Visual Basic

CHAPTER 10

配重殘值法

(Method of Weighted Residuals)

張 榮 興 博士
豐映科技股份有限公司
E-mail: chang.ronhsin@msa.hinet.net

蘭若斯利用正交多項式作為微分方程式的近似解
經魏德森及史都華改進成為最廣泛被使用的正交配置法

設計問題 D-X

圓柱形觸媒的徑向質傳及反應

圓柱形觸媒粒子是化學工業上最常被使用的反應促進媒介，因此，其反應及質量傳遞相關分析也備受重視。考慮圓柱形觸媒粒子的徑向滲透、質量傳遞及等溫催化反應，所建立之數學模式為

$$\frac{d^2y}{dx^2}+\frac{1}{x}\frac{dy}{dx}=\Phi^2 f(y)$$

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

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

其中 y 為無因次濃度 $(y=C/Cs)$；x 為無因次半徑 $(x=r/R)$。已知反應動力可利用二次反應模式表示。

$$f(y)=y^n\quad,\quad n=2$$

試求圓柱形觸媒粒子的有效度係數 (Effectiveness Factor, η) 與希笠模數 (Thiele Modulus, Φ) 之關係。其中 Φ 及 η 之定義分別為

$$\Phi=R\sqrt{\frac{C_0^{n-1}}{D_e}}$$

$$\eta=\frac{1}{V\cdot f(1)}\int_v f(y)dV\equiv\int_0^1 f(y)dx^2$$

配重殘值法 (Method of Weighted Residuals, MWR) *的基本觀念，是利用一組多項式表示近似於問題的正確解，將正確解與近似解的誤差殘值，乘以適當的配重函數後，使其總和或積分值最小化，以達到最接近正確解的目標。*工程界採用 MWR 配重殘值法解決工程分析問題，約自 1960 年代開始，目前已有許多專書 [1 , 2 , 3] 詳細討論這種方法，其應用廣見於流體力學、熱傳遞、質量傳遞、反應工程等各領域，是一種相當有效率的數學工具。配重殘值法中，又以數學配置法最易於使用，近年來已成爲解微分方程式的數值方法中最廣泛被使用的方法之一。

第一節 配重殘值法基本原理

Visual Basic

爲了說明配重殘值法的基本想法與處理方法，我們首先以一個最簡單的邊界值常微分方程式爲例加以說明。

$$
\begin{aligned}
&\theta''(x) = x \quad 0 < x < 1 \\
&\quad BC1 \qquad \theta(0) = 1 \\
&\quad BC2 \qquad \theta'(1) + \theta(1) = 1
\end{aligned}
\tag{10-1.1}
$$

假設 θ'' 及 θ' 在 $0 < x < 1$ 的區間內都是連續函數，則以上方程式的解也必然是一個連續函數。根據本書第三章的說明，任何連續函數都可以利用一個多項式來代表；因此，方程式 (10-1.1) 的解，可以利用一無限級數的和表示之。

$$\theta = \sum_{i=0}^{\infty} C_i \ x^i \equiv \sum_{i=0}^{\infty} C_i u_i(x)$$

其中 $u_i(x)$ 稱爲基礎函數 (Basis Function)。由於 x 的範圍爲 $0 < x < 1$，當 i 值增大時，x^i 即快速變小；因此，爲了簡化起見，考慮當 $i > N$ 後 x^i 即可忽略，θ 可以假設爲利用一個 (N+1) 次多項式近似之。

$$\theta \equiv \theta_N = \sum_{i=0}^{N+1} C_i \ x^i \equiv \sum_{i=0}^{N+1} C_i u_i(x) \tag{10-1.2}$$

由於 θ_N 爲方程式 (10-1.1) 的近似解，因此，θ_N 也要滿足微分方程式的兩個邊界條件，BC1 及 BC2；將 θ_N 代入邊界條件 BC1 中，得到：

$$\text{BC1} \quad \theta_N(0) = C_0 = 1 \tag{10-1.3}$$

將 θ_N 代入邊界條件 BC2 中，得到：

$$\text{BC2}\quad \theta'_N(1)+\theta_N(1)=\sum_{i=1}^{N+1} iC_i+\sum_{i=0}^{N+1} C_i$$
$$=[C_1+\sum_{i=2}^{N+1} i\,C_i]+[C_0+C_1+\sum_{i=2}^{N+1} C_i]$$
$$=0$$

或將此方程式整理後得到

$$C_1=-\frac{1}{2}-\frac{1}{2}\sum_{i=2}^{N+1}(i+1)\,C_i \tag{10-1.4}$$

將方程式 (10-1.3) 及 (10-1.4) 代回方程式 (10-1.2)，則 θ 的近似解表示式可以改寫成為

$$\theta_N=1-\frac{1}{2}〔1+\sum_{i=2}^{N+1}(i+1)\,C_i〕\,x+\sum_{i=2}^{N+1} C_i\,x^i \tag{10-1.5}$$

此方程式中含有 $C_2, C_3, \cdots\cdots, C_{N+1}$ 總共 N 個未定係數，因此，需要再利用 N 個滿足微分方程式 (10-1.1) 的條件來決定這些係數。

為了簡化說明起見，首先考慮 $N=1$ 的情況，亦即 $\theta\equiv\theta_1=\sum_{i=0}^{2} C_i\,x^i$，則方程式 (10-1.5) 可簡化成

$$\theta_1=1-\frac{x}{2}+C_2(x^2-\frac{3}{2}x) \tag{10-1.6}$$

以上步驟為建立微分方程式近似解表示式的方法。其次要考慮如何決定未定係數 $C_2, C_3, \cdots\cdots, C_{N+1}$。為了簡化說明起見，將微分方程式寫成 $L\{x, y\}=0$ 的形式；若近似多項式 (10-1.2) 完全滿足原微分方程式，則代入微分方程式將得到 $L\{x, \theta_N\}=0$。但由於如前面所說明 θ_N 為近似解，將 θ_N 代入微分方程式 $L\{x, y\}$ 中，結果不一定能得到零。因此，定義 θ_N 代入微分方程式所得到的殘值為 $R=L\{x, \theta_N\}$。

將方程式 (10-1.6) 代入原方程式，得到殘值為

$$R=\theta''_1-x=2\,C_2-x \tag{10-1.7}$$

若 θ_1 為正確解，則殘值 $R=0$。但由方程式 (10-1.7) 可以知道，由於 x 位於 0 與 1 間 $(0<x<1)$，因此，我們若選定某一特定 C_2 值，必能使殘值 R 在 $0<x<1$ 間的某一點

時爲零，而在其他 x 位置的殘值 R 則可能大於或小於零。可是需注意由於 θ_N 只是一個近似解，如果殘值 R 在每一 x 位置都等於零，當然是最好的解答，否則也可以退而求其次，要求殘值 R 在所考慮區間 $0<x<1$ 內的某一種平均值爲零，亦即，讓殘值 R 的配重積分爲零，亦不失爲一良好的近似解。這種觀念以數學式表示，即爲

$$\int_0^1 W_j\ R\,dx = 0 \quad , \quad j = 1, 2, \cdots, N \tag{10-1.8}$$

亦即，只要給定配重函數 W_j，利用方程式 (10-1.8) 執行積分後，即可求解未定係數 C_i。然後，代回 θ_N 的表示式 (10-1.2)，即可求得微分方程式的近似解 θ_N。以上所說明的就是配重殘值法的基本想法與步驟，加以整理如下：

1. 利用一個多項式 $\theta_N = \sum_{i=0}^{N+1} C_i\ x^i$ 作爲微分方程式 $L\{x, y\} = 0$ 的近似解；

2. 使多項式滿足微分方程式的邊界條件，並簡化 θ_N 表示式；

3. 代入微分方程式 $L\{x, y\} = 0$，求出殘值的表示式 $R = \{x, \theta_N\}$；

4. 定義配重函數 W_j；

5. 使配重殘值的積分式爲零，即 $\int_0^1 W_j\ R\,dx = 0$，$j = 1, 2, \cdots, N$；可以建立 N 個未定係數 C_i 的聯立方程式；

6. 解聯立方程式，找出近似多項式的未定係數 C_i；

7. 代回原近似多項式 $\theta_N = \sum_{i=0}^{N+1} C_i\ x^i$，建立微分方程式的近似解；

8. 利用適當內插法，計算微分方程式在特定位置的解。

配重殘值法 (Methods of Weighted Residuals) 的基本想法如前面的介紹，但配重函數 $W_j(x)$ 有多種不同的選擇。依所選擇配重函數 $W_j(x)$ 的不同，對應的配重殘值法即有不同的名稱及不同的特性，以下分別簡要介紹之。

1. 配置法 (Collocation Method)

配置法的配重函數 $W_j(x)$ 定義爲

$$W_j(x) = \delta(x - x_j) \quad ; \quad j = 1, 2, 3, \ldots\ldots, N = \begin{Bmatrix} 1 & ; & x = x_j \\ 0 & ; & x \neq x_j \end{Bmatrix} \tag{10-1.9}$$

其中 x_j 稱爲配置點 (Collocation points)。δ 稱爲 Dirac Delta 函數，假設殘值 $R = L\{x, \theta_N\}$ 函數在所考慮的控制體積 V 之外不存在，則配重殘值積分式可以寫

成

$$\int_V R(x)\delta(x-x_j)dx = R(x_j) \equiv 0$$

這種方法表示在配置點位置上的殘值一定為零。當 N 值增大時，殘值 $R(x)$ 會在越多的點上為零，使得近似解趨近於實際解。

蘭若斯 (Lanczos, C. 1938) 選擇柴比雪夫多項式 (Chebyshev Polynomial) 作為基礎函數 $u_i(x)$ ，柴比雪夫多項式是一種正交多項式。蘭若斯並利用柴比雪夫多項式的根作為配置點，這種方法即稱為正交配置法 (Orthogonal Collocation)。

2. 巴諾夫 - 葛勒金法 (Bubnov-Galerkin Method)

在配重殘值法中最有名的方法，可能就是巴諾夫 - 葛勒金法。在這種方法中，定義配重函數 $W_j(x)$ 為基礎函數 $u_j(x)$ 對未定係數 C_j 的導函數，以數學式表示為

$$W_j(x) = \frac{\partial \theta_N}{\partial C_j} \equiv u_j(x) \quad ; \quad j = 1, 2, 3, \ldots\ldots, N \qquad \textbf{(10-1.10)}$$

在這種方法中，$u_j(x)$ 為一組基礎函數之一，使得在所考慮控制體積內，任何函數都可以利用這組基礎函數表示之， $f = \sum a_i u_i$ 。因此，當 N 趨近於無窮大時，近似解在所考慮的空間內即能代表正確解。

3. 高斯 - 李根德最小平方法 (Gauss-Legendre Least Square Method)

高斯 - 李根德最小平方法的基本觀念，是利用殘值的平方積分式為零，找出讓殘值最小化的未定係數。 $\mathrm{Min[I}(c)] = \mathrm{Min}\left[\int_v R^2(x)dV\right]$。亦即，

$$\frac{\partial \mathrm{I}}{\partial C_j} = 2\int_v R(x)\frac{\partial R(x)}{\partial C_j}dV = 0 \quad , \quad j = 1, 2, \cdots, N$$

由以上關係式，可以建立 N 組代數方程式。這種方法極具有數學意義，理論上是強迫讓誤差值的平方和達最小化。由上一方程式，比較配重函數之定義，可知這種方法的配重函數 $W_j(x)$ 為

$$W_j(x) = \frac{\partial R}{\partial C_j} \quad ; \quad j = 1, 2, \ldots\ldots, N \qquad \textbf{(10-1.11)}$$

最小平方法在許多工程應用上常被使用，但由於使用殘值的平方積分式，也很容易產生很困擾的代數方程式。

3. 慣量法 **(Method of Meoments)**

慣量法最先被應用在非線性滲透 (Non-linear Diffusion) 及流體力學中的層流境界層 (Laminar Boundary Layer) 問題的解析上，基本上要求殘值的連續階次慣量爲零。亦即配重函數爲

$$W_j(x) = x^{j-1} \;;\quad j = 1, 2, \ldots\ldots, N \tag{10-1.12}$$

第一階近似慣量法的結果，與考慮整個區間視爲一個子區間的區間法相同，這種近似法在層流境界層問題的解析上亦稱爲馮卡門及薄豪森近似法 (von Karman & Pohlhausen Similarity)。

4. 副區間法 **(Subdomain Method)**

副區間法是將原來的區間 (0 , 1) 分割成 N 個副區間，並定義配重函數爲

$$W_j(x) = \begin{cases} 1 \;;\quad x_{j-1} < x < x_j ;\quad j = 1, 2, \ldots\ldots, N \\ 0 \;;\quad x \text{ 在其他範圍} \end{cases} \tag{10-1.13}$$

例題 10-1 配重殘值法

試利用各種不同的配重殘值法解邊界值常微分方程式 (BVP-ODE)

$$\theta''(x) = x \;,\quad 0 < x < 1$$

$$\text{BC1}\quad \theta(x) = 1$$

$$\text{BC2}\quad \theta'(1) + \theta(1) = 0$$

【註】正確解爲 $\theta = 1 - \frac{5}{6}x + \frac{1}{6}x^3$

解：

1. N = 1；$\theta = 1 - \frac{x}{2} + C_2(x^2 - \frac{3}{2}x)$；殘值表示式爲

$$R = \theta_1''(x) - x = 2C_2 - x$$

根據 MWR 的基本原理：$\int_0^1 WRdx = 0$，以下以各種不同配重函數，建立未定係數 C_2 的值。

(1) 配置法：配重函數爲 $W = \delta(x - x_1)$

令配置點 $x_1 = 0.5$；得 $R = 2C_2 - 0.05 = 0$，即 $C_2 = 0.25$，近似函數爲

$$\theta_1 = 1 - \frac{7}{8}x + \frac{1}{4}x^2$$

若令 $x_1 = 0.6$ ；則得 $C_2 = 0.3$ 。近似函數為

$$\theta_1 = 1 - 0.95x + 0.3x^2$$

注意：配置法所得結果會因配置點位置的選擇而異。

(2) 巴諾夫 - 葛勒金法：配重函數為 $W(x) = \dfrac{\partial\theta_1}{\partial C_2} = x^2 - \dfrac{3}{2}x$

$$\int_0^i WRdx = \int_0^1 (x^2 - \frac{3}{2}x)(2C_2 - x)dx = -\frac{5}{6}C_2 + \frac{1}{4} = 0$$

故 $C_2 = 0.3$ ； $\theta_1 = 1 - 0.95x + 0.3x^2$ 。

(3) 最小平方法：配重函數為 $W(x) = \dfrac{\partial R}{\partial C_2} = 2$

$$\int_0^1 WRdx = \int_0^1 2(2C_2 - x)dx = 4C_2 - 1 = 0$$

故 $C_2 = 0.25$ ； $\theta_1 = 1 - \dfrac{7}{8}x + \dfrac{1}{4}x^2$ 。

(4) 慣量法：配重函數為 $W_1(x) = x^0 = 1$

$$\int_0^1 WRdx = \int_0^1 (2C_2 - x)dx = 2C_2 - \frac{1}{2} = 0$$

得到 $C_2 = \dfrac{1}{4}$ ； $\theta_1 = 1 - \dfrac{7}{8}x + \dfrac{1}{4}x^2$ 。

(5) 副區間法：配重函數為 $W(x) = 1; \quad 0 < x < 1$

$$\int_0^1 WRdx = \int_0^1 (2C_2 - x)dx = 2C_2 - \frac{1}{2}$$

得到 $C_2 = \dfrac{1}{4}$ ； $\theta_1 = 1 - \dfrac{7}{8}x + \dfrac{1}{4}x^2$ 。

2. N = 2；考慮二階近似函數，即取 N = 2；由方程式 (10-1.5) 得二階近似解為

$$\theta_2 = 1 - \frac{1}{2}x + C_2(x^2 - \frac{3}{2}x) + C_3(x^3 - 2x)$$

代回原微分方程式，得到殘值為 $R = \theta_2'' - x = 2C_2 + 6C_3x - x$

(1) 配置法：配重函數為 $W_j = \delta(x - x_j)$ ； $j = 1, 2$

令配置點爲 $x_1 = \frac{1}{3}$ 及 $x_2 = \frac{2}{3}$；則得配置點殘值爲

$$R_{x=\frac{1}{3}} = 2C_2 + 6C_3 \cdot \frac{1}{3} - \frac{1}{3} = 0$$

$$R_{x=\frac{2}{3}} = 2C_2 + 6C_3 \cdot \frac{2}{3} - \frac{2}{3} = 0$$

解之，得 $C_2 = 0$，且 $C_3 = \frac{1}{6}$，代回原近似函數方程式，得到微分方程式的二階近似解爲

$$\theta_2 = 1 - \frac{5}{6}x + \frac{1}{6}x^3$$

與正確解相同。

(2) 巴諾夫 - 葛勒金法：配重函數爲 $W_j(x) = \frac{\partial \theta_N}{\partial C_j}$

$$W_1(x) = \frac{\partial \theta_2}{\partial C_2} = x^2 - \frac{3}{2}x$$

$$W_2(x) = \frac{\partial \theta_2}{\partial C_3} = x^3 - 2x$$

配重積分式可以分成兩組

$$\int_0^1 W_1 R dx = \int_0^1 (x^2 - \frac{3}{2}x)(2C_2 + 6C_3 x - x)dx = -\frac{5}{6}C_2 - \frac{3}{2}C_3 + \frac{1}{4} = 0$$

$$\int_0^1 W_2 R dx = \int_0^1 (x^3 - 2x)(2C_2 + 6C_3 x - x)dx = -\frac{3}{2}C_2 - \frac{14}{5}C_3 + \frac{7}{15} = 0$$

解以上二聯立方程式，得到 $C_2 = 0, C_3 = \frac{1}{6}$。亦即近似解爲

$$\theta_2 = 1 - \frac{5}{6}x + \frac{1}{6}x^3$$

亦與正確解完全相同。

表 10.1　配置法與葛勒金法準確度之比較

x	正確解	N = 1		N = 2	
		配置法*	葛勒金法	配置法	葛勒金法
0.1	0.9168	0.908	0.908	0.9168	0.9168
0.2	0.8347	0.822	0.822	0.8347	0.8347
0.3	0.7545	0.742	0.742	0.7545	0.7545
0.4	0.6773	0.668	0.668	0.6773	0.6773
0.5	0.6042	0.600	0.600	0.6042	0.6042
0.6	0.5360	0.538	0.538	0.5360	0.5360
0.7	0.4738	0.482	0.482	0.4738	0.4738
0.8	0.4187	0.432	0.432	0.4187	0.4187
0.9	0.3715	0.388	0.388	0.3715	0.3715

* 配置法 N = 1 時，配置點選用 $x = 0.6$；若配置點選用 $x = 0.5$，則結果略差。

由本例可發現各種配重殘值法均能很快地獲得此問題的近似解。但一般而言較低階的近似解，以葛勒金法能獲得較佳的結果。高階近似解則各方法所得結果都相當準確，但由於配置法只需解一組代數聯立方程式，即可求得係數 C_i ，而其他方法則均須先作積分處理，因此，N 值較大時，配置法最具使用潛力，而且程式設計也最容易。因此，本章其餘各節將著重於各種數學配置法的介紹。

第二節　配重殘值法的應用

Visual Basic

爲了說明配重殘值法的基本使用方法，其次再以**圓柱狀**觸媒粒子的徑向質傳滲透及等溫 n 次非可逆反應模式爲例，說明配重殘值法的應用方法。

$$\frac{d^2y}{dx^2}+\frac{1}{x}\frac{dy}{dx}-\Phi^2 y^n=0 \qquad \textbf{(10-2.1)}$$

BC1　$x=0$時，　$\dfrac{dy}{dx}=0$

BC2　$x=1$時，　$y=1$

其中 $y=C/C_s$ 爲無因次濃度， $x=r/R$ 爲無因次半徑， Φ 稱爲希笠數 (Thiele modulus)。觸媒粒子的有效度係數爲

$$\eta=\int_0^1 y\,dx^2 \qquad \textbf{(10-2.2)}$$

最低階配重殘值法 $y_1(x)$

當 $n=1$ 時，考慮滿足兩個邊界條件即方程式 (10-2.1) 的最低階近似函數爲 $y_1(x)=1+a_1(1-x^2)$，代入方程式 (10-2.1) 得到殘值表示式爲

$$R_1(a_1, x)=L\{y_1, x\}-L\{y, x\}=-4a_1-\Phi^2\left[1+a_1(1-x^2)\right] \tag{10-2.3}$$

根據配重殘值法的定義

$$\iiint_v WRdV=0=\int WR\cdot 2\pi rLdr=\pi L\int WRdr^2$$

或

$$\int_0^1 R_1(a_1, x)W(x)dx^2=0 \tag{10-2.4}$$

1. 配置法

配重函數爲 $W_j(x)=\delta(x-x_j)$, $j=1$ ；令 $x_1=1/2$ ，代入方程式 (10-2.4) 得到 $R_1(a_1, x_1)=-4a_1-\Phi^2\left[1+\frac{3}{4}a_1\right]=0$，故得到未定係數 a_1 值及一階近似函數 $y_1(x)$ 爲

$$a_1=-4\Phi^2/(16+3\Phi^2)\quad ;\quad y_1=\frac{1-\frac{1}{4}(\frac{\Phi}{2})^2+(\frac{\Phi x}{2})^2}{1+\frac{3}{4}(\frac{\Phi}{2})^2} \tag{10-2.5}$$

觸媒粒子的有效度係數爲 $\eta_1=\dfrac{1+\frac{1}{4}(\frac{\Phi}{2})^2}{1+\frac{3}{4}(\frac{\Phi}{2})^2}$ 。

2. 葛勒金法

配重函數爲 $W_j(x)=\partial y_N/\partial a_j$ ，或 $W(x)=\partial y_1/\partial a_1=(1-x^2)$ ；代入方程式 (10-2.4) 得到 $\int_0^1 R_1(a_1, x)(1-x^2)dx^2=0$，積分後得到未定係數 a_1 值及一階近似函數 $y_1(x)$ 爲

$$a_1=-3\Phi^2/(12+2\Phi^2)\quad ;\quad y_1=\frac{1-\frac{1}{3}(\frac{\Phi}{2})^2+(\frac{\Phi x}{2})^2}{1+\frac{2}{3}(\frac{\Phi}{2})^2} \tag{10-2.6}$$

觸媒粒子的有效度係數爲 $\eta_1 = \dfrac{1+\frac{1}{6}(\frac{\Phi}{2})^2}{1+\frac{2}{3}(\frac{\Phi}{2})^2}$ 。

3. 副區間法

配重函數爲 $W_j(x)=1$，$x_{j-1}<x<x_j$；其他區間 $W_j(x)=0$。代入方程式 (10-2.4) 得到 $\int_0^1 R_1(a_1\,,x)dx^2=0$，積分後得到未定係數 a_1 值及一階近似函數 $y_1(x)$ 爲

$$a_1=-2\Phi^2/(8+\Phi^2)\quad;\quad y_1=\frac{1-\frac{1}{2}(\frac{\Phi}{2})^2+(\frac{\Phi x}{2})^2}{1+\frac{1}{2}(\frac{\Phi}{2})^2} \tag{10-2.7}$$

