

數學標計

計 算 数 学

复旦大学数学系 編著

上海科学技术出版社

內 容 提 要

本书系复旦大学数学系数学专业革新教材之一。内容包括程序設計,偏微分方程数值解,程序組織等三部分。可作綜合大学数学专业計算数学課程的教材,讲授 24 学时,并可作高等院校有关专业的参考书。

計 算 数 学

复旦大学数学系 編著

上海科学技术出版社出版

(上海瑞金二路450号)

上海市书刊出版业营业许可证出093号

新华书店上海发行所发行 各地新华书店經售

商务印书館上海厂印刷

*

开本 850×1168 1/32 印張 3 16/32 字數 82,000

1960年6月第1版 1960年6月第1次印刷

印數 1—4,000

統一書号: 13119·370

定 价: (十) 0.42 元

編輯說明

我們受到全國持續躍進的大好形勢的鼓舞和推動，積極響應了黨的號召，在兩年來教育大革命已經取得偉大成績的基礎上，掀起了一个聲勢浩大的教學改革的群眾運動。通過這個運動，我們揭露了現在教學體系、教學內容和教學方法上陳舊落後的狀況，抓住訂方案、編大綱、寫教材、搞試驗等重要環節，試圖建立一套以馬克思列寧主義、毛澤東思想為指導的、反映現代科學發展水平的、理論聯繫實際的新的教學體系和內容，以及與之相適應的教學方法，使培養人才的工作更好地貫徹黨的社會主義建設總路綫的精神。

作為這種新的探索和嘗試，我們在教學改革運動中，師生結合，提供了“關於綜合大學數學專業課程革新的建議”，編寫了一套可供綜合大學數學專業試用的基礎課程教材。全套教材包括數學分析(一)、數學分析(二)、泛函分析、高等代數、綫性規劃和計算實習、計算數學、數理邏輯與控制論、常微分方程、數學物理方程、一般力學、連續介質力學、統計數學(包括信息論)等十二種，尚有物理學一種，因力量所限，未能及時編出。

根據“關於綜合大學數學專業課程革新的建議”的精神，我們力圖使這套教材具有以下几个特點：

一、在選材上，注意克服資產階級教育思想的影響，體現為社會主義建設服務和反映現代科學發展的要求。中國數學會提出的“數學發展的方向必須以解決尖端技術和重大工程、現代物理、自動化、國民經濟和大量計算任務中的數學問題為綱”，具體地說明了社會主義建設和現代科學發展對數學的要求，我們即以此作為選材的主要標準，同時，也考慮到基礎課的某些特殊要求，適當注

意了根据理論与实践之間的直接联系与間接联系,当前需要与长远需要等关系,来确定材料的取舍和不同材料的主次安排。根据这个精神,我們精簡了原来基础課内容中一部分不必要的古典内容,添加了一部分現代材料,还增加了一些新課程。

二、在材料处理上,注意克服过去課程設置各自为政、互不联系的缺点,体现科学知識的綜合与分类的辯証統一的关系。特别是近年来,邊緣学科大量出現,科学发展在原有基础上愈来愈明显地趋向新的更高级的綜合,我們想力求使这套教材适应这个趋势。

具体說来,对那些条件已經成熟的、可以綜合处理的内容,即加以統一处理,例如将泛函分析与实变函数、积分方程以及綫性代数中的部分内容統一处理,在泛函分析中加以綜合;对那些綜合趋势已經比較明显,但独立設課条件尚不成熟的,也分別情况,注意在有关課程間建立密切的有机联系,若干材料还重新另行配置,例如原来理論力学中振动理論的一部分内容,这次就移到常微分方程中去了。

三、在材料的处理与闡述上,以辯証唯物主义观点为武器,破除形而上学和唯心主义对数学教学的影响。数学的研究对象是客观世界的量的側面,所以它具有較多的抽象性,在研究方法上也較多地运用邏輯上的演繹推証。这些特点,本来应该有利于深刻地闡明問題的本質,但唯心主义者却总是加以歪曲,企图在引出抽象概念时,掩盖其实践来源,在形式論証中,避免闡述問題的本質。在这套教材中,我們力求消除这些唯心主义观点的影响。具体說来,对某些与生产实践有着更加直接联系的課程(如数学物理方程),既吸收已經严格建立数学理論的材料,也采用在实践中有广泛应用而理論上尚未成熟的材料,重新加以組織,恢复这門課程本来的生动活潑的面貌。各門課程中,对重要数学概念与問題的引进,都尽量闡明它們的直接和間接的实践来源;闡述論証过程中,

