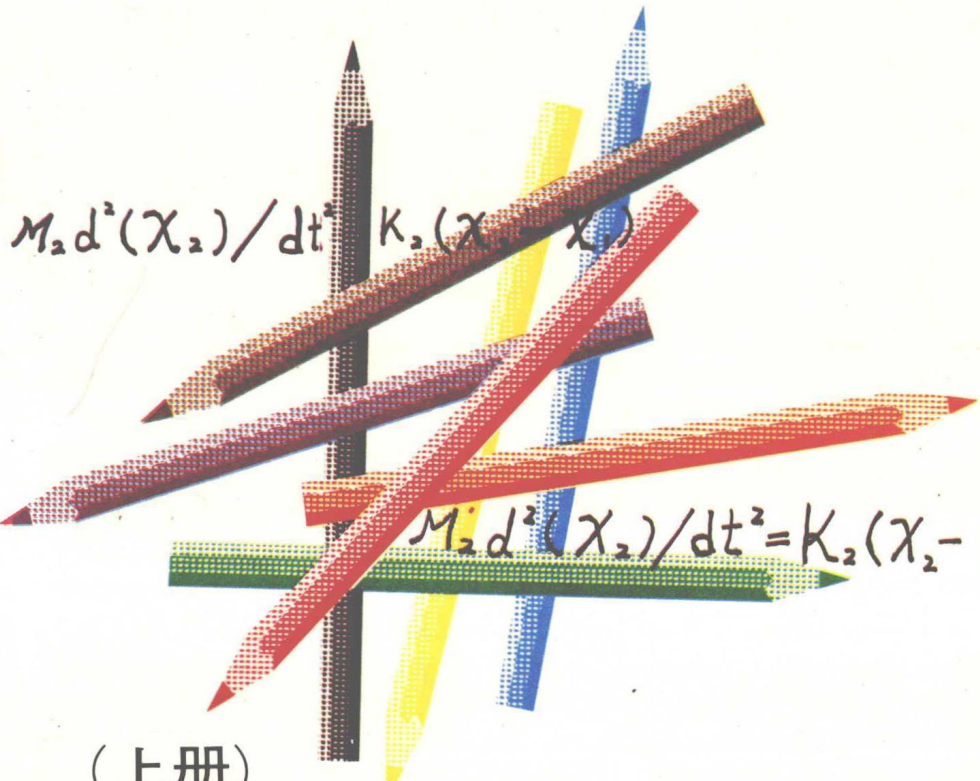


BASIC

數值計算程式集


$$M_2 d^2(x_2)/dt^2 = K_2(x_2 - x_1)$$

$$M_2 d^2(x_2)/dt^2 = K_2(x_2 - x_1)$$

(上册)

洪欽銘・陳景堂 譯

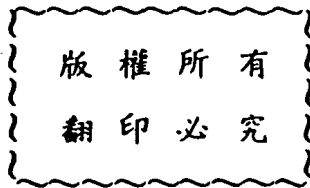
BASIC

數值計算程式集

(上冊)

洪欽銘・陳景堂 譯

儒林圖書公司 印行



BASIC 數值計算程式集 (上冊)

譯者：洪欽銘、陳景堂
發行人：楊鏡秋
出版者：儒林圖書有限公司
地址：台北市重慶南路一段111號
電話：3812302 3110883 3140111
郵政劃撥：0106792-1號
吉豐印刷廠有限公司承印
板橋市三民路二段正隆巷46弄7號
行政院新聞局局版台業字第1492號

1984年11月20日初版
定價新台幣180元正

原 序

BASIC 快速地成為最廣泛使用的計算機語言之一，它是由 Dartmouth College 的 J.G.Kemeny 教授與其同僚所發展出來，目的是要引入一種計算機語言，它能比其他語言，例如 FORTRAN，更快地進入熟練的程度。BASIC 由於有簡易的語法與交談的特性，因此已非常地成功。BASIC 的各種版本可在大的分時電腦系統、迷你電腦，以及最近在微電腦中找到。很可能，其使用性將繼續成長，特別是在微電腦的領域。

