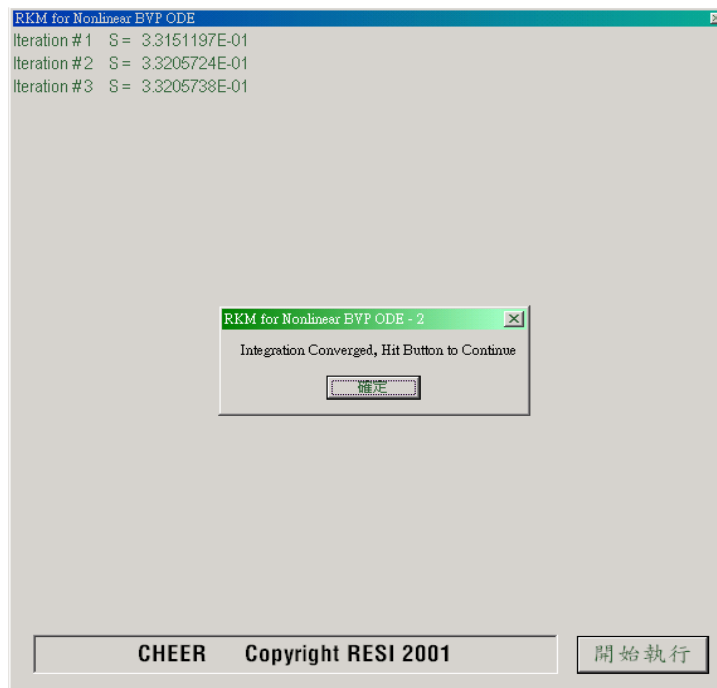
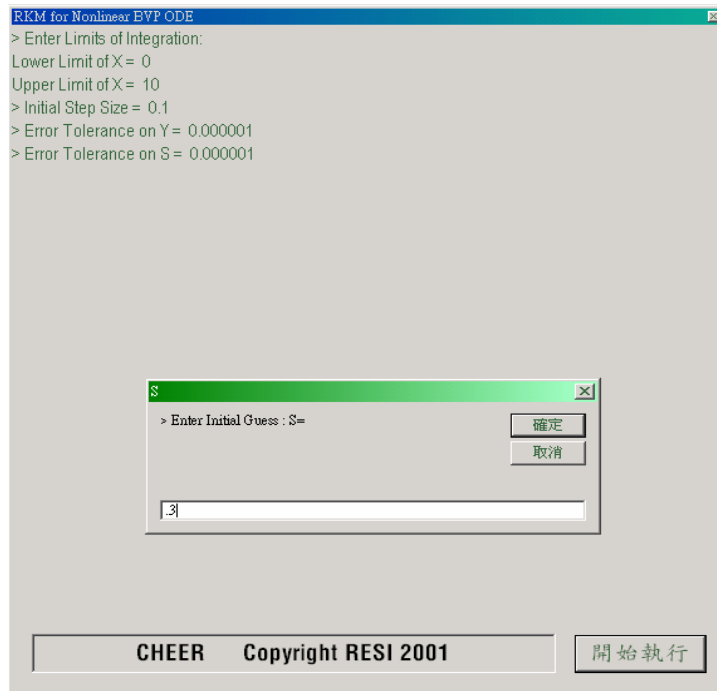
If H <= 0 Then
    ConvergentFlag% = 1
    H = Hsav
    HOLD = Hsav
End If
Loop While ConvergentFlag% <> 1
Call RKMFinalResults(N, NN, W, YS, IPrt$)
End Sub

Public Sub RKMFinalResults(N, NN, W, YS, IPrt$)
' ***** FINAL VALUES
If IPrt$ <> "N" Then
    For I = 1 To N + 1
        Print Format(W(I), "0.000000E+");
        If (Int(I / 5) - I / 5) = 0 Then Print
    Next I
    Print
End If
For I = 1 To NN
    YS(I) = W(I)
Next I
'
' CHEER-2001    by Ron-Hsin Chang
'
End Sub

```

1. 副程式使用方法詳第八章阮奇庫塔摩森法說明。
2. 本例所使用副程式與例 9-2 相同，建議讀者比較主程式及所定義函數部分之差異，應該很容易可以了解這類問題的解法。
3. 在本例所使用程式中，已將前例中的變數 **ThieleModulus** 除去，讀者應注意副程式 **Private Sub DeriveFunction(Y, DY)**，並與例 9-2 作比較，應可了解必要時如何增加變數之作法。

程式執行結果



第 9 章 常微分方程式 — 邊界值問題

RKM for Nonlinear BVP ODE

```

***** X ***** Y *****
0.000000E+00 0.000000E+00 0.000000E+00 3.3205724E-01
5.000000E-01 4.1492969E-02 1.6588519E-01 3.3091087E-01
1.000000E+00 1.6557183E-01 3.2977992E-01 3.2300702E-01
1.500000E+00 3.7013858E-01 4.8678910E-01 3.0258041E-01
2.000000E+00 6.5002431E-01 6.2976550E-01 2.6675147E-01
2.500000E+00 9.9631091E-01 7.5125942E-01 2.1741153E-01
3.000000E+00 1.3968079E+00 8.4604414E-01 1.6136029E-01
3.500000E+00 1.8376981E+00 9.1304008E-01 1.0777262E-01
4.000000E+00 2.3057457E+00 9.5551792E-01 6.4234115E-02
4.500000E+00 2.7901335E+00 9.7951399E-01 3.3980884E-02
5.000000E+00 3.2832727E+00 9.9154160E-01 1.5906794E-02
5.500000E+00 3.7805708E+00 9.9687852E-01 6.5785949E-03
6.000000E+00 4.2796197E+00 9.9897258E-01 2.4020470E-03
6.500000E+00 4.7793209E+00 9.9969877E-01 7.7412164E-04
7.000000E+00 5.2792373E+00 9.9992131E-01 2.2017976E-04
7.500000E+00 5.7792163E+00 9.9998157E-01 5.5266537E-05
8.000000E+00 6.2792116E+00 9.9999598E-01 1.2243142E-05
8.500000E+00 6.7792106E+00 9.9999903E-01 2.3818402E-06
9.000000E+00 7.2792103E+00 9.9999960E-01 4.0716385E-07
9.500000E+00 7.7792101E+00 9.9999969E-01 6.0961536E-08
1.000000E+01 8.2792099E+00 9.9999971E-01 7.9525573E-09
    
```

RKM for Nonlinear BVP ODE - 2

Ready to Print U & V

CHEER

RKM for Nonlinear BVP ODE

```

-----ETA-----U-----V-----
0.000000E+00 0.000000E+00 0.000000E+00
5.000000E-01 1.6588519E-01 4.1449624E-02
1.000000E+00 3.2977992E-01 1.6420808E-01
1.500000E+00 4.8678910E-01 3.6004508E-01
2.000000E+00 6.2976550E-01 6.0950669E-01
2.500000E+00 7.5125942E-01 8.8183765E-01
3.000000E+00 8.4604414E-01 1.1413245E+00
3.500000E+00 9.1304008E-01 1.3579422E+00
4.000000E+00 9.5551792E-01 1.5163259E+00
4.500000E+00 9.7951399E-01 1.6176794E+00
5.000000E+00 9.9154160E-01 1.6744353E+00
5.500000E+00 9.9687852E-01 1.7022611E+00
6.000000E+00 9.9897258E-01 1.7142158E+00
6.500000E+00 9.9969877E-01 1.7187211E+00
7.000000E+00 9.9992131E-01 1.7202119E+00
7.500000E+00 9.9998157E-01 1.7206454E+00
8.000000E+00 9.9999598E-01 1.7207563E+00
8.500000E+00 9.9999903E-01 1.7207812E+00
9.000000E+00 9.9999960E-01 1.7207861E+00
9.500000E+00 9.9999969E-01 1.7207870E+00
1.000000E+01 9.9999971E-01 1.7207871E+00
    