觸媒粒子的有效度係數爲 $\eta_1=\dfrac{1}{1+\frac{1}{2}(\frac{\Phi}{2})^2}$。

4. 慣量法

配重函數爲 $W_j(x)=x^{2(j-1)}=x^0=1$；代入方程式 (10-2.4) 得到 $\int_0^1 R_1(a_1\,,x)dx^2=0$，積分後得到未定係數 a_1 值及一階近似函數 $y_1(x)$ 爲

$$a_1=-2\Phi^2/(8+\Phi^2)\quad;\quad y_1=\frac{1-\frac{1}{2}(\frac{\Phi}{2})^2+(\frac{\Phi x}{2})^2}{1+\frac{1}{2}(\frac{\Phi}{2})^2} \tag{10-2.8}$$

觸媒粒子的有效度係數爲 $\eta_1=\dfrac{1}{1+\frac{1}{2}(\frac{\Phi}{2})^2}$。

5. 最小平方法

配重函數爲殘值對未定係數之微分，$W_j(x)=\partial R_N/\partial a_j$，或 $W(x)=\partial R_1/\partial a_1=-4-\Phi^2(1-x^2)$；將配重函數代入方程式 (10-2.4) 得到 $\int_0^1 R_1(a_1\,,x)[-4-\Phi^2(1-x^2)]dx^2=0$，積分後得到未定係數 a_1 值及一階近似函數 $y_1(x)$ 爲

$$a_1=\frac{-\Phi^2(1+\frac{1}{8}\Phi^2)}{4+\Phi^2+\frac{1}{12}\Phi^4}\quad;\quad y_1=\frac{1-\frac{1}{6}(\frac{\Phi}{2})^4+(1+\frac{1}{2}(\frac{\Phi}{2})^2)(\frac{\Phi x}{2})^2}{1+(\frac{\Phi}{2})^2+\frac{1}{3}(\frac{\Phi}{2})^4} \tag{10-2.9}$$

觸媒粒子的有效度係數爲 $\eta_1=\dfrac{1+\frac{1}{2}(\frac{\Phi}{2})^2+\frac{1}{12}(\frac{\Phi}{2})^4}{1+(\frac{\Phi}{2})^2+\frac{1}{3}(\frac{\Phi}{2})^4}$。

方法比較：

當 $\Phi = 2$ 時，比較各種方法所得結果，如表 10.2 所示。可以發現葛勒金法結果最佳。

表 **10.2** 一階配重殘值法所得結果之比較

x	正確解	泰勒級數	配置法	葛勒金法	副區間法	最小平方法
0.000	0.439	0.50	0.429	0.400	0.333	0.357
0.200	0.456	0.52	0.451	0.424	0.360	0.383
0.400	0.512	0.58	0.520	0.496	0.440	0.460
0.600	0.611	0.68	0.634	0.616	0.573	0.589
0.800	0.768	0.82	0.794	0.784	0.760	0.769
1.000	1.000	1.00	1.000	1.000	1.000	1.000
有效度	0.6978	0.7500	0.7143	0.7000	0.6667	0.6786
誤　差		7.48%	2.36%	0.32%	–4.46%	–2.75%

高階配重殘值法 $y_N(x)$

當 $n=1$ 時，考慮滿足兩個邊界條件及方程式 (10-2.1) 的 N 階近似函數為 $y_N = \phi_0 + \sum_{i=1}^{N} a_i\phi_i$，其中 ϕ_0 為滿足微分方程式邊界條件的函數，ϕ_i 為滿足微分方程式均勻邊界條件 (homogeneous boundary conditions) 的函數。就方程式 (10-2.1) 而言，

$$\phi_0 = 1 \quad , \quad \phi_i = (1-x^2)x^{2(i-1)} \quad , \quad i = 1, 2, \cdots \qquad \textbf{(10-2.10)}$$

或將 N 階近似函數寫成下式，其中令 $u = x^2$, $p = \Phi^2/4$

$$\begin{aligned} y_N(x) &= 1 + (1-x^2)\sum_{i=1}^{N} a_i x^{2(i-1)} \\ y_N(u) &= 1 + (1-u)\sum_{i=1}^{N} a_i u^{i-1} \end{aligned} \qquad \textbf{(10-2.11)}$$

方程式 (10-2.1) 可以改寫成

$$u\frac{d^2y}{du^2}+\frac{dy}{du}-py=0$$

$$\text{BC:}\quad 在\ u=1\ 處,\quad y=1\qquad \frac{\partial^k y}{\partial u^k}=有限值,\ k\ge 0 \tag{10-2.12}$$

$$\eta_N=\int_0^1 y_N dx^2=\int_0^1 y_N du$$

將方程式 (10-2.11) 代入方程式 (10-2.12)，經整理後，得到

$$\begin{aligned}R_N(\underline{a},u)&=L\{y_N,u\}-L\{y,u\}\\&=\sum_{i=1}^N a_i\left[(i-1)^2u^{i-2}-i^2u^{i-1}\right]-p\left[1+(1-u)\sum_{i=1}^N a_iu^{i-1}\right]\end{aligned} \tag{10-2.13}$$

根據配重殘值法的定義

$$\iiint_v WRdV=0=\int WR\cdot 2\pi rLdr=\pi L\int WRdr^2$$

即 $\int_0^1 R_1(a_1,x)W(x)dx^2=0$ 或再寫成 u 的函數爲

$$\int_0^1 R_N(\underline{a},u)W_j(u)du=0,\quad i=1,2,\cdots,N \tag{10-2.14}$$

將方程式 (10-2.13) 代入方程式 (10-2.14)，積分結果可以寫成以下的矩陣形式，

$$\begin{gathered}(\underline{\underline{A}}+p\cdot\underline{\underline{B}})\underline{a}=p\cdot\underline{C}\\ \sum_{i=1}^N(A_{ji}+pB_{ji})a_i=pC_j\end{gathered} \tag{10-2.15}$$

其中矩陣 $\underline{\underline{A}}$, $\underline{\underline{B}}$ 及 $\underline{C}$ 分別因配重函數 $W_j(u)$ 的定義而會有改變。解方程式 (10-2.15) 得到未定係數陣列爲

$$\underline{a}=(\underline{\underline{A}}+p\cdot\underline{\underline{B}})^{-1}\cdot p\cdot\underline{C} \tag{10-2.16}$$

將此結果代入方程式 (10-2.11)，即可得到微分方程式的近似解。同時，也可以求得觸媒的有效度係數爲

$$\eta_N=\int_0^1 y_N du=\int_0^1\left[1+(1-u)\sum_{i=1}^N a_iu^{i-1}\right]du=1+\sum_{i=1}^N\frac{a_i}{i(i+1)} \tag{10-2.17}$$

1. 配置法

配重函數爲 $W_j(u)=\delta(u-u_j)$, $j=1,2,\cdots,N$ ；代入方程式 (10-2.13) 及方程式 (10-2.14)，得到

$$R_N(\underline{a},u_j)=\sum_{i=1}^{N}a_i\left[(i-1)^2u_j^{i-2}-i^2u_j^{i-1}\right]-p\left[1+(1-u_j)\sum_{i=1}^{N}a_iu_j^{i-1}\right]=0$$

或

$$\sum_{i=1}^{N}\left\{\left[(i-1)^2u_j^{i-2}-i^2u_j^{i-1}\right]-p\left[(1-u_j)u_j^{i-1}\right]\right\}a_i=p$$

與方程式 (10-2.15) 比較，得到矩陣$\underline{\underline{A}}$, $\underline{\underline{B}}$ 及$\underline{C}$分別爲

$$\begin{aligned}&A_{ji}=\left[(i-1)^2u_j^{i-2}-i^2u_j^{i-1}\right]\\&B_{ji}=(u_j-1)u_j^{i-1}\\&C_j=1\end{aligned}\qquad\textbf{(10-2.18)}$$

若考慮使用等間距配置點，由於 $u_0=0$, $u_{N+1}=1$ ，則 $u_j=j/(N+1)$ ，代回方程式 (10-2.17)，得到

$$\begin{aligned}&A_{ji}=\left[(i-1)^2(\frac{j}{N+1})^{i-2}-i^2(\frac{j}{N+1})^{i-1}\right]\\&B_{ji}=(\frac{j-N-1}{N+1})(\frac{j}{N+1})^{i-1}\\&C_j=1\end{aligned}\qquad\textbf{(10-2.19)}$$

2. 葛勒金法

配重函數爲 $W_j(u)=\partial y_N/\partial a_j=\phi_j(u)=(1-u)u^{j-1}$ ；代入方程式 (10-2.13) 及方程式 (10-2.14)，得到 $\int_0^1 R_N(\underline{a},u)(1-u)u^{j-1}du=0$ ，積分展開後，得到矩陣$\underline{\underline{A}}$ ，$\underline{\underline{B}}$，及$\underline{C}$分別爲

$$\begin{aligned}&A_{ji}=\int_0^1\left[(i-1)^2u^{i-2}-i^2u^{i-1}\right](1-u)u^{j-1}du=\frac{i+j-2ij}{(i+j-2)(i+j-1)(i+j)}\\&B_{ji}=\int_0^1\left[-(1-u)^2u^{i+j-2}\right]du=\frac{-2}{(i+j-1)(i+j)(i+j+1)}\\&C_j=\int_0^1(1-u)u^{j-1}du=\frac{1}{j(j+1)}\end{aligned}\qquad\textbf{(10-2.20)}$$

3. 副區間法

配重函數爲 $W_j(u)=1, u_{j-1}<u<u_j$；其他區間 $W_j(u)=0$。代入方程式 (10-2.14)，得到 $\int_{u_{j-1}}^{u_j} R_N(\underline{a},u)du=0$。將方程式 (10-2.13) 代入，積分後得到得到矩陣 $\underline{\underline{A}}$, $\underline{\underline{B}}$，及 $\underline{C}$ 分別爲

$$A_{ji}=\int_{u_{j-1}}^{u_j}\left[(i-1)^2u^{i-2}-i^2u^{i-1}\right]du=(i-1)\left[u_j^{i-1}-u_{j-1}^{i-1}\right]-i\left[u_j^i-u_{j-1}^i\right]$$

$$B_{ji}=\int_{u_{j-1}}^{u_j}\left[-(1-u)u^{i-1}\right]du=\frac{1}{i+1}\left[u_j^{i+1}-u_{j-1}^{i+1}\right]-\frac{1}{i}\left[u_j^i-u_{j-1}^i\right] \qquad \textbf{(10-2.21)}$$

$$C_j=\int_{u_{j-1}}^{u_j}du=u_j-u_{j-1}$$

若考慮使用等間距區間，由於 $u_0=0, u_{N+1}=1$，則 $u_j=j/(N+1)$。

4. 慣量法

配重函數爲 $W_j(u)=u^{(j-1)}, j=1,2,\cdots,N$；代入方程式 (10-2.13) 及方程式 (10-2.14)，得到

$$\int_0^1 R_N(\underline{a},u)u^{j-1}du=0$$

$$=\int_0^1\left\{\sum_{i=1}^N a_i\left[(i-1)^2u^{i-2}-i^2u^{i-1}\right]-p\left[1+(1-u)\sum_{i=1}^N a_iu^{i-1}\right]\right\}u^{j-1}du$$

積分後得到矩陣 $\underline{\underline{A}}$, $\underline{\underline{B}}$，及 $\underline{C}$ 分別爲

$$A_{ji}=\int_0^1\left[(i-1)u^{i-2}-i^2u^{i-1}\right]u^{j-1}du=\frac{(i-1)^2}{(i+j-2)}-\frac{i^2}{(i+j-1)} \qquad \textbf{(10-2.22)}$$

$$B_{ji}=\int_0^1(u-1)u^{i-1}u^{j-1}du=\frac{-1}{(i+j-1)(i+j)}$$

$$C_j=\int_0^1 u^{j-1}du=\frac{1}{j}$$

5. 最小平方法

配重函數爲殘值對未定係數之微分，$W_j(u)=\partial R_N/\partial a_j$，或 $W_j(u)=(j-1)^2u^{j-2}-j^2u^{j-1}-p(1-u)u^{j-1}$；將此配重函數代入方程式 (10-2.14)，並將方程式 (10-2.13) 代入，積分後可以得到矩陣 $\underline{\underline{A}}$, $\underline{\underline{B}}$，及 $\underline{C}$。由於積分冗長，不予列出。

計算方法與使用策略

配重殘值法的計算方法，可以歸納如下：

1. 利用一個多項式 $y_N = \phi_0 + \sum_{i=1}^{N} a_i \phi_i$ 作爲微分方程式 $L\{y,u\}=0$ 的近似解；
2. 代入微分方程式 $L\{y,u\}=0$，求出殘值的表示式 $R = L\{y_N, u\} - L\{y,u\}$；
3. 選擇使用的配重殘值法，定義配重函數 W_j；
4. 使殘值的配重積分值爲零

$$\int_0^1 W_j(u)\, R_N(\underline{a}, u)\, du = 0,\quad j = 1, 2, \cdots, N$$

5. 建立N個未定係數陣列$\underline{a}$的聯立方程式；

$$(\underline{\underline{A}} + p \cdot \underline{\underline{B}})\underline{a} = p \cdot \underline{C}$$

$$\sum_{i=1}^{N} (A_{ji} + pB_{ji})a_i = pC_j$$

6. 利用數值方法解聯立方程式，找出近似多項式的未定係數陣列$\underline{a}$；
7. 代回原近似多項式 $y_N = \phi_0 + \sum_{i=1}^{N} a_i \phi_i$，建立微分方程式的近似解；
8. 利用適當內插法，計算微分方程式在特定位置的解。

以上所述之計算邏輯如圖 10.1 所示；利用 Excel 或高斯消去法，都可以很輕易地建立所需的程式。利用 Excel 所建立程式如圖 10.2 所示。程式編寫細節可參考本書所附光碟。

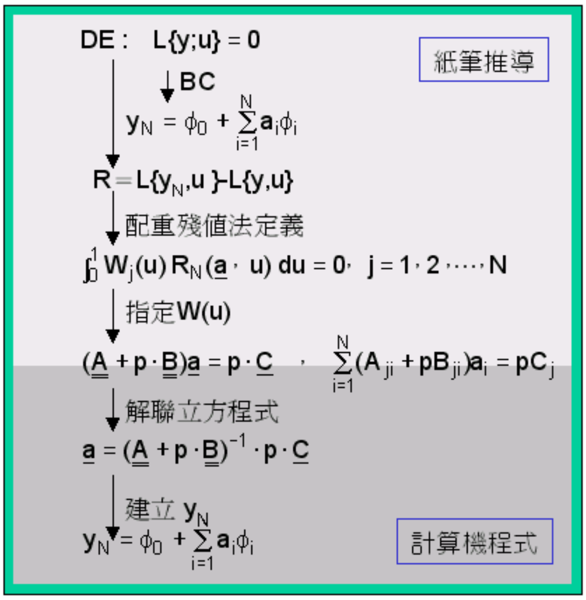


圖 10.1　配重殘值

法的邏輯圖

	A	B	C	D	E	F	G	H	I	J	K	L
1	數學配置法											
2	配置點數:	3	1	2	3		配置點數:	3	1	2	3	
3	Matrix A	1	- 1.0000	-	0.4375		Matrix B	1	-0.7500	-0.1875	-0.0469	
4		2	- 1.0000	- 1.0000	- 0.2500			2	-0.5000	-0.2500	-0.1250	
5		3	- 1.0000	- 2.0000	- 2.0625			3	-0.2500	-0.1875	-0.1406	
6												
7	配置點數:	3	1				$A_{ji} = \left[(i-1)^2 (\frac{j}{N+1})^{i-2} - i^2 (\frac{j}{N+1})^{i-1}\right]$					
8	Vector C	1	1.0000				$B_{ji} = (\frac{j-N-1}{N+1})(\frac{j}{N+1})^{i-1}$					
9		2	1.0000									
10		3	1.0000				$C_j = 1$					
11												
12		J/I	1	2	3			J/I	1	2	3	
13	A+pB	1	-1.7500	-0.1875	0.3906		$(A+pB)^{-1}$	1	-0.8867	0.5813	-0.2562	
14		2	-1.5000	-1.2500	-0.3750			2	1.3005	-1.9919	0.5696	
15		3	-1.2500	-2.1875	-2.2031			3	-0.7882	1.6480	-0.8742	
16												
17		I	a		=a/[i(i+1))							
18	Vect a	1	-0.5616		-0.2808		$\underline{a} = (\underline{A} + p \cdot \underline{B})^{-1} \cdot p \cdot \underline{C}$					
19		2	-0.1218		-0.0203							
20		3	-0.0143		-0.0012							
21												
22			有效度 =		0.697716		$\eta_N = 1 + \sum_{i=1}^{N} \frac{a_i}{i(i+1)}$					
23												

24	程式說明	
25	1	C3 = (C$2-1)^2*($B3/(B2+1))^(C$2-2)-C$2^2*($B3/($B$2+1))^(C$2-1)
26	2	C3..E5 直接複製 C3 公式即可
27	3	I3 = ($H3-$H$2-1)/($H$2+1)*($H3/(H2+1))^(I$2-1)
28	4	I3..K5 直接複製 I3 公式即可
29	5	C13 = C3+I3
30	6	C13..E15 直接複製 C13 公式即可
31	7	I13..K15 = Minverse(C13..E15) 然後按<Ctrl><Shift><Enter>
32	8	C18..C20 = MMULT(I13:K15,C8:C10) 然後按<Ctrl><Shift><Enter>
33	9	E18 = C18/B18/(B18+1)
34	10	E18..E20 直接複製 E18 公式即可
35	11	E22 = 1+SUM(E18:E20)

圖 **10.2** (a) 利用 Excel 編寫配重殘值法的範例，配置法。將微分方程式的求解，變成簡單的矩陣運算。利用 Excel 即可輕易地完成任務，N=3 所得結果與正確解相當一致。

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	數學配置法															
2	配置點數=	5	1	2	3	4	5		配置點數=	5	1	2	3			
3	Matrix A	1	- 1.000	0.333	0.417	0.176	0.055		Matrix B	1	-0.833	-0.139	-0.023	-5.000	-5.000	
4		2	- 1.000	- 0.333	0.333	0.407	0.284			2	-0.667	-0.222	-0.074	-2.000	-2.000	
5		3	- 1.000	- 1.000	- 0.250	0.250	0.438			3	-0.500	-0.250	-0.125	-1.000	-1.000	
6		4	- 1.000	- 1.667	- 1.333	- 0.741	- 0.198			4	-0.333	-0.222	-0.148	-0.500	-0.500	
7		5	- 1.000	- 2.333	- 2.917	- 3.009	- 2.797			5	-0.167	-0.139	-0.116	-0.200	-0.200	
8																
9	配置點數=	5	1													
10	Vector C	1	1.000						$A_{ji}=\left[(i-1)^2(\frac{j}{N+1})^{i-2}-i^2(\frac{j}{N+1})^{i-1}\right]$							
11		2	1.000													
12		3	1.000						$B_{ji}=(\frac{j-N-1}{N+1})(\frac{j}{N+1})^{i-1}$							
13		4	1.000													
14		5	1.000						$C_j=1$							
15																
16		J/I	1	2	3	4	5			J/I	1	2	3	4	5	
17	A+pB	1	-1.833	0.194	0.394	-4.824	-4.945		$(A+pB)^{-1}$	1	2.134	-9.212	11.505	-5.974	0.985	
18		2	-1.667	-0.556	0.259	-1.593	-1.716			2	-2.665	12.096	-16.601	8.421	-1.373	
19		3	-1.500	-1.250	-0.375	-0.750	-0.563			3	2.365	-9.668	13.044	-6.439	0.683	
20		4	-1.333	-1.889	-1.481	-1.241	-0.698			4	-1.215	2.821	-1.072	-1.465	0.932	
21		5	-1.167	-2.472	-3.032	-3.209	-2.997			5	0.275	0.370	-2.835	3.462	-1.274	
22																
23		I	a		=a/[i(i+1))											
24	Vect a	1	-0.562		-0.281				$\underline{a}=(\underline{\underline{A}}+p\cdot\underline{\underline{B}})^{-1}\cdot p\cdot\underline{C}$							
25		2	-0.121		-0.020											
26		3	-0.015		-0.001											
27		4	0.001		0.000											
28		5	-0.001		0.000											
29																
30			有效度 =	0.697517					$\eta_N=1+\sum_{i=1}^{N}\frac{a_i}{i(i+1)}$							
31																

32	程式說明	
33	1	C3 = (C$2-1)^2*($B3/(B2+1))^(C$2-2)-C$2^2*($B3/($B$2+1))^(C$2-1)
34	2	C3..G7 直接複製 C3 公式即可
35	3	K3 = =($J3-$J$2-1)/($J$2+1)*($J3/(J2+1))^(K$2-1)
36	4	K3..O7 直接複製 K3 公式即可
37	5	C17 = C3+K3
38	6	C17..G21 直接複製 C17 公式即可
39	7	K17..O21 = Minverse(C17..G21)　然後按<Ctrl><Shift><Enter>
40	8	C24..C28 =MMULT(K17:O21,C10:C14) 然後按<Ctrl><Shift><Enter>
41	9	E24 =C24/B24/(B24+1)
42	10	E24..E28 直接複製 E24 公式即可
43	11	E30 = 1+SUM(E24:E28)

圖 **10.2**　(b) 利用 Excel 編寫配重殘值法的範例，配置法。N=5 所得結果。

第三節　數學配置法

Visual Basic

上一節中，我們利用簡單的線性問題爲例，說明了各種 MWR 配重殘值法及其特性，本節中則將針對配置法作較詳盡的討論。使用配置法的時候，近似函數及配置點的選擇方法大致可分成三類：

1. **內部配置法 (Interior Collocation)**

   所選用的近似函數滿足邊界條件，配置點選擇位於微分方程式所考慮的區間內，以決定未定係數。

2. **邊界配置法 (Boundary Collocation)**

   所選用的近似函數滿足原微分方程式，而配置點則在邊界上選取，以決定未定係數。

3. **混和配置法 (Mixed Collocation)**

   所選用的近似函數爲任意函數，其未定係數由所考慮區間內及邊界上選定的配置點來決定。

配置法常見於邊界值常微分方程式、特徵值問題 (Eigenvalue Problem) 及偏微分的求解。以下再利用幾個例題加以說明。

例題 10-2　一次元熱傳導 (K≠ 常數)

一平板厚度爲 L，兩側溫度分別爲 T_0 及 T_1。其能量平衡方程式爲

$$\frac{d}{dx}\left[k(T)\frac{dT}{dx}\right]=0;\quad 0<x<L \tag{10-3.1}$$

BC1　$T(0)=T_0$

BC2　$T(L)=T_1$

此平板的熱傳導率與溫度關係爲 $k(T)=k_0+\alpha(T-T_0)$，試求其溫度分布。

解：

令無因次溫度及無因次長度分別爲

$$\theta = \frac{T - T_0}{T_1 - T_0} \qquad \xi = x / L$$

則原微分方程式可以改寫成

$$\frac{d}{d\xi}〔(1 + a\theta)\frac{d\theta}{d\xi}〕= 0\,; \quad 0 < \xi < 1 \qquad \textbf{(10-3.2)}$$

$$\theta(0) = 1$$

$$\theta(1) = 1$$

其中 $a = \alpha(T - T_0) / k_0$；爲了簡化說明起見，令 $a = 1$，則滿足邊界條件的基礎函數，可以寫成

$$\theta_{\mathrm{N}} = \xi + \sum_{j=1}^{N} C_j (\xi^{j+1} - \xi) \qquad \textbf{(10-3.3)}$$

