

计算机数值应用法

李宗义 编著

修订第六版

本书荣获台湾 1978 年度青年研究著作奖

台湾复文书局

大學用書

計算機數值應用法

李宗義編著

復文書局

計算機數值應用法

著作權執照台內著字第19074號

版權所有

翻印必究

中華民國七十二年十月修訂五版

中華民國七十二年九月修訂六版

修訂版特價 300 元

編著者： 李 宗 義

地 址：新竹市高峯路110巷1弄19號

發行者： 吳 主 和

發行所： 復 文 書 局

地址：臺南市東門路421巷28號

門市部地址：台南市林森路二段63號

電話：(062)370003・386937

郵政劃撥帳戶32104號

No.28, LANE421 DONG-MEN
ROAD TAINAN TAIWAN
CHINA

TEL: (062)370003・386937

本書局經行政院新聞局核准登記發給
出版事業登記證局版台業字第0370號

序

電子計算機應用於工業生產、商業事務及科學研究等方面，可使複雜繁難的分析與計算迅速而正確地解決，實為值得推廣使用的新工具。計算機雖為各界爭相使用，但礙於計算機語言的隔閡，如何將問題轉換成可以借助計算機處理的形式遂成為使用者必須探討的課題。本書旨在介紹應用計算機之 FORTRAN 程式解數學問題的方法。

近年來數值分析的英文書籍雖多，但深入淺出的中文本仍少，更鮮有對程式設計詳論者。雖云現代學生已習於原文書籍的閱讀，但究不若本國文字之親切、迅速與易於了解。筆者有鑒於此，乃參考國內外著述及有關論文編著此書。全書共分九章，舉凡一般數值分析上的典型方法均已包羅。各數值方法均附程式實例，並加解說，期使讀者明瞭理論之餘更能應用，以收事半功倍之效。

書中的程式計劃大部份是筆者在成功大學時利用 IBM 1130 完成的，另一些則是筆者在中正理工學院時利用 CDC 7000 處理的，亦有直接節錄自原文書籍者。因此，讀者直接使用時，對於輸入、輸出及格式指述應稍加留意，以配合所用的計算機。本書係假設讀者已對 FORTRAN 語言有了相當的了解，如讀者尚不諳於程式設計，可以參閱筆者另著“電腦語言 FORTRAN”（第二十三版）當有所幫助。

本書出版，波折再三，誠所謂差字不易付梓維艱，況筆者不敏冗務繁瑣，郭公夏五勢所難免，尚祈讀者不吝指正！

李 宗 義

六十六年於新竹



再版序

目前，計算機語言FORTRAN 的課程在國內各大專院校中已算是相當的普遍了，幾乎文理工商各系的同學都有機會接觸。但是礙於計算機的時間就是金錢，每位同學的實習機會還是不夠充分。如果不是自己多下功夫，那麼一學期下來，大概除了幾個名詞以外，別無概念可言，至於要應用到求解問題上，就更覺心有餘而力不足了。

數值方法不是一門什麼大學問，卻是研究大學問必備的小技巧，今天不論那一行，沒有用不著它的，有它來效犬馬之勞，可以省上萬分氣力，何樂而不為？可惜許多人在FORTRAN 語言學了一知半解之後，便畫地自限裹足不前，卒至前功盡棄。請記住，數值方法只不過是些巧妙的加減乘除運算，把一切的積微分、方程式等解析簡化到了極致，在你了解了基本程式設計之後，它將帶領您走入應用的領域。希望每位同學都能更上層樓，磨練一下自己的思考力。有朝一日書到用時，它便會給您帶來無比的方便。

近年來硬體科學的不斷突破，各類硬體元件的價格急遽下降，使得計算機成了大衆化的產品，尤其微計算機的誕生，更帶來了莫大的震撼。軟體科學的有待努力，相對的成了當前最急切的課題，時代在加速進步，眼看別人的機器一代一代的蕃衍，我們在學習使用方法上，委實是加緊努力的時刻了。