專業性的微電腦內含 BASIC 軟體，使個人電腦成為可能，這在幾年前是無法想像的。它們所提供的不但方便而且價格低，可代替大的分時系統。對大量的使用情況，專業性的微電腦可提供較快的響應，以及較低的計算成本。BASIC 有兩個主要缺點：它通常使用解譯程式 (interpreter)，因此，執行較慢；BASIC 的使用者只能使用有限的基本函數，而大的分時系統有很豐富的庫存副程式。而 FORTRAN 等語言，則在這方面較為優秀。

FORTRAN 原始碼整個被組合成可執行的最佳化目的碼，此目的碼係該計算機所使用者。目的碼的執行相當地快。BASIC 一般是碰到一個敘述才組合一個敘述。因此，FORTRAN 的執行遠較 BASIC 快。不過，對數學性的運算，BASIC 與 FORTRAN 的速率差二到十倍，因為，二者都取用相似的浮點副程式。

由於已存在了一段時間，FORTRAN 已成為科學界最廣泛使用的語言。因而，大的庫存副程式例如 SPSS (Statistical Package for the Social Sciences；參考資料 24) 在許多情況下，要放棄 FORTRAN 在大電腦系統中優越的速率與豐富的庫存副程式，而在微電腦中執行 BASIC 是不利的。

然而，BASIC 這兩個特殊的缺點，將來會被減小。在許多微電腦中BASIC的編譯程式將變成可供使用者選用。如此，使用者可在程式發展時，取得BASIC解譯程式交談式的優點，而執行時，有BASIC編譯程式高速率的優點。用BASIC來寫科學副程式的趨勢正快速地增加。這本書即為此種貢獻。

在準備這本書時有一些考慮。第一是範圍，作為一個美國公司的物理學家與工程師，作者已提出各種技術性的研究，每一個都需要不同的手法與技巧。基於這些經驗與對目前大學課程所提出分析方法的研究，我們選擇的主題示於目錄中。這些尚不完整。例如，統計技巧與它們相關的副程式很難選擇。未來的幾冊將延伸庫存副程式，以包含統計程式以及別的。

本書提出的型態是另一個要考慮的。作為庫存副程式集，我們希望能適合廣泛的對象，包含高中與大學的學生，以及專業人員，如果提供簡單的程式列表是不夠的。我們所選擇的型態是，先討論可能使用的副程式，接著為一簡明的數學描述，最後以副程式列表及示範例結束。提供了參考書目，使高級的程式員，可對使用的步驟學得更多，並得加以擴充。在某些情況下，超過本書範圍，但可能有某些讀者會使用到的一些進一步的應用，作者也作了建議。

另一個考慮的是要使用那一種版本的BASIC。有許多種版本的BASIC，其中包含由Microsoft公司與North Star公司所提供者。在這本書中，選用了Microsoft與North Star BASIC的子集(subset)，使它儘可能與各種系統相容。有些微電腦的製造廠商提供的BASIC解譯程式，使用它們自己的商標，但實際上其來源却是Microsoft，例如，Apple，Commodore，Ohio Scientific與Radio Shack均屬此類。所以，使用這些主要版本的子集，使我們僅需稍作修改，即可將所提出的副程式轉移到大部份的計算機。但未完全使用Microsoft與North Star BASIC的所有功能所造成的缺點，會反映在程式的大小與效率。例如，敘述分界符號(de-

limiter) 或許無法與其它的 BASIC 解譯程式相容，因此我們在一列中未使用多重的敘述。

爲了幫助使用者，在本文中所提出，以 North Star 爲主的副程式，也在附錄中以連接的形式提出。簡潔的 North Star 副程式也被提出，以改進記憶體的使用。這本書中所出現的軟體，可由任何機器 - 可讀 (machine-readable) 之型式提供，包含卡帶、紙帶與磁片。最後，附錄 3 提供了一些知識，指導你如何將既有程式轉換成其它型式的 BASIC。

我們已付出相當的心血，使本書所包含的知識，能使許多擁有含 BASIC 解譯程式的微電腦使用者，可極其輕易的使用。希望，這本書的材料能加強你電腦系統的應用。

F. R. Ruckdeschel

目 錄

原 序	V
第1章 簡 介	1
1.1 範 圍	1
1.2 內 容	2
1.3 結 構	3
1.4 需 求	3
1.5 執行速度	5
第2章 繪圖副程式	7
2.1 一維資料的圖形	8
2.2 函數的圖形	15
2.3 二維資料的圖形	19
第3章 複變數	27
3.1 複數平面	27
3.2 複變數的運算	34
3.3 $Z = X + i Y$ 的乘冪與次方根	41
3.4 球座標轉換	50
第4章 向量與矩陣的運算	55