插入若干必要的描述性材料；得到的結論，也闡明它在實踐中直接和間接的作用。本學期我系幾門主要基礎課程，都初步做到了減少學時、提高質量，據了解，主要是在教學過程中初步體現了這個精神。因此，根據我們一些不成熟的經驗，要徹底解決這個問題，除在教材內容上盡量克服這些唯心主義觀點的影響以外，還要注意在教學方法中消除這些影響。

徹底實現教學改革，建立一套新的體系，是一個艱巨複雜的任務，也需要一個較為長期的時間來摸索。我們所作的一些嘗試，僅僅是個開端，既受到思想水平和科學水平的限制，又缺乏較充分的實踐經驗，某些課程的教材，還是在師生結合、邊學邊寫的情況下編出來的。因此不論在處理原則上或者在處理方法上都還不夠成熟。有不少問題，例如如何在教材中反映我國社會主義建設實際中所提出的數學問題、如何在有關各課程間建立更密切的有機聯繫等等，在編寫過程中，也還把握不定，處理不盡適當。我們懇切地希望同志們批評和指正。

上海科學技術出版社和商務印書館上海印刷廠對這套教材的迅速出版，給了極大的支持，我們在這裡表示衷心的感謝。

復旦大學數學系

1960年5月

序

最近几年来,計算数学这門学科有了很快的发展。計算数学的迅速发展是和快速数字电子計算机的出現分不开的。电子計算机具有空前的計算速度,为許多以前无法解决的問題提供了很好的解决条件。为了充分發揮电子計算机的作用,彻底解决那些过去无法解决的問題,迫切地需要新的計算数学理論来指导。因此,計算数学获得了迅速发展的动力,形成了一門独立的学科。

随着我国社会主义建設的高速度发展,工农业生产、国防建設和科学研究經常提出大量的数学問題,它們需要通过复杂的計算才能得到有效的解决,特别是科学技术向高精尖方向发展所提出的問題更加复杂,格外需要有新的計算数学理論来指导,才能得到解决。因此計算数学的发展,既是直接服务于国家建設,又是促进数学和其他科学,特别是尖端学科,如高速空气动力学、原子能反应堆、自动控制、宇宙火箭理論等等的发展。这样,計算数学在国民經济和科学发展中就有了非常突出的意义。鉴于以上这些,在綜合大学的数学专业中設立計算数学这門課是完全必要的。

計算数学的发展大致有两个方向:一个是計算方法方面。它是研究各种不同类型問題的算法,就其內容而言,和分析数学、应用数学以及統計数学相接近,是在傳統的計算数学基础上发展起来的。这方面目前的主流是綜合实际問題,研究适宜于电子計算机的計算方法,如綫性和非綫性微分方程的数值解法,高阶代数方程的数值解法,非綫性代数方程組和超越方程組的数值解法等。另一个是机器数学方面。它是研究計算方法在計算机上实现的問題,

主要是程序設計,其內容和数理邏輯相接近,目前的主流是研究如何最有效地使用电子計算机,包括程序自动化、自动翻譯、程序控制和生产自动化等。在計算数学这門課程中,我們介紹了程序設計的理論和应用,也介紹了适宜于电子計算机使用的解偏微分方程問題的差分方法。在程序設計方面我們力求簡單明了,在教学上注意把教师講授和同学編制具体程序結合起来,使同学在短期内具备編制程序的技能。在叙述解偏微分方程的差分法时,既注意到理論的严格性,也注意到在实际問題应用中的灵活性。至于其他的計算方法,在計算实习和数学物理方程等課程中已有介紹,在此不再重复。在叙述程序設計或偏微分方程数值解法时,当引入一个概念和提出一种方法时,都指出它們是为了解决何种实际需要而产生的,并在此基础上启示这門学科某些新理論所賴以产生的思想。这样,就揭示了所叙述的概念、方法和理論的本質,使讀者能更好地掌握。本教材力求將計算方法和程序設計貫穿在一起,使它們相輔相成,便于讀者全面掌握計算数学的內容。