本書再版之前，筆者又校訂一次，更正了一些誤謬之處，尤其是程式內容的排印，較難做到百分之百的完美，如果您發現了什麼文字或資料上的缺失，尚希來函指正，不勝感激之至。

李 宗 義

中華民國六十八年十月於紐約州立大學

目 次

第一章 FORTRAN 語言概要

1-1	計算機簡介.....	1
1-2	FORTRAN 簡介.....	2
1-3	Fortran 程式的設計效率.....	30
1-4	繪圖副程式 PLOT.....	42

第二章 數值方法之誤差與不穩定性

2-1	引言.....	51
2-2	絕對誤差與相對誤差.....	55
2-3	固有誤差.....	56
2-4	截尾誤差.....	57
2-5	浮點運算.....	58
2-6	捨入誤差.....	66
2-7	誤差傳播.....	71
2-8	誤差估計.....	80
2-9	條件數.....	82
2-10	不穩定性.....	86
習題		106

第三章 方程式之根

3-1	半間距法.....	111
3-2	直接疊代法.....	117

3-3 假位法與割線法	128
3-4 穆勒法	147
3-5 韋格斯坦法	155
3-6 牛頓—拉福森法	162
3-7 利用綜合除法的牛頓法	167
3-8 葛列菲根平方法	180
3-9 白努利法	194
3-10 林—貝爾斯多法	202
3-11 重根之求法	215
習題	220

第四章 矩陣與聯立方程式

4-1 矩陣簡介	223
4-2 矩陣之基本運算	235
4-3 行列式值	243
4-4 聯立線性方程式與柯拉瑪法則	250
4-5 高斯直接消去法	251
4-6 高斯—約旦消去法	259
4-7 高斯—約旦法求行列式值	265
4-8 亞可比疊代法	268
4-9 高斯—賽德疊代法	274
4-10 柯勞特法與杜立妥法	279
4-11 柯列斯基法	287
4-12 緩衝法	292
4-13 聯立線性方程式之解法比較	301
4-14 矩陣方程式	303

4-15 帶型聯立方程式之解法.....	308
習題	316

第五章 反矩陣與特徵值問題

5-1 高斯—約旦法求反矩陣.....	319
5-2 節省儲存位置的高斯—約旦法.....	324
5-3 柯列斯基法求對稱矩陣之反矩陣.....	330
5-4 利用三角矩陣（柯列斯基法）求一般矩陣之反矩陣.....	336
5-5 疊代法求反矩陣.....	342
5-6 利用柯勞特法解聯立方程式以求反矩陣.....	347
5-7 分割法求反矩陣.....	353
5-8 交換法求反矩陣.....	365
5-9 特徵方程式與特徵值.....	386
5-10 乘方法求特徵值及特徵向量.....	389
5-11 拉堤蕭瑟法求特徵值與特徵向量.....	400
5-12 丹尼列夫斯基法求特徵方程式、特徵值及特徵向量.....	415
5-13 亞可比法求特徵值與特徵向量.....	437
習題	454

第六章 插值法與近似法

6-1 函數之差商與差分.....	458
6-2 線性插值法.....	466
6-3 等間距插值公式.....	469
6-4 差分表與牛頓插值程式計劃.....	477
6-5 高斯及司徒令插值程式計劃.....	483
6-6 拉格朗日插值法.....	494

4 計算機數值應用法

6-7 牛頓差商插值公式與艾特肯法	498
6-8 逆插值法	502
6-9 近似法	505
6-10 最小二乘方近似法	508
6-11 最小二乘方多項式	514
6-12 柴比西夫多項式近似法	521
習題	536

第七章 數值微分與積分

7-1 微分與積分運算子	537
7-2 數值微分	541
7-3 外推導數法	551
7-4 數值積分	557
7-5 梯形積分法則	564
7-6 不定積分之應用	570
7-7 牛頓-寇次積分法	580
7-8 辛普森積分法	591
7-9 羅勃格積分法	597
7-10 高斯積分	608
7-11 積分法之比較	619
7-12 數值多重積分	622
習題	638