4.1	向量運算	56
4.2	矩陣的加法與乘法	67
4.3	其它的矩陣運算	82
4.4	矩陣的座標變換	89
4.5	行列式	91
4.6	反矩陣	97
4.7	解線性方程組	104
4.8	特性多項式與特徵值	108
4.9	自乘法 (Porer Method) 求特徵值	114
4.10	矩陣的乘冪與微分方程式	120
第5章 亂數產生器		129
5.1	均勻分配, RND	131
5.2	線性分配	136
5.3	常態分配	139
5.4	布阿松分配	143
5.5	二項式分配	147
5.6	指數分配	150
5.7	費米分配	153
5.8	柯西分配	156
5.9	伽瑪分配	158
5.10	貝他分配	161
5.11	韋伯分配	164
第6章 基本的級數逼近		169
6.1	泰勒級數展開式	169
6.2	逼近級數展開式	173
6.3	最佳級數逼近	176

6.4	最小平方迴歸	178
6.5	推 廣	200
附錄1 副程式交互索引 		203
A	軟體索引 (依列號序)	205
B	函數索引	209
附錄2 副程式列表		213
A	示範與副程式所有程式列表	215
B	簡潔 (濃縮) 的副程式列表	253
附錄3 轉換至其他的BASIC 以及所有 程式列表		273
參考資料		315

第1章

簡 介

1.1 範 圍

BASIC 數值計算程式集綜合收集了以 BASIC 寫成的程式，它提供許多與大型分時電腦通用的庫存函數。大部份的副程式，都使用應用數學、數值分析與統計學中的標準方法。這些解析方法已利用 North Star 及 (或) Microsoft BASIC 語言的部份集 (subset)，轉換成電腦程式。

每一個副程式均安排有相關問題的說明，數學原理的描述，程式列表以及至少一個以上的使用例題。在一些單元中，討論如何擴展這些程式的使用。在所有的單元中，筆者所作的努力是清楚的提出知識，使執行時的困難減到最小。

雖然這本書並未企圖成為一本很完整的軟體庫存程式集，但它確也提供一系列科學程式師感興趣的程式。所撰寫的這些材料都是筆者費心想過的，讀者的範圍可由剛學習如何寫程式的工科學生，到希望充分了解所用技巧來源的專業人員。

這本書並不打算提供 BASIC 程式的講授。不過，由於程式內充分使用 REM (註解) 敘述，程式列表可很輕易的了解。由於使用者了解程式的邏輯，必要時，可更改程式以適合特殊的需要。然而，這些內部的註解有反面的影響，即佔用記憶體以及減慢執行的速率。爲了克服此點，這本書所提供的所有程式，均以簡潔的形式 (去掉 REM

敘述)重製一份於附錄2中。

這本書是一序列科學副程式叢書中的上冊 (Volume I)。上冊中所處理的是繪圖副程式、矩陣數及副近技巧。下冊 (Volume II) 考慮的是逼近、迴歸、插值、尋根及最佳化。這些都是各種科學的共同主題。

1.2 內 容

在準備這本書時，內容曾作若干選擇。最後內容的選定，是基於程式使用的評估，以及它們與一般科學計算方法的關係。

繪圖程式通常是需要的，在第二章提出幾種顯示圖形的方法。每一個程式要求程式師只能使用高解析度的 X Y 繪圖機。

第三章討論複變數，在一個單元裏，複數代數的規則被用來轉換直角座標與極座標。在另一個單元裏，考慮了 $Z = X + i Y$ 的乘冪。所提出的副程式可進一步使用於解析函數的多項式展開。

第四章包含一些屬於矩陣運算的副程式，它提供許多在一般微電腦 BASIC 解譯程式中找不到的函數，例如反矩陣。同時，也提出用矩陣代數方法解方程式的技巧。

第五章討論在計算機模擬中最常使用的函數之一，即亂數產生器。所提供的副程式可產生均勻與常態分配的亂數序列。許多的 BASIC 解譯程式祇包含均勻分配的亂數函數，而很少提供其它的分配函數。常態分配在許多應用中特別有用，諸如蒙地卡羅分析與訊號/雜訊比之計算。