若取 N = 2，其殘值爲

$$R = \frac{d}{d\xi}〔(1 + a\theta_N)\frac{d\theta_N}{d\xi}〕$$

$$= 〔1 + \xi + C_1(\xi^2 - \xi) + C_2(\xi^3 - \xi)〕\cdot(2C_1 + 6C_2\xi) + \qquad \textbf{(10-3.4)}$$

$$〔1 + C_1(2\xi - 1) + C_2(3\xi^2 - 1)〕^2$$

配置點取爲 $\xi_1 = \dfrac{1}{3}$ 及 $\xi_2 = \dfrac{2}{3}$，代入殘值表示式，並讓殘值在配置點上爲 0，則得

$$C_1 = -0.5992$$

$$C_2 = 0.1916$$

故得 θ 的二階近似解爲

$$\theta_2 = 1.4076\xi - 0.5992\xi^2 + 0.1916\xi^3$$

或溫度分布近似解爲

$$T = T_0 + (T_1 - T_0)〔1.4076(\frac{x}{L}) - 0.5992(\frac{x}{L})^2 + 0.1916(\frac{x}{L})^3〕$$

例題 10-3　非線性微分方程式

試利用數學配置法解非線性微分方程式。

$$\frac{d^2\theta}{dx^2}=\exp(\theta)\,;\quad 0<\theta<1 \tag{10-3.5}$$

BC1 $\theta(0)=0$

BC2 $\theta(1)=0$

解：

滿足邊界條件的基礎函數可為

$$\theta_N=\sum_{j=1}^{N}C_j(x^{j+1}-x) \tag{10-3.6}$$

考慮最簡單的情況，N = 1，即 $\theta_1=C_1(x^2-x)$ 。則殘值為

$$R=\frac{d\theta_1}{dx^2}-\exp(\theta_1)=2C_1-\exp〔C_1(x^2-x)〕 \tag{10-3.7}$$

取配置點為 $x=0.5$ ；令 R 在配置點上為 0，得到

$$2C_1-\exp〔-\frac{1}{4}C_1〕=0$$

利用牛頓法求得上式的解為 $C_1=0.44712$ 。故 θ_1 為

$$\theta_1=0.44712x(x-1)$$

與正確解之比較如下表，結果顯示用最簡單的近似解，效果還不錯呢！

x	0.1	0.2	0.3	0.4	0.5
正確解	−0.0414	−0.0733	−0.0958	−0.1092	−0.1137
θ_1	−0.0402	−0.0715	−0.0939	−0.1073	−0.1178
誤差%	2.8	2.4	1.99	1.73	1.69

第四節　正交配置法

Visual Basic

配置法使用方便，但其準確度卻決定於基礎函數的選擇、N 值的大小及配置點的選擇。1938 年蘭若斯 (Lanczos) 提議利用正交多項式 (Orthogonal Polynomials) 作為基礎函數，並以正交多項式的根作為配置點；後來再經過魏德森及史都華 (Villadsen and Stewart) 的改進，目前已成為被使用得最廣泛的配置法，通稱為「正交配置法」

(Orthogonal Collocation Method)。

正交配置法基本上是以正交多項式作爲基礎函數；如果基礎函數的最高次數爲 N，則利用 N+1 次的正交多項式的根作爲配置點。由於正交多項式具有許多易於處理的數學特性，因此，正交配置法基本原理雖與前述配置法完全一致，但使用上卻可善用正交多項式的特性，使處理方法更爲簡化，更適於利用計算機來求解。

假設一微分方程式的近似解，可以利用一組正交多項式 $P(x)$ 的線性和來表示，

$$y_N(x)=\sum_{i=1}^{N+2}\alpha_i\ P_{i-1}(x) \tag{10-4.1}$$

在以上的方程式中總共有 N+2 個未定係數 α_i，但近似解 y_N 的下註標只寫成 N，是由於邊界值常微分方程式 (BVP ODE) 的兩個邊界條件已經可以決定兩個 α_i 值，只需要再利用 N 個內部配置點即可以決定所有 α_i 值，故寫成 y_N；其中，N 代表所需的內部配置點數。

正交多項式 $P(x)$ 可以寫成多項式的一般表示式

$$P_m(x)=\sum_{j=0}^{m}\beta_j\ x^j \tag{10-4.2}$$

其正交性質爲

$$\int_a^b W(x)P_k(x)P_m(x)dx=0;\quad k=0,1,2,\cdots,m-1 \tag{10-4.3}$$

其中 (a,b) 爲邊界值常微分方程式的定義範圍，通常都先利用座標變換轉移成 (0 , 1)，以便處理。

將方程式 (10-4.2) 代入方程式 (10-4.1) 中，可將基礎函數 y_N 改寫成以下的方式，以利計算處理：

$$y_N=\sum_{i=1}^{N+2}d_i x^{i-1} \tag{10-4.4}$$

假設可以利用適當方法得到配置點，則在配置點 x_j 位置上的函數值 y_N 爲

$$y_N(x_j)=\sum_{i=1}^{N+2}d_i\ x_j^{i-1};\quad j=1,2,\cdots,N+2 \tag{10-4.5}$$

$$x_1=0\,,\ x_{N+2}=1$$

而在配置點 x_j 位置上，y_N 的一次及二次導函數，亦可分別由方程式 (10-4.4) 對 x

作微分，得到

$$\frac{1}{dx}y_N(x_j)=\sum_{i=1}^{N+2}d_i(i-1)x_j^{i-2} \tag{10-4.6}$$

$$\frac{d^2}{dx^2}y_N(x_i)=\sum_{i=1}^{N+2}d_i(i-1)(i-2)x_j^{i-3} \tag{10-4.7}$$

令矩陣$\underline{\underline{Q}}$，$\underline{\underline{C}}$及$\underline{\underline{D}}$分別爲

$$\begin{aligned}Q_{ji}&=x_j^{i-1}\\ C_{ji}&=(i-1)x_j^{i-2}\\ D_{ji}&=(i-1)(i-2)x_j^{i-3}\end{aligned} \tag{10-4.8}$$

令陣列 $\underline{y}=[y_N(x_1)\quad y_N(x_2)\quad \cdots \quad y_N(x_N)]^T,\underline{d}=[d_1\quad d_2\quad \cdots\quad d_N]^T$，則方程式 (10-4.5) , (10-4.6) 及 (10-4.7) 分別改寫成矩陣符號後，得到

$$\underline{y}=\underline{\underline{Q}}\underline{d} \tag{10-4.9}$$

$$\frac{d}{dx}\underline{y}=\underline{\underline{C}}\underline{d} \tag{10-4.10}$$

$$\frac{d^2}{dx^2}\underline{y}=\underline{\underline{D}}\underline{d} \tag{10-4.11}$$

根據方程式 (10-4.8) 可知，若已知配置點位置，則矩陣$\underline{\underline{Q}}$，$\underline{\underline{C}}$及$\underline{\underline{D}}$都可以很快的計算出來。而由方程式 (10-4.9)，可以將未定義的係數陣列$\underline{d}$表示成配置點上的函數值$\underline{y}$的關係

$$\underline{d}=\underline{\underline{Q}}^{-1}\underline{y} \tag{10-4.12}$$

將上式代回方程式 (10-4.10) 及方程式 (10-4.11) 中，可以分別得到

$$\frac{d}{dx}\underline{y}=\underline{\underline{C}}\underline{d}=\underline{\underline{C}}\underline{\underline{Q}}^{-1}\underline{y}\equiv\underline{\underline{A}}\underline{y} \tag{10-4.13}$$

$$\frac{d^2}{dx^2}\underline{y}=\underline{\underline{D}}\underline{d}=\underline{\underline{D}}\underline{\underline{Q}}^{-1}\underline{y}\equiv\underline{\underline{B}}\underline{y} \tag{10-4.14}$$

根據方程式 (10-4.8) 可知，若選定一組配置點x_j，則Q_{ji}, C_{ji}及D_{ji}均可直接求得，而上二方程式中$\underline{\underline{A}}=\underline{\underline{C}}\underline{\underline{Q}}^{-1}$, $\underline{\underline{B}}=\underline{\underline{D}}\underline{\underline{Q}}^{-1}$也都可以利用矩陣操作很快的求得。因此，可將微分方程式中的一次及二次導函數項，都表示成配置點位置的函數值$\underline{y}$的線性函數，而將原微分方程式轉化成一組聯立代數方程式。矩陣 $\underline{\underline{A}}$ 及 $\underline{\underline{B}}$ 即稱爲**微分操作矩陣**。

利用微分操作矩陣，可以將微分方程式改寫成函數值 $\underline{y}$ 的線性函數，亦即變成一組線性聯立方程式。利用高斯消去法或適當數值方法，即可求得在配置點處的函數值 $\underline{y}$。

以上所說明的正交配置法基本邏輯，可以歸納如下：

1. 利用正交多項式的解，得到一組配置點 x_j；

2. 利用方程式 (10-4.8) 及配置點 x_j，建立矩陣 $\underline{\underline{Q}}$ ，$\underline{\underline{C}}$ 及 $\underline{\underline{D}}$；

3. 建立一次微分操作矩陣 $\frac{d}{dx}\underline{y} \equiv \underline{\underline{A}}\underline{y}$，其中 $\underline{\underline{A}} = \underline{\underline{C}}\underline{\underline{Q}}^{-1}$；

4. 建立二次微分操作矩陣 $\frac{d^2}{dx^2}\underline{y} \equiv \underline{\underline{B}}\underline{y}$，其中 $\underline{\underline{B}} = \underline{\underline{D}}\underline{\underline{Q}}^{-1}$；

5. 利用邊界條件建立 y_1 及 y_{N+2} 的表示式（ 在本章第五節中詳細說明 ）；

6. 將微分操作矩陣代入原微分方程式，建立函數值 $\underline{y}$ 的聯立方程式；

7. 利用高斯消去法或其他適當數值方法，求解函數值 $\underline{y}$ 的聯立方程式；

8. 利用適當內插法，建立特定點之函數值 y。

例題 10-4　利用配置法解邊界值常微分方程式

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

$$\text{BC1}\quad y(0)=0$$

$$\text{BC2}\quad y(1)=1$$

解：

爲了說明起見，我們只考慮最簡單的情況 $N=1$，則 $N+2$ 個配置點分別爲 $x_1 = 0$, $x_2 = 1/2$ 及 $x_3 = 1$。根據方程式 (10-4.8) 得 $\underline{\underline{Q}}$ ，$\underline{\underline{C}}$ 及 $\underline{\underline{D}}$ 分別爲

$$Q_{ji} = x_j^{i-1} \quad ; \quad \underline{\underline{Q}} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & \frac{1}{2} & \frac{1}{4} \\ 1 & 1 & 1 \end{bmatrix}$$

$$C_{ji} = (i-1)x_j^{i-2} \quad ; \quad \underline{\underline{C}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$

$$D_{ji}=(i-1)(i-2)x_j^{i-3} \quad ; \quad \underline{\underline{D}}=\begin{bmatrix}0&0&2\\0&0&2\\0&0&2\end{bmatrix}$$

矩陣 $\underline{\underline{Q}}$ 的反矩陣 $\underline{\underline{Q}}^{-1}$ 可利用高斯約旦法求得爲

$$\underline{\underline{Q}}^{-1}=\begin{bmatrix}1&0&0\\-3&4&-1\\2&-4&2\end{bmatrix}$$

因此，微分操作矩陣可分別根據其定義求得。

$$\underline{\underline{A}}=\underline{\underline{C}}\underline{\underline{Q}}^{-1}=\begin{bmatrix}-3&4&-1\\-1&0&0\\1&-4&3\end{bmatrix}$$

$$\underline{\underline{B}}=\underline{\underline{D}}\underline{\underline{Q}}^{-1}=\begin{bmatrix}4&-8&4\\4&-8&4\\4&-8&4\end{bmatrix}$$

代回原微分方程式得到

$$(1+y_j)\left[\sum_{i=1}^{N+2}B_{ji}y_i\right]+\left[\sum_{i=1}^{N+2}A_{ji}y_i\right]^2=0; \quad j=1,2,\cdots,N+2$$

由於 $y_1=0$, $y_{N+2}=y_3=1$，只剩下 y_2 未知，因此，只考慮 $N=1$ 且 $j=2$ 的情況，由上式得

$$(1+y_2)(4y_1-8y_2+4y_3)+(-y_1+y_3)^2=0$$

將 y_1 及 y_3 代入，得 $(1+y_2)(4-8y_2)+1=0$ ，由與 $y_2>0$ ，因此，可以得到 $y_2=$ 0.57916。

利用這種處理方式，我們可獲得各配置點位置的 y 值。若需其他位置的 y 值，則可利用本書第三章所介紹的插值法求之。需要求 y 的積分值，則可利用本書第六章所介紹的高斯積分法 (Gauss Quadrature) 求之。

第五節　正交配置法邊界條件的處理

Visual Basic

微分方程式的邊界條件通常可分爲三大類，本節將以實例來說明使用正交配置法時，邊界條件的處理方法。

考慮一常微分方程式爲例

$$\frac{d^2y}{dx^2}+3\frac{dy}{dx}+2y=4x \tag{10-5.1}$$

其正確解爲

$$y=C_1e^{-2x}+C_2e^{-x}+2x-3 \tag{10-5.2}$$

利用上一節所介紹的處理方法，方程式 (10-5.1) 可以被改寫成

$$\left(\sum_{i=1}^{N+2}B_{ji}y_i\right)+3\left(\sum_{i=1}^{N+2}A_{ji}y_i\right)+2y_j=4x_j\,;\quad j=1,2,\cdots,N+1 \tag{10-5.3}$$

或將 y_1 及 y_{N+1} 移到方程式右側，改寫成

$$\sum_{i=2}^{N+1}(B_{ji}+3A_{ji}+2\delta_{ji})y_j=4x_j-B_{j1}y_1-3A_{j1}y_1-B_{j,N+2}y_{N+2}-3A_{j,N+2}y_{N+2} \tag{10-5.4}$$

其中 δ_{ji} 稱爲 Kronecker Delta，定義爲

$$\delta_{ji}=\begin{cases}1 & ; \quad i=j\\ 0 & ; \quad i\neq j\end{cases} \tag{10-5.5}$$

方程式 (10-5.4) 中，y_1 及 y_{N+2} 通常利用邊界條件決定之，以下分別利用各種不同的邊界條件加以說明。

1. 邊界條件 (I - I)

兩側邊界條件均爲第一類 (Dirichlet) 邊界條件：

$$\begin{aligned}&\text{BC1}\quad y(0)=0\,;\; x=0\\ &\text{BC2}\quad y(1)=1\,;\; x=1\end{aligned} \tag{10-5.6}$$

上式可改寫成 $y_1=0,\,y_{N+2}=1$，代入方程式 (10-5.4) 得到

$$\sum_{i=2}^{N+1}(B_{ji}+3A_{ji}+2\delta_{ji})y_i=4x_j-B_{j,N+2}-3A_{j,N+2}$$

或利用矩陣符號簡寫成

$$(\underline{\underline{B}} + 3\underline{\underline{A}} + 2\underline{\underline{I}})\underline{y} = \underline{f}$$

或進一步簡化，並寫成以下的矩陣方程式

$$\underline{\underline{M}}\,\underline{y} = \underline{f} \tag{10-5.7}$$

其中

$$M_{ji} = B_{ji} + 3A_{ji} + 2\delta_{ji}$$
$$f_i = 4x_i - B_{i,N+2} - 3A_{i,N+2}$$

方程式 (10-5.7) 代表一組線性聯立方程式，可利用高斯消去法解之。

2. 邊界條件 (I - II)

原點側邊界條件爲第一類邊界條件，在 $x=1$ 位置之邊界條件爲第二類邊界條件：

$$\text{BC1} \quad y(0)=0\text{；}\ x=0$$
$$\text{BC2} \quad \frac{d}{dx}y(1)=0\text{；}\ x=1 \tag{10-5.8}$$

若以 y_i 表示，則得到

$$y_1 = 0$$
$$\sum_{i=1}^{N+2} A_{N+2,i}\,y_i = 0$$

由上式求得 y_{N+2} 爲

$$y_{N+2} = -\frac{1}{A_{N+2,N+2}}\sum_{i=2}^{N+1} A_{N+2,i}\,y_i$$

代回方程式 (10-5.4)，得到

$$\sum_{i=2}^{N+1}\left[B_{ji} + 3A_{ji} + 2\delta_{ji} - \frac{A_{N+2,j}}{A_{N+2,N+2}}(B_{i,N+2} + 3A_{i,N+2})\right]y_i = 4x_j$$
$$j = 2, 3, \cdots, N+1$$

或寫成

$$\underline{\underline{M}}\,\underline{y} = \underline{f} \tag{10-5.9}$$

其中

$$M_{ij}=B_{ij}+3A_{ij}+2\delta_{ij}-\frac{A_{N+2,\,j}}{A_{N+2,\,N+2}}(B_{i,\,N+2}+3A_{i,\,N+2})$$

$$f_i=4x_i$$

3. 邊界條件 (I - III)

原點側邊界條件爲第一類邊界條件，在 $x=1$ 位置之邊界條件爲第三類邊界條件：

$$\text{BC1} \quad y(0)=0\ ; \ x=0$$

$$\text{BC2} \quad \frac{dy}{dx}+y=0\ ; \ x=1 \tag{10-5.10}$$

BC2 可利用一次導函數操作矩陣表示爲

$$\sum_{i=1}^{N+2} A_{N+2,i}\, y_i + y_{N+2} = 0$$

整理之，並利用 BC1 $(y_1=0)$。可以得到

$$y_{N+2} = -\frac{1}{1+A_{N+2,N+2}} \sum_{i=2}^{N+1} A_{N+2,i}\, y_i$$

代入方程式 (10-5.4) 得到

$$\sum_{i=2}^{N+1} \left[B_{ji} + 3A_{ji} + 2\delta_{ji} - \frac{A_{N+2,i}(B_{j,N+2}+3A_{j,N+2})}{1+A_{N+2,N+2}} \right] y_i = 4x_j$$

或以矩陣符號表示爲

$$\underline{\underline{M}}\,\underline{y} = \underline{f} \tag{10-5.11}$$

其中

$$M_{ij} = B_{ij} + 3A_{ij} + 2\delta_{ij} - \frac{A_{N+2,j}(B_{j,N+2}+3A_{j,N+2})}{1+A_{N+2,N+2}}$$

$$f_i = 4x_i$$

4. 邊界條件 (II - I)

原點側邊界條件爲第二類邊界條件，在 $x=1$ 位置之邊界條件爲第一類邊界條件：

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

$$\text{BC2} \quad y=1\ ; \ x=1 \tag{10-5.12}$$

仿以上處理，得到

$$y_1 = -\frac{1}{A_{11}}(A_{1,N+2} + \sum_{i=2}^{N+1} A_{1i} y_i) \qquad y_{N+2} = 1$$

或

$$\underline{\underline{M}}\,\underline{y} = \underline{f} \tag{10-5.13}$$

$$M_{ij} = B_{ij} + 3A_{ij} + 2\delta_{ij} - \frac{A_{ij}(B_{i1} + 3A_{i1})}{A}$$

$$f_i = 4x_i - B_{i,N+2} - 3A_{i,N+2} + \frac{A_{1,N+2}(B_{i1} + 3A_{i1})}{A_{11}}$$

5. **邊界條件 (II - II)**

兩側邊界條件均為第二類邊界條件：

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

$$\text{BC2} \quad \frac{dy}{dx} = 0 \;；\; x = 1 \tag{10-5.14}$$

仿以上處理，得 y_1 及 y_{N+2} 分別為

$$y_1 = \sum_{i=2}^{N+1} \left[\frac{A_{N+2,N+2}A_{1i} - A_{1,N+2}A_{N+2,i}}{A_{N+2,1}A_{1,N+2} - A_{1i}A_{N+2,N+2}} \right] y_i$$

$$y_{N+2} = \sum_{i=2}^{N+1} \left[\frac{A_{11}A_{N+2,i} - A_{N+2,1}A_{1i}}{A_{N+2,1}A_{1,N+2} - A_{11}A_{N+2,N+2}} \right] y_i$$

或將上式中括號內的表示式以 $M_i^{(1)}$ 及 $M_i^{(N+2)}$ 表示，並寫成

$$y_1 = \sum_{i=2}^{N+1} M_i^{(1)} y_i$$

$$y_{N+2} = \sum_{i=2}^{N+1} M_i^{(N+2)} y_i$$

代入方程式 (10-5.4) 得到

$$\underline{\underline{M}}\,\underline{y} = \underline{f} \tag{10-5.15}$$

$$M_{ij} = B_{ij} + 3A_{ij} + 2\delta_{ij} + (B_{i1} + 3A_{i1})M_j^{(1)} + (B_{i,N+2} + 3A_{i,N+2})M_j^{(N+2)}$$

$$f_i = 4x_i$$

6. 邊界條件 (II - III)

原點側邊界條件為第二類邊界條件，在 $x=1$ 位置之邊界條件為第三類邊界條件：

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

$$\text{BC2}\quad \frac{dy}{dx}+y=0\ ;\ x=1 \qquad \textbf{(10-5.16)}$$

BC1 及 BC2 分別可用微分操作矩陣寫成：

$$\sum_{i=1}^{N+2} A_{1i}y_i=0$$

$$\sum_{i=1}^{N+2} A_{N+2,i}y_i+y_{N+2}=0$$

或整理成

$$A_{11}y_1+A_{1.\mathrm{N}+2}y_{N+2}=-\sum_{i=2}^{N+1}A_{1i}y_i$$

$$A_{N+2,1}y_1+(A_{N+2,N+2}+1)y_{N+2}=-\sum_{i=2}^{N+1}A_{N+2,i}y_i$$

解以上聯立方程式，得 y_1 及 y_{N+2} 分別為

$$y_1=\sum_{i=2}^{N+1}\left[\frac{A_{1,N+2}A_{N+2,i}-A_{1j}-A_{1j}A_{N+2,N+2}}{A_{11}+A_{11}A_{N+2,N+2}-A_{1,N+2}A_{N+2,1}}\right]y_i\equiv\sum_{i=2}^{N+1}M_i^{(1)}y_i$$

$$y_{N+2}=\sum_{i=2}^{N+1}\left[\frac{A_{N+2,1}A_{1i}-A_{11}A_{N+2,i}}{A_{11}+A_{11}A_{N+2,N+2}-A_{1,N+2}A_{N+2,1}}\right]y_i\equiv\sum_{i=2}^{N+1}M_i^{(N+2)}y_i$$

將上式代入方程式 (10-5.4) 得到

$$\underline{\underline{M}}\,\underline{y}=\underline{f} \qquad \textbf{(10-5.17)}$$

其中

$$M_{ij}=(B_{ij}+3A_{ij}+2\delta_{ij})+(B_{i1}+3A_{i1})M_j^{(1)}+(B_{i,N+2}+3A_{i,N+2})M_j^{(N+2)}$$

$$f_i=4x_i$$

7. 邊界條件 (III - III)

兩側邊界條件均為第三類邊界條件：

$$\text{BC1}\quad k_1\frac{\partial y}{\partial x}+h_1 y=\xi_1\ ;\ x=0$$

$$\text{BC2}\quad -k_2\frac{\partial y}{\partial x}+h_2 y=\xi_2\ ;\ x=1 \qquad \textbf{(10-5.18)}$$