第八章 常微分方程式

8-1 前言	641
初值問題	

8-2 泰勒法	645
8-3 奧衣勒法	651
8-4 較準確的兩種奧衣勒法	656
8-5 阮奇—庫塔法	662
8-6 聯立微分方程式與高階微分方程式	676
8-7 麥恩預測—修正法	692
8-8 海明法	705
8-9 亞當默頓法	720
邊界值問題	
8-10 試差法	728
8-11 有限差分法	741
習題	767

第九章 偏微分方程式

9-1 引言	771
9-2 橢性偏微分方程式	773
9-3 拋物性偏微分方程式	785
9-4 雙曲性偏微分方程式	796
習題	808

第十章 附 錄

10-1 各計算機系統之特性	811
10-2 FORTRAN 流程圖之符號	811
10-3 一般的 FORTRAN IV 庫存函數	813
10-4 相異數型間之運算關係	816

英文名詞索引	817
--------------	-----

第一章 FORTRAN 語言概要

1-1 計算機簡介

一般而言，計算機可分為數位計算機（Digital Computer）類比計算機（Analog Computer）及混型計算機（Hybrid Computer）。通常，數位計算機準確度較高而計算速度較差，類比計算機適得其反。混型計算機結合了前二者的優點於一身。

每一數位計算機均由五大單元組成：輸入（Input）、儲存（Storage）、控制（Control）、算術－邏輯（Arithmetic - Logical）和輸出（Output）單元。它們分別擔任不同的重要工作。算術－邏輯單元與控制單元是計算機的核心，兩者組成中央處理單元，簡稱CPU（Central Processing Unit or Computing Unit）。以下僅就五大部門的功能略述之。

1 輸入（Input）

欲使用計算機解決問題，必先將資料輸入計算機內。所有駕馭計算機的指令和有關數據，都以指述（Statements）及數字（Numbers）的形式打在卡片（Card）、紙帶（Paper Tape）或錄於磁帶（Magnetic Tape）、磁碟（Disk）等媒介上，經由輸入機件轉變成電磁信號而被存於計算機內。目前最新的發展是操作員可以寫一些資料或繪出圖形而透過輸入機件使計算機獲得

所要的資料，甚至可以利用語言、聲音直接與計算機互通款曲。

2 儲存 (Storage)

儲存部門專司計算機的記憶 (Memory) 功能。各種指令 (Instructions) 及資料都以符號存於此部門中，直到有指令指示時方才更改或除去。電腦與人腦在這方面有所差異，電腦有輔助機件如磁帶、磁碟可以與之相連而儲存許多資料，但是輔助機件大部份以機械動作來操作，遠較計算機內的符號運算為慢。

3 算術－邏輯 (Arithmetic - Logical)

此一部門可履行算術運算加、減、乘、除以及簡單的邏輯運算如：數目是否為零？數目之符號是正抑負？算術－邏輯單元可以回答此類詢問。如果需要解決複雜的問題，數值分析人員便常常按數值方法中的步驟重複地詢問上述問題。

4. 控制 (Control)

此一部門在中央處理單元內，是計算機的神經中樞，控制各種不同的操作，掌握輸入、輸出、儲存及算術－邏輯運算，並負責各單元間的資料傳輸。它告訴機器該履行什麼動作、在什麼地方進行、什麼時候進行等。

5. 輸出 (Output)

沒有輸出單元，計算機便失去了價值，這是與使用者聯絡的機構。輸出的資料可打在卡片上、紙帶上；錄於磁帶上或印在報表紙上。無論使用何種媒介，輸出的資料應該盡量使程式計劃員以外的人也能了解。因此，最好對輸出結果加以註解或說明，以免需要此資料的人不知所云。

1-2 FORTRAN 簡介

數位計算機是一種極複雜的電子機件，能接受、儲存並執行你的指令。但這些指令必須按照特定的方式書寫。事實上，計算機所真正而且唯一能接受的語言是機械語言（Machine Language）。此種語言所構成的指令表面上是一序列的數碼（Code），實際上代表著電磁信號，這些信號也就控制著計算機的內部動作，所以計算機能對該類數碼有特定的反應。