在第六章中所討論的級數逼近，是一種初步性的介紹。一些重要函數的逼近，所使用的級數與電腦程式師經常使用的泰勒展開式略微不同。所提出的方法可提供較高的準確性與較快的執行速度。同時，也討論其它級數逼近的產生，以適合特殊的需要。

1.3 結 構

每一個副程式都是以特定的形式提出。首先是副程式科學使用的討論，接著是程式所用數學方法的說明。在許多單元裏，數學方法本身即能引起讀者的興趣，若欲進一步作更完整的討論，可在所附的參考資料中找到。再接著下去為副程式輸入與輸出特性的描述，包括記憶體所需空間以及執行速度。同時也列出副程式中所用到的函數與變數。副程式的列表也被印出，其後並附有例子。在某些單元中，也考慮副程式如何應用到別的或是較複雜問題，同時，也注意到程式綜合性的應用。後面這一項很重要，因為甚至一寫得很好的程式也可能有無法順利執行的時候。

筆者選擇上面的結構，是在看完幾本目前有關軟體書籍之後。某些書祇強調數學技巧，而很少討論可能的使用。而有些書，主要是程式列表，而沒有詳盡的說明。有趣的是，有些精彩的教科書却是針對可程式計算器而寫，特別是Hewlett-Packard 系列（參考資料4、5、12、13與26）。在某些情形中，可程式計算器手冊內所提出的古典演算法則，很適合計算機的執行，特別是涉及疊代方法時。

1.4 需 求

每一節中所提出的程式，其硬體與軟體的需求均被列出。所提出的大多數副程式，僅需在作業系統以外再佔有2 K以下的位元組。一些含維數宣告陣列資料的程式，對記憶體的需求，視資料集的多寡而定，因此可能非常大。

雖然無法滿足某些表格式之顯示，但我們仍企圖將終端機的列寬要求定在32字元以下。幾乎在所有的情況中，64字元的顯示已是足夠了。所有示範程式的輸入資料，均假設是由鍵盤輸入，所有執行時間

之估計，是假設在 IMSAI 8080 下執行，其時序信號為 2 MHz 而無等待狀態。

在 Microsoft 與 North Star BASIC 中可用的全部敘述與運算，並未被充分利用，特別是，未利用多重敘述於一列的功能，同時，輸入輸出的格式均保持簡單的形式。我們也使用少許幾個函數，雖然他們可以用本書中所出現的副程式來取代。例如，三角函數在某些程式中被用到，但是我們也提供他們的級數展開式。

程式以副程式的形式提出，也就是說，他們以 RETURN 敘述作為結尾。由於我們希望將軟體組合成套裝的庫存副程式，因此各程式，由列號 40000 開始，被依序排下。這些列號應是夠大了，而可遠離主呼叫 (main calling) 程式。

在使用副程式時，要特別小心，避免主程式與所用副程式重複使用變數名列。我們選擇下列規則：

- FOR/NEXT 迴路：六個一般常用的保留字：I < n > , J < n > , K < n > , L < n > , M < n > , 與 N < n > , 其中 n 為隨意的一個數值，它被放在第二個位置上。由這些變數延伸的陣列也包含在此規則內，例如，I4 (K) 也可在副程式中找到。
- 執行變數：六個被使用的執行變數為：U < n > , V < n > , W < n > , X < n > , Y < n > , 與 Z < n > , 以及由這些變數延伸的陣列。
- 參數：使用了五個參數：A < n > , B < n > , C < n > , D < n > , 與 E < n > , 以及由它們延伸成的陣列。

與副程式運算有關的進一步資料，可在書中找到。

1.5 執行速度

解譯程式的執行速度天生就比較慢。然而，如第六章所說明的，這裏所提出的數學性副程式，比相似的編譯程式慢四倍。在各種BASIC 中，執行速度的差別相當大，一基本的對照列於表 1.1（參考資料）。