利用微分操作矩陣 $\underline{A}$ ，即 $\frac{dy}{dx}(x_j)=\sum_{i=1}^{N+2}A_{ji}\,y(x_i)$ ，可將邊界條件改寫成

$$k_1\sum_{i=1}^{N+2}A_{1i}y_i+h_1y_1=\xi_1$$

$$-k_2\sum_{i=1}^{N+2}A_{N+2\,,\,i}y_i+h_2y_{N+2}=\xi_2$$

整理成 y_1 及 y_{N+2} 的聯立線性方程式為

$$(k_1A_{11}+h_1)y_1+(k_1A_{1\,,\,N+2})y_{N+2}=(\xi_1-k_1\sum_{i=1}^{N+1}A_{1i}y_i)$$

$$(-k_2A_{N+2\,,\,1})y_1+(-k_2A_{N+2\,,\,N+2}+h_2)y_{N+2}=(\xi_2+k_2\sum_{i=2}^{N+1}A_{N+2\,,\,i}y_i)$$

由以上二方程式，可聯立解得 y_1 及 y_{N+2} 分別為

$$\begin{aligned}y_1=&\left[(-k_2\xi_1A_{N+2\,,\,N+2}+h_2\xi_1-k_1\xi_2A_{1\,,\,N+2})\right.\\&\left.-\sum_{i=2}^{N+1}(-k_1k_2A_{N+2\,,\,N+2}A_{1i}+k_1h_2A_{1i}+k_1k_2A_{1\,,\,N+2}A_{N+2\,,\,i})y_i\right]/\Delta\end{aligned}$$

$$\begin{aligned}y_{N+2}=&\left[(k_1\xi_2A_{11}+h_1\xi_2+k_2\xi_1A_{N+2\,,\,1})\right.\\&\left.-\sum_{i=2}^{N+1}(-k_1k_2A_{11}A_{N+2\,,\,i}-k_2h_1A_{N+2\,,\,i}+k_1k_2A_{N+2\,,\,1}A_{1i})y_i\right]/\Delta\end{aligned}$$

$$\Delta=-k_1k_2A_{11}A_{N+2\,,\,N+2}+k_1h_2A_{11}-k_2h_1A_{N+2\,,\,N+2}+h_1h_2+k_1k_2A_{1\,,\,N+2}A_{N+2\,,\,1}$$

或以符號代表，簡寫成

$$y_1\equiv\sum_{i=2}^{N+1}M_i^{(1)}y_i+\alpha$$

$$y_{N+2}\equiv\sum_{i=2}^{N+1}M_i^{(N+2)}y_i+\beta$$

將上式代入方程式 (10-5.4)，可以得到

$$\underline{\underline{M}}\,\underline{y} = \underline{f} \qquad \textbf{(10-5.19)}$$

其中

$$M_{ij} = (B_{ij} + 3A_{ij} + 2\delta_{ij}) + (B_{i1} + 3A_{i1})M_j^{(1)} + (B_{i,N+2} + 3A_{i,N+2})M_j^{(N+2)}$$
$$f_i = 4x_i - (B_{i1} + 3A_{i1})\alpha - (B_{i,N+2} + 3A_{i,N+2})\beta$$

以上係根據常微分方程式 $\frac{d^2y}{dx^2} + 3\frac{dy}{dx} + 2y = 4x$ 為例，所做的演導與說明，讀者針對特定微分方程式求解時，可根據相同作法作適當處理。

第六節　雅可必多項式與操作矩陣

Visual Basic

正交配置法係採用正交多項式作為基礎函數，利用正交多項式的特性，簡化數學配置法的處理程序。常用的正交多項式包括李根德多項式 (Legendre Polynomial) 及本節所介紹的雅可必多項式 (Jacobi Polynomial) 等。以下針對雅可必多項式的特性及如何利用其特性撰寫程式，作簡要說明。

1. 雅可必多項式的基本定義

雅可必多項式是一種典型的正交多項式，其基本定義為

$$\int_0^1 x^\beta (1-x)^\alpha P_j(x) P_N(x)dx = 0\,; \quad j = 0, 1, 2, \cdots, N-1 \qquad \textbf{(10-6.1)}$$

或表示成

$$P_N^{(\alpha,\beta)}(x) = \sum_{i=0}^{N} (-1)^{N-i} \gamma_i x^i \qquad i = 1, 2, \cdots, N \qquad \textbf{(10-6.2)}$$

$$\gamma_i = \frac{N-i+1}{i} \cdot \frac{N+i+\alpha+\beta}{\alpha+\beta}\gamma_{i-1} \quad , \quad \gamma_0 = 1$$

有關雅可必多項式的詳細說明，請參考工程數學書籍。

2. 雅可必多項式的羅德芮格表示法

以羅德芮格方程式 (Rodrigues' Formula) 表示，雅可必多項式可以寫成

$$P_N^{(\alpha,\beta)}(1-x)^\alpha x^\beta = \frac{(-1)^N \Gamma(\beta+1)}{\Gamma(N+\beta+1)} \frac{d^N}{dx^N}[(1-x)^{N+\alpha} x^{N+\beta}] \qquad \textbf{(10-6.3)}$$

$$P_N^{(\alpha,\beta)}(x)=\sum_{i=0}^{N}(-1)^{N-i}\gamma_i x^i$$

$$\gamma_i=C_i^N\frac{\Gamma(N+i+\alpha+\beta+1)\Gamma(\beta+1)}{\Gamma(N+\alpha+\beta+1)\Gamma(i+\beta+1)}$$

3. 雅可必多項式的循序計算公式表示法

雅可必多項式的循序計算公式 (Recurrence formula) 為

$$p_N=[x-g_N(N,\alpha,\beta)]p_{N-1}-h_N(N,\alpha,\beta)p_{N-2}\;;\quad p_0=0 \qquad \textbf{(10-6.4)}$$

$$p_N=\frac{P_N}{\gamma_{NN}}\quad,\qquad \gamma_{NN}=N\text{ 次多項式的首項係數}\gamma_N$$

$$g_1=\frac{\beta+1}{\alpha+\beta+2}\;;\quad g_N=\frac{1}{2}[1-\frac{(\alpha+\beta)(\alpha-\beta)}{(2N+\alpha+\beta-1)^2-1}]\quad,\quad N>1$$

$$h_1=0\;;\quad h_2=\frac{\alpha\beta+\alpha+\beta+1}{(\alpha+\beta+2)^2(\alpha+\beta+3)}$$

$$h_N=\frac{(N-1)(N+\alpha-1)(N+\beta-1)(N+\alpha+\beta-1)}{(2N+\alpha+\beta-1)(2N+\alpha+\beta-2)^2(2N+\alpha+\beta-3)}\quad,\quad N>2$$

4. 常見的雅可必多項式

常見的雅可必多項式如下表所示：

α	β	N=0	N=1	N=2	N=3
0	0	1	$2x-1$	$6x^2-6x+1$	$20x^3-30x^2+12x-1$
1	0	1	$3x-1$	$10x^2-8x+1$	$35x^3-45x^2+15x-1$
2	0	1	$4x-1$	$15x^2-10x+1$	$56x^3-63x^2+18x-1$
0	1	1	$\frac{3}{2}x-1$	$\frac{25}{6}x^2-5x+1$	$\frac{35}{4}x^3-15x^2+\frac{15}{2}x-1$
1	1	1	$2x-1$	$6x^2-6x+1$	$14x^3-21x^2+9x-1$
2	1	1	$\frac{5}{2}x-1$	$\frac{49}{6}x^2-7x+1$	$21x^3-28x^2+\frac{21}{2}x-1$

例題 10-5　雅可必多項式之計算

試求 $P_{10}^{(0,0)}(0.98)$ 的值。

解：

$$g_N = 1/2$$

$$h_N = \frac{(N-1)^2}{4(2N-1)(2N-3)} \qquad N \ge 2$$

$$\gamma_{NN}^{(0,0)} = \frac{\Gamma(2N+1)}{\Gamma(N+1)\Gamma(N+1)} = \frac{20!}{10!10!} = 184756$$

$$P_{10}^{(0,0)}(0.98) = -0.2552433289 \quad (\text{正確解})$$
$$= -0.2552433288 \quad (\text{循環式計算結果})$$
$$= -0.255252 \quad (\text{多項式計算結果})$$

5. 雅可必多項式與配置點

由雅可必多項式的循序計算公式 (Recurrence formula) $p_N = [x - g_N(N, \alpha, \beta)]\, p_{N-1} - h_N(N, \alpha, \beta) p_{N-2}$, $p_0 = 0$ ；利用牛頓法求雅可必多項式的根作爲配置點 x_i 時，假設已經知道 $x_1, x_2, \cdots, x_k$ 共 k 個解，要求第 $k+1$ 個解時，可以利用輔助函數 $G_{N-k} = p_N(x) / \prod_{i=1}^{k}(x - x_i)$ ，將已知的解 $x_1, x_2, \cdots, x_k$ 由多項式中先行去除掉。再利用牛頓法解輔助函數 G_{N-K} ，得到迭代式爲

$$x_{k+1,i} = x_{k+1,i-1} - \delta(x) \tag{10-6.5}$$

其中

$$\delta(x) = \frac{G_{N-k}}{G_{N-k}^{(1)}} = \frac{p_N / p_N'}{1 - \frac{p_N}{p_N'} \sum_{i=1}^{k} \frac{1}{x - x_i}}$$

$$0 < x_1 < x_2 < \cdots\cdots < x_k < \cdots\cdots < 1$$

$$p_N' = [x - g_N] p_{N-1}' - h_N p_{N-2}' + p_{N-1}$$

執行時，令 $x_{k+1,0}$ 爲略大於 x_k 的數字，即 $x_{k+1,0} = x_k + \varepsilon$ ；其中 ε 爲一個很小的值，例如 10^{-4} 。雅可必多項式利用本節 [3] 的方程式計算，於計算求得 x_{k+1} 後，讓 k 增加 1，再重複以上步驟。利用這種方法所建立的副程式如下所示。

```
' =========================
' ROOTS OF JACOBI POLYNOMIAL
' =========================
'       Diff1 (I) = G (I)
'       Diff2 (I) = H (I)
```

```
'
AlphaPlusBeta = Alpha + Beta
AlphaMinusBeta = Beta - Alpha
AlphaBeta = Beta * Alpha
Diff1 (1) = (AlphaMinusBeta / (AlphaPlusBeta + 2) + 1) / 2
Diff2 (1) = 0

' Calculate G (I) & H (I)
If N >= 2 Then
        For I = 2 To N
                IM1 = I - 1
                Z = AlphaPlusBeta + 2 * IM1
                Diff1 (I) = (AlphaPlusBeta * AlphaMinusBeta / Z / (Z + 2) + 1) / 2
                If I = 2 Then
                        Diff2 (I) = (AlphaPlusBeta + AlphaBeta + IM1) / Z / Z / (Z + 1)
                Else
                        Z = Z * Z
                        Y = IM1 * (AlphaPlusBeta + IM1)
                        Y = Y * (AlphaBeta + Y)
                        Diff2 (I) = Y / Z / (Z - 1)
                End If
        Next I
End If

' Newton's Method to find ROOT
X = 0
For I = 1 To N
        Do
                XD = 0
                XN = 1
                XE = 0
                XM = 0
                For J = 1 To N
                        XP = (Diff1 (J) - X) * XN - Diff2 (J) * XD
                        XQ = (Diff1 (J) - X) * XM - Diff2 (J) * XE - XN
                        XD = XN
                        XE = XM
                        XN = XP
                        XM = XQ
                Next J
                ZC = 1
                Z = XN / XM
                If I > 1 Then
                        For J = 2 To I
                                ZC = ZC - Z / (X - Root (J - 1) )
                        Next J
                End If
                Z = Z / ZC
```

```
            X = X - Z
        Loop While (Abs (Z) > 0.000000001)
        Root (I) = X
        X = X + 0.0001
Next I

' To include ROOT at x= 0 and x=1
If (N1 = 1) Then
        For I = 1 To N
            J = N + 1 - I
            Root (J + 1) = Root (J)
        Next I
        Root (1) = 0
End If
If (N2 = 1) Then Root (ND) = 1

' Print ROOTs for the Polynomial
Print
Print "** COLLOCATION POINTS:"
Print
For I = 1 To ND
        Print Format (Root (I) , "     0.000000E+00     ")
Next I
```

6. 雅可必多項式的導函數

假設利用 [5] 中所介紹的牛頓法，可以得到雅可必多項式的 N+1 個根，x_j，$j=1,2,\cdots,N+1$，則可將雅可必多項式寫成

$$p_{N+1}(x)=\prod_{i=1}^{N+1}(x-x_i) \tag{10-6.6}$$

或改寫成循序計算式

$$\begin{aligned} &p_0(x)=1 \\ &p_j(x)=(x-x_j)\,p_{j-1}(x) \quad ; \quad j=1,2,\cdots,N+1 \end{aligned} \tag{10-6.7}$$

將上式微分，可以得到

$$p'_j(x)=(x-x_j)\,p'_{j-1}(x)+p_{j-1}(x) \tag{10-6.8}$$

$$p''_j(x)=(x-x_j)\,p''_{j-1}(x)+2p'_{j-1}(x) \tag{10-6.9}$$

$$p'''_j(x)=(x-x_j)\,p'''_{j-1}(x)+3p''_{j-1}(x) \tag{10-6.10}$$

由於 $p_0'(x) = p_0''(x) = p_0'''(x) = 0$，因此，利用以上方程式，可以很快的求得在各配置點位置上的導函數值。寫成副程式範例如下。

```
‘ Prepare Derivatives of the Polynomial
For I = 1 To ND
        X = Root (I)
        Diff1 (I) = 1
        Diff2 (I) = 0
        Diff3 (I) = 0
        For J = 1 To ND
                If J <> I Then
                        Y = X - Root (J)
                        Diff3 (I) = Y * Diff3 (I) + 3 * Diff2 (I)
                        Diff2 (I) = Y * Diff2 (I) + 2 * Diff1 (I)
                        Diff1 (I) = Y * Diff1 (I)
                End If
        Next J
Next I
```

7. 雅可必多項式與拉格蘭奇內插法

參考本書第三章數值內插法的說明，若已知 (x_i , y_i)；$i = 1, 2, \cdots, N+1$，則 $y_N(x)$ 可以利用拉格蘭奇內插法 (Lagrangian Interpolation) 寫成

$$y_N(x) = \sum_{i=1}^{N+1} y(x_i)\ell_i(x) \qquad , \qquad \ell_i(x) = \frac{p(x)}{(x - x_i)p'(x_i)} \tag{10-6.11}$$

利用這種方法所建立的內插法副程式如下所示。

```
' ========================
' LAGRANGIAN INTERPOLATION
' ========================
POL = 1
For I = 1 To ND
        YVA = X1 - Root (I)
        XP (I) = 0
        If YVA = 0 Then XP (I) = 1
        POL = POL * YVA
Next I
If POL <> 0 Then
        For I = 1 To ND
                XP (I) = POL / Diff1 (I) / (X1 - Root (I) )
        Next I
End If
```

8. 雅可必多項式與微分操作矩陣

根據拉格蘭奇內插方程式 (10-6.11)，取微分，可以得到 $y(x)$ 的一次及二次導函數分別為

$$\frac{dy}{dx}(x_j)=\sum_{i=1}^{N+1}\ell'_i(x_j)y(x_i)\equiv\sum_i A_{ji}y(x_i) \tag{10-6.12}$$

$$\frac{d^2y}{dx^2}(x_j)=\sum_{i=1}^{N+1}\ell''_i(x_j)y(x_i)\equiv\sum_i B_{ji}y(x_i) \tag{10-6.13}$$

其中

$$\ell'_i(x_i)=\frac{1}{2}\cdot\frac{p''_{N+1}(x_i)}{p'_{N+1}(x_i)} \qquad @\, x=x_i$$

$$\ell''_i(x_i)=\frac{1}{3}\cdot\frac{p'''_{N+1}(x_i)}{p'_{N+1}(x_i)} \qquad @\, x=x_i$$

$$\ell'_i(x_j)=\frac{1}{(x_j-x_i)}\cdot\frac{p''_{N+1}(x_j)}{p'_{N+1}(x_i)} \qquad @\, x=x_j\neq x_i$$

$$\ell''_i(x_j)=2\ell'_i(x_j)\left[\ell'_j(x_j)-\frac{1}{(x_j-x_i)}\right] \qquad @\, x=x_j\neq x_i$$

其中雅可必多項式的導函數 $p_{N+1}^{(m)}(x_i)$ 可以利用方程式 (10-6.8)、(10-6.9) 及 (10-6.10) 求之。利用這種方法所建立的副程式如下所示。

```
Sub DefMatrix (ID, ND, N1, N2, Diff1, Diff2, Diff3, Root, Vect, A, B)
' ==============================
' DERIVATIVE OPERATION MATRICES
' ==============================
For I = 1 To ND
        ID = 1
        Call OpMatrix (I, ID, ND, N1, N2, Diff1, Diff2, Diff3, Root, Vect)
        For J = 1 To ND
                A (I, J) = Vect (J)
        Next J
        ID = 2
        Call OpMatrix (I, ID, ND, N1, N2, Diff1, Diff2, Diff3, Root, Vect)
        For J = 1 To ND
                B (I, J) = Vect (J)
        Next J
Next I
End Sub
```

```
Sub OpMatrix (I, ID, ND, N1, N2, Diff1, Diff2, Diff3, Root, Vect)
' ================
' OPERATION MATRIX
' ================
' --ENTRY POINT
        For J = 1 To ND
                If J = Index Then
                        If ID = 1 Then
                                Vect (J) = Diff2 (Index) / Diff1 (Index) / 2
                        Else
                                Vect (J) = Diff3 (Index) / Diff1 (Index) / 3
                        End If
                Else
                        Y = Root (Index) - Root (J)
                        Vect (J) = Diff1 (Index) / Diff1 (J) / Y
                        If ID = 2 Then Vect (J) = Vect (J) * (Diff2 (Index) / Diff1 (Index) - 2 / Y)
                End If
        Next J
End Sub
```

9. 雅可必多項式與高斯數值積分

積分操作可以採用本書第六章所介紹之高斯積分法。

$$I \equiv \int_0^1 W(x)y(x)dx = \sum_{i=1}^{N} \omega_i y_i \quad \textbf{(10-6.14)}$$

其中，求出積分配重函數 ω_i ，即可計算積分結果。高斯雅可必配重函數爲

$$W(x) = x^{\beta}(1-x)^{\alpha} \quad \textbf{(10-6.15)}$$

$$\omega_i = \frac{(x_i)^{2\xi_1-1}(1-x_i)^{2\xi_2-1}}{[p'_{N+\xi_1+\xi_2}(x_i)]^2}(2N+\alpha+\beta+1)C_N^{(\alpha,\beta)} \quad \textbf{(10-6.16)}$$

$$i = [0], 1, 2, \cdots, N, [N+1]$$

若 $x=0$ 爲積分配置點，則 $\xi_1 = 1$ ，否則 $\xi_1 = 0$ 。

若 $x=1$ 爲積分配置點，則 $\xi_2 = 1$ ，否則 $\xi_2 = 0$ 。

10. 雷道與羅伯特數值積分

除了高斯積分法以外，雷道與羅伯特 (Radau and Lobatto) 積分法也是較常被使用的數值積分方法。其配重函數爲

$$[0,1] \qquad \omega_i = \frac{(2N+\alpha+\beta+2)C_N^{(\alpha+1,\beta)}}{x_i[p'_{N+1}(x_i)]^2}\cdot\begin{cases}1 & i\neq N+1\\ \dfrac{1}{1+\alpha} & i=N+1\end{cases}$$

$$[0,1) \qquad \omega_i = \frac{(2N+\alpha+\beta+2)C_N^{(\alpha,\beta+1)}}{(1-x_i)[p'_{N+1}(x_i)]^2}\cdot\begin{cases}\dfrac{1}{1+\beta} & i=0\\ 1 & i\neq 0\end{cases}$$

$$[0,1] \qquad \omega_i = \frac{(2N+\alpha+\beta+3)C_N^{(\alpha+1,\beta+1)}}{[p'_{N+2}(x_i)]^2}\cdot\begin{cases}\dfrac{1}{1+\beta} & i=0\\ 1 & i\neq 0, N+1\\ \dfrac{1}{1+\alpha} & i=N+1\end{cases}$$

(10-6.17)

其中第一及第二式稱爲雷道數值積分配重，第三式稱爲羅伯特數值積分配重。

第七節　線性問題的程式規劃

Visual Basic

根據以上諸節的介紹，我們可擬定出利用正交配置法解微分方程式的幾項步驟：

1. 紙上作業

- 由邊界條件續導出 y_1 及 y_{N+2} 的表示式（ 如本章第五節的說明 ）。
- 利用微分操作矩陣 $\underline{\underline{A}}$ 及 $\underline{\underline{B}}$，將微分方程式轉換成

$$\underline{\underline{M}}\,\underline{y} = \underline{f}$$

並寫下 $\underline{\underline{M}}$ 及 $\underline{f}$ 的表示方法。

2. 程式規劃

- 決定使用的 N 值（ 通常選用 4～10 間即可得到相當正確的解 ）。
- 決定使用的正交級數種類。
- 設計程式找出正交多項式 $P_N(x)$ 的根 x_j，$j=1,2,\cdots,N$ ，作爲配置點。
- 設計程式求出微分操作矩陣 $\underline{\underline{A}}$ 及 $\underline{\underline{B}}$ 。
- 建立 $\underline{\underline{M}}$ 及 $\underline{f}$ ，並將微分方程式改寫成聯立方程式。
- 利用高斯消去法解線性聯立方程式 $\underline{\underline{M}}\,\underline{y} = \underline{f}$ ，求得配置點上的函數值 $\underline{y}$ 。
- 利用內插法以適當 x 間距列印 y 值。

例題 10-6　圓柱狀觸媒的徑向質傳及反應

考慮圓柱狀觸媒粒子的徑向質傳滲透及等溫一次非可逆反應模式。

$$\frac{d^2y}{dx^2}+\frac{1}{x}\frac{dy}{dx}-\Phi^2 y=0 \qquad \textbf{(10-7.1)}$$

BC1　$x=0$ 時，$\frac{dy}{dx}=0$

BC2　$x=1$ 時，$y=1$

其中 $y=C/C_s$ 爲無因次濃度，$x=r/R$ 爲無因次半徑，Φ 稱爲希笠數。

試利用正交配置法求出 $\Phi=4$ 時，觸媒粒子內的穩定態濃度分布函數 $y(x)$，並求此觸媒粒子的有效度係數。

$$\eta=\int_0^1 y dx^2$$

解：

由 BC1 可知濃度分布對 $x=0$ 對稱，因此，可將座標轉換成 $\xi=x^2$，使處理過程簡化。

令 $\xi=x^2$，則原微分方程式可被轉換成

$$\xi\frac{d^2y}{d\xi^2}+\frac{dy}{d\xi}=\frac{\Phi^2}{4}y \qquad \textbf{(10-7.2)}$$

BC1　$\xi=0$ 時，$y=$ 有限値

BC2　$\xi=1$ 時，$y=1$

利用微分操作矩陣 $\underline{A}$ 及 $\underline{B}$，將上式寫成

$$\sum_{j=1}^{N+1}(\xi_1 B_{ij}+A_{ij}-\frac{\Phi^2}{4}\delta_{ij})y_j=0;\quad i=1,2,3,\cdots,N$$

$$y_{N+1}=1$$

TOP-DOWN 設計

- 副程式 Jacobi, DefMatrix, RadauLabotto 及 LagrangianInterpolation 的詳細說明，請見本節對應部分之說明。
- 副程式 GAUSS 請見本書第五章說明。