使用機械語言必須告訴計算機每一個該做的細節，這些細節是根據最基本的數學和邏輯來編排。因此，使用此種語言求解較複雜的問題時，程式計劃相當冗長。為了簡化計算機應用方面的程式，免去繁瑣的指令編排，乃產生了近似英語的計算機語言，可將普通的數學語言轉換成機械語言。換言之，你能利用計算機本身來組成冗長的機械語言指令。

在程式計劃上，最重要也最常用的符號語言（Symbolic Language）之一便是 FORTRAN。此字代表 FORMula TRANslation。FORTRAN IV 是 FORTRAN 語言系統的第四代產物。此種語言的指述極類似於一般的代數方程式與公式，如 $X = Y + \text{SQRT}(Z)$ ，其意義是告訴計算機取 Z 的平方根加上 Y ，所得結果稱為 X 。我們可以想見，計算機必須先將上述式子變成以機械語言組合的基本步驟，爾後方能執行運算。這是一種翻譯的工作，仰賴一個機械語言的程式計劃，稱之為 FORTRAN 編譯器（Compiler）。通常，此編譯器儲存於 CPU 的磁碟中，當某些控制卡讀入計算機時，會令控制單元將 FORTRAN 編譯器自磁碟移至主儲存器（Main Storage），以進行編譯工作。在編譯（Compilation）過程中，會有一個檢錯（Error - Detecting）的計劃同時動作，此計劃可以找出你程式計劃中的許多錯誤。這些錯誤通常是計劃書寫時的筆誤、打卡時的疏忽或卡片次序上的不正確等所造成。其後，計算機會在報表紙上將錯誤消息（Error Message）

印出，你可參照計算機手冊找出所印錯誤號碼對應的錯誤原因，適當地加以改正。有些錯誤在執行過程中才會發現，但是計劃本身邏輯上的錯誤或其他非語言規則上的錯誤是無法被發現的。此外，在執行時，編譯器尚可提供必要的輸出一輸入、算術、函數等副計劃。我們所設計的 FORTRAN 程式計劃稱為原始計劃（Source Program），經過 FORTRAN 編譯器編譯後，成為目的計劃（Object Program）。事實上，我們無法看到機械語言所組成的計劃，因為它都是在計算機內部組合、儲存和執行，而我們也沒必要自計算機中複抄一份目的計劃。

1-2.1 FORTRAN 指述

FORTRAN IV 語言之指述可分以下五類：

1 算術指述（Arithmetic Statements）

履行等號右側的算術運算，將所得結果給予等號左側的變數名稱。

2 控制指述（Control Statements）

用以控制各指述的執行次序。

3 輸入及輸出指述（Input and Output Statements）

達成計算機內外資料的傳遞。

4 說明指述（Specification Statements）

對其他指述或待處理之資料加以描述或說明。

5 副計劃指述（Subprogram Statements）

用以定義並達成主副計劃間之連繫。

前三者稱為可執行指述（Executable Statements）；後兩者稱為不可執行指述（Nonexecutable Statements），屬於一種描述性質的指述。

1-2.2 符號紙 (Coding Form)

通常先將程式計劃寫在符號紙上，符號紙的一橫行對應於一張卡片，使用規則簡述於下：

- 1 程式計劃以指述為單位，每一指述自新的橫行開始。
- 2 1 ~ 5 欄 (直行) 用以寫指述號碼 (Statement Number)，此號碼用作指述的參考數字，同一程式計劃中不得有重複者。號碼大小有 1 ~ 99999 者，亦有 1 ~ 32767 者，隨計算機之特性而異，詳細資料參閱附錄。
- 3 7 ~ 72 欄寫 Fortran 指述，如一指述含有 66 個符號以上，可自次一橫行的第 7 欄繼續寫，唯該連續橫行的第 6 欄應註上一個非零的符號，不得留為空白。一個指述可以連續寫多少橫行是有一定限制的，隨計算機類型不同而異，有的計算機壹指述可佔 6 橫行，有些可佔 20 橫行不等。詳細資料參閱附錄。
- 4 73 ~ 80 橫行供程式計劃者作備註之用，其資料不被編譯進入計算機內。
- 5 寫數據時 1 ~ 80 欄均可使用。
- 6 第 1 欄寫字母 C 時，該卡片成為註解卡 (Comment Card)，其內容對程式計劃不發生影響，故均用作程式計劃之描述或段落說明。