```

CHEER Copyright RESI 2001

結果討論

1. 利用本節所介紹的方法， s 只需迭代三次即可收斂。雖然利用此方法需聯立解六個聯立一次常微分方程式，但由於收斂快，因此，所需計算時間仍相當短。
2. 投射法有許多處理方式，Carnahan et.al. [7] 利用線性內插投射法解同一問題，共需迭代 20 次， s 值才會收斂至 1.0×10^{-6} 的準備度。所需計算時間約為本程式的 5 倍。茲介紹線性內插投射法如下：

(1) 假設 $s = s_L$ 使 $R(s_L) < 0$ ， $s = s_R$ 使 $R(s_R) > 0$ 。

(2) 利用線性內插，求 $R(s) = 0$ 時的 s 值，

$$s_{k+1} = s_1 - \frac{R(s_L)}{R(s_L) - R(s_R)}(s_R - s_L)$$

(3) 積分 IVP 方程式 (9-2.16)，求新的 $R(s)$ 值。

3. 解本問題時，假設 $\eta_{\max} = 10$ 。事實上，當 $\eta = 6$ 時，假設 $f' = 1$ 的誤差約為 0.1%，已可將此處的流體流速視為 u_∞ 。U 及 V 與 η 的關係如圖 9.3 所示。

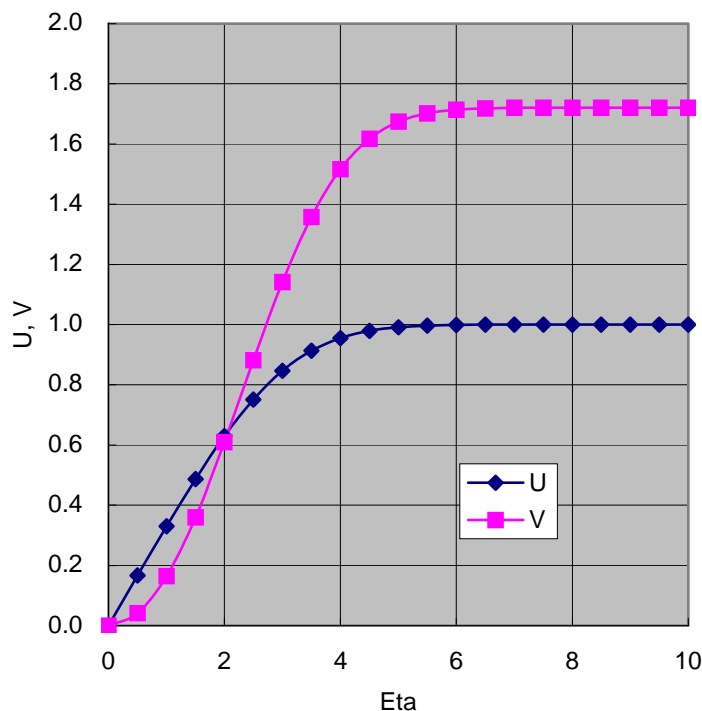


圖 9.3 計算所得 U 及 V 與 η 關係

第三節 有限差分法

Visual Basic

我們在第六章中曾討論了利用有限差分法求導函數近似值的作法。如果我們將微分方程式中的導函數利用這種表示方法來取代，則可將微分方程式轉換成差分方程式，所得的解可作為原微分方程式的近似解；這種方法就稱為有限差分法 (Finite Difference Method)。

有限差分法與投射法各有長處，優劣甚難論斷，但一般認為差分法有時略較投射法易於使用。

有限差分法基本上將 BVP 微分方程式改寫成差分方程式，將微分方程式的求解變成較簡單的聯立代數方程式，例如，考慮 BVP 微分方程式

$$y'' + p(x)y' + q(x)y = g(x); \quad a < x < b \quad (9-3.1)$$

$$\text{BC1} \quad x = a; \quad y = A$$

$$\text{BC2} \quad x = b; \quad y = B$$

首先將 $[a, b]$ 區間分割成 n 等份，使

$$x_i = x_0 + ih; \quad i = 1, 2, \dots, n \quad (9-3.2)$$

其中 $x_0 = a, x_n = b$ ，且

$$h = \frac{b-a}{n} \quad (9-3.3)$$

x_i 稱為節點 (node)，表示要求解 y_i 的位置； h 稱為積分間距 (step size)。

函數 $y(x)$ 利用泰勒級數展開，得

$$y(x \pm h) = y(x) \pm h \frac{\partial y}{\partial x}(x) + \frac{h^2}{2!} \frac{\partial^2 y}{\partial x^2}(x) \pm \frac{h^3}{3!} \frac{\partial^3 y}{\partial x^3}(x) + O(h^4) \quad (9-3.4)$$

因此， $y(x)$ 的一次導函數可利用一階差分式表示為：

$$\begin{aligned} y'(x) &= \frac{y(x+h) - y(x)}{h} - \frac{h}{2} y''(x) - \frac{h^2}{6} y'''(x) + \dots \\ &\equiv \frac{y(x+h) - y(x)}{h} + O(h) \quad \text{正向差分} \end{aligned} \quad (9-3.5)$$

$$\begin{aligned}
 y'(x) &= \frac{y(x) - y(x-h)}{h} + \frac{h}{2} y''(x) - \frac{h^2}{6} y'''(x) + \dots & (9-3.6) \\
 &\equiv \frac{y(x) - y(x-h)}{h} + O(h) \quad \text{逆向差分}
 \end{aligned}$$

若將上二式相加，則得 $y'(x)$ 的二階差分式：

$$\begin{aligned}
 y'(x) &= \frac{y(x+h) - y(x-h)}{2h} - \frac{h^2}{6} y'''(x) + \dots & (9-3.7) \\
 &\equiv \frac{y(x+h) - y(x-h)}{2h} + O(h^2)
 \end{aligned}$$

$y(x)$ 的二次導函數差分式可由 (9-3.4) 正負符號兩種表示式相加，可以得到

$$y(x+h) + y(x-h) = 2y(x) + h^2 y''(x) + O(h^4)$$

整理之，得到 $y''(x)$ 的差分近似表示式為

$$y''(x) = \frac{y(x+h) - 2y(x) + y(x-h)}{h^2} + O(h^2) \quad (9-3.8)$$

利用方程式 (9-3.7) 及 (9-3.8)，可將微分方程式 (9-3.1) 改寫成二階準確度的有限差分近似式

$$\begin{aligned}
 \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + p(x_i) \frac{y_{i+1} - y_{i-1}}{2h} + q(x_i) y_i &= g(x_i) \\
 i &= 1, 2, \dots, N-1 & (9-3.9)
 \end{aligned}$$

其中

$$y_0 = A, y_N = B$$

將上式乘以 $2h^2$ ，整理後得

$$\beta_i y_{i-1} + \alpha_i y_i + \gamma_i y_{i+1} = f(x_i), \quad i = 1, 2, \dots, N-1 \quad (9-3.10)$$

其中

$$\begin{aligned}
 \beta_i &= 2 - hp(x_i) \\
 \alpha_i &= -4 + 2h^2 q(x_i) \\
 \gamma_i &= 2 + hp(x_i) \\
 f(x_i) &= 2h^2 g(x_i)
 \end{aligned}$$

將方程式 (9-3.10) 改寫成矩陣符號為

$$\underline{A} \underline{y} = \underline{F} \quad (9-3.11)$$

其中

$$\underline{y} = [y_1, y_2, \dots, y_{N-1}]^T$$

$$\underline{F} = [f(x_1) - \beta_1 A, f(x_2), f(x_3), \dots, f(x_{N-1}) - \gamma_{N-1} B]^T$$

$$\underline{A} = \begin{bmatrix} \alpha_1 & \gamma_1 & 0 & 0 & & & & 0 \\ \beta_2 & \alpha_2 & \gamma_2 & 0 & & & & \\ 0 & \beta_3 & \alpha_3 & \gamma_3 & & & & \\ & & \cdot & \cdot & \cdot & & & \\ & & & & & \beta_{N-2} & \alpha_{N-2} & \gamma_{N-2} \\ 0 & & & & & 0 & \beta_{N-1} & \alpha_{N-1} \end{bmatrix}$$

矩陣 \underline{A} 稱為三對角線矩陣。矩陣方程式 $\underline{A}\underline{y} = \underline{F}$ 的解法可利用本書第四章所介紹的高斯消去法或三對角線矩陣方程式解法求之，即可求得近似解 \underline{y} 。

1. 等間距有限差分法一次導函數 hy' 之表示式

編號	乘數	不同表示式的係數						誤差	
101	1	-1	1					-1/2	$hy'(\xi)$
102		-1	1					1/2	
103	1/2	-3	4	-1				1/3	$h^2 y''(\xi)$
104		-1	0	1				-1/6	
105		1	-4	3				1/3	
106	1/6	-11	18	-9	2			-1/4	$h^3 y^{iv}(\xi)$
107		-2	-3	6	-1			1/12	
108		1	-6	3	2			-1/12	
109		-2	9	-18	11			1/4	
110	1/12	-25	48	-36	16	-3		1/5	$h^4 y^{v}(\xi)$
111		-3	-10	18	-6	1		-1/20	
112		1	-8	0	8	-1		1/30	
113		-1	6	-18	10	3		-1/20	
114		3	-16	36	-48	25		1/5	
115	1/60	-137	300	-300	200	-75	12	-1/6	$h^5 y^{vi}(\xi)$
116		-12	-65	120	-60	20	-3	1/30	
117		3	-30	-20	60	-15	2	-1/60	
118		-2	15	-60	20	30	-3	1/60	
119		3	-20	60	-120	65	12	-1/30	
120		-12	75	-200	300	-300	137	1/6	

編號	乘數	不同表示式的係數							誤差	
121	1/60	-147	360	-450	400	-225	72	-10	1/7	$h^6 y^{vii}(\xi)$
122		-10	-77	150	-100	50	-15	2	-1/42	
123		2	-24	-35	80	-30	8	-1	1/105	
124		-1	9	-45	0	45	-9	1	-1/140	
125		1	-8	30	-80	35	24	-2	1/105	
126		-2	15	-50	100	-150	77	10	-1/42	
127		10	-72	225	-400	450	-360	147	1/7	

2. 等間距有限差分法一次導函數 $h^2 y''$ 之表示式

編號	乘數	不同表示式的係數					誤差	
201	1	1	-2	1			-	$hy'''(\xi)$
202		1	-2	1			-1/12	$h^2 y^{iv}(\xi)$
203		1	-2	1				$hy'''(\xi)$
204	1/6	12	-30	24	-6		11/12	$h^2 y^{iv}(\xi)$
205		6	-12	6	0		-1/12	
206		0	6	-12	6		1/12	
207		-6	24	-30	12		11/12	
208	1/24	70	-208	228	-112	22	-5/6	$h^3 y^v(\xi)$
209		22	-40	12	8	-2	1/12	$h^4 y^{vi}(\xi)$
210		-2	32	-60	32	-2	1/90	
211		-2	8	12	-40	22	-1/12	
212		22	-112	228	-208	70	5/6	

3. 等間距有限差分法一次導函數 $h^3 y'''$ 之表示式

編號	乘數	不同表示式的係數						誤差	
301	1	-1	3	-3	1			-2/3	$hy^{iv}(\xi)$
302		-1	3	-3	1			-1/2	
303		-1	3	-3	1			1/2	
304		-1	3	-3	1			3/2	
305	1/2	-5	18	-24	14	-3		7/4	$h^2 y^v(\xi)$
306		-3	10	-12	6	-1		1/4	
307		-1	2	0	-2	1		-1/4	
308		1	-6	12	-10	3		1/4	
309		3	-14	24	-18	5		7/4	

4. 等間距有限差分法一次導函數 $h^4 y^{(4)}$ 之表示式

編號	乘數	不同表示式的係數							誤差	
401	1	1	-4	6	-4	1			-2	$hy'(\xi)$
402		1	-4	6	-4	1			-1	
403		1	-4	6	-4	1			-1/6	$h^2 y^{(4)}(\xi)$
404		1	-4	6	-4	1			1	$hy'(\xi)$
405		1	-4	6	-4	1			2	

陰影部分表示基準點。例如第 115 號方程式為

$$y'_j = y'(x_j) = \frac{1}{60h} [-137y_j + 300y_{j+1} - 300y_{j+2} + 200y_{j+3} - 75y_{j+4} + 12y_{j+5}]$$

以上各表所列資料是利用有限差分法解常微分方程式時常見的差分近似式歸納表，列出所有表示式的目的，是讓使用者建立邊界條件表示式時，易於使用。其中最常被使用的表示式，依表中所列，整理如下：

1. 一次導函數

$$y'(x_i) = \frac{y_{i+1} - y_i}{h} + O(h)$$

$$y'(x_i) = \frac{y_i - y_{i-1}}{h} + O(h)$$

$$y'(x_i) = \frac{y_{i+1} - y_{i-1}}{2h} + O(h^2)$$

$$y'(x_i) = \frac{-3y_i + 4y_{i-1} - y_{i-2}}{2h} + O(h^2)$$

$$y'(x_i) = \frac{3y_i - 4y_{i-1} + y_{i-2}}{2h} + O(h^2)$$

$$y'(x_i) = \frac{-y_{i-2} + 8y_{i-1} - 8y_i + y_{i+1}}{12h} + O(h^4)$$

2. 二次導函數

$$y''(x_i) = \frac{y_i - 2y_{i+1} + y_{i+2}}{h^2} + O(h)$$

$$y''(x_i) = \frac{y_i - 2y_{i-1} + y_{i-2}}{h^2} + O(h)$$

$$y''(x_i) = \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + O(h^2)$$

$$y''(x_i) = \frac{-y_{i+3} + 4y_{i+2} - 5y_{i+1} + 2y_i}{h^2} + O(h^2)$$

$$y''(x_i) = \frac{2y_i - 5y_{i-1} + 4y_{i-2} - y_{i-3}}{h^2} + O(h^2)$$

$$y''(x_i) = \frac{-y_{i-2} + 16y_{i+1} - 30y_i + 16y_{i-1} - y_{i-2}}{12h^2} + O(h^4)$$

3. 三次導函數

$$y'''(x_i) = \frac{y_{i+2} - 2y_{i+1} + 2y_{i-1} - y_{i-2}}{2h^3} + O(h^2)$$

$$y'''(x_i) = \frac{-3y_{i+4} + 14y_{i+3} - 24y_{i+2} + 18y_{i+1} - 5y_i}{2h^3} + O(h^2)$$

$$y'''(x_i) = \frac{5y_i - 18y_{i-1} + 24y_{i-2} - 14y_{i-3} + 3y_{i-4}}{2h^3} + O(h^2)$$

4. 四次導函數

$$y^{iv}(x_i) = \frac{y_{i+2} - 4y_{i+1} + 6y_i - 4y_{i-1} + y_{i-2}}{h^4} + O(h^2)$$

例題 9-4 有限差分法

利用有限差分法重解例 9-1，並與理論解作比較。

$$y'' - 3y = 2x, \quad 0 < x < 1$$

BC1 $x=0$, $y=0$

BC2 $x=1$, $y=1$

解：

令 $N=5, h=0.2$, 則微分方程式可寫成有限差分形式

$$\frac{1}{0.04} [y_{i-1} - 2y_i + y_{i+1}] - 3y_i = 2x_i, \quad i=1, 2, 3, 4$$

$$x_i = a + ih = 0.2i, \quad x_0 = 0, \quad x_5 = 1$$

將邊界條件放入差分式中，可得以下四個聯立線性方程式

$$\begin{aligned} -2.12y_1 + y_2 &= 0.016 \\ y_1 - 2.12y_2 + y_3 &= 0.032 \\ y_2 - 2.12y_3 + y_4 &= 0.048 \\ y_3 - 2.12y_4 &= -0.936 \end{aligned}$$

程式列印： (三對角線矩陣方程式解法詳見第四章)