由于我們缺乏实际工作經驗,因此在本教材里这方面的反映是不够的。这一缺陷有待于今后在解决实际問題中不断地补充改进。

总的說来,我們的目的是要在短短的教学時間內,使讀者能掌握計算数学的基本理論,特別是利用电子計算机来解决生产中提出的复杂計算問題,为偉大祖国的社会主义建設服务。

复旦大学数学系計算数学編写小組

1960年5月

目 录

序.....	1
第一章 程序設計初步.....	1
§1 机器結構.....	1
§2 数制.....	3
§3 指令系統.....	5
§4 各类算子,框图,邏輯图.....	12
§5 子程序.....	18
§6 比例因子.....	26
第二章 偏微分方程的数值解法.....	29
§1 差分方法的提出.....	29
§2 橢圓型方程的差分方法.....	31
§3 拋物型和双曲型方程的差分方法.....	58
§4 拟綫性双曲型一阶方程組的特征綫法.....	68
第三章 程序組織.....	77
§1 程序設計一例.....	77
§2 程序組織基本內容.....	82
§3 輸入輸出.....	84
附录.....	89
一、严格主元高斯消去法解綫性代数方程組的标准程序(M-3).....	89
二、常微分方程初值問題的龙格庫达方法.....	98
三、偏导数用差分表示的公式.....	101

第一章 程序設計初步

近代科学和技术的发展,給数学提出了既复杂又繁重的任务;另一方面,由于无綫电和电子学的发展,电子計算机的出現,提供了高速运算的設備,使人們能够摆脱大量的計算劳动,大大提高了生产效率。为了运用計算机,有必要使广大的工程技术人员、科学工作者和高等学校学生熟悉电子数字計算机的基本結構和运用方法,了解它广泛使用的可能性。这样就必須要介紹人們与机器互通的語言,使机器能按照它进行运算。这一套給机器指出运算、运算方向及参与运算的数的性质的語言,就是指令系統。完成一个計算問題的一系列指令就称为程序。如何按具体机器的指令系統,組織成程序,使机器能化最少時間得到正确的結果,这样的一門学科就是程序設計。

§1 机器結構

一般的电子計算机由下列部分組成:

一、存貯器: 用来存貯代碼(表示数量或表示机器的語言——指令),它由一些編了号的单元所組成。单元的編号称为地址,通常一个单元中存貯一个代碼(包括 31 个或 39 个或 43 个数位),但有些时候可以把表示一个量的数存放在两个单元中。当准确度要求不高,而又需存貯大量数据及指令时,可以在一个单元中人工地放二个或三个数。它的特点是:每个单元中的代碼平常不改变,但当送入另一个代碼时,則原来的代碼就消失而存貯了新的代碼,一个代碼送出存貯单元时,原单元保留原状。存貯器可以由

磁鼓或磁芯組成，磁鼓運轉時速度較慢，磁芯的工作效率較高。

二、運算器：對送入運算器中的代碼能進行 $+$ ， $-$ ， $:$ ， $\times$ ， $\wedge$ ， $\vee$ 等等運算。

三、控制器：根據計算程序，實現機器各部分的聯繫，指揮各部分工作。

四、人工控制器：可以破壞機器的自動工作，按照人的意志使機器開始計算，停止工作，改變存貯內容及改變工作順序等等。

五、輸入器與輸出器：輸入計算問題的初始信息（數及程序）及輸出計算結果。

六、外存貯器：是機器的附加存貯器，它的存貯量較大，但與運算器聯繫很少，甚至沒有聯繫。可與存貯器交換代碼，它由磁鼓或磁帶組成，有些機器可以沒有外存貯器。

機器的各部分是用代碼綫路，及控制綫路相互聯繫的。

通常指令的形式有：

$$\theta, A_1, A_2, \dots, A_k.$$

其中 θ 表運算碼， $A_1, A_2, \dots, A_k$ 表示參加運算的數的地址，要存貯結果的地址以及下一步工作指令的地址號碼等，這種有 k 個地址的機器稱為 k 地址的。