1-2.3 常數 (Constants)

Fortran IV 語言中常數有三種，即整常數 (Integer Constants)、實常數 (Real Constants) 和雙倍精確度常數 (Double - Precision Constants)。

1. 整數 (Integer)

任一數除正負號外不帶小數點及其他符號者屬之，又名定點常數（Fixed Point Constant）。其最大值及最小值有所限制，隨計算機不同而異，詳細資料參閱附錄。

2 實數（Real）

帶小數點之常數屬之。又名浮點常數（Floating Point Constant），此種常數除了小數點、正負號、指數符號外不可有其他符號或標點。實數的範圍亦隨計算機類型之相異而有不同的限制，詳細資料參閱附錄。指數型常數以E表示指數，例如 $2.15E + 02$ 代表 $2.15 \times 10^2 = 215$ 。指數符號E之後連正負號最多三位。

3 雙倍精確度常數（Double - Precision）

此種常數亦有兩類，一種與普通實常數類似，唯有效位數為實常數的兩倍。另一種相當於指數型的實常數，唯以D代表指數符號，例如 $8.7D + 03$ 代表 $8.7 \times 10^3 = 8700$ 。

1-2.4 變數（Variables）

變數係程式計劃中具有可變數值的量。其名稱用以代表該可變量。變數名稱可以用字母及數字組成，但中間不得有空白且第一個符號必須是字母，符號數有一定的限制，各型計算機不同，通常是1～6個符號組成，詳細資料參閱附錄。錢幣符號（\$）在某些計算機系統如RAX系統（IBM/360 Remote Access Computing System）中視同字母。

變數亦有實數、整數、雙倍精確度三種型態。未經型態指述（Type Statement）特別聲明時，直接觀察變數名稱的第一個字母，如屬I, J, K, L, M, N中任一時，表示該變數為整變數，否則該變數屬實變數或雙倍精確度變數，此稱之為內隱型態（Implicit

Type)。型態指述係專為宣告某些變數所屬之數型者，經該類指述宣告之變數即不再以名稱之首字母限定其數型，此稱之為外顯型態 (Explicit Type)。例如

INTEGER A, B

此指述使得 A, B 兩變數成為整變數。此外尚有其他的型態指述用以定義實變數及雙倍精確度變數，在稍後將分別述及。

1-2.5 陣列 (Arrays)

任何有關矩陣 (Matrices)、向量 (Vectors) 或帶下標 (Subscripts) 變數之計算都需用及陣列。陣列可有一因次 (One - Dimensional)、二因次 (Two - Dimensional) 及三因次 (Three - Dimensional) 者。今舉例說明 Fortran 語言中陣列的使用法。

[例] $A(1), A(2), \dots, A(I), A(I+1), \dots, A(M)$

其中 A 代表陣列名稱，1, 2, ..., I, I+1, ..., M 是下標，表示某特定變數在陣列中的位置。在數學中我們常用 $a_1, a_2, \dots, a_i, a_{i+1}, \dots, a_m$ 來代表一陣列，在 Fortran 語言中則常表為例 1 的形式。

[例] $B(1, 1) \quad B(1, 2) \quad \dots \quad B(1, N)$
 $B(2, 1) \quad B(2, 2) \quad \dots \quad B(2, N)$
 $\dots$
 $B(I, 1) \quad B(I, 2) \quad \dots \quad B(I, J) \quad \dots \quad B(I, N)$
 $\dots$
 $B(M, 1) \quad B(M, 2) \quad \dots \quad B(M, N)$

其中 B 是陣列名稱，下標代表特定變數在數列中的位置。在