```

*****
'      SOLUTION OF B.V.P.-O.D.E
'      BY FINITE DIFFERENCE METHOD
*****
'
'      CHEER 2001
'
Sub FiniteDifferenceMethod(Xpos, Ypos)
'
'      y'' + P(x) y' + Q(x) y = g(x)
'      BC1  x= lower  y=A
'      BC2  x= upper  y=B
'
Dim Alpha(100), Beta(100), Gama(100), Z(100)
'
' TRIDIAGONAL MATRIX EQUATION
' Alpha,Beta,Gama=COEFFICIENT VECTOR
' Z=CONSTANT VECTOR
'/RETURN : SOLUTION
'
' Boundar Conditions to be defined by User
'
Lower = 0
Upper = 1
A = 0
B = 1
    
```

```

Do
    Cls
    Print
    Nmesh% = Val(InputBox("> NUMBER OF MESHES  N=", "", Nmesh%, Xpos, Ypos))
    Print "Number of Meshes N = "; Nmesh%

    Print
    PNT$ = InputBox("> Print Out the Matrix Equation  <Y/N>", "Print Equation", "N", Xpos, Ypos)

    Neqn% = Nmesh% - 1
    Call CoeffMaker(Alpha, Beta, Gama, Z, Neqn%, A, B, Lower, Upper, PNT$)
    Call FiniteDifference(Neqn%, A, B, Alpha, Beta, Gama, Z)

    ContinueYN$ = InputBox("Run Next Job <Y/N>", "", "N", Xpos, Ypos)
Loop Until ContinueYN$ = "N" Or ContinueYN$ = "n"
End Sub

Sub EqnCoeff(P, Q, G, x)
'
' Equation Coefficients
' To Be Defined by User
'
P = 0
Q = -3
G = 2 * x
End Sub

Sub CoeffMaker(Alpha, Beta, Gama, Z, Neqn%, A, B, Lower, Upper, PNT$)
'
' Convert 2nd Order ODE to Finite Difference Form
'
h = (Upper - Lower) / (Neqn% + 1)
For I = 1 To Neqn%
    x = Lower + h * I
    Call EqnCoeff(P, Q, G, x)
    Alpha(I) = -4 + 2 * h * h * Q
    Beta(I) = 2 - h * P
    Gama(I) = 2 + h * P
    Z(I) = 2 * h * h * G
Next I
Z(1) = Z(1) - Beta(1) * A
Z(Neqn%) = Z(Neqn%) - Gama(Neqn%) * B
Beta(1) = 0
Gama(Neqn%) = 0

If PNT$ <> "Y" Then Exit Sub
'
' Print Matrix Equation
' Eqn # 1
AB$ = "
Print Format(Alpha(1), " 0.000E+00 ");

```

```

Print Format(Gama(1), " 0.000E+00 ");
For I = 3 To Neqn%
  Print AB$;
Next I
Print Format(Z(1), " 0.000E+00")
'
' Eqn #2 -- N-1
For I = 2 To Neqn% - 1
  If I > 2 Then
    For J = 3 To I
      Print AB$;
    Next J
  End If
  Print Format(Beta(I), " 0.000E+00 ");
  Print Format(Alpha(I), " 0.000E+00 ");
  Print Format(Gama(I), " 0.000E+00 ");
  If I < Neqn% - 1 Then
    For J = I + 1 To Neqn% - 1
      Print AB$;
    Next J
  End If
  Print Format(Z(I), " 0.000E+00")
Next I
'
' Eqn #N
For I = 1 To Neqn% - 2
  Print AB$;
Next I
Print Format(Beta(Neqn%), " 0.000E+00 ");
Print Format(Alpha(Neqn%), " 0.000E+00 ");
Print Format(Z(Neqn%), " 0.000E+00")
MsgBox ("Ready to Solve the Tridiagonal Matrix Equation")
End Sub

Sub FiniteDifference(Neqn%, A, B, Alpha, Beta, Gama, Z)
'-----
'ECHO
'
' SOLVE & PRINT RESULT
Call Jacobi(Neqn%, Alpha, Beta, Gama, Z)
Cls
Print "**** SOLUTION:"
Print
Print Format(A, " 0.000000E+00 ")
For I = 1 To Neqn%
  Print Format(Z(I), " 0.000000E+00 ")
Next I
Print Format(B, " 0.000000E+00 ")
MsgBox ("Hit Button to Continue")
End Sub

```

```

Sub Jacobi(Neqn%, Alpha, Beta, Gama, Z)
'SUBROUTIN JACOBI
Gama(1) = Gama(1) / Alpha(1)
Z(1) = Z(1) / Alpha(1)
'--LU DECOMPOSITION
For I = 2 To Neqn%
    Alpha(I) = Alpha(I) - Beta(I) * Gama(I - 1)
    Gama(I) = Gama(I) / Alpha(I)
    Z(I) = (Z(I) - Beta(I) * Z(I - 1)) / Alpha(I)
Next I
'--BACK SUUBSTITUTION
For I = Neqn% - 1 To 1 Step -1
    Z(I) = Z(I) - Gama(I) * Z(I + 1)
Next I
End Sub