- 本程式可用於解一次元滲透—反應問題；幾何形狀包括平板、圓柱體及球形座標。

程式列印

```
'****************************************"
' SOLUTION OF LINEAR B.V.P.-O.D.E.
' BY    ORTHOGONAL    COLLOCATION
'****************************************"
'
' Program Developed by Dr. Ron Hsin Chang
' Copyright 1986, 2001
'
' Example 10-6
'
Sub OrthogonalCollocationBVPODE (Xpos, Ypos)
Dim AC (50, 50) , Z (50) , Coef (50) , XP (50) As Double
Dim Diff1 (50) , Diff2 (50) , Diff3 (50) , Root (50) As Double
Dim A (50, 50) , B (50, 50) , W (50) As Double
Dim Vect (50) As Double
'
'-- Enter Basic Data for Collocation Method
'
Call CollocationBasicData (Xpos, Ypos, GeoFact, Alpha, Beta, N, N1, N2, NoColloPoints, Diff1, Diff2,
Diff3, Root, Vect, A, B, W)
'
' ==DEFINE THE PROBLEM
'
Cls
THM = Val (InputBox ("> THIELE MODULUS = ?", "THIELE MODULUS", THM, Xpos, Ypos) )
For I = 1 To NoColloPoints
        For J = 1 To NoColloPoints
                AC (I, J) = 4 * (Root (I) * B (I, J) + (GeoFact + 1) / 2 * A (I, J) )
        Next J
        AC (I, I) = AC (I, I) - THM * THM
        Z (I) = 0
Next I
For J = 1 To NoColloPoints
        AC (NoColloPoints, J) = 0
Next J

AC (NoColloPoints, NoColloPoints) = 1
Z (NoColloPoints) = 1
NEQN% = NoColloPoints
'
' ELIMINATION to find the Solution
'
Call Gauss (NEQN%, AC, Z, Coef, Flag)
'
' Print Out the Results
'
Print
Print "** SOLUTION "
```

```
Print
For I = 1 To NEQN%
        Print Format (Coef (I) , "     0.000000E+00") ;
        If (Int (I / 5) - I / 5) = 0 Then Print
Next I
Print
'
' ==INTERPOLATION AT FIXED POINTS
'
Print
Print
Print "** SOLUTION AT FIXED GRIDPOINTS"
Print
For IT = 1 To 11
        X = (IT - 1) / 10
        X1 = X * X
        Call LagrangianInterpolation (NoColloPoints, Root, X1, XP, Diff1)
        YVA = 0
        For J = 1 To NoColloPoints
                YVA = YVA + XP (J) * Coef (J)
        Next J
        Print Format (X, "0.00E+00         ") ;
        Print Format (YVA, "       0.000000E+00")
Next IT
'
' Calculate the Effectiveness Factor
'
EFF = 0
For I = 1 To NoColloPoints
        EFF = EFF + W (I) * Coef (I)
Next I
Print ">>> EFFECTIVENESS FACTOR =";
Print Format (EFF, " 0.00000000E+00")
Print
Print "============END============"
End Sub


Sub CollocationBasicData (Xpos, Ypos, GeoFact, Alpha, Beta, N, N1, N2, NoColloPoints, Diff1, Diff2, Diff3, Root, Vect, A, B, W)
'
'       Basic Data for Collocation Method
'
'       GeoFact S      = 0-planar; 1-cylinder; 2-sphere"
'       Alpha                    = 1
'       Beta                     = (GeoFact - 1) /2
'       N                        = Number of Interior Collocation Points
'       N1                       = 1 if x=0 is a collocation point, otherwise N1=0
'       N2                       = 1 if x=1 is a collocation point, otherwise N2=0
'       NoColloPoints            = Total number of collocation points, = N + N1 + N2
'       Diff1                    = First order derivative vector
'       Diff2                    = Second order derivative vector
'       Diff3                    = Third order derivative vector
```

```
'      Root                  = Root of Jacobi Polynomial, Collocation Points
'      Vect                  = Operation vector
'      A                     = First order derivative matrix
'      B                     = Second order derivative matrix
'      W                     = Radau Lobatto quadrature weights
'
Cls
Call GeoFactor (Xpos, Ypos, GeoFact, Alpha, Beta, N, N1, N2, NoColloPoints)
'
' ==CALL JACOBI to find collocation points
'
Call Jacobi (Alpha, Beta, NoColloPoints, N, N1, N2, Diff1, Diff2, Diff3, Root)
MsgBox ("Ready to Continue")
'
' ==CALL DEFMAT to define operation matrix
'
Cls
Call DefMatrix (ID, NoColloPoints, N1, N2, Diff1, Diff2, Diff3, Root, Vect, A, B)
MsgBox ("Ready to Continue")
'
' ==RADAU AND LOBATTO QUADRATURE
'
Call RadauLobatto (Xpos, Ypos, N1, N2, NoColloPoints, Root, Diff1, Vect, W, Alpha, Beta)
MsgBox ("Ready to Continue")
End Sub


Sub GeoFactor (Xpos, Ypos, GeoFact, Alpha, Beta, N, N1, N2, ND)
'
'** DEFINE THE JACOBI POLYNOMIAL "
'      Generally, ALPHA=1, BETA= (S-1) /2"
'      GeoFact S = 0-planar; 1-cylinder; 2-sphere"
'
Print "** GEOMETRIC FACTOR **"
Print " S = 0 for Planar"
Print " S = 1 for Cylinder"
Print " S = 2 for Sphere"
Print

GeoFact = Val (InputBox ("    S          = ", "Geometric Factor", GeoFact, Xpos, Ypos) )
Print "Geometric Factor S = "; GeoFact

Alpha = Val (InputBox ("    Alpha = ", "Alpha", 1, Xpos, Ypos) )
Print "Alpha = "; Alpha

Beta = Val (InputBox ("    Beta = ", "BETA", (GeoFact - 1) / 2, Xpos, Ypos) )
Print "Beta = "; Beta

N = Val (InputBox (">> NUMBER OF COLLOCATION POINTS:", "Collocation Points", N, Xpos,
Ypos) )
Print "Number of Collocation Points N = "; N

Print
```

```
Print "      BOUNDARY COLLOCATION POINT (S) :"
Print "      0=not a collocation point"
Print "      1= is a collocation point"

N1 = Val (InputBox ("Collocation at X=0         (0/1) ? ", "Collocation @ X=0", N1, Xpos, Ypos) )
Print "N1 = "; N1

N2 = Val (InputBox ("Collocation at X=1         (0/1) ? ", "Collocation @ X=1", N2, Xpos, Ypos) )
Print "N2 = "; N2

ND = N + N1 + N2
End Sub


Sub RadauLobatto (Xpos, Ypos, N1, N2, NoColloPoints, Root, Diff1, Vect, W, AlphaP, BetaP)
'
' ==RADAU AND LOBATTO QUADRATURE
'
Cls
Print
Print ">> RADAU & LOBATTO QUADRATURE"
Print "ID=1 Radau quad. weights including x=1"
Print "ID=2 Radau quad. weights including x=0"
Print "ID=3 Lobatto quad wieghts including both"
Print "WEIGHT FUNCTION W (X) = (1-X) ^APLHA*X^BETA"
Print
If N1 * N2 = 1 Then
        ID = 3
        Alpha = AlphaP - 1
        Beta = BetaP - 1
ElseIf N1 = 1 Then
        ID = 2
        Alpha = AlphaP
        Beta = BetaP - 1
Else
        ID = 1
        Alpha = AlphaP - 1
        Beta = BetaP
End If
Print "ID = "; ID

Alpha = Val (InputBox ("    Alpha = ", "Alpha", Alpha, Xpos, Ypos) )
Print "Alpha = "; Alpha

Beta = Val (InputBox ("    Beta = ", "BETA", Beta, Xpos, Ypos) )
Print "Beta = "; Beta
Print

Call RadauQuadratureWeights (ID, NoColloPoints, Root, N1, N2, Diff1, Vect, Alpha, Beta)
Print "** VECTOR W:"
Print
For I = 1 To NoColloPoints
        W (I) = Vect (I)
```

```
        Print Format (W (I) , "       0.0000E+00") ;
Next I
End Sub


Sub Jacobi (Alpha, Beta, ND, N, N1, N2, Diff1, Diff2, Diff3, Root)
' =========================
' ROOTS OF JACOBI POLYNOMIAL
' =========================
AlphaPlusBeta = Alpha + Beta
AlphaMinusBeta = Beta - Alpha
AlphaBeta = Beta * Alpha

Diff1 (1) = (AlphaMinusBeta / (AlphaPlusBeta + 2) + 1) / 2
Diff2 (1) = 0

If N >= 2 Then
        For I = 2 To N
                IM1 = I - 1
                Z = AlphaPlusBeta + 2 * IM1
                Diff1 (I) = (AlphaPlusBeta * AlphaMinusBeta / Z / (Z + 2) + 1) / 2
                If I = 2 Then
                        Diff2 (I) = (AlphaPlusBeta + AlphaBeta + IM1) / Z / Z / (Z + 1)
                Else
                        Z = Z * Z
                        Y = IM1 * (AlphaPlusBeta + IM1)
                        Y = Y * (AlphaBeta + Y)
                        Diff2 (I) = Y / Z / (Z - 1)
                End If
        Next I
End If
'
X = 0
For I = 1 To N
        Do
                XD = 0
                XN = 1
                XE = 0
                XM = 0
                For J = 1 To N
                        XP = (Diff1 (J) - X) * XN - Diff2 (J) * XD
                        XQ = (Diff1 (J) - X) * XM - Diff2 (J) * XE - XN
                        XD = XN
                        XE = XM
                        XN = XP
                        XM = XQ
                Next J
                ZC = 1
                Z = XN / XM
                If I > 1 Then
                        For J = 2 To I
                                ZC = ZC - Z / (X - Root (J - 1) )
                        Next J
```

```
            End If
            Z = Z / ZC
            X = X - Z
        Loop While (Abs (Z) > 0.000000001)
        Root (I) = X
        X = X + 0.0001
    Next I

    If (N1 = 1) Then
        For I = 1 To N
            J = N + 1 - I
            Root (J + 1) = Root (J)
        Next I
        Root (1) = 0
    End If
    If (N2 = 1) Then Root (ND) = 1

    Print
    Print "** COLLOCATION POINTS:"
    Print
    For I = 1 To ND
        Print Format (Root (I) , "     0.000000E+00     ")
    Next I

    For I = 1 To ND
        X = Root (I)
        Diff1 (I) = 1
        Diff2 (I) = 0
        Diff3 (I) = 0
        For J = 1 To ND
            If J <> I Then
                Y = X - Root (J)
                Diff3 (I) = Y * Diff3 (I) + 3 * Diff2 (I)
                Diff2 (I) = Y * Diff2 (I) + 2 * Diff1 (I)
                Diff1 (I) = Y * Diff1 (I)
            End If
        Next J
    Next I
    End Sub


    Sub DefMatrix (ID, ND, N1, N2, Diff1, Diff2, Diff3, Root, Vect, A, B)
    ' ============================
    ' DERIVATIVE OPERATION MATRICES
    ' ============================
    For I = 1 To ND
        ID = 1
        Call OpMatrix (I, ID, ND, N1, N2, Diff1, Diff2, Diff3, Root, Vect)
        For J = 1 To ND
            A (I, J) = Vect (J)
        Next J
        ID = 2
        Call OpMatrix (I, ID, ND, N1, N2, Diff1, Diff2, Diff3, Root, Vect)
```

```
        For J = 1 To ND
            B (I, J) = Vect (J)
        Next J
    Next I

    Print
    Print "** MATRIX A:"
    Print
    For I = 1 To ND
        For J = 1 To ND
            Print Format (A (I, J) , "    0.00000E+00") ;
        Next J
        Print
    Next I
    Print

    Print "** MATRIX B:"
    Print
    For I = 1 To ND
        For J = 1 To ND
            Print Format (B (I, J) , "    0.00000E+00") ;
        Next J
        Print
    Next I
    End Sub


    Sub OpMatrix (Index, ID, ND, N1, N2, Diff1, Diff2, Diff3, Root, Vect)

    ' ================
    ' OPERATION MATRIX
    ' ================
    ' --ENTRY POINT
    If ID = 3 Then
        Call GaussQuadratureWeights (ND, Root, N1, N2, Diff1, Vect)
    Else
        For J = 1 To ND
            If J = Index Then
                If ID = 1 Then
                    Vect (J) = Diff2 (Index) / Diff1 (Index) / 2
                Else
                    Vect (J) = Diff3 (Index) / Diff1 (Index) / 3
                End If
            Else
                Y = Root (Index) - Root (J)
                Vect (J) = Diff1 (Index) / Diff1 (J) / Y
                If ID = 2 Then Vect (J) = Vect (J) * (Diff2 (Index) / Diff1 (Index) - 2 / Y)
            End If
        Next J
    End If
    End Sub
```

```
Sub GaussQuadratureWeights (ND, Root, N1, N2, Diff1, Vect)
' ===========================
' GAUSSIAN QUADRATURE WEIGHTS
' ===========================
Y = 0
For J = 1 To ND
        X = Root (J)
        AX = X * (1 - X)
        If N1 = 0 Then AX = AX / X / X
        If N2 = 0 Then AX = AX / (1 - X) / (1 - X)
        Vect (J) = AX / Diff1 (J) / Diff1 (J)
        Y = Y + Vect (J)
Next J
For J = 1 To ND
        Vect (J) = Vect (J) / Y
Next J
End Sub


Sub RadauQuadratureWeights (ID, ND, Root, N1, N2, Diff1, Vect, Alpha, Beta)
' ========================
' RADAU QUADRATURE WEIGHTS
' ========================
Sum = 0
For I = 1 To ND
        X = Root (I)
        Select Case ID
                Case 1
                        AX = X
                        If N1 = 0 Then AX = 1 / AX
                Case 2
                        AX = 1 - X
                        If N2 = 0 Then AX = 1 / AX
                Case 3
                        AX = 1
        End Select
        Vect (I) = AX / Diff1 (I) / Diff1 (I)
Next I

If ID <> 2 Then Vect (ND) = Vect (ND) / (1 + Alpha)
If ID > 1 Then Vect (1) = Vect (1) / (1 + Beta)

For I = 1 To ND
        Sum = Sum + Vect (I)
Next I

For I = 1 To ND
        Vect (I) = Vect (I) / Sum
Next I
End Sub


Sub LagrangianInterpolation (ND, Root, X1, XP, Diff1)
```

```
' ========================
' LAGRANGIAN INTERPOLATION
' ========================
POL = 1
For I = 1 To ND
        YVA = X1 - Root (I)
        XP (I) = 0
        If YVA = 0 Then XP (I) = 1
        POL = POL * YVA
Next I
If POL <> 0 Then
        For I = 1 To ND
                XP (I) = POL / Diff1 (I) / (X1 - Root (I) )
        Next I
End If
End Sub


Sub Gauss (N, A, Z, Coef, Flag)
'
'       ********************
'           SUBROUTINE GAUSS
'       ********************
'
Flag = 0

'-----START

For I = 1 To N - 1
        BIG = Abs (A (I, I) )
        L% = I: Rem L%= BIG ELEM.
        IP1 = I + 1

'-----CHECK FOR BIG

        For J = IP1 To N
                If (Abs (A (J, I) ) > BIG) Then
                        BIG = Abs (A (J, I) )
                        L% = J
                End If
        Next J
        If (BIG = 0!) Then
                Flag = 1
                Exit For
        End If

'-----PIVOT

        If (L% <> I) Then
                For J = 1 To N
                        HOLD = A (L%, J)
                        A (L%, J) = A (I, J)
                        A (I, J) = HOLD
```

```
            Next J
            HOLD = Z (L%)
            Z (L%) = Z (I)
            Z (I) = HOLD
        End If

'-----ELIMINATION

        For J = IP1 To N
            TEMP = A (J, I) / A (I, I)
            For K = IP1 To N
                A (J, K) = A (J, K) - TEMP * A (I, K)
            Next K
                Z (J) = Z (J) - TEMP * Z (I)
        Next J
Next I

'-----BACK SUBSTITUTION

If Flag = 0 Then
        If (A (N, N) = 0) Then
            Flag = 1
        Else
            Coef (N) = Z (N) / A (N, N)
            For I = N - 1 To 1 Step -1
                TEMP = 0
                For J = I + 1 To N
                    TEMP = TEMP + A (I, J) * Coef (J)
                Next J
                Coef (I) = (Z (I) - TEMP) / A (I, I)
            Next I
        End If
End If

'-----RETURN TO USER'S PROGRAM

If (Flag = 1) Then
        Print "ERROR: Matrix singular !!"
End If
End Sub
```

副程式使用方法說明

1. 副程式 **Sub OrthogonalCollocationBVPODE (Xpos, Ypos)**
   爲使用者必須針對所要處理的問題所自行編寫的程式部分。包括呼叫副程式 CollocationBasicData，建立操作矩陣方程式，呼叫線性方程式求解副程式 Gauss，利用副程式 RadauLobatto 求積分值，及利用副程式 LagrangianInterpolation 作線性

內插。

2. 副程式 **Sub CollocationBasicData (Xpos, Ypos, GeoFact, Alpha, Beta, N, N1, N2, NoColloPoints, Diff1, Diff2, Diff3, Root, Vect, A, B, W)**

用於宣告問題之參數。並主動呼叫以下副程式

GeoFactor (Xpos, Ypos, GeoFact, Alpha, Beta, N, N1, N2, NoColloPoints)
Jacobi (Alpha, Beta, NoColloPoints, N, N1, N2, Diff1, Diff2, Diff3, Root)
DefMatrix (ID, NoColloPoints, N1, N2, Diff1, Diff2, Diff3, Root, Vect, A, B)
RadauLobatto (Xpos, Ypo, N1, N2, NoColloPoints, Root, Diff1, Vect, W, Alpha, Beta)

Xpos 及 Ypos 爲對話方塊絕對位置。

3. 副程式 **Sub GeoFactor (Xpos, Ypos, GeoFact, Alpha, Beta, N, N1, N2, ND)**

用於宣告問題之參數。

呼叫後，傳回 GeoFact, Alpha, Beta, N, N1, N2,及配置點數 ND。

GeoFact = 0	平板
GeoFact = 1	圓柱體
GeoFact = 2	球體
N1=1	x=0 爲配置點
N2=1	x=1 爲配置點

4. 副程式 **Sub Jacobi (Alpha, Beta, ND, N, N1, N2, Diff1, Diff2, Diff3, Root)**

用於計算配置點。

使用時先給定 Alpha, Beta, N, N1, N2，及配置點數 ND。

呼叫副程式後，傳回配置點 Root(I) 及微分操作陣列 Diff1, Diff2 及 Diff3。

5. 副程式 **Sub RadauLobatto (Xpos, Ypos, N1, N2, NoColloPoints, Root, Diff1, Vect, W, Alpha, Beta)**

用於計算 Radau & Lobatto 積分配重函數。

使用時先給定 N1, N2，配置點數 NoColloPoints，配置點 Root (I) 及一階微分操作陣列 Diff1 (I)。Alpha 及 Beta 爲 Jacobi 副程式中之 Alpha 及 Beta。

呼叫副程式後，傳回配置點位置之積分配重函數 W(I)。

此程式中，計算 Radau & Lobatto 配重積分式所使用的 Alpha 及 Beta 於程式中修

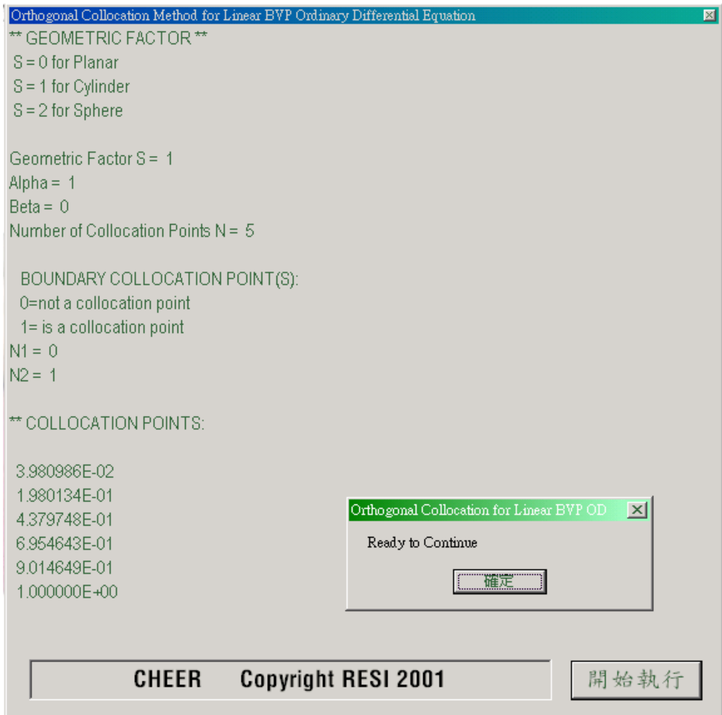
改之，會傳回由使用者確認。

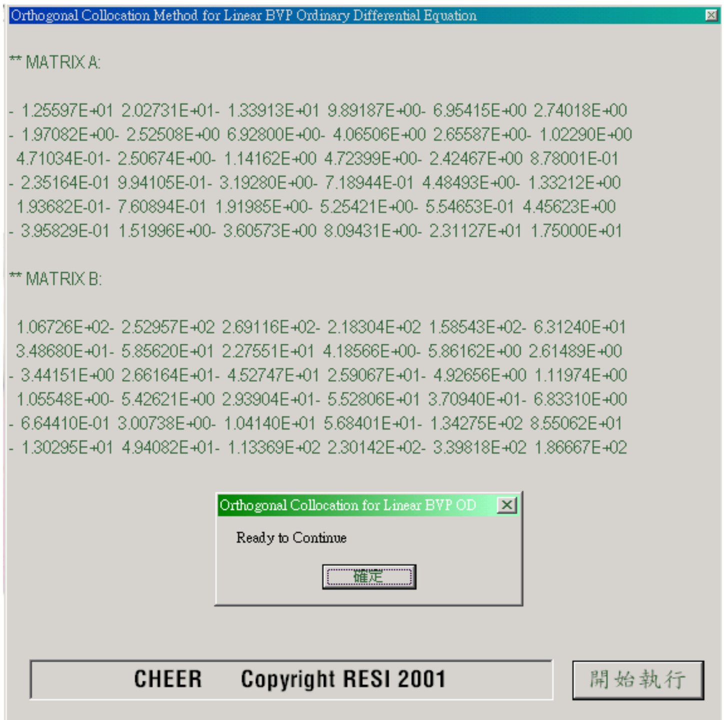
6. **副程式 Sub DefMatrix (ID, ND, N1, N2, Diff1, Diff2, Diff3, Root, Vect, A, B)**
   用於計算微分操作矩陣 A 及 B。
   使用時先給定 N1, N2，配置點數 ND，配置點 Root(I)，微分操作陣列 Diff1(I)，Diff2(I), Diff3(I) 及 Vect(I)。
   呼叫副程式後，傳回微分操作矩陣 A 及 B。