本書所提供的副程式時間對照，是在標準的 8 位的 North Star BASIC 上執行的。14 位的 BASIC 執行慢 40 %。很明顯的，利用其他的 BASIC，至少可使執行速度增加兩倍以上。要有兩倍的改進是可能的，只要用 4 MHz 的微處理機取代本書中所用的 2 MHz 微處理機。所以，在每一章中，所提供的值，可被認為是一上限。

執行時間（毫秒）

函 數	MIT8K	OSI8K	NorthStar	NorthStar +
	(8080)	(6502)	(8080)	浮點電路板
乘	4	2	5	2
除	7	3	16	2
正弦 / 餘弦	23	17	99	11
對數	19	14	99	9
指數	28	22	73	8

表 1.1 在各種 BASIC 解譯程式中，某些函數執行速度（毫秒）之對照

第2章

繪圖副程式

在工業與商業計算機程式設計中，常常須要的輸出能力就是用圖形顯示資料與函數。下面三節將介紹能半自動地繪出所予資料與函數的軟體。這些副程式被設計成能配合大部份的 ASCII (American Standard Code for Information Interchange; 美國標準資訊交換碼) 終端機。最好能配置硬式拷貝 (hard copy) 裝置，視訊終端機有一定的列與寬，而我們的副程式只想繪出近乎方形的圖形，爲了克服視訊終端機列寬的限定，使用者可設定一系列的顯示寬度，以使圖形配置在螢幕內 (繪圖程式進一步的資料，見參考資料 28)。

所提出的第一個繪圖程式處理等間隔的資料值。第二個程式用來繪函數的圖形，它與等距離資料程式只有些微的差異。第三個程式較複雜，它用來顯示座標組的資料點。

繪圖軟體被設計成三個功能區段。第一個區段是呼叫程式，資料在此產生與組織，它可能是一個較大程式的一部份，或者如後所示，可能是一單純的資料來源。利用 GOSUB 敘述，此程式可呼叫實際的繪圖副程式，此副程式包含兩個部份：資料列印副程式與座標 (Y) 軸列印副程式。這樣的劃分是有意的，因爲同一個座標軸列印副程式可用在我們所考慮的三個例子中，在這三個例子中有兩個使用同樣的主繪圖副程式。因而，在後面三節裏有三個不同的呼叫程式，一般由使用者提供，只有兩個不同的繪圖副程式，及一個座標軸列印副程式。這種模組化結構，令使用者有相當適應性，不覺得太複雜。

這裏所提出的程式有一些經常可見的輸出特性。軟體欲產生一近乎方形的圖形，而其橫座標軸（X 軸）的長度長於縱座標軸（Y 軸）的長度。二方向的比例是可調整的，不過，最後的圖形依所用終端機字與字以及列與列間之距離而定。

軟體也定出 $Y = 0$ 的位置，同時，如果數值落在所繪資料的範圍之內，軟體即適當地移動 Y 軸。

爲使終端機繪圖的解析度能爲最佳，資料均依一定的比例安排，使得圖形的界限完全由資料的極值來決定。要不要用此一特性，可由使用者決定。

執行時間受制於提供資料給繪圖副程式的程式以及終端機的傳遞速率。我們建議圖形的寬度大於 10 而小於 132。

在後面各節的三個例子中，首先我們看的是最簡單的例子：描繪使用者提供的資料點，而 X 軸係等間隔。在第二個例子中，使用同一個繪圖副程式，而欲顯示的資料點由呼叫程式中的函數產生。第三個例子被視爲最重要，用來描繪二維的資料集，由於這一類的資料，其 X 或 Y 座標通常並非等距離，所以，須引入一個不同的繪圖副程式。

2.1 一維資料的圖形

在這一章將提出的所有繪圖程式，其方法並不須具備深奧的數學基礎。它只是充分地利用 PRINT 與 TAB 函數，要徹底了解此程式，最好順著所示的程式說明看下去。

一維繪圖副程式示於程式 2.1b，由列號 40000 開始（第一個有效的副程式敘述）。程式 2.1a 要求終端機輸入資料，並呼叫繪圖副程式以備繪圖。繪圖副程式需要下列明確的資料：

- L：繪圖時所用的終端機列寬
- N：欲顯示資料點的數目
- D (I)：N 個資料點