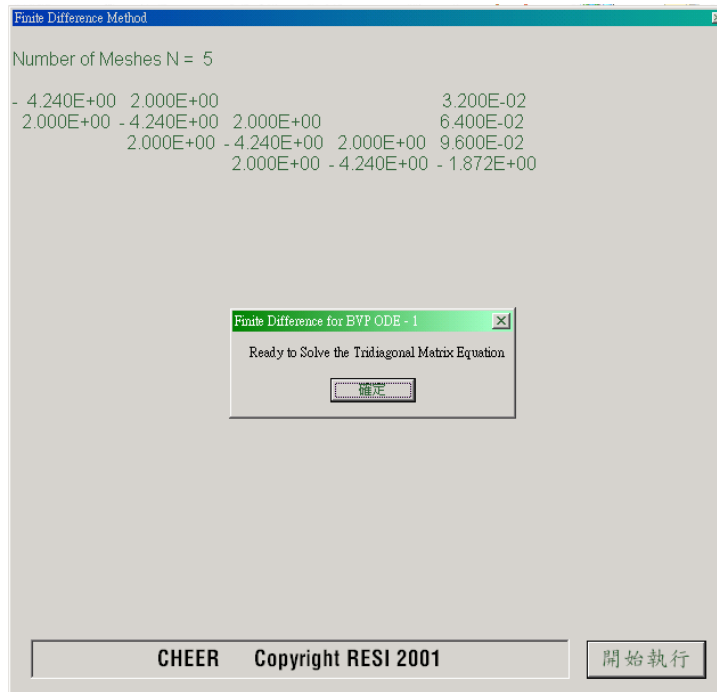
```

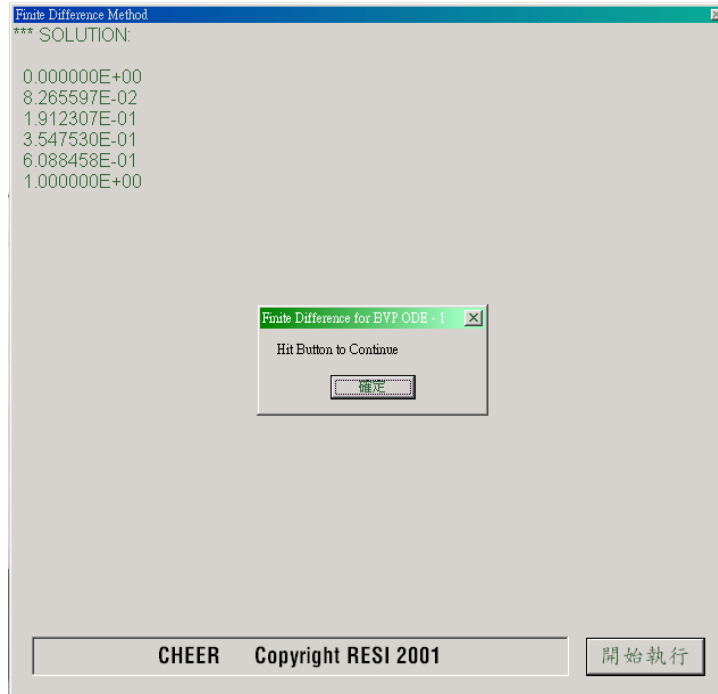
```

Private Sub Start_Click()
Xpos = 8500
Ypos = 6000
Call FiniteDifferenceMethod(Xpos, Ypos)
End Sub

```

程式執行結果





第四節 邊界條件與有限差分法

Visual Basic

以上介紹了如何將微分方程式利用差分法表示，並且整理出各種差分表示法供參考使用。除了微分方程式的差分法以外，要解微分方程式，還得了解如何將邊界條件改寫成對應的差分表示法。以下以線性二次常微分方程式為例加以說明。

$$y'' + p(x)y' + q(x)y = g(x) \quad (9-4.1)$$

$$\text{BC1 } x=0, \quad k_1 y' + \eta_1 y = a$$

$$\text{BC2 } x=1, \quad -k_2 y' + \eta_2 y = b$$

仿上節說明，可將微分方程式 (9-4.1) 改寫成差分近似式

$$\beta_i y_{i-1} + \alpha_i y_i + \gamma_i y_{i+1} = f(x_i), \quad i=1, 2, \dots, N-1 \quad (9-4.2)$$

邊界條件則可以有不同的處理方式，以下分別說明之。

一階近似法 $O(h)$

邊界條件中的導函數 $y'(0)$ 及 $y'(1)$ 分別用一階正向差分及逆向差分表示，

$$y'(0) = \frac{y_1 - y_0}{h} \quad (9-4.3)$$

$$y'(1) = \frac{y_N - y_{N-1}}{h} \quad (9-4.4)$$

代回 BC1 及 BC2，經整理後，可分別求得 y_0 及 y_N 為

$$y_0 = \left(\frac{ah}{h\eta_1 - k_1}\right) - \left(\frac{k_1}{h\eta_1 - k_1}\right)y_1 \equiv \xi_1 - \xi_2 y_1 \quad (9-4.5)$$

$$y_N = \left(\frac{bh}{h\eta_2 - k_2}\right) - \left(\frac{k_2}{h\eta_2 - k_2}\right)y_{N-1} \equiv \delta_1 - \delta_2 y_{N-1} \quad (9-4.6)$$

將 y_0 及 y_N 代回方程式 (9-4.2)，並將 (9-4.2) 改寫成矩陣型式，得到

$$\underline{A}\underline{y} = \underline{F} \quad (9-4.7)$$

其中

$$\underline{y} = [y_1 \quad y_2 \quad \cdots \quad y_{N-1}]^T$$

$$\underline{F} = [f(x_1) - \beta_1 \xi_1 \quad f(x_2) \quad f(x_3) \quad \cdots \quad f(x_{N-1}) - \gamma_{N-1} \delta_1]^T$$

$$\underline{A} = \begin{bmatrix} (\alpha_1 - \beta_1 \xi_2) & \gamma_1 & 0 & & & 0 \\ \beta_2 & \alpha_2 & \gamma_2 & & & \\ & \cdot & \cdot & \cdot & & \\ & & & \beta_{N-2} & \alpha_{N-2} & \gamma_{N-2} \\ 0 & & & 0 & \beta_{N-1} & (\alpha_{N-1} - \gamma_{N-1} \delta_2) \end{bmatrix}$$

$$\alpha_i = -4 + 2h^2 q(x_i), \quad i = 1, 2, \dots, N-1$$

$$\beta_i = 2 - hp(x_i), \quad \gamma_i = 2 + hp(x_i)$$

$$f(x_i) = 2h^2 g(x_i)$$

$$\xi_1 = \frac{ah}{h\eta_1 - k_1}, \quad \xi_2 = \frac{k_1}{h\eta_1 - k_1}, \quad \delta_1 = \frac{bh}{h\eta_2 - k_2}, \quad \delta_2 = \frac{k_2}{h\eta_2 - k_2}$$

$$x_i = x_0 + ih, \quad x_0 = 0$$

方程式 (9-4.7) 基本上與方程式 (9-3.11) 的形式完全相仿，仍為三對角線矩陣方程

式，可以利用相同方法解之。

二階近似法

邊界條件中的導函數 $y'(0)$ 及 $y'(1)$ ，若利用準確度為 $O(h^2)$ 的中央差分式表示，則需分別引入假節點 (fictitious nodes) x_{-1} 及 x_{N+1} ，和對應的函數值 y_{-1} 及 y_{N+1} ：

$$y'(0) = \frac{y_1 - y_{-1}}{2h} \quad (9-4.8)$$

$$y'(1) = \frac{y_{N+1} - y_{N-1}}{2h} \quad (9-4.9)$$

代回 BC1 及 BC2，分別求解假設的 y_{-1} 及 y_{N+1} ，得到

$$y_{-1} = \left(\frac{2h\eta_1}{k_1}\right)y_0 + y_1 - \left(a\frac{2h}{k_1}\right) \equiv \xi_1 y_0 + y_1 - \xi_2 \quad (9-4.10)$$

$$y_{N+1} = y_{N-1} + \left(\frac{2h\eta_2}{k_2}\right)y_N - \left(b\frac{2h}{k_2}\right) \equiv y_{N-1} + \delta_1 y_N - \delta_2 \quad (9-4.11)$$

代回方程式 (9-4.2)，並令 $i=0, 1, 2, \dots, N$ ，改寫成矩陣型式後得到

$$\underline{A}y = \underline{F} \quad (9-4.12)$$

其中

$$\underline{y} = [y_0 \quad y_1 \quad y_2 \quad \dots \quad y_N]^T$$

$$\underline{F} = [f(x_0) + \beta_0 \xi_2 \quad f(x_1) \quad f(x_2) \quad \dots \quad f(x_N) + \gamma_N \delta_2]^T$$

$$\underline{A} = \begin{bmatrix} (\alpha_0 + \beta_0 \xi_1) & (\gamma_0 + \beta_0) & 0 & & & & & & & & 0 \\ \beta_1 & \alpha_1 & \gamma_1 & & & & & & & & \\ & & & \ddots & & & & & & & \\ & & & & \ddots & & & & & & \\ & & & & & \beta_{N-1} & \alpha_{N-1} & \gamma_{N-1} & & & \\ & & & & & & & & & & \\ 0 & & & & & & 0 & (\beta_N + \gamma_N) & (\alpha_N + \gamma_N \delta_1) & & \end{bmatrix}$$

$$\alpha_i = -4 + 2h^2 q(x_i), \quad i = 0, 1, 2, \dots, N$$

$$\beta_i = 2 - hp(x_i), \quad \gamma_i = 2 + hp(x_i), \quad f(x_i) = 2h^2 g(x_i)$$

$$\xi_1 = \frac{2h\eta_1}{k_1}, \quad \xi_2 = \frac{2ha}{k_1}, \quad \delta_1 = \frac{2h\eta_2}{k_2}, \quad \delta_2 = \frac{2hb}{k_2}$$

$$x_i = x_0 + ih, \quad x_0 = 0$$

例題 9-5 散熱片熱傳分析

為提高熱交換器的傳熱效率，常在傳熱面上裝設散熱片，增加傳熱面積，以促進熱量的傳遞。考慮一圓管外裝有一環狀散熱片，如圖 9.4 所示。定義此系統的參數及典型的參數值如下：

- L = 散熱片的高度 ($\frac{1}{2}$ 吋)
- b = 散熱片的厚度 ($\frac{1}{16}$ 或 $\frac{1}{32}$ 吋)
- r_1 = 散熱片底部管子的外徑 ($\frac{1}{2}$ 吋)
- k = 金屬的熱傳導係數 ($65 \text{ Btu/hr} \cdot ^\circ\text{F} \cdot \text{ft}^2$)
- T_B = 散熱片底部圓管面積溫度 (350°F)
- T_A = 周圍空氣溫度 (60°F)

利用能量平衡可得散熱片之徑向溫度分布微分方程式為

$$r \frac{d^2T}{dr^2} + \frac{dT}{dr} - \frac{2\eta}{kb} r(T - T_A) = 0 \quad (9-4.13)$$

BC1 $r = r_1$, $T = T_B$

BC2 $r = r_2$, $-k \frac{dT}{dr} = \eta(T - T_A)$

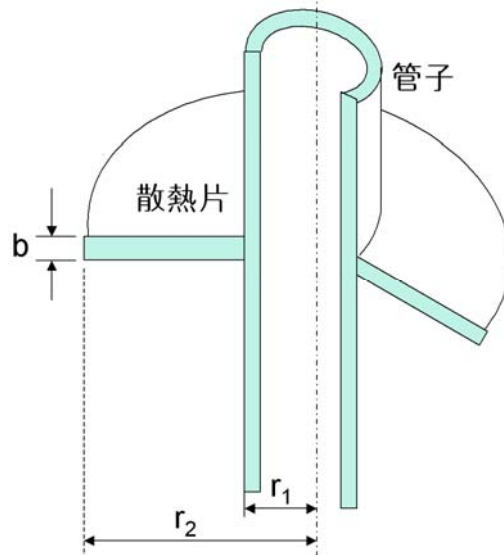


圖 9.4 散熱片熱傳分析

試利用有限差分法計算散熱片內的溫度分布，並求散熱片末端溫度。

解：

定義無因次溫度 θ 及無因次半徑 Z 為

$$\theta = \frac{T - T_A}{T_B - T_A}, \quad Z = \frac{r}{r_1}$$

則原微分方程式 (9-4.13) 可以被轉換成

$$Z \frac{d^2 \theta}{dZ^2} + \frac{d\theta}{dZ} - \alpha Z \theta = 0$$

$$\begin{aligned} \text{BC1} \quad Z = 1, \quad \theta = 1 \\ \text{BC2} \quad Z = \frac{r_2}{r_1}, \quad -k \frac{d\theta}{dZ} = \eta r_1 \theta \end{aligned} \tag{9-4.14}$$

其中

$$\alpha = \frac{2\eta r_1^2}{kb}$$

將微分方程式 (9-4.14) 改寫成差分方程式，可以得到

$$Z_i \frac{\theta_{i+1} - 2\theta_i + \theta_{i-1}}{h^2} + \frac{\theta_{i+1} - \theta_{i-1}}{2h} - \alpha Z_i \theta_i = 0, \quad i = 1, 2, \dots, N-1$$

或整理後，得到

$$\begin{aligned} [2Z_i + h] \theta_{i+1} + [-4Z_i - 2\alpha h^2 Z_i] \theta_i + [2Z_i - h] \theta_{i-1} = 0 \\ i = 1, 2, \dots, N-1 \end{aligned} \tag{9-4.15}$$

邊界條件利用本節所介紹方法分別為

$$\begin{aligned} \text{BC1} : \theta_0 = 1 \\ \text{BC2} : -k \frac{\theta_{N+1} - \theta_{N-1}}{2h} = \eta r_1 \theta_N \end{aligned}$$

其中 θ_{N+1} 為假設的節點溫度，由上式可以得到

$$\begin{aligned} \theta_{N+1} &= \theta_{N-1} - \frac{2\eta h r_1}{k} \theta_N \\ &\equiv \theta_{N-1} - \xi h \theta_N \end{aligned}$$

改寫成矩陣型式，可以得到

$$\underline{A}\underline{\theta} = \underline{F}$$

其中

$$\underline{\theta} = [\theta_1 \quad \theta_2 \quad \cdots \quad \theta_N]$$

$$\underline{F} = [-b_1 \quad 0 \quad 0 \quad \cdots \quad 0]$$

$$\underline{A} = \begin{bmatrix} a_1 & c_1 & 0 & & & 0 \\ b_2 & a_2 & c_2 & & & \\ & \cdot & \cdot & \cdot & & \\ & & \cdot & \cdot & & \\ & & & b_{N-1} & a_{N-1} & c_{N-1} \\ 0 & & & 0 & (b_N + c_N) & (a_N - c_N \xi h) \end{bmatrix}$$

$$a_i = -2Z_i(2 + \alpha h^2), b_i = 2Z_i - h, c_i = 2Z_i + h$$

$$\xi = \frac{2\eta r_1}{k}, \alpha = \frac{2\eta r_1^2}{kb}, h = \left(\frac{r^2}{r_1} - 1\right) / N, Z_i = 1 + ih$$

由於所得方程式與例 9-4 型式相同，可利用類似的程式解之。

程式列印