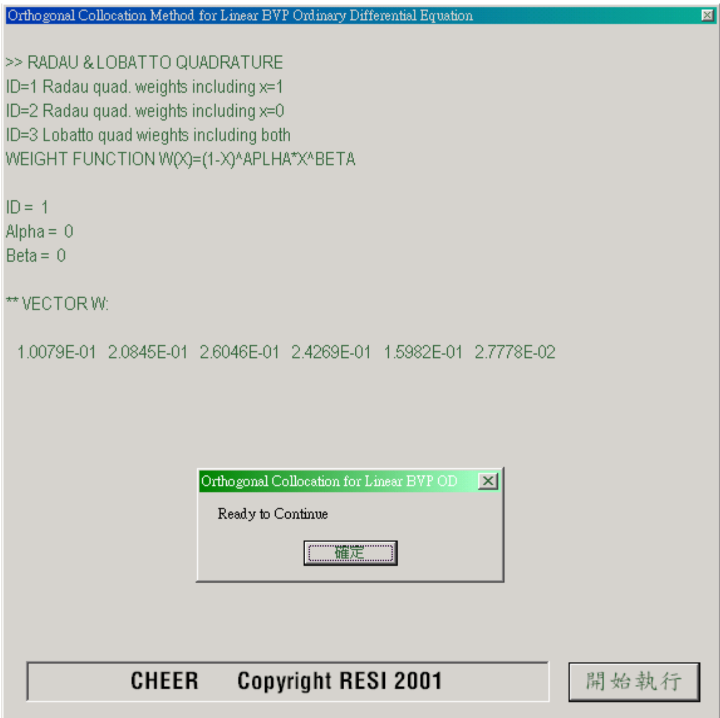
7. **副程式 Sub LagrangianInterpolation (ND, Root, X1, XP, Diff1)**
   用於計算內插值。詳本書第三章。

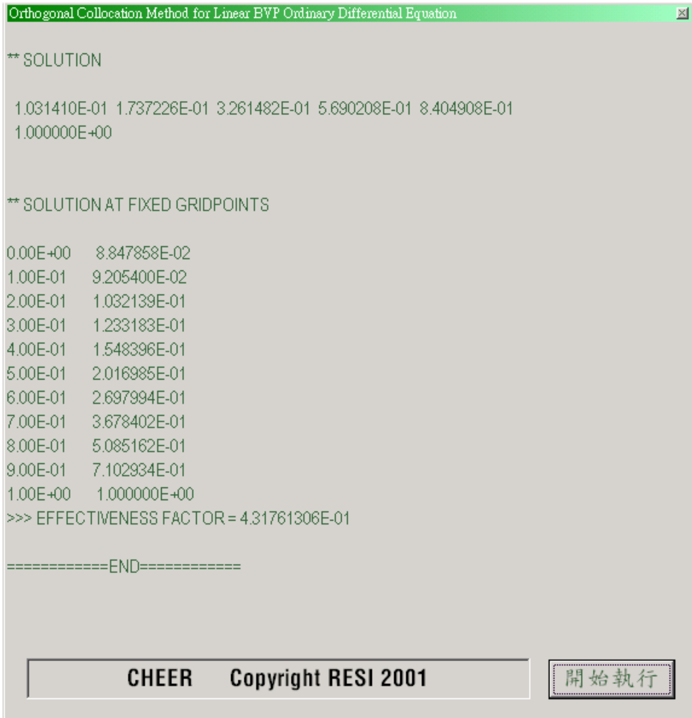
8. **副程式 Sub Gauss (N, A, Z, Coef, Flag)**
   用於解聯立線性方程式，詳本書第四章。
   使用時先給定方程式數目 N，係數矩陣 A，及常數陣列 Z。
   呼叫副程式後，傳回錯誤旗號 Flag 及解答 Coef(I)。

程式執行結果




Orthogonal Collocation Method for Linear BVP Ordinary Differential Equation
** MATRIX A:
- 1.25597E+01 2.02731E+01- 1.33913E+01 9.89187E+00- 6.95415E+00 2.74018E+00
- 1.97082E+00- 2.52508E+00 6.92800E+00- 4.06506E+00 2.65587E+00- 1.02290E+00
4.71034E-01- 2.50674E+00- 1.14162E+00 4.72399E+00- 2.42467E+00 8.78001E-01
- 2.35164E-01 9.94105E-01- 3.19280E+00- 7.18944E-01 4.48493E+00- 1.33212E+00
1.93682E-01- 7.60894E-01 1.91985E+00- 5.25421E+00- 5.54653E-01 4.45623E+00
- 3.95829E-01 1.51996E+00- 3.60573E+00 8.09431E+00- 2.31127E+01 1.75000E+01
** MATRIX B:
1.06726E+02- 2.52957E+02 2.69116E+02- 2.18304E+02 1.58543E+02- 6.31240E+01
3.48680E+01- 5.85620E+01 2.27551E+01 4.18566E+00- 5.86162E+00 2.61489E+00
- 3.44151E+00 2.66164E+01- 4.52747E+01 2.59067E+01- 4.92656E+00 1.11974E+00
1.05548E+00- 5.42621E+00 2.93904E+01- 5.52806E+01 3.70940E+01- 6.83310E+00
- 6.64410E-01 3.00738E+00- 1.04140E+01 5.68401E+01- 1.34275E+02 8.55062E+01
- 1.30295E+01 4.94082E+01- 1.13369E+02 2.30142E+02- 3.39818E+02 1.86667E+02
Orthogonal Collocation for Linear BVP OD
Ready to Continue
確定
CHEER Copyright RESI 2001
開始執行


Orthogonal Collocation Method for Linear BVP Ordinary Differential Equation
>> RADAU & LOBATTO QUADRATURE
ID=1 Radau quad. weights including x=1
ID=2 Radau quad. weights including x=0
ID=3 Lobatto quad wieghts including both
WEIGHT FUNCTION W(X)=(1-X)^APLHA*X^BETA
ID = 1
Alpha = 0
Beta = 0
** VECTOR W:
1.0079E-01 2.0845E-01 2.6046E-01 2.4269E-01 1.5982E-01 2.7778E-02
Orthogonal Collocation for Linear BVP OD
Ready to Continue
確定
CHEER Copyright RESI 2001
開始執行



結果討論

1. 不同座標系統的滲透—反應問題，只需改變 S 的輸入值，利用本程式即可直接求解。
2. 應注意在本問題中已作座標轉換，令 $\xi = x^2$；轉換後的方程式解，在轉回原方程式時，要先作 $\xi = x^2$ 處理。
3. 測試結果顯示 N 值選擇 3 以上，所得結果都相當一致。且由於計算簡單，利用單準及倍準計算所得結果並無明顯差別。
4. 若所考慮問題的統制方程式不是下列型式時，

$$\xi\frac{d^2y}{d\xi^2} + \frac{(1+s)}{2}\frac{dy}{d\xi} = \frac{\Phi^2}{4}y$$

請仿本例先演導建立 $\underline{\underline{M}}\,\underline{y} = \underline{f}$ 的方程式，並修正以上所附的程式中 Sub OrthogonalCollocationBVPODE 中的使用者定義程式區塊--DEFINE THE PROBLEM 部分即可。

5. 由於有效度係數 (Effectiveness factor) 的定義，在本問題中為

$$\eta = \int_0^1 y dx^2 = \int_0^1 y d\xi$$

即積分式配重函數 $W(x)=1$，故程式執行時令配重函數為 $W(x)=(1-x)^{\alpha}x^{\beta}$，其中的參數 α 及 β 均為 0。

第八節　非線性常微分方程式

Visual Basic

非線性常微分方程式基本上分成三大類：(1) 線性常微分方程式，邊界條件非線性；(2) 非線性常微分方程式，邊界條件為線性；(3) 非線性常微分方程式，非線性邊界條件。但仿以上各節處理後，這三類非線性常微分方程式均可變成一組聯立的非線性代數方程式，利用本書所介紹的牛頓拉福森法或割線法即可仿以上各節求得 $\underline{y}$。以下利用實例說明之。

例題 10-7　圓柱狀觸媒的徑向質傳及反應（設計問題 D-X）

考慮例 10-6 所討論的質傳滲透及等溫反應系統，但所進行反應為等溫二次非可逆反應，其數學模式為：

$$\frac{d^2y}{dx^2} + \frac{1}{x}\frac{dy}{dx} - \Phi^2 y^2 = 0 \tag{10-8.1}$$

BC1　$x=0$ 時，$\dfrac{dy}{dx}=0$

BC2　$x=1$ 時，$y=1$

試設計程式利用正交配置法求出 $\Phi=4$ 時，觸媒粒子內的穩定態濃度分布函數 $y(x)$，並求其有效度係數。

$$\eta = \int_0^1 y dx^2$$

解：

仿例 10-6，進行座標變換 $\xi = x^2$，將方程式 (10-8.1) 轉換成

$$\xi\frac{d^2y}{d\xi^2} + \frac{dy}{d\xi} = \frac{\Phi^2}{4} y^2 \tag{10-8.2}$$

BC1　$\xi=0$ 時，$y=$ 有限值

BC2　$\xi = 1$時，$y = 1$

利用微分操作矩陣 $\underline{\underline{A}}$ 及 $\underline{\underline{B}}$，將上式改寫成

$$\sum_{j=1}^{N+1} (\xi_1 B_{ij} + A_{ij}) y_j = \frac{\Phi^2}{4} \delta_{ij} y_j^{\,2}\,; \quad i = 1, 2, 3, \cdots, N \qquad \textbf{(10-8.3)}$$

$$y_{N+1} = 1$$

建立一組聯立非線性代數方程式。

TOP-DOWN 設計

- 副程式 Jacobi, DefMatrix, RadauLabotto 及 LagrangianInterpolation 的詳細說明，請見以上說明。
- 副程式 Sub Newton 是非線性聯立方程式的求解附程式，其說明詳見本書第五章。
- 函數副程式標準型式為

$$DZ(I) = \sum_{j=1}^{N} C(i\,,\,j) Z_j - f_i$$

即 $\underline{\underline{F}} \equiv \underline{\underline{M}}\, \underline{y} - \underline{f}$

- 割線法迭代所需的初值，本程式利用線性函數提供，為 $Z_i = i / N$ 。

程式列印

```
' ==========================================
' SOLUTION OF NONLINEAR B.V.P.-O.D.E
' BY ORTHOGONAL COLLOCATION
' ==========================================
'
' Program Developed by Dr. Ron Hsin Chang
' Copyright 1986, 2001
'
' Example 10-7
'
Sub OrthogonalCollocationBVPODE (Xpos, Ypos)
'
Dim AC (50, 50) , Z (50) , COEF (50) , XP (50) As Double
Dim Diff1 (50) , Diff2 (50) , Diff3 (50) , Root (50) As Double
Dim A (50, 50) , B (50, 50) , W (50) As Double
Dim Vect (50) , DZ (50) As Double
Cls
'
' ==Enter Basic Data
```

```
'
Call CollocationBasicData (Xpos, Ypos, GeoFact, Alpha, Beta, N, N1, N2, NoColloPoints, Diff1, Diff2,
Diff3, Root, Vect, A, B, W)
'
' ==DEFINE THE PROBLEM
'
Cls
Thm = Val (InputBox ("> THIELE MODULUS = ?", "THIELE MODULUS", Thm, Xpos, Ypos) )
For I = 1 To NoColloPoints - 1
        For J = 1 To NoColloPoints
                AC (I, J) = Root (I) * B (I, J) + (GeoFact + 1) / 2 * A (I, J)
        Next J
Next I
For J = 1 To NoColloPoints - 1
        AC (NoColloPoints, J) = 0
Next J
AC (NoColloPoints, NoColloPoints) = 1
NEQN% = NoColloPoints
ID = 0

'        -----------------------------
'            CALL NEWTON RAPHSON SECANT
'        -----------------------------
'


Call Newton (NEQN%, NoColloPoints, AC, Z, DZ, Thm, Itmax, ED, EZ, EX)

Print
Print "** SOLUTION "
Print
For I = 1 To NEQN%
        Print Format (Z (I) , "    0.00000E+00") ;
        If (Int (I / 5) - I / 5) = 0 Then Print
Next I
Print
'
' ==INTERPOLATION AT FIXED POINTS
'
Print
Print
Print "** SOLUTION AT FIXED GRIDPOINTS"
Print
Print "**X********Y***"
For IT = 1 To 11
        X = (IT - 1) / 10
        X1 = X * X
        Call LagrangianInterpolation (NoColloPoints, Root, X1, XP, Diff1)
        YVA = 0
        For J = 1 To NoColloPoints
                YVA = YVA + XP (J) * Z (J)
        Next J
        Print Format (X, " 0.##        ") ;
        Print Format (YVA, "      0.000000E+00")
Next IT

' ==EFFECTIVENESS FACTOR
EFF = 0
```

```
For I = 1 To NoColloPoints
        EFF = EFF + W (I) * Z (I) * Z (I)
Next I
Print ">>> EFFECTIVENESS FACTOR =";
Print Format (EFF, " 0.00000000E+00")
Print
Print "===========END==========="
End Sub


Sub DefineFunction (ND, AC, Z, DZ, Thm)
' ===================
' FUNCTION SUBROUTINE
' ===================

For IK = 1 To ND - 1
        DZ (IK) = 0
        For JK = 1 To ND
                DZ (IK) = DZ (IK) + AC (IK, JK) * Z (JK)
        Next JK
        DZ (IK) = DZ (IK) - Thm * Thm / 4 * Z (IK) * Z (IK)
Next IK

DZ (ND) = 0

For JK = 1 To ND
        DZ (ND) = AC (ND, JK) * Z (JK)
Next JK

DZ (ND) = DZ (ND) - 1
End Sub


Sub CollocationBasicData (Xpos, Ypos, GeoFact, Alpha, Beta, N, N1, N2, NoColloPoints, Diff1,
Diff2, Diff3, Root, Vect, A, B, W)
'
'       Basic Data for Collocation Method
'
'       GeoFact S                = 0-planar; 1-cylinder; 2-sphere"
'       Alpha                   = 1
'       Beta                    = (GeoFact - 1) /2
'       N                       = Number of Interior Collocation Points
'       N1                      = 1 if x=0 is a collocation point, otherwise N1=0
'       N2                      = 1 if x=1 is a collocation point, otherwise N2=0
'       NoColloPoints           = Total number of collocation points, = N + N1 + N2
'       Diff1                   = First order derivative vector
'       Diff2                   = Second order derivative vector
'       Diff3                   = Third order derivative vector
'       Root                    = Root of Jacobi Polynomial, Collocation Points
'       Vect                    = Operation vector
'       A                       = First order derivative matrix
'       B                       = Second order derivative matrix
'       W                       = Radau Lobatto quadrature weights
'
Cls
Call GeoFactor (Xpos, Ypos, GeoFact, Alpha, Beta, N, N1, N2, NoColloPoints)
'
```

```
' ==CALL JACOBI to find collocation points
'
Call Jacobi (Alpha, Beta, NoColloPoints, N, N1, N2, Diff1, Diff2, Diff3, Root)
MsgBox ("Ready to Continue")
'
' ==CALL DEFMAT to define operation matrix
'
Cls
Call DefMatrix (ID, NoColloPoints, N1, N2, Diff1, Diff2, Diff3, Root, Vect, A, B)
MsgBox ("Ready to Continue")
'
' ==RADAU AND LOBATTO QUADRATURE
'
Call RadauLobatto (Xpos, Ypos, N1, N2, NoColloPoints, Root, Diff1, Vect, W, Alpha, Beta)
MsgBox ("Ready to Continue")
End Sub


Sub GeoFactor (Xpos, Ypos, GeoFact, Alpha, Beta, N, N1, N2, ND)
'
'    BASIC DATA for Collocation Method
'
'    ** DEFINE THE JACOBI POLYNOMIAL "
'      Generally, ALPHA=1, BETA= (S-1) /2"
'      S=0-planar; 1-cylinder; 2-sphere"
'
Print "** GEOMETRIC FACTOR **"
Print " S = 0 for Planar"
Print " S = 1 for Cylinder"
Print " S = 2 for Sphere"
Print

GeoFact = Val (InputBox ("    S          = ", "Geometric Factor", GeoFact, Xpos, Ypos) )
Print " S selected = "; GeoFact

Alpha = Val (InputBox ("    Alpha = ", "Alpha", 1, Xpos, Ypos) )
Print " Alpha = "; Alpha

Beta = Val (InputBox ("    Beta = ", "BETA", (GeoFact - 1) / 2, Xpos, Ypos) )
Print " Beta =    "; Beta

N = Val (InputBox (">> NUMBER OF COLLOCATION POINTS:", "Collocation Points", N, Xpos,
Ypos) )
Print " Number of interior collocation Points N = "; N

Print "      BOUNDARY COLLOCATION POINT (S) :"
Print "      0=not a collocation point"
Print "      1= is a collocation point"

N1 = Val (InputBox ("Collocation at X=0         (0/1) ? ", "Collocation @ X=0", N1, Xpos, Ypos) )
Print "N1 = "; N1

N2 = Val (InputBox ("Collocation at X=1         (0/1) ? ", "Collocation @ X=1", N2, Xpos, Ypos) )
Print "N2 = "; N2

ND = N + N1 + N2
End Sub
```

```
Sub RadauLobatto (Xpos, Ypos, N1, N2, NoColloPoints, Root, Diff1, Vect, W, AlphaP, BetaP)
'
' ==RADAU AND LOBATTO QUADRATURE
'
Cls

Print
Print ">> RADAU & LOBATTO QUADRATURE"
Print "ID=1 Radau quad. weights including x=1"
Print "ID=2 Radau quad. weights including x=0"
Print "ID=3 Lobatto quad wieghts including both"
Print "WEIGHT FUNCTION W (X) = (1-X) ^APLHA*X^BETA"
Print

If N1 * N2 = 1 Then
        ID = 3
        Alpha = AlphaP - 1
        Beta = BetaP - 1
ElseIf N1 = 1 Then
        ID = 2
        Alpha = AlphaP
        Beta = BetaP - 1
Else
        ID = 1
        Alpha = AlphaP - 1
        Beta = BetaP
End If
Print "ID = "; ID

Alpha = Val (InputBox ("    Alpha = ", "Alpha", Alpha, Xpos, Ypos) )
Print "Alpha = "; Alpha

Beta = Val (InputBox ("    Beta = ", "BETA", Beta, Xpos, Ypos) )
Print "Beta = "; Beta
Print

Call RadauQuadratureWeights (ID, NoColloPoints, Root, N1, N2, Diff1, Vect, Alpha, Beta)
Print
Print "** VECTOR W:"
Print
For I = 1 To NoColloPoints
        W (I) = Vect (I)
        Print Format (W (I) , "      0.00000E+00") ;
Next I
End Sub


Sub Newton (NEQN%, NoColloPoints, AC, Z, DZ, Thm, Itmax, ED, EZ, EX)
Cls
Call InitialGuess (NoColloPoints, Z)
Call ErrorCriteria (Itmax, ED, EX, EZ)
Call NewtonRaphsonSecant (NEQN%, NoColloPoints, AC, Z, DZ, Thm, Itmax, ED, EZ, EX)
End Sub
```

```
Sub InitialGuess (ND, Z)
'
' ==INITIAL GUESSES
'
For I = 1 To ND
        Z (I) = (I / ND)
Next I
End Sub


Sub ErrorCriteria (Itmax, ED, EX, EZ)
'
' SET ERROR CRITERION
'
' ED=Value of determinant considered to be zero for checking whither the matrix is singular,
'          usually taken as 1E-30
' EX=Step length for partial derivative estimations, function evaluated at (1+-EX) *Z
' EZ=Acceptable relative error in Z
' ITMAX=Maximum number of iterations allowed
'
Itmax = 30
ED = 1E-32
EX = 0.00000001
EZ = 0.00000001
End Sub


Sub NewtonRaphsonSecant (N, ND, AC, Z, DZ, Thm, Itmax, ED, EZ, EX)
'
'      ----------------------------------
'          SUBROUTINE NEWTON RAPHSON SECANT
'      ----------------------------------
'
Dim AJ (50, 50) , BB (50) , Z2 (50) , MP (50) , MQ (50) As Double
Iter% = 0

Do
        Call Jacobian (N, ND, AC, AJ, BB, Z, DZ, EX, Thm)
        Call Trixee (N, AJ, BB, DZ, ED, DET, ConvergentFlag)

        If ConvergentFlag = 0 Then
                Print "** Matrix singular in nonlinear equations solver, DELTA = "; DET
                N = -N
                Exit Do
        End If

        Call DefineFunction (ND, AC, Z, DZ, Thm)

        For L1 = 1 To N
                Z2 (L1) = 0
                For KH = 1 To N
                        Z2 (L1) = Z2 (L1) + AJ (L1, KH) * DZ (KH)
                Next KH
                Z2 (L1) = Z (L1) - Z2 (L1)
        Next L1

        For KS = 1 To N
```

```
                If (Abs ( (Z2 (KS) - Z (KS) ) / Z (KS) ) - EZ) <= 0 Then
                        ConvergentFlag = ConvergentFlag
                Else
                        ConvergentFlag = ConvergentFlag + 1
                End If
        Next KS

        If ConvergentFlag = 1 Then
                Exit Do
        Else
                Iter% = Iter% + 1
                For I = 1 To N
                        Z (I) = Z2 (I)
                Next I
        End If
Loop While Iter% < Itmax

If ConvergentFlag > 1 Then
        Print "** Poor convergence in nonlinear system equation solver, ITER = "; Iter%
        N = -N
End If
End Sub


Sub Jacobian (N, ND, AC, AJ, BB, Z, DZ, EX, Thm)
'
'       -----------------------
'          SUBROUTINE JACOBIAN
'       -----------------------
'
For I = 1 To N
        XX = Z (I)
        Z (I) = XX * (1 - EX)
        X1 = Z (I)
        Call DefineFunction (ND, AC, Z, DZ, Thm)
        For J = 1 To N
                BB (J) = DZ (J)
        Next J
        Z (I) = XX * (1 + EX)
        X2 = Z (I)
        Call DefineFunction (ND, AC, Z, DZ, Thm)
        For J = 1 To N
                AJ (J, I) = (DZ (J) - BB (J) ) / (X2 - X1)
        Next J
        Z (I) = XX
Next I
End Sub


Sub Trixee (N, AJ, BB, DZ, ED, DET, KSG)
'
'       ---------------------
'          SUBROUTINE TRIXEE
'       ---------------------
'       SIMULTANEOUS EQNS. SOLVER
'
Dim MP (50) , MQ (50) As Double
```

```
DET = 1

'    -- DETERMINE PIVOTAL ELEMENT

For K = 1 To N
        PIV = 0
        For I = K To N
                For J = K To N
                        If (Abs (AJ (I, J) ) > Abs (PIV) ) Then
                                  PIV = AJ (I, J)
                                  MP (K) = I
                                  MQ (K) = J
                        End If
                Next J
        Next I
        DET = DET * PIV
        If (Abs (DET) <= ED) Then
                KSG = 0
                Exit For
        End If
        KSG = 1

'    -- TRANSPOSITION OF THE PIVOTAL ROW WITH THE KTH ROW

        If (MP (K) <> K) Then
                For J = 1 To N
                        K1 = MP (K)
                        Call SWAP (AJ (K1, J) , AJ (K, J) )
                Next J
        End If

'    -- TRANSPOSITION OF THE PIVOTAL COLUMN WITH THE KTH COLUMN

        If (MQ (K) <> K) Then
                For I = 1 To N
                        K2 = MQ (K)
                        Call SWAP (AJ (I, K2) , AJ (I, K) )
                Next I
        End If

'    -- JORDAN TRANSFORMATION

        For J = 1 To N
                If J = K Then
                        BB (J) = 1 / PIV
                        DZ (J) = 1
                Else
                        BB (J) = -AJ (K, J) / PIV
                        DZ (J) = AJ (J, K)
                End If
                AJ (K, J) = 0
                AJ (J, K) = 0
        Next J
        For I = 1 To N
                For J = 1 To N
                        AJ (I, J) = AJ (I, J) + DZ (I) * BB (J)
                Next J
```

```
        Next I
Next K

'    REARRANGEMENT OF THE MATRIX

If KSG = 1 Then
        For K = 1 To N
                K3 = N - K + 1
                If (MP (K3) <> K3) Then
                        DET = -DET
                        For I = 1 To N
                                K4 = MP (K3)
                                Call SWAP (AJ (I, K4) , AJ (I, K3) )
                        Next I
                End If
                If MQ (K3) <> K3 Then
                        DET = -DET
                        For J = 1 To N
                                K5 = MQ (K3)
                                Call SWAP (AJ (K5, J) , AJ (K3, J) )
                        Next J
                End If
        Next K
End If
End Sub


Sub Jacobi (Alpha, Beta, ND, N, N1, N2, Diff1, Diff2, Diff3, Root)
' ==========================
' ROOTS OF JACOBI POLYNOMIAL
' ==========================

AlphaPlusBeta = Alpha + Beta
AlphaMinusBeta = Beta - Alpha
AlphaBeta = Beta * Alpha

Diff1 (1) = (AlphaMinusBeta / (AlphaPlusBeta + 2) + 1) / 2
Diff2 (1) = 0

If N >= 2 Then
        For I = 2 To N
                IM1 = I - 1
                Z = AlphaPlusBeta + 2 * IM1
                Diff1 (I) = (AlphaPlusBeta * AlphaMinusBeta / Z / (Z + 2) + 1) / 2
                If I = 2 Then
                        Diff2 (I) = (AlphaPlusBeta + AlphaBeta + IM1) / Z / Z / (Z + 1)
                Else
                        Z = Z * Z
                        Y = IM1 * (AlphaPlusBeta + IM1)
                        Y = Y * (AlphaBeta + Y)
                        Diff2 (I) = Y / Z / (Z - 1)
                End If
        Next I
End If
'
X = 0
For I = 1 To N
```

```
        Do
            XD = 0
            XN = 1
            XE = 0
            XM = 0
            For J = 1 To N
                XP = (Diff1 (J) - X) * XN - Diff2 (J) * XD
                XQ = (Diff1 (J) - X) * XM - Diff2 (J) * XE - XN
                XD = XN
                XE = XM
                XN = XP
                XM = XQ
            Next J
            ZC = 1
            Z = XN / XM
            If I > 1 Then
                For J = 2 To I
                    ZC = ZC - Z / (X - Root (J - 1) )
                Next J
            End If
            Z = Z / ZC
            X = X - Z
        Loop While (Abs (Z) > 0.000000001)
        Root (I) = X
        X = X + 0.0001
    Next I

    If (N1 = 1) Then
        For I = 1 To N
            J = N + 1 - I
            Root (J + 1) = Root (J)
        Next I
        Root (1) = 0
    End If
    If (N2 = 1) Then Root (ND) = 1

    Print
    Print "** COLLOCATION POINTS:"
    Print
    For I = 1 To ND
        Print Format (Root (I) , "    0.000000E+00")
    Next I

    For I = 1 To ND
        X = Root (I)
        Diff1 (I) = 1
        Diff2 (I) = 0
        Diff3 (I) = 0
        For J = 1 To ND
            If J <> I Then
                Y = X - Root (J)
                Diff3 (I) = Y * Diff3 (I) + 3 * Diff2 (I)
                Diff2 (I) = Y * Diff2 (I) + 2 * Diff1 (I)
                Diff1 (I) = Y * Diff1 (I)
            End If
        Next J
    Next I
```

```
End Sub


Sub DefMatrix (ID, ND, N1, N2, Diff1, Diff2, Diff3, Root, Vect, A, B)
' ==============================
' DERIVATIVE OPERATION MATRICES
' ==============================

For I = 1 To ND
        ID = 1
        Call OpMatrix (I, ID, ND, N1, N2, Diff1, Diff2, Diff3, Root, Vect)
        For J = 1 To ND
                A (I, J) = Vect (J)
        Next J
        ID = 2
        Call OpMatrix (I, ID, ND, N1, N2, Diff1, Diff2, Diff3, Root, Vect)
        For J = 1 To ND
                B (I, J) = Vect (J)
        Next J
Next I

Print
Print "** MATRIX A:"
Print
For I = 1 To ND
        For J = 1 To ND
                Print Format (A (I, J) , "    0.00000E+00") ;
        Next J
        Print
Next I
Print
Print "** MATRIX B:"
Print
For I = 1 To ND
        For J = 1 To ND
                Print Format (B (I, J) , "    0.00000E+00") ;
        Next J
        Print
Next I
End Sub


Sub OpMatrix (I, ID, ND, N1, N2, Diff1, Diff2, Diff3, Root, Vect)
' ================
' OPERATION MATRIX
' ================

' --ENTRY POINT
If ID = 3 Then
        Call GaussQuadratureWeights (ND, Root, N1, N2, Diff1, Vect)
Else
        For J = 1 To ND
                If J = I Then
                        If ID = 1 Then
                                Vect (J) = Diff2 (I) / Diff1 (I) / 2
                        Else
                                Vect (J) = Diff3 (I) / Diff1 (I) / 3
```

```
                End If
            Else
                Y = Root (I) - Root (J)
                Vect (J) = Diff1 (I) / Diff1 (J) / Y
                If ID = 2 Then Vect (J) = Vect (J) * (Diff2 (I) / Diff1 (I) - 2 / Y)
            End If
        Next J
End If
End Sub


Sub GaussQuadratureWeights (ND, Root, N1, N2, Diff1, Vect)
' ===========================
' GAUSSIAN QUADRATURE WEIGHTS
' ===========================
Y = 0
For J = 1 To ND
        X = Root (J)
        AX = X * (1 - X)
        If N1 = 0 Then AX = AX / X / X
        If N2 = 0 Then AX = AX / (1 - X) / (1 - X)
        Vect (J) = AX / Diff1 (J) / Diff1 (J)
        Y = Y + Vect (J)
Next J
For J = 1 To ND
        Vect (J) = Vect (J) / Y
Next J
End Sub


Sub RadauQuadratureWeights (ID, ND, Root, N1, N2, Diff1, Vect, Alpha, Beta)
' ========================
' RADAU QUADRATURE WEIGHTS
' ========================
Sum = 0
For I = 1 To ND
        X = Root (I)
        Select Case ID
            Case 1
                AX = X
                If N1 = 0 Then AX = 1 / AX
            Case 2
                AX = 1 - X
                If N2 = 0 Then AX = 1 / AX
            Case 3
                AX = 1
        End Select
        Vect (I) = AX / Diff1 (I) / Diff1 (I)
Next I
If ID <> 2 Then Vect (ND) = Vect (ND) / (1 + Alpha)
If ID > 1 Then Vect (1) = Vect (1) / (1 + Beta)
For I = 1 To ND
        Sum = Sum + Vect (I)
Next I
For I = 1 To ND
        Vect (I) = Vect (I) / Sum
Next I
```

```
End Sub


Sub LagrangianInterpolation (ND, Root, X1, XP, Diff1)
' ========================
' LAGRANGIAN INTERPOLATION
' ========================
POL = 1
For I = 1 To ND
        YVA = X1 - Root (I)
        XP (I) = 0
        If YVA = 0 Then XP (I) = 1
        POL = POL * YVA
Next I
If POL <> 0 Then
        For I = 1 To ND
                XP (I) = POL / Diff1 (I) / (X1 - Root (I) )
        Next I
End If
End Sub


Sub SWAP (PAR1, PAR2)
Temp = PAR1
PAR1 = PAR2
PAR2 = Temp
End Sub
```