為了了解機器及指令系統，可通俗地以人們用算盤進行 $+$ $-$ $\times$ 來作譬喻：一本簿子好象存貯器，算盤好象運算器，大腦好象控制器， $+$ $-$ $\times$ 就是指令系統。

機器的種類很多，現在我國製造的有三地址的和二地址的。在蘇聯，電子計算機的種類就更多，象 B9CM 和箭牌電子計算機是三地址的，M-3 是二地址的，烏拉爾型是單地址的。其他社會主義國家和資本主義國家，也有種種不同名稱和不同地址個數的機器。但機器的種類雖多，它們的結構原理基本上是相同的，只是綫路構造有些不同，因而指令系統也各各不同。

下面是电子计算机的结构示意图。

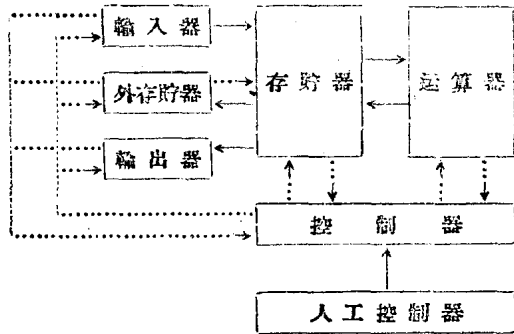


图 1-1

—→表示代码路线,→表示控制路线。

§2 数 制

机器的电路只有“通”与“不通”两种物理状态, 故而一般电子计算机要求用二进位数制来进行运算。机器采用二进位数制比采用十进位数制, 可以节省许多元件。由于二进位数的记录较长, 书写时可改用八进位数制, 对照关系如下:

0	1	2	3	4	5	6	7
000	001	010	011	100	101	110	111

它们是由三位二进位数组成。

通常数的形式, 可表成:

$$x = \pm M \cdot 2^P, \quad \frac{1}{2} \leq |M| < 1,$$

M 称为数的尾数, P 称为数的阶数。在机器里如果 P 是固定的数, 即小数点的位置固定, 这种机器就称为定点机。常见的是

$$P=0, \quad -1 < x < 1.$$

当 P 可以变化, 即小数点的位置可以改变时, 这种机器就称为浮

点机。例如 M-3 型是定点的,

$$-1 < x < 1, \quad P=0.$$

箭牌是浮点的,

$$-63 \leq P \leq 63, \quad \frac{1}{2} \leq |M| < 1.$$

現在談談數制的換算:

$$[\text{例 1}] \quad 1246 = 1 \times 10^3 + 2 \times 10^2 + 4 \times 10^1 + 6 \times 10^0.$$

其中 1, 2, 4, 6 都小于 10, 改為二進位的數為:

$$\begin{aligned} 1 \cdot 2^{10} + 0 \cdot 2^9 + 0 \cdot 2^8 + 0 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 \\ + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1, \end{aligned}$$

$$\therefore 1246_{(10)} = 10001011110_{(2)}.$$

其中每一位數都小于 2. 用八進制表示時為:

$$\begin{array}{cccc} 2 & 1 & 3 & 6 \\ 010 & 001 & 011 & 110 \end{array}$$

$$\therefore 1246_{(10)} = 2136_{(8)} = 10001011110_{(2)}.$$

四位十進位數在機器里需要 40 個電子管表達, 而 $9999 < 2^{14}$, 故二進位數只要 14 個電子管就夠了, 因而元件可大大的節省。

$$\begin{aligned} [\text{例 2}] \quad 0.25_{(10)} &= 2 \cdot 10^{-1} + 5 \cdot 10^{-2} = 0 \cdot 2^{-1} + 1 \cdot 2^{-2} \\ &= 0.01_{(2)} = 0.2_{(8)}. \end{aligned}$$

十進位整數翻成八進位數時, 可以除 8 取余, 然後用商除 8 取余, 一直繼續到商為 0 時結束。

$$x_{10} - \left[\frac{x_{10}}{8} \right] 8 = \alpha_0, \quad \left[\frac{x_{10}}{8} \right] = a_1, \quad a_1 - \left[\frac{a_1}{8} \right] 8 = \alpha_1, \dots$$

$$a_i - \left[\frac{a_i}{8} \right] 8 = \alpha_i \quad \text{到 } \alpha_i = 0 \text{ 為止。}$$