```

/ *****
/          SOLUTION OF B.V.P.-O.D.E
/          BY FINITE DIFFERENCE METHOD
/ *****
/
Sub FiniteDifferenceMethod(Xpos, Ypos)
Dim Alpha(100), Beta(100), Gama(100), Z(100)
' TRIDIAGONAL MATRIX EQUATION
' Alpha,Beta,Gama=COEFFICIENT VECTOR
' Z=CONSTANT VECTOR
'/RETURN : SOLUTION
'
L = 0.5 / 12           'Height of Fin
B = 1 / 16 / 12      'Thickness of Fin
R1 = 0.5 / 12        'Outside Diameter of Pipe
K = 65               'Thermal Conductivity
ETA = 5              'Heat Transfer Coeff. from Fin to Ambient
R2 = R1 + L          'Outside Diameter of Fin
ALFA = 2 * ETA * R1 * R1 / K / B
PHI = 2 * ETA * R1 / K
TB = 350
TA = 60
TREF = TB - TA
    
```

```

Do
    Cls
    Print
    NEQN% = Val(InputBox("> Number of Meshes N=", "", NEQN%, Xpos, Ypos))
    Print "Number of Meshes N = "; NEQN%

    H = (R2 / R1 - 1) / NEQN%
    For I = 1 To NEQN%
        ZI = 1 + I * H
        Alpha(I) = -2 * ZI * (2 + ALFA * H * H)
        Beta(I) = 2 * ZI - H
        Gama(I) = 2 * ZI + H
        Z(I) = 0
    Next I

    Z(1) = -Beta(1)
    Beta(NEQN%) = Beta(NEQN%) + Gama(NEQN%)
    Alpha(NEQN%) = Alpha(NEQN%) - Gama(NEQN%) * PHI * H
    Beta(1) = 0
    Gama(NEQN%) = 0

    Call FiniteDifference(NEQN%, Alpha, Beta, Gama, Z)
    MsgBox ("Ready to Print Temperature Profile")

    Cls
    T = TB - TA
    Print
    Print "*** TEMPERATURE PROFILE:"
    Print
    Print " R/R1          T (F)"
    For I = 1 To NEQN%
        Print Format(1 + I * H, " 0.000E+00 ");
        Print Format(Z(I) * TREF + TA, " 0.000000E+00")
    Next I
    Print
    ContinueYN$ = InputBox("Run Next Job <Y/N>", "", "N", Xpos, Ypos)
Loop Until ContinueYN$ = "N" Or ContinueYN$ = "n"
End Sub

```

Sub FiniteDifference(NEQN%, Alpha, Beta, Gama, Z)