副程式使用方法說明

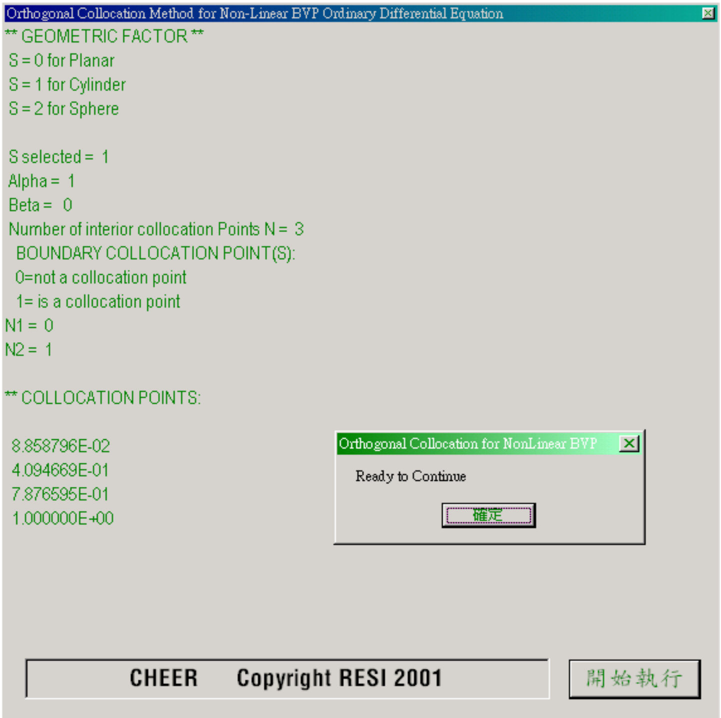
1. 副程式 **Sub OrthogonalCollocationBVPODE (Xpos, Ypos)**
   為使用者必須針對所要處理的問題所自行編寫的程式部分。包括呼叫副程式 CollocationBasicData，建立操作矩陣方程式，呼叫非線性方程式求解副程式 Newton，利用副程式 RadauLobatto 求積分值，及利用副程式 LagrangianInterpolation 作線性內插。

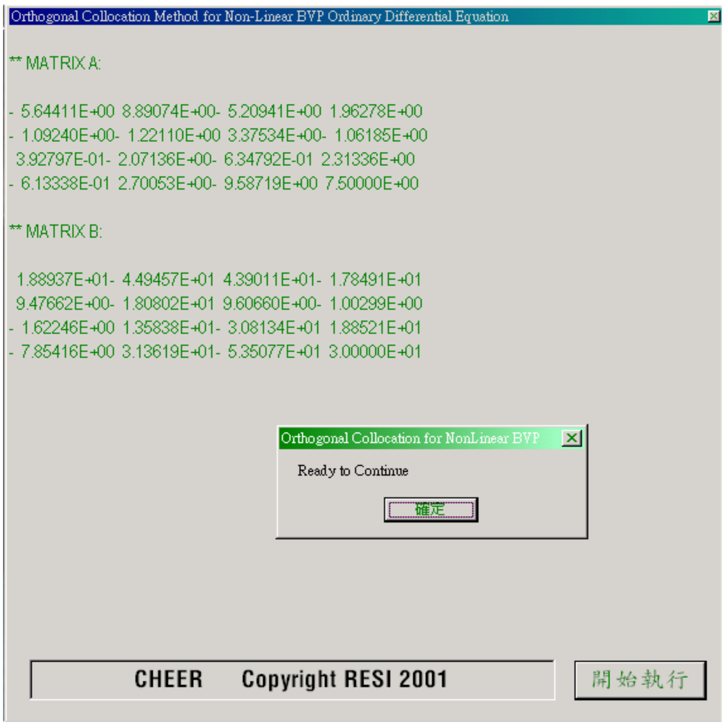
2. 副程式 **Sub DefineFunction (ND, AC, Z, DZ, Thm)**
   為使用者必須針對所要處理的問題所自行編寫的函數定義部分。

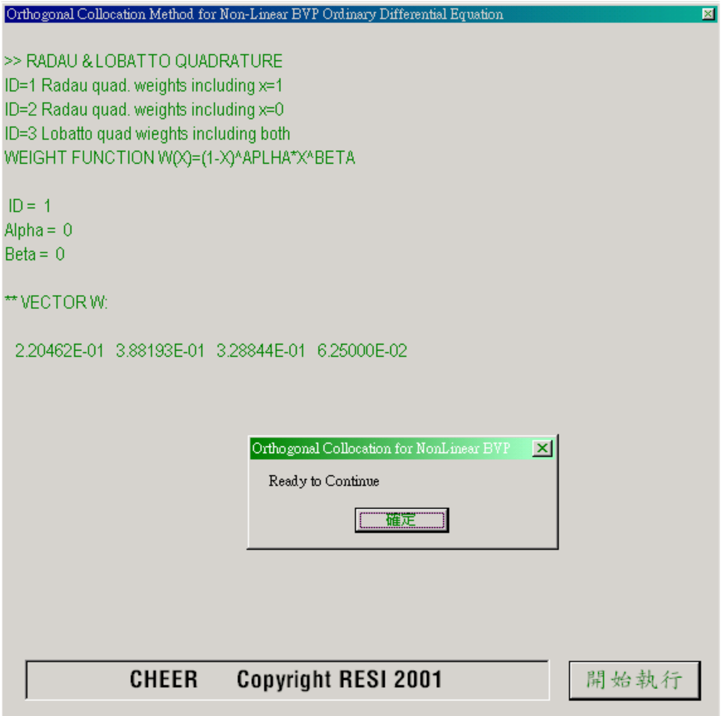
3. 副程式 **Sub Newton (NEQN%, NoColloPoints, AC, Z, DZ, Thm, Itmax, ED, EZ, EX)**
   利用牛頓拉福森割線法解非線性聯立方程式。詳見本書第五章說明。

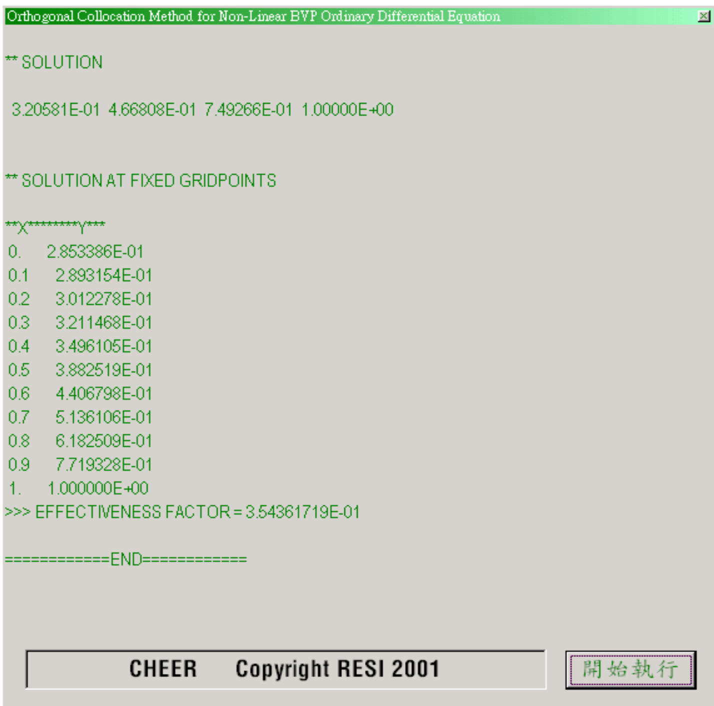
程式執行結果

內部配置點數 **N=3**

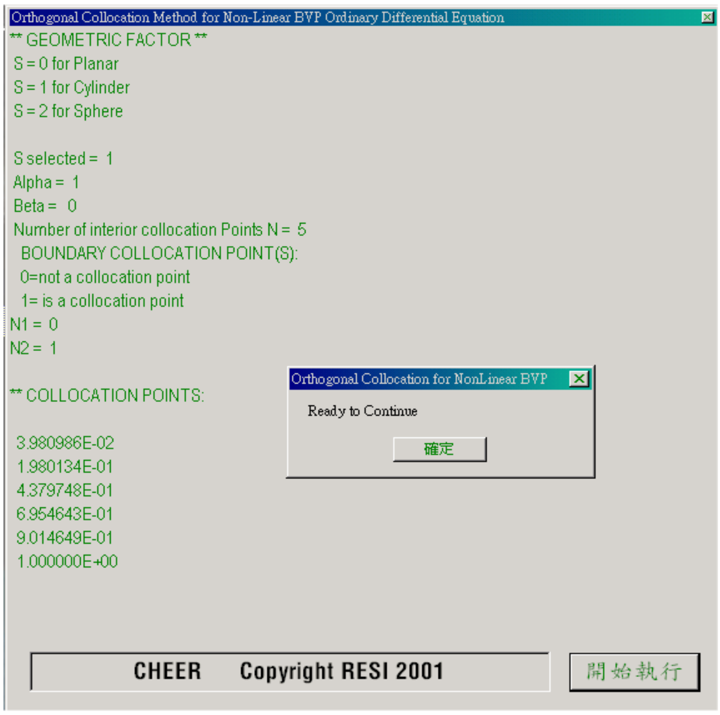

Orthogonal Collocation Method for Non-Linear BVP Ordinary Differential Equation
** GEOMETRIC FACTOR **
S = 0 for Planar
S = 1 for Cylinder
S = 2 for Sphere
S selected = 1
Alpha = 1
Beta = 0
Number of interior collocation Points N = 3
BOUNDARY COLLOCATION POINT(S):
0=not a collocation point
1= is a collocation point
N1 = 0
N2 = 1
** COLLOCATION POINTS:
8.858796E-02
4.094669E-01
7.876595E-01
1.000000E+00
Orthogonal Collocation for NonLinear BVP
Ready to Continue
確定
CHEER Copyright RESI 2001
開始執行

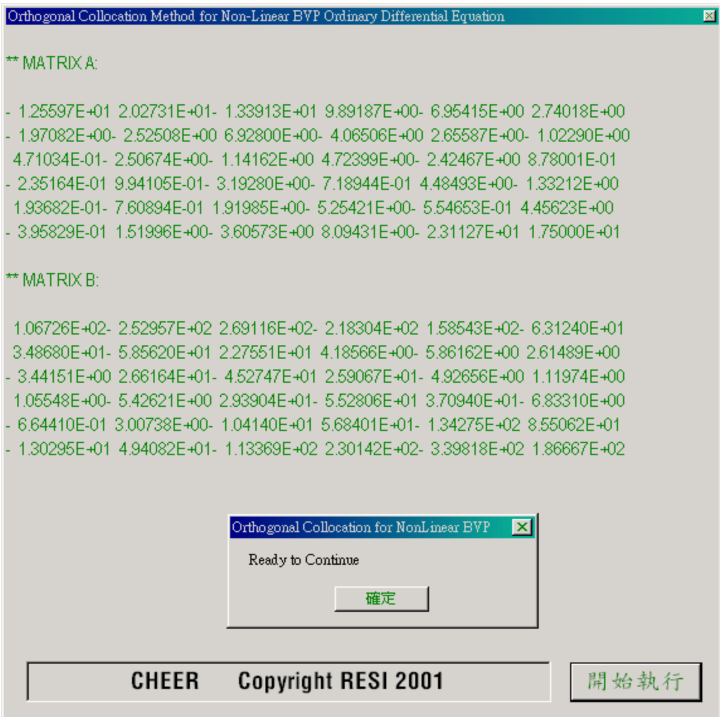

Orthogonal Collocation Method for Non-Linear BVP Ordinary Differential Equation
** MATRIX A:
- 5.64411E+00 8.89074E+00- 5.20941E+00 1.96278E+00
- 1.09240E+00- 1.22110E+00 3.37534E+00- 1.06185E+00
3.92797E-01- 2.07136E+00- 6.34792E-01 2.31336E+00
- 6.13338E-01 2.70053E+00- 9.58719E+00 7.50000E+00
** MATRIX B:
1.88937E+01- 4.49457E+01 4.39011E+01- 1.78491E+01
9.47662E+00- 1.80802E+01 9.60660E+00- 1.00299E+00
- 1.62246E+00 1.35838E+01- 3.08134E+01 1.88521E+01
- 7.85416E+00 3.13619E+01- 5.35077E+01 3.00000E+01
Orthogonal Collocation for NonLinear BVP
Ready to Continue
確定
CHEER Copyright RESI 2001
開始執行

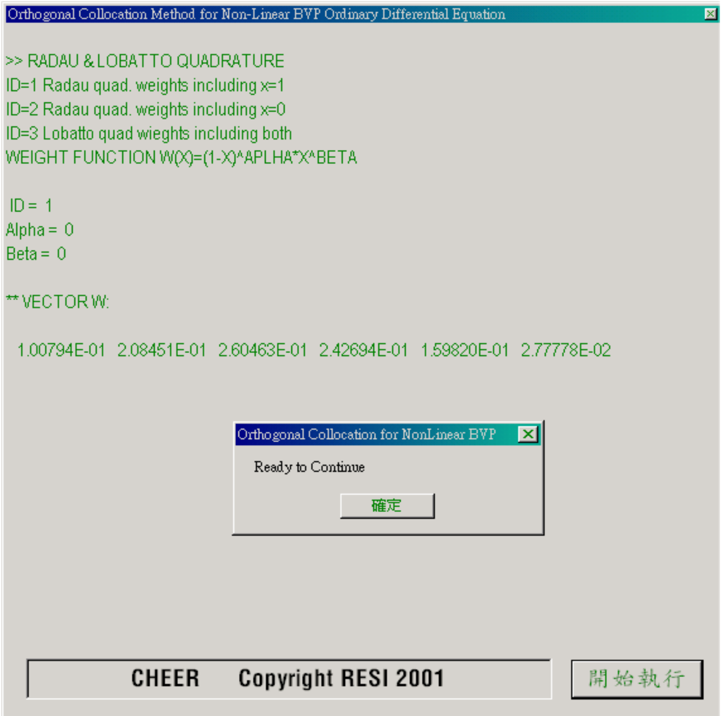

Orthogonal Collocation Method for Non-Linear BVP Ordinary Differential Equation
>> RADAU & LOBATTO QUADRATURE
ID=1 Radau quad. weights including x=1
ID=2 Radau quad. weights including x=0
ID=3 Lobatto quad wieghts including both
WEIGHT FUNCTION W(X)=(1-X)^APLHA*X^BETA
ID = 1
Alpha = 0
Beta = 0
** VECTOR W:
2.20462E-01 3.88193E-01 3.28844E-01 6.25000E-02
Orthogonal Collocation for NonLinear BVP
Ready to Continue
確定
CHEER Copyright RESI 2001
開始執行

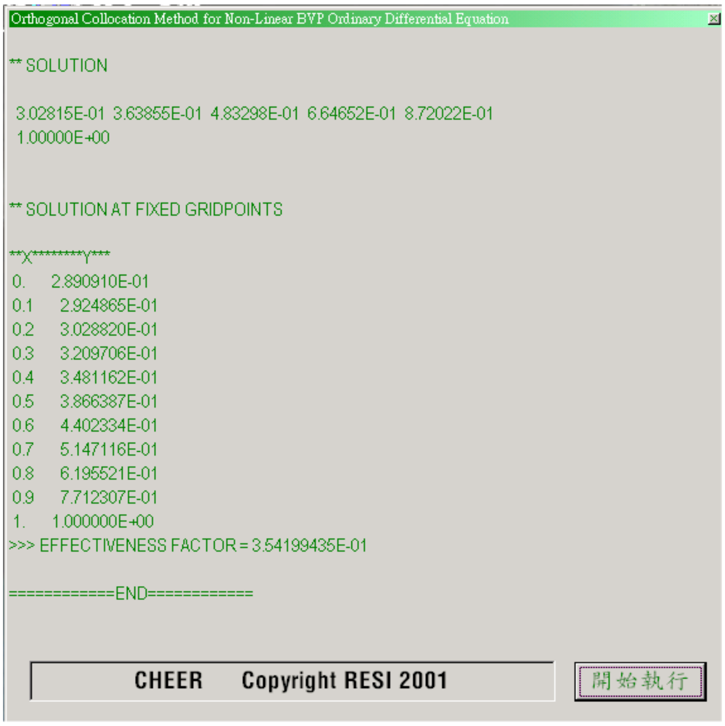

Orthogonal Collocation Method for Non-Linear BVP Ordinary Differential Equation
** SOLUTION
3.20581E-01 4.66808E-01 7.49266E-01 1.00000E+00
** SOLUTION AT FIXED GRIDPOINTS
X*******Y***
0. 2.853386E-01
0.1 2.893154E-01
0.2 3.012278E-01
0.3 3.211468E-01
0.4 3.496105E-01
0.5 3.882519E-01
0.6 4.406798E-01
0.7 5.136106E-01
0.8 6.182509E-01
0.9 7.719328E-01
1. 1.000000E+00
>>> EFFECTIVENESS FACTOR = 3.54361719E-01
============END============
CHEER Copyright RESI 2001
開始執行