十進位小數翻成八進位數時, 可以乘 8 取整, 其後余數再乘 8 取整, 一直到所需的位數為止。

$$\begin{aligned}
 [\text{例 3}] \quad 0.0625_{(10)} &= 0.0625 \times 8 \times 8^{-1} = (0.5 \times 8) \times 8^{-2} \\
 &= 4 \times 8^{-2} = 0.04_{(8)}.
 \end{aligned}$$

§3 指令系統

这里介绍 M-3 的指令系統。M-3 机器共有 2048 个存貯单元，每个单元有 31 个代碼，它可以表示数或指令，表示形式为如下：

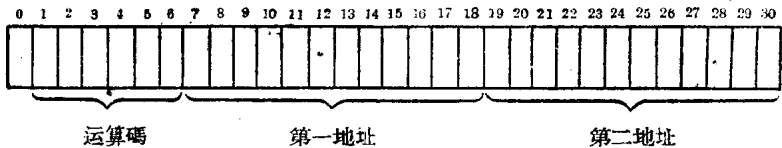


图 1-2

第 0 位是数或指令的符号位，指令的符号不影响运算的完成。运算碼由 2 个八进位組成，地址由 4 个八进位組成，写书的形式如下：

$$\pm \theta_1 \theta_2 \quad A_1 \quad A_2$$

設現在要进行运算： $a+b$ ，則在机器里应有单元存放 a 及 b ，同时亦应有单元存放結果，另外要有具体的指令来执行 $a+b$ ，它們串連起来就成下面程序。

[例 1] 0001 a
 0002 b
 $\rightarrow 0003 + 00 \quad 0001 \quad 0002 \quad a+b \Rightarrow 0002$
 0004 + 37 0000 0000 Ω

这里 0001, 0002, 0003, 0004 都是单元的編号，叫做地址，在 0001 里放 a ，在 0002 里放 b ，0003 里是具体执行 $a+b$ 的代碼，0004 是表示工作結束机器停止工作。

我們現在来具体地介绍机器完成这一运算的过程，及如何使机器接受这个任务的概况：

一、将上面四行写在专用的程序紙上，由穿孔机打成机器認

識的格式，即在紙帶上穿孔。

二、把紙帶放在機器的輸入口，使四個數分別送入 0001, 0002, 0003, 0004 里去，人工操作至此為止。

三、機器里實現從 → 開始這條指令，即執行 $a+b$ 這個運算，它的過程可分為八步：

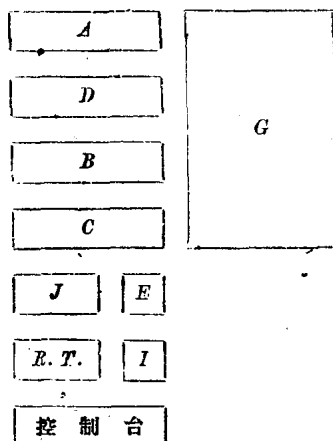


圖 1-3

1. 執行運算的指令地址 0003 放在啟動寄存器 I 中，由 I 將 0003 送入選擇寄存器 E 中。

2. 存貯器 G 根據 E 中的 0003 將 0003 的內容送入寄存器 C 中。

3. 一般情況下 I 自動加 1，得 0004，運算碼 00 進入譯碼器 J ，第一地址 0001 從 C 進入 E 。

4. G 將 0001 的內容送入 C ，而 C 在未接到 0001 的內容前，將第二地址 0002 送 E 。

5. C 將 0001 的內容送寄存器 A 。

6. G 按 E 里的 0002 將 0002 的內容送 C 。

7. C 的內容送 B 。

8. J 里的運算碼 00 起作用，使 A 及 B 的內容加到累加器 D 中去進行加法，結果由 D 送入 B 再送入 C ，然後按操作碼（運算碼）的要求，根據 E 內的 0002 把 C 的內容送入 0002 中去。

這時一條指令工作完畢，順次做 I 中所示的地址 0004，看到 37 就停止工作。這時譯碼器 J 中的兩位八進位數即操作碼起了關鍵性的作用，它按本身當時的狀況，指示 A, B, D 對具體的數進行運算，其結果送入存貯器中某一單元，或者保留在 B 里。機器