```

'-----
' ECHO
' SOLVE & PRINT RESULT
Call Jacobi(NEQN%, Alpha, Beta, Gama, Z)
Cls
Print "**** SOLUTION:": Print
For I = 1 To NEQN%
    Print Format(Z(I), " 0.0000E+00 ");
    If (Int(I / 5) = I / 5) Then Print
Next I
Print

```

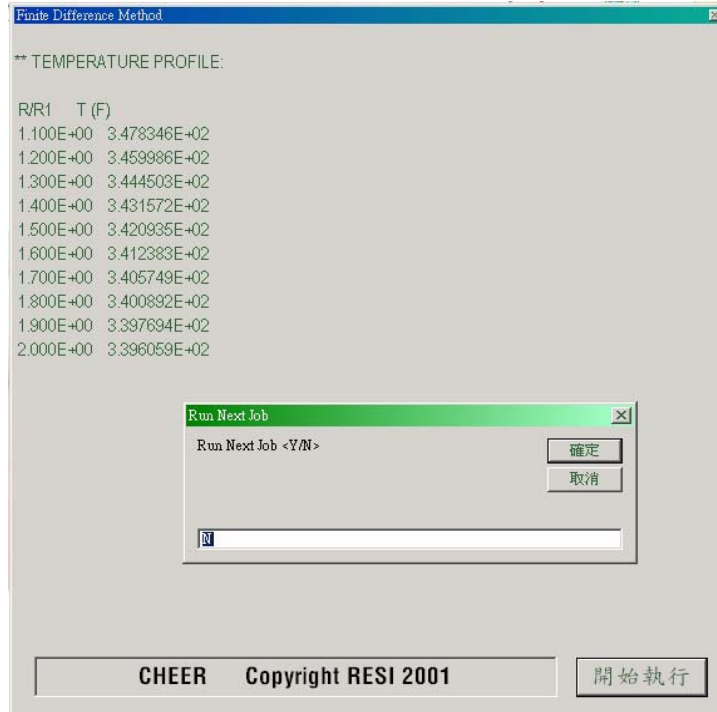
```

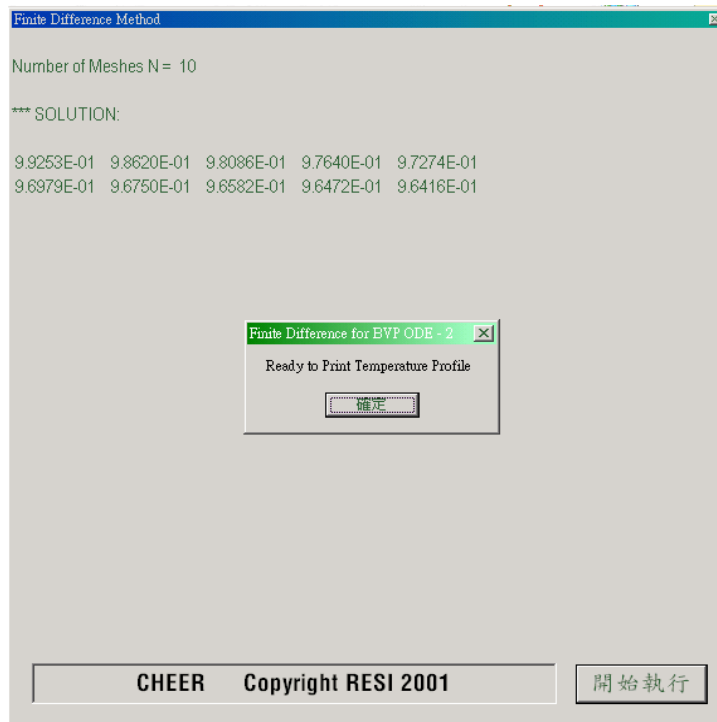
End Sub

Sub Jacobi(NEQN%, Alpha, Beta, Gama, Z)
'
' Solution of Tridiagonal Matrix Equations
'
Gama(1) = Gama(1) / Alpha(1)
Z(1) = Z(1) / Alpha(1)
' --LU DECOMPOSITION
For I = 2 To NEQN%
    Alpha(I) = Alpha(I) - Beta(I) * Gama(I - 1)
    Gama(I) = Gama(I) / Alpha(I)
    Z(I) = (Z(I) - Beta(I) * Z(I - 1)) / Alpha(I)
Next I
' --BACK SUBSTITUTION
For I = NEQN% - 1 To 1 Step -1
    Z(I) = Z(I) - Gama(I) * Z(I + 1)
Next I
End Sub
    