內部配置點數 **N=5**


Orthogonal Collocation Method for Non-Linear BVP Ordinary Differential Equation
** GEOMETRIC FACTOR **
S = 0 for Planar
S = 1 for Cylinder
S = 2 for Sphere
S selected = 1
Alpha = 1
Beta = 0
Number of interior collocation Points N = 5
BOUNDARY COLLOCATION POINT(S):
0=not a collocation point
1= is a collocation point
N1 = 0
N2 = 1
** COLLOCATION POINTS:
3.980986E-02
1.980134E-01
4.379748E-01
6.954643E-01
9.014649E-01
1.000000E+00
Orthogonal Collocation for NonLinear BVP
Ready to Continue
確定
CHEER Copyright RESI 2001
開始執行


Orthogonal Collocation Method for Non-Linear BVP Ordinary Differential Equation
** MATRIX A:
- 1.25597E+01 2.02731E+01- 1.33913E+01 9.89187E+00- 6.95415E+00 2.74018E+00
- 1.97082E+00- 2.52508E+00 6.92800E+00- 4.06506E+00 2.65587E+00- 1.02290E+00
4.71034E-01- 2.50674E+00- 1.14162E+00 4.72399E+00- 2.42467E+00 8.78001E-01
- 2.35164E-01 9.94105E-01- 3.19280E+00- 7.18944E-01 4.48493E+00- 1.33212E+00
1.93682E-01- 7.60894E-01 1.91985E+00- 5.25421E+00- 5.54653E-01 4.45623E+00
- 3.95829E-01 1.51996E+00- 3.60573E+00 8.09431E+00- 2.31127E+01 1.75000E+01
** MATRIX B:
1.06726E+02- 2.52957E+02 2.69116E+02- 2.18304E+02 1.58543E+02- 6.31240E+01
3.48680E+01- 5.85620E+01 2.27551E+01 4.18566E+00- 5.86162E+00 2.61489E+00
- 3.44151E+00 2.66164E+01- 4.52747E+01 2.59067E+01- 4.92656E+00 1.11974E+00
1.05548E+00- 5.42621E+00 2.93904E+01- 5.52806E+01 3.70940E+01- 6.83310E+00
- 6.64410E-01 3.00738E+00- 1.04140E+01 5.68401E+01- 1.34275E+02 8.55062E+01
- 1.30295E+01 4.94082E+01- 1.13369E+02 2.30142E+02- 3.39818E+02 1.86667E+02
Orthogonal Collocation for NonLinear BVP
Ready to Continue
確定
CHEER Copyright RESI 2001
開始執行


Orthogonal Collocation Method for Non-Linear BVP Ordinary Differential Equation
>> RADAU & LOBATTO QUADRATURE
ID=1 Radau quad. weights including x=1
ID=2 Radau quad. weights including x=0
ID=3 Lobatto quad wieghts including both
WEIGHT FUNCTION W(X)=(1-X)^APLHA*X^BETA
ID = 1
Alpha = 0
Beta = 0
** VECTOR W:
1.00794E-01 2.08451E-01 2.60463E-01 2.42694E-01 1.59820E-01 2.77778E-02
Orthogonal Collocation for NonLinear BVP
Ready to Continue
確定
CHEER Copyright RESI 2001
開始執行


Orthogonal Collocation Method for Non-Linear BVP Ordinary Differential Equation
** SOLUTION
3.02815E-01 3.63855E-01 4.83298E-01 6.64652E-01 8.72022E-01
1.00000E+00
** SOLUTION AT FIXED GRIDPOINTS
X*******Y***
0. 2.890910E-01
0.1 2.924865E-01
0.2 3.028820E-01
0.3 3.209706E-01
0.4 3.481162E-01
0.5 3.866387E-01
0.6 4.402334E-01
0.7 5.147116E-01
0.8 6.195521E-01
0.9 7.712307E-01
1. 1.000000E+00
>>> EFFECTIVENESS FACTOR = 3.54199435E-01
============END============
CHEER Copyright RESI 2001
開始執行

結果討論

1. 表 10.3 顯示配置點數不同時，比較計算所得結果之正確性。由計算結果可以發現，$N \geq 3$以上，誤差均小於 0.1%。N=5 時，誤差已經小於10^{-6}。取更高的 N 值並無太大意義。

表 **10.3**　配置點數與解答之正確性比較

X	Y			
	N=3	N=4	N=5	N=19
0.0	0.2853386	0.2898014	0.2890910	0.2891837
0.1	0.2893154	0.2930179	0.2924865	0.2925483
0.2	0.3012278	0.3030042	0.3028820	0.3028801
0.3	0.3211468	0.3207469	0.3209706	0.3209323
0.4	0.3496105	0.3478497	0.3481162	0.3480984
0.5	0.3882519	0.3866154	0.3866387	0.3866624
0.6	0.4406798	0.4404462	0.4402334	0.4402563
0.7	0.5136106	0.5148471	0.5147116	0.5146922
0.8	0.6182509	0.6194011	0.6195521	0.6195367
0.9	0.7719328	0.7711624	0.7712307	0.7712564
1.0	1.0000000	1.0000000	1.0000000	1.0000000
有效係數	0.3543617	0.35420556	0.3541994	0.3541992
Err%	0.0459%	0.0018%	0.0001%	0.0000%

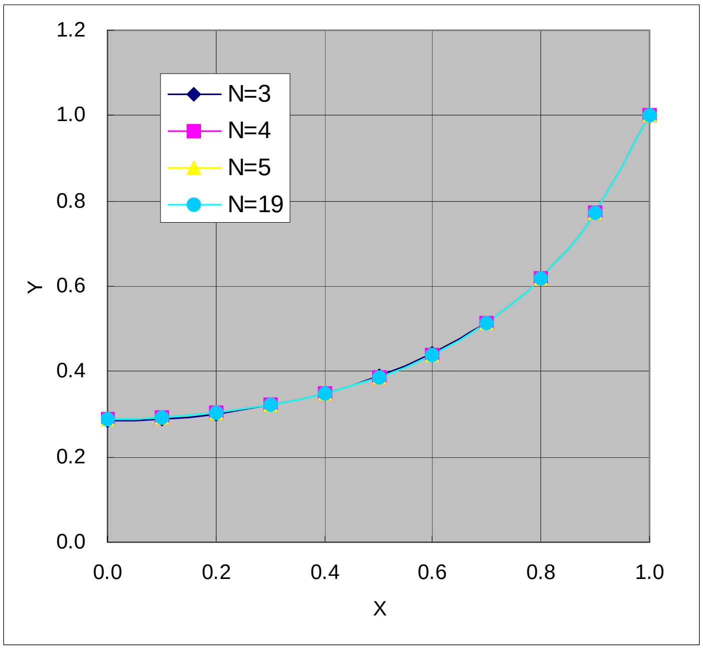


圖 10.3　解答之正確性比較

2. 微分方程式 $\frac{d^2y}{dx^2}+\frac{1}{x}\frac{dy}{dx}-\Phi^2y^2=0$ 計算所需使用的最小 N 值，可以利用數量級分析作判斷。假設函數的數量級可以用 O[--]表示：

$$O(\frac{d^2y}{dx^2}+\frac{1}{x}\frac{dy}{dx})=O(\Phi^2y^2)=\Phi^2O(y^2)\approx\Phi^2 \tag{10-8.4}$$

又因方程式的解可以表示成

$$y_N=1+(1-x^2)\sum_{i=1}^{N}a_ix^{2(i-1)}=-a_Nx^{2N}+\cdots\cdots$$

由於

$$\text{Max}(\frac{d^2y}{dx^2})\sim(2N)(2N-1)\text{Max}(x^{2N-2})\sim(2N)(2N-1)$$

$$\text{Max}(\frac{1}{x}\frac{dy}{dx})\sim(2N)\text{Max}(x^{2N-2})\sim(2N)$$

因此，

$$\text{Max}(\frac{d^2y}{dx^2}+\frac{1}{x}\frac{dy}{dx})\sim4N^2 \tag{10-8.5}$$

由方程式 (10-8.4) 及 (10-8.5) 可以得到 $4N^2\approx\Phi^2$ 或 $N>\frac{\Phi}{2}$ 即可得到相當正確的解答。

3. 根據本例題我們可發現非線性邊界值問題的解法，基本上與線性問題的邏輯完全相同，唯一的差別在於聯立方程式的解法不同而已。但由於非線性代數方程式必須利用迭代法求解，因此，解題所需計算次數遠較線性問題多。

4. 在本程式中，已經將一次元滲透及反應問題的一般模式寫成

$$\frac{d^2y}{dx^2}+\frac{s}{x}\frac{dy}{dx}=\Phi^2f(y) \tag{10-8.6}$$

利用座標轉換 $\xi=x^2$，得到

$$\xi\frac{d^2y}{d\xi^2}+\frac{s+1}{2}\frac{dy}{d\xi}=\frac{\Phi^2}{4}f(y)$$

$s=0$ 為直交座標，$s=1$ 為圓柱座標，$s=2$ 為球形座標。

5. 有效係數 η 可改寫成

$$\eta=\frac{1}{f(1)}\int_0^1 f(y)dx^{(s+1)}=\frac{s+1}{2}\frac{1}{f(1)}\int_0^1 f(y)\xi^{(\frac{s-1}{2})}d\xi=\sum_i W_i f(y_i)/f(1) \quad \textbf{(10-8.7)}$$

參考文獻

Visual Basic

1. Finlayson, B. A., “The Method of Weighted Residuals and Variational Principles” Academic Press, (1972).

2. Villadsen, J., and M. L. Michelsen; “Solution of Differential Equation Models by Polynomial Models”, Prentice-Hall, (1978).

3. Finalyson, B. A., “Nonlinear Analysis in Chemical Engineering”. (1980).

4. Davis, M. E., “Numerical Methods and Modeling for Chemical Engineers”, John Wiley, New York, (1984).

5. Aris, R. “The Mathematical Theory of Diffusion and Reaction in Permeable Catalysts”. Oxford, Clarendon Press (1975).

6. Carberry, J. J., “Chemical and Catalytic Reaction Engineering” McGraw-Hill, New York, (1976).

7. Weisz, P. B., and J. S. Hicks, “The Behavior of Porous Catalyst Particles in View of Internal Mass and Heat Diffusion Effects”, Chem. Eng. Sci., 17, 265 (1962).

習　題

Visual Basic

1. 考慮例 10-2，$\frac{d}{d\xi}[(1+\psi\theta)\frac{d\theta}{d\xi}]=0\,;0<\xi<1,\theta(0)=1,\theta(1)=1$；假設方程式中的參數 $\psi=1$時；

(a) 試證明若利用數學配置法求解，且只取一個配置點時，可以得到

$$\theta_1=1.317\xi-0.317\xi^2$$

(b) 試證明若利用數學配置法求解，且當 N=2 時，所得到結果爲

$$\theta_2 = 1.4076\xi - 0.5992\xi^2 + 0.1916\xi^3$$

(c) 試證明若利用葛勒金法且只取 N=1 時，所得到結果為

$$\theta_1 = 1.326\xi - 0.326\xi^2$$

(d) 試證明若利用慣量法且取 N=1 時，所得到結果為

$$\theta_1 = \frac{4}{3}\xi - \frac{1}{3}\xi^2$$

(e) 試證明若利用慣量法且當 N=2 時，所得到結果為

$$\theta_2 = \frac{3}{2}\xi - \frac{3}{4}\xi^2 + \frac{1}{4}\xi^3$$

(f) 試作表比較各方法之差異。何者最佳？

2. 考慮例 10-2，$\frac{d}{d\xi}[(1+\psi\theta)\frac{d\theta}{d\xi}] = 0\,;\,0<\xi<1\,,\,\theta(0)=1\,,\,\theta(1)=1$；假設方程式中的參數 $\psi=1$時；若假設基礎函數的階次 N>2，且配重積分函數採用葛勒金法；

(a) 試仿照本章第二節所介紹的方式，將未定係數方程式寫成下列形式：

$$(\underline{\underline{A}} + p\cdot\underline{\underline{B}})\underline{a} = p\cdot\underline{C}$$

(b) 利用高斯消去法求解陣列$\underline{a}$，並建立θ與ξ關係。

(c) 利用Excel試算表程式求解陣列$\underline{a}$，並建立θ與ξ關係。

3. 若 $\psi=10$，試重做問題 2。

4. 利用正交配置法重解問題 2。

5. 試算表程式對於重複性的運算是相當容易使用的工具，仿圖 10.2(a) 及圖 10.2(b) 的試算表程式，解同一問題

$$u\frac{d^2y}{du^2} + \frac{dy}{du} - py = 0$$

$$BC:\ \text{在}u=1\text{處},\quad y=1 \qquad \frac{\partial^k y}{\partial u^k} = \text{有限值}, k\ge 0$$

$$\eta_N = \int_0^1 y_N\,dx^2 = \int_0^1 y_N\,du$$

(a) 修改程式為 N=10，並與 N=3 及 N=5 的結果作比較。

(b) 修改程式為使用葛勒金法，N=3, 5, 10。

(c) 修改程式為使用慣量法，N=3, 5, 10。

(d) 比較各種方法所得結果的差異。

6. 假設一微分方程式 $L(y,u)=0$ 的邊界條件為

$$\text{BC1} \qquad -k_1\frac{\partial y}{\partial u}+h_1 y=\xi_1 \qquad @u=0$$

$$\text{BC2} \qquad -k_2\frac{\partial y}{\partial u}+h_2 y=\xi_2 \qquad @u=1$$

令此微分方程式的 N 階近似解為 $y_N=\sum_{i=0}^{N+1}a_i u^i$, $Nu_j=h_j/k_j$

(a) 試將微分方程式的 N 階近似解為 $y_N=\sum_{i=0}^{N+1}a_i u^i$ 代入邊界條件 BC1 及 BC2 中，並求解 a_0 及 a_1。

(b) 試證明微分方程式的 N 階近似解可以寫成以下一般式

$$y_N=\varphi_{0,1}+\varphi_{1,1}u+\sum_{i=2}^{N+1}a_i\left[u^i+\varphi_{0,2}(i+Nu_2)+\varphi_{1,2}(i\,u+u\,Nu_2)\right]$$

其中

$$\varphi_{0,1}=\frac{\xi_1 h_2+\xi_1 k_2+\xi_2 k_1}{h_1h_2+h_1k_2+k_1h_2}$$

$$\varphi_{0,2}=\frac{-k_1k_2}{h_1h_2+h_1k_2+k_1h_2}$$

$$\varphi_{1,1}=\frac{h_1\xi_2-h_2\xi_1}{h_1h_2+h_1k_2+k_1h_2}$$

$$\varphi_{1,2}=\frac{-h_1k_2}{h_1h_2+h_1k_2+k_1h_2}$$

$$Nu_2=\frac{h_2}{k_2}$$

(c) 利用以上結果，考慮在下列邊界條件時

$$\text{BC1} \quad dy/dx=0 \quad , \quad @x=0$$

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

試證明微分方程式的 N 階近似解可以用下列方程式表示

$$y_N = 1 + (1 - x^2)\sum_{i=1}^{N} a_i x^{i-1}$$

7. 核反應器所使用的燃料棒，基本上是由圓柱狀可分裂的放射性物質外覆鋯合金所構成，燃料棒的溫度分布可利用以下的微分方程式描述：

$$k_e\left[\frac{d^2T}{dr^2} + \frac{1}{r}\frac{dT}{dr}\right] = -S(r)$$

其中 k_e 為多孔狀氧化鈾粒子的有效熱傳導係數，r 表半徑位置，S 為分裂反應所伴生的能量。假設

$$\frac{S(r)}{k_e T_0} = 1 + 0.5\left(\frac{r}{R}\right)^2$$

$r = 0$ 時，$\dfrac{dT}{dr} = 0$ ；$r = R$ 時，$T = T_0$

(a) 令 $\theta = \dfrac{T - T_0}{T_0}$ ，$\xi = r / R$ ，將原方程式轉換成

$$\frac{d^2\theta}{d\xi^2} + \frac{1}{\xi}\frac{d\theta}{d\xi} = -1 - \frac{1}{2}\theta^2$$

BC1　$\xi = 0$ 時，$\dfrac{d\theta}{d\xi} = 0$

BC2　$\xi = 0$ 時，$\theta = 1$

(b) 試證明 θ 的理論解為

$$\theta = -\frac{\xi^4}{32} - \frac{\xi^2}{4} + \frac{9}{32}$$

(c) 試利用配重殘值法解此問題，並與理論解作比較。

8. 近年來由於空氣污染管制法執行愈來愈嚴格，化工界急於尋找最有效率且經濟的觸媒，以便由汽車廢氣中，有效地將未完全燃燒的碳氫化合物、一氧化碳及一氧化氮去除。經過研究發現，活性鉬金屬可將這三種污染物去除，而在這類觸媒上，CO 的氧化或 NO 的還原，均可利用 Langmuir-Hinshelwood 模式描述

$$R(C) = \frac{C(k' + C)}{(1 + kC)^2}$$

球形觸媒粒子的質傳及反應模式爲

$$\frac{1}{r^2}\frac{d}{dr}(r^2\frac{dC}{dr})=\Phi^2 R(C)$$

BC1　$r=0$ 時，$\frac{dC}{dr}=0$

BC2　$r=1$ 時，$C=1$

試利用正交配置法求其濃度分布及觸媒粒子的有效度係數與希笠模數 Φ 的關係。Φ 值範圍由 0.1 至 100，所得結果並作圖於全對數紙上。

9. 考慮問題 8，若反應物濃度極低時，則 $R(C)\cong k'C$，則微分方程式可簡化成

$$\frac{1}{r^2}\frac{d}{dr}(r^2\frac{dC}{dr})=\Phi^2 C$$

BC1　$r=0$ 時，$\frac{dC}{dr}=0$

BC2　$r=1$ 時，$-\frac{dC}{dr}=B_i〔C-1〕$

其中 BC2 是考慮外界質傳所得結果，試利用正交配置法解下列情況。

	A	B	C	D	E
K	1	10	100	100	100
k'	0	0	0	1	10

10. 考慮問題 8，由於活性鉬金屬價格相當昂貴，因此，有人建議爲了提高觸媒使用效率，所使用觸媒只有外層殼狀部分均勻地含浸活性金屬，而內部則完全沒有活性金屬原子。其數學模式分爲二部分

<u>含浸觸媒的活性層</u>

$$\frac{1}{r^2}\frac{d}{dr}(r^2\frac{dC_1}{dr})=\Phi^2 C_1$$

BC1　$r=0$ 時，$-\frac{dC_1}{dr}=B_i〔C_1-1〕$

BC2　$r=\rho$ 時，$\frac{dC_1}{dr}=\frac{dC_2}{dr}$

<u>只有觸媒單體的中心層</u>

$$\frac{1}{r^2}\frac{d}{dr}(r^2\frac{dC_2}{dr})=0$$

BC3　$r=\rho$ 時，$C_1=C_2$

BC4　$r=0$ 時，$\frac{dC_2}{dr}=0$

若 B_i 及 Φ 值仍選擇與問題 8 相同，試比較其有效度係數及濃度分布情況。

11. 絕熱管狀反應器的軸向熱傳導及滲透問題，可以利用以下聯立微分方程式表示 [6]：

$$\frac{1}{Pe}\frac{d^2C}{dx^2}-\frac{dC}{dx}-R(C,T)=0$$

$$\frac{1}{Bo}\frac{d^2T}{dx^2}-\frac{dT}{dx}-\beta R(C,T)=0$$

BC1　$x=0$ 時，$\frac{1}{Pe}\frac{dC}{dx}=C-1$

BC2　$x=0$ 時，$\frac{1}{Bo}\frac{dT}{dx}=T-1$

BC3　$x=1$ 時，$\frac{dC}{dx}=0$

BC4　$x=1$ 時，$\frac{dT}{dx}=0$

(a) 試利用正交配置法將上列方程式改寫成 $\underline{\underline{M}}\,\underline{y}=\underline{f}$ 的型式，並求出 M_{ij} 及 f_i 的表示式。

(b) 若 $\beta=-0.05$, $Pe=Bo=10$, $E=18$，且

$$R(C,T)=4C\exp[E(1-1/T)]$$

試建立程式計算無因次濃度 C 及溫度 T 的分布情況。

(c) 若 $Pe=Bo=100$，試重作 (b) 部分。

12. 球形觸媒若考慮內部濃度及溫度梯度對其活性的影響 [6]，則質量及能量平衡方程式分別可寫成

$$\frac{d^2C}{dx^2}+\frac{2}{x}\frac{dC}{dx}=\Phi^2C\cdot\exp〔\gamma(1-\frac{1}{T})〕$$

$$\frac{d^2T}{dx^2}+\frac{2}{x}\frac{dT}{dx}=-\beta\Phi^2C\cdot\exp〔\gamma(1-\frac{1}{T})〕$$

BC1　$x=0$ 時，$\dfrac{dC}{dx}=0$

BC2　$x=0$ 時，$\dfrac{dT}{dx}=0$

BC3　$x=1$ 時，$C=1$

BC4　$x=1$ 時，$T=1$

以上方程式可合併成

$$\frac{d^2C}{dx^2}+\frac{2}{x}\frac{dC}{dx}=\Phi^2 C\cdot\exp\left[\frac{\alpha\beta(1-C)}{1+\beta(1-C)}\right]$$

BC1　$x=0$ 時，$\dfrac{dC}{dx}=0$

BC2　$x=1$ 時，$C=1$

Weisz and Hicks[7] 發現 $\alpha=30, \beta=0.4$，且 $\Phi=0.3$ 時，會存在三組解。

(a) 利用正交配置法將上式改寫成以下型式：

$$\underline{\underline{M}}\,\underline{y}=\underline{f}$$

(b) 所得的非線性聯立方程式，試利用不同的啓始假設值，搜尋找出三組解。

13. 重做例 10-7 所考慮之二次非可逆反應

$$\xi\frac{d^2y}{d\xi^2}+\frac{dy}{d\xi}=\frac{\Phi^2}{4}y^2$$

BC1　在 $\xi=0$ 時，$y=$ 有限值

BC2　在 $\xi=1$ 時，$y=1$。

試建立程式求以下結果；

(a) 不同 Φ 值時的濃度分布曲線

(b) η 與 Φ 之關係（$\eta=\int_0^1 y\,dx^2$）

考慮 Φ 值範圍由 0.1 至 100，所得結果並作圖於全對數紙上。

14. 試利用正交配置法解例 9-5。

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

BC1　$r=r_1$，$T=T_B$

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

或寫成無因次方程式

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

BC1　$Z = 1$　,　$\theta = 1$

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

其中

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

15. 試利用正交配置法解例 9-6。請嘗試改變正交配置法中的α及β值，使配置點大部分集中在接近$x = 1$處，否則將使濃度 y 變成不合理的負值 (y 必須大於或等於 0)。

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

BC1　$x = 0$　，　$\frac{dy}{dx} = 0$

BC2　$x = 1$　，　$y = 1$