```

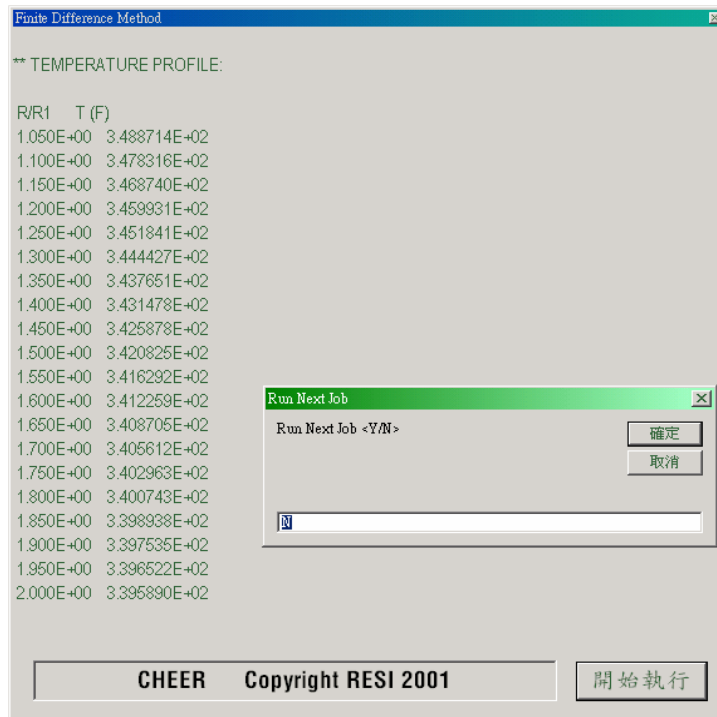
執行結果

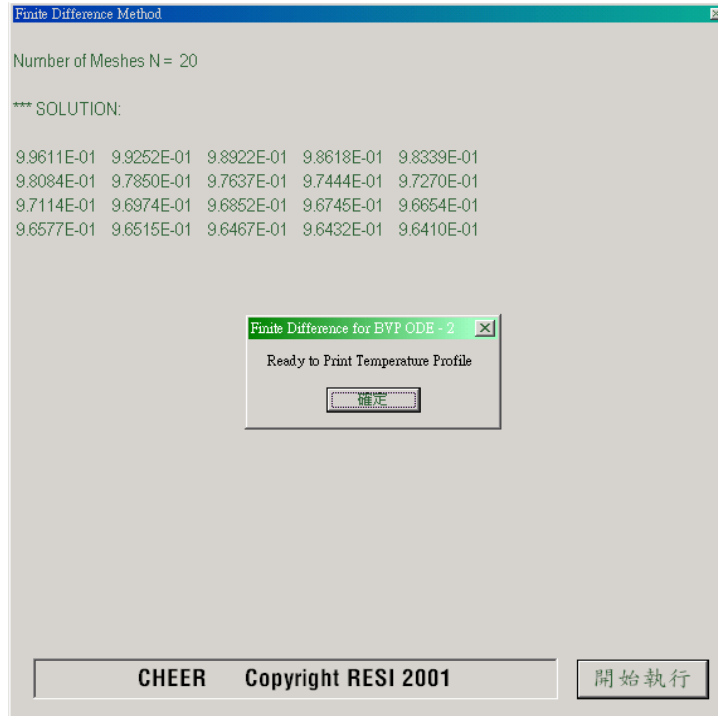
格子點數 N=10 的結果





格子點數 N=20 的結果





第五節 非線性常微分方程式

Visual Basic

將非線性常微分方程式改寫成差分方程式，所使用的方法，與前面所討論的線性常微分方程式作法完全一樣。但所得到的差分方程式通常為非線性聯立方程式，因此，無法像線性問題般直接用三對角線矩陣方程式求解，而必須利用牛頓迭代法或其他數值方法才能求得近似解。

考慮二次非線性常微分方程式

$$Ly \equiv y'' + f(x, y, y') = 0, \quad a < x < b \quad (9-5.1)$$

$$\text{BC1} \quad x = a, \quad y = A$$

$$\text{BC2} \quad x = b, \quad y = B$$

使用二階差分近似法，則方程式 (9-5.1) 可以被改寫成以下的差分方程式

$$Ly_i = \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + f(x_i, y_i, \frac{y_{i+1} - y_{i-1}}{2h}) = 0 \quad (9-5.2)$$

$$i = 1, 2, \dots, N - 1$$

$$y_0 = A, y_N = B$$

若邊界條件為第二類或第三類邊界條件，亦可仿上節方式處理之。方程式 (9-5.2) 通常為聯立非線性方程式，令

$$\underline{y} = [y_1 \quad y_2 \quad \dots \quad y_{N-1}]^T$$

$$\Phi_i(\underline{y}) = Ly_i$$

$$\underline{\Phi}(\underline{y}) = [\Phi_1(\underline{y}) \quad \Phi_2(\underline{y}) \quad \dots \quad \Phi_{N-1}(\underline{y})]^T$$

則方程式 (9-5.2) 可簡寫為

$$\underline{\Phi}(\underline{y}) = 0 \tag{9-5.3}$$

利用牛頓迭代法解所得到的方程式 (9-5.3)，可以得到

$$\underline{y}^{(k+1)} = \underline{y}^{(k)} - \frac{\underline{\Phi}(\underline{y}^{(k)})}{\partial \underline{\Phi}(\underline{y}^{(k)}) / \partial \underline{y}^{(k)}} \tag{9-5.4}$$

其中 $\partial \underline{\Phi}(\underline{y}^{(k)}) / \partial \underline{y}^{(k)}$ 亦可利用數值方法求之，也就是利用牛頓割線法解方程式 (9-5.3)。此外，亦可利用連續迭代法求 \underline{y} 。以下以實際例子加以說明。

例題 9-6 零次恆溫催化反應

以下看似相當容易的常微分方程式，用於描述一球形觸媒粒子內進行零次常溫反應，且主成分趨近於平衡時的濃度分布情況。

$$\frac{d^2 y}{dx^2} + \frac{2}{x} \frac{dy}{dx} - 80 \left(1 - \frac{10^{-6}}{y}\right) = 0 \tag{9-5.5}$$

BC1 $x = 0$, $\frac{dy}{dx} = 0$

BC2 $x = 1$, $y = 1$

試求濃度分布函數 $y(x)$ 。

解：

利用二階差分式將原方式改寫成差分方程式，得到

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + \frac{2}{x_i} \frac{y_{i+1} - y_{i-1}}{2h} - 80\left(1 - \frac{10^{-6}}{y_i}\right) = 0$$

$$i = 1, 2, \dots, N-1 \quad (9-5.6)$$

邊界條件 BC1 仿上節處理方式，引入一假節點 x_{-1} ，則 BC1 可改寫成差分式。

$$\frac{y_1 - y_{-1}}{2h} = 0$$

或 $y_{-1} = y_1$ 。當 $x=0$ 時，根據 L'Hospital 定理，原微分方程式第二項可改寫成

$$\lim_{x \rightarrow 0} \left(\frac{2}{x} \frac{dy}{dx} \right) = 2 \frac{d^2 y}{dx^2}$$

因此，當 $x=0$ 時，原微分方程式變成

$$3 \frac{d^2 y}{dx^2} - 80\left(1 - \frac{10^{-6}}{y}\right) = 0; \quad x=0 \quad (9-5.7)$$

或寫成差分式為

$$3 \frac{y_1 - 2y_0 + y_{-1}}{h^2} - 80\left(1 - \frac{10^{-6}}{y_0}\right) = 0 \quad (9-5.8)$$

由於 $y_{-1} = y_1$ ，代入方程式 (9-5.8)，整理之。並將 (9-5.8) 也改寫成 (9-5.3) 型式，得到

$$\Phi_0(\underline{y}) = \frac{6}{h^2} y_1 - \frac{6}{h^2} y_0 - 80\left(1 - \frac{10^{-6}}{y_0}\right) = 0$$

$$\Phi_i(\underline{y}) = \left(\frac{1}{h^2} + \frac{1}{x_i h_i}\right) y_{i+1} - \frac{2}{h^2} y_i + \left(\frac{1}{h^2} - \frac{1}{h x_i}\right) y_{i-1} - 80\left(1 - \frac{10^{-6}}{y_i}\right) = 0 \quad (9-5.9)$$

$$i = 1, 2, \dots, N-1$$

$$y_N = 1$$

以上計有 N 個聯立方程式，可求解 y_0, y_1, \dots, y_{N-1} ， N 個未知數。

參考文獻

Visual Basic

1. Schlichting, H., "Boundary Layer Theory", 4th ed., McGraw-Hill, New York, (1960).
2. Hall, G., and J. M. Watt, "Modern Numerical Methods for Ordinary Differential Equations", Oxford Univ. Press, Inc., New York, (1976).
3. Lapidus, L., and J. H. Seinfeld, "Numerical Solution of Ordinary Differential Equations", Academic Press, New York, (1971).
4. Caberry, J. J., "Chemical and Catalytic Reaction Engineering", McGraw-Hill, New York, (1976).
5. Deuflhard, P., "Recent Advances in Multiple Shooting Techniques" in "Computational Techniques for Ordinary Differential Equations", Edited by Gladwell, I., and D. K. Sayers, Academic, London, (1980).
6. Scott, M. R., and H. A. Watts, "Computational Solutions of Linear Two Point Boundary Value Problems via Ortho-normalization", SIAM K. Numer. Ana., 14, 40 (1977).
7. Nogotov, E. F., "Applications of Numerical Heat Transfer" Hemisphere, Washington, (1978).
8. Anderson, D. A., J. C. Tannehill, and R. H. Pletcher, "Computational Fluid Mechanics and Heat Transfer", McGraw Hill, New York, (1984).
9. Shoup, T. E., "Applied Numerical Methods for the Microcomputer" Prentice-Hall, London, (1984).
10. Davis, M. E., "Numerical Methods and Modeling for Chemical Engineers" John Wiley, New York, (1984).
11. Whitaker, S., "Elementary Heat Transfer Analysis" (1976).
12. Rohsenow, W. M., and H. Y. Choi, "Heat, Mass, and Momentum Transfer", Prentice-Hall, Englewood Cliffs, N.J., (1961).

習題

Visual Basic

1. 試利用有限差分法重解例 9-2。

$$\frac{d^2 y}{dx^2} + \frac{2}{x} \frac{dy}{dx} = \Phi^2 R(y)$$

BC1 $x=0$ 時, $\frac{dy}{dx} = 0$

BC2 $x=1$ 時, $y=1$

2. 試利用投射法重解例 9-5。

$$Z \frac{d^2 \theta}{dZ^2} + \frac{d\theta}{dZ} - \alpha Z \theta = 0$$

BC1 $Z=1, \theta=1$

BC2 $Z = \frac{r_2}{r_1}, -k \frac{d\theta}{dZ} = \eta r_1 \theta$

3. 試利用有限差分法解設計問題 D-IX。

$$f f'' + 2 f''' = 0$$

BC1 $\eta=0$ 時, $f=0$

BC2 $\eta=0$ 時, $f'=0$

BC3 $\eta=\infty$ 時, $f'=1$

4. 試利用第八章例 8-8 所提供的 STIFF2000 副程式結合投射法解本章的例 9-6。若有困難，請試著找出可能解決的方法。

$$\frac{d^2 y}{dx^2} + \frac{2}{x} \frac{dy}{dx} - 80 \left(1 - \frac{10^{-6}}{y}\right) = 0$$

BC1 $x=0, \frac{dy}{dx} = 0$

BC2 $x=1, y=1$

5. 解聯立方程式 (9-5.9)，並繪製 y vs. x 的變化圖。試著改變差分間距，觀察所得結果變化。

$$\begin{aligned}\Phi_0(\underline{y}) &= \frac{6}{h^2} y_1 - \frac{6}{h^2} y_0 - 80\left(1 - \frac{10^{-6}}{y_0}\right) = 0 \\ \Phi_i(\underline{y}) &= \left(\frac{1}{h^2} + \frac{1}{x_i h_i}\right) y_{i+1} - \frac{2}{h^2} y_i + \left(\frac{1}{h^2} - \frac{1}{h x_i}\right) y_{i-1} - 80\left(1 - \frac{10^{-6}}{y_i}\right) = 0 \quad (9-5.9) \\ & i = 1, 2, \dots, N-1 \\ & y_N = 1\end{aligned}$$

6. 考慮線性二次常微分方程式

$$Ly \equiv y'' + p(x)y' + q(x)y = f(x), \quad a < x < b \quad (9-P6.1)$$

$$\text{BC1} \quad x = a \text{ 時, } -k_1 \frac{dy}{dx} + h_1 y = \alpha$$

$$\text{BC2} \quad x = b \text{ 時, } k_2 \frac{dy}{dx} + h_2 y = \beta$$

(a) 試證明此方程式的解，可以假設成以下方式

$$y(x) = y_0(x) + A_1 y_1(x) + A_2 y_2(x) \quad (9-P6.2)$$

其中 $y_0(x)$ 稱爲此微分方程式的特定解 (Particular Solution)，

$$\begin{aligned}Ly_0 &\equiv y'' + p(x)y' + q(x)y = f(x), \quad a < x < b \\ y(a) &= \eta_0 \quad y'(a) = \eta'_0\end{aligned} \quad (9-P6.3)$$

$y_1(x)$ 及 $y_2(x)$ 分別爲以下均勻方程式 $y'' + p(x)y' + q(x)y = 0$ 的輔助解 (complementary Solution)。

$$\begin{aligned}Ly_1 &\equiv y'' + p(x)y' + q(x)y = 0, \quad a < x < b \\ y(a) &= \eta_1 \quad y'(a) = \eta'_1\end{aligned} \quad (9-P6.4)$$

$$\begin{aligned}Ly_2 &\equiv y'' + p(x)y' + q(x)y = 0, \quad a < x < b \\ y(a) &= \eta_2 \quad y'(a) = \eta'_2\end{aligned} \quad (9-P6.5)$$

(b) 假設以上三組微分方程式由 $x = a$ 積分到 $x = b$ 的解，分別爲 $y_0(b)$, $y'_0(b)$, $y_1(b)$, $y'_1(b)$, $y_2(b)$, $y'_2(b)$ ，請將結果代入原微分方程式的邊界條件，並求出假設參數 A_1 及 A_2 。

(c) 根據以上的分析，可建立邊界值常微分方程式的輔助函數法數值解策略如下：

- i. 建立的三個初值常微分方程式，其中 η 值可任意假設。
- ii. 積分常微分方程式 (9-P6.3)。

- iii. 積分常微分方程式 (9-P6.4)。
- iv. 積分常微分方程式 (9-P6.5)。
- v. 利用 (b) 所建立的方程式，求出 A_1 及 A_2 。
- vi. 利用方程式 (9-P6.2)，建立常微分方程式的解。

(d) 根據以上的輔助函數法數值解策略，試撰寫程式，重解例 9-1 的微分方程式。

7. 利用例 9-2 所附程式，作適當修改，求解以下方程式。

$$\frac{d^2 y}{dx^2} + \frac{2}{x} \frac{dy}{dx} = \Phi^2 R(y)$$

BC1 $x=0$ 時， $\frac{dy}{dx} = 0$

BC2 $x=1$ 時， $y=1$

(a) 當化學反應為一次反應時 $R(y) = y$ ，試求有效度係數與希笠模數 $\Phi = r_0 \sqrt{\frac{k}{D}}$ 的

關係（請用全對數作圖， $\text{Log}(\eta)$ vs. $\text{Log}(\Phi)$ ， Φ 範圍為 0.1 到 10），並與理論解 $\eta = \frac{3}{\Phi} \left[\frac{1}{\tanh(\Phi)} - \frac{1}{\Phi} \right]$ 作比較。

(b) 若反應為二次反應時， $\Phi = r_0 \sqrt{\frac{kC_0}{D}}$ ， $R(y) = y^2$ ，重複以上計算及作圖。

8. 在例 9-5 中，假設金屬傳熱面與周圍空氣間的熱傳係數 η 為定值。但如果將管子水平放置，根據 Rohsenow and Choi [12] 利用無因次分析得到 $\eta = 0.21(T - T_A)^{1/3}$ ，單位與例 9-5 同。試將此條件列入考慮後，重作例 9-5。

$$Z \frac{d^2 \theta}{dZ^2} + \frac{d\theta}{dZ} - \alpha Z \theta = 0$$

BC1 $Z=1$ ， $\theta=1$

BC2 $Z = \frac{r_2}{r_1}$ ， $-k \frac{d\theta}{dZ} = \eta r_1 \theta$

9. 例 9-5 的散熱片若裝在半徑極大的管面上，或裝在一平面上時，如圖 9.5 所示，則能量平衡方程式為

$$\frac{d^2 T}{dZ^2} = \frac{\eta}{Kb} (T - T_A)$$

$$\begin{aligned} \text{BC1} \quad Z = 0 \quad T = T_B \\ \text{BC2} \quad Z = L \quad \frac{dT}{dZ} = 0 \end{aligned}$$

求散熱片內的溫度分布。

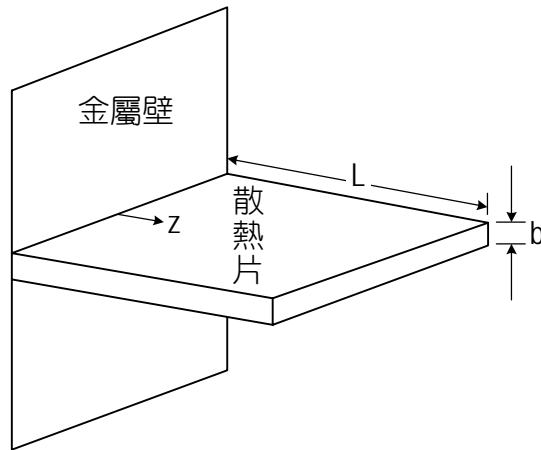


圖 9.5 平板散熱片

10. 重作問題 9，但是將邊界條件 2 更改為

$$\text{BC2} \quad Z = L \quad -k \frac{dT}{dZ} = \eta(T - T_A)$$

結果與問題 9 有何差異？

11. 若問題 9 中的傳熱係數 $\eta = 0.21(T - T_A)^3$ ，試重新計算溫度分布，並與問題 8 比較。
12. 試利用投射法及有限差分法解以下的微分方程式，並比較所需計算時間及準確度之關係。

$$Z(1-Z) \frac{d^2y}{dZ^2} + (1-2Z) \frac{dy}{dZ} - aZy = 0$$

$$\text{BC1} \quad Z = \frac{1}{3} \quad y = 1$$

$$\text{BC2} \quad Z = \frac{2}{3} \quad \frac{dy}{dz} = -a(\sin \theta)y$$

其中 $a = 0.3078$, $a \sin \theta = 0.00961$

13. 試解微分方程式

$$\frac{d^2 y}{dx^2} - \frac{1}{y} \left(\frac{dy}{dx}\right)^2 - y \frac{dy}{dx} + 1 = 0$$

BC1 $x=0$, $y=y_0$ ($= 0.1, 0.2$ 或 0.3)

BC2 $x \rightarrow \infty$, $\frac{dy}{dx} \rightarrow \frac{1}{\sqrt{2x}}$

此微分方程式可用於描述高黏度流體之簾式塗裝操作。

14. 設計問題 D-IX 若連熱傳現象也同時加以考慮，假設平板溫度為 T_w ，而離平板無限遠處的流體溫度為 T_∞ ，則由能量平衡可得

$$u \frac{\partial T}{\partial x} + v \frac{\partial T}{\partial y} = \frac{k}{\rho C_p} \frac{\partial^2 T}{\partial y^2}$$

BC1 $y=0$, $T=T_w$

BC2 $y \rightarrow \infty$, $T=T_\infty$

其中 k 為流體的熱傳導率， ρ 為流體密度， C_p 為流體比熱。令

$$\theta = \frac{T - T_\infty}{T_w - T_\infty}$$

則仿設計問題 D-IX，可將以上的微分方程式簡化成

$$\frac{d^2 \theta}{d\eta^2} + \frac{f \text{Pr}}{2} \frac{d\theta}{d\eta} = 0$$

BC1 $\eta=0$, $\theta=1$

BC2 $\eta \rightarrow \infty$, $\theta=0$

其中 f 及 η 同設計問題 IX 所定義， $\text{Pr} = \frac{\rho C_p \nu}{k}$ ，試作 θ 與 η 的關係圖，並令 $\text{Pr} = 0.1, 1, 10, 100$ 及 1000 。

15. 一水平圓管用於輸送高溫高壓流體，由於流體壓力極高，因此，使用相當厚的鋼管。若流體溫度為 780°C ，空氣溫度 20°C 。管子內徑 2.5 公分，外徑 5.2 公分。試求其溫度分布情況。

16. 為了減少熱量的損失，問題 15. 所使用的鋼管利用絕熱材料包覆，此絕熱材料熱傳導係數為 0.083，試求外管壁溫度及管內溫度分布情況。絕熱材料厚 5 公分。

16. 請嘗試用各種方法求解下列問題，並檢討各種解法所遭遇的困難及解決方法：

(a) $\frac{d^2 y}{dx^2} - 10^6 y^2 = 0$

BC1 $x = 0, y' = 0$

BC2 $x = 1, y = 1$

(b) $\frac{d^2 y}{dx^2} - \Phi^2 y^{1/3} = 0$

BC1 $x = 0, y' = 0$

BC2 $x = 1, y = 1$

$\phi = 1, 10, 100$

18. 在兩片水平板間且下面平板溫度較高的情況下，流體在平板間的穩定度問題，可以利用以下的常微分方程式表示之

$$(D^2 - k^2)^3 W = -Ra \cdot k^2 W \quad (9-P18.1)$$

BC $W = 0 \quad @ z = \pm \frac{1}{2}$

$DW = 0 \quad @ z = \pm \frac{1}{2}$

$(D^2 - k^2)^2 W = 0 \quad @ z = \pm \frac{1}{2}$

其中

$$D = \frac{d}{dz}, \quad D^n = \frac{d^n}{dz^n}, \quad k = \text{參數}, \quad Ra = PrGr$$

Pr = 普朗特常數 Prandtl No.

Gr = 葛氏常數 Grashoff No.

W = 流體在 z 方向的速度擾動幅度

此微分方程式是一個特徵值問題 (Eigen Value Problem)，對某一特定 k 值，只有在 Ra 等於某些特定值時，W 的解才會存在。

當 k = 3.117 時，Ra 範圍約在 1700 ± 50，請利用程式嘗試 Ra 值直到得到正確的 Ra 值為止。

(a) 令

$$W_1 = W$$

$$W_2 = W' = W_1'$$

$$W_3 = W'' = W_2'$$

$$W_4 = W''' = W_3'$$

$$W_5 = W^{(4)} = W_4'$$

$$W_6 = W^{(5)} = W_5'$$

試將方程式 (10-P18.1) 轉化成矩陣微分方程式

$$\underline{W}' = \underline{A}\underline{W}$$

$$\underline{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ k^6 - Ra k^4 & 0 & -3k^4 & 0 & 3k^2 & 0 \end{bmatrix}$$

$$W_1 = 0$$

$$\text{BC } @ z = \pm 1/2 \quad W_2 = 0$$

$$W_5 = 2k^2 W_3 - k^4 W_1$$

- (b) 假設 $Ra = r$ (起始值為 1700)，且在 $z = -1/2$ 處， $W_3 = a, W_4 = b, W_6 = c$ 。利用阮奇庫塔費勃格法 (RKF45) 積分至 $z = 1/2$ ，得到

$$W_1 = f_1(r, a, b, c)$$

$$W_2 = f_2(r, a, b, c)$$

$$W_5 - 2k^2 W_3 + k^4 W_1 = f_3(r, a, b, c)$$

- (c) 定義誤差為 $F = \frac{1}{2}(f_1^2 + f_2^2 + f_3^2)$ ，積分計算出 F 後，檢驗 F 是否滿足誤差容忍度之要求。

- (d) 假設 $Ra = r + \delta r$ ，且在 $z = -1/2$ 處， $W_3 = a + \delta a, W_4 = b + \delta b, W_6 = c + \delta c$ 。再利用阮奇庫塔費勃格法 (RKF45) 積分至 $z = 1/2$ 。並利用步驟 (b) 所得結果，建立 f 之導函數：

$$\frac{\partial f_i}{\partial r} = \frac{\delta f_i}{\delta r}, \quad \frac{\partial f_i}{\partial a} = \frac{\delta f_i}{\delta a}, \quad \frac{\partial f_i}{\partial b} = \frac{\delta f_i}{\delta b}, \quad \frac{\partial f_i}{\partial c} = \frac{\delta f_i}{\delta c}$$

- (e) 利用牛頓拉福森法，取得新的 Ra, a, b 及 c 值。重複步驟 (b) 至 (e) 直到收斂為止。
- (f) 請利用以上策略撰寫程式，並作數值分析探討。