的八步大概完成了下列工作：

- (1) 取那些单元或寄存器中的内容进行运算。
- (2) 进行怎样的运算或操作。
- (3) 运算结果送存贮器或留寄存器 B 。
- (4) 指出下一次该做的那条指令。

机器的指令系統就要按照这四个任务来规定的，具体指令分类如下：

1. 运算指令：运算(或操作)的种类由 θ_2 表示，有

+	0	加法	} 运算时没有捨入误差
-	1	减法	
:	2	除法	
×	3	乘法	
$\wedge$	6	邏輯乘法	

邏輯乘法是按位的操作，结果的符号是根据 B 或者 A_2 的符号而决定的。它的运算規則如下：

$$0 \wedge 0 = 0, \quad 0 \wedge 1 = 0, \quad 1 \wedge 0 = 0, \quad 1 \wedge 1 = 1.$$

$$[\text{例 2}] \quad 110010100 \wedge 010110000 = 010010000.$$

θ_1 表示存放在何处的数来参加运算和运算的结果放在何处(一般运算的结果保存在寄存器 C 和 B 中)， θ_1 共有如下几种：

$$\left. \begin{array}{ll}
 0\theta_2 & (A_2)\theta_2(A_1) \Rightarrow A_2 \Rightarrow B \\
 1\theta_2 & (A_2)\theta_2(A_1) \Rightarrow B \\
 2\theta_2 & (B)\theta_2(A_1) \Rightarrow A_2 \Rightarrow B \\
 3\theta_2 & (B)\theta_2(A_1) \Rightarrow B \\
 4\theta_2 & (A_2)\theta_2(A_1) \Rightarrow A_2 \Rightarrow B \Rightarrow \Pi \\
 5\theta_2 & |(A_2)|\theta_2|(A_1)| \Rightarrow B \\
 6\theta_2 & (B)\theta_2(A_1) \Rightarrow A_2 \Rightarrow B \Rightarrow \Pi \\
 7\theta_2 & |(B)|\theta_2|(A_1)| \Rightarrow B
 \end{array} \right\} \quad (1-1)$$

这里 (A_i) 表示 A_i 中的内容, (B) 表示 B 中的内容, “ $\Rightarrow$ ” 表示送, “ Π ” 表示印出, A_2 表示单元的地址。由上述运算指令中, 清晰的看到操作碼不同时, 机器的八步也就不同。

现在具体来看指令的形式及其作用:

```

00   $A_1 \ A_2 \ (A_2) + (A_1) \Rightarrow A_2 \Rightarrow B$ 
01   $A_1 \ A_2 \ (A_2) - (A_1) \Rightarrow A_2 \Rightarrow B$ 
02   $A_1 \ A_2 \ (A_2) \div (A_1) \Rightarrow A_2 \Rightarrow B$ 
03   $A_1 \ A_2 \ (A_2) \times (A_1) \Rightarrow A_2 \Rightarrow B$ 
06   $A_1 \ A_2 \ (A_2) \wedge (A_1) \Rightarrow A_2 \Rightarrow B$ 

```

其余的 $\theta_1 \theta_2 \ A_1 \ A_2$ 同样可以列出較詳細的表格, 这些指令工作完毕后, 机器自动取这条指令所在单元地址的下一单元地址工作(参閱机器八步的第三步)。

由(1-1)可得出一个比較简单的記憶方法:

- (1) θ_1 逢双送 A_2 , 逢单送 B 。
- (2) θ_1 为 0, 1, 4, 5 时是 A_1, A_2 参与运算, 2, 3, 6, 7 时是 B, A_1 参与运算。
- (3) 0 与 4 及 2 与 6 作用相同, 但 4, 6 还要印出。
- (4) θ_1 为 5, 7 时是绝对值运算。

现在回过去看看上面的例子, 如果要求把結果印出, 而能复算的話, 則具体程序为:

```

0001  37  0000  0000
→0002  10  0004  0005
0003  64  0001  0006
0004           a
0005           b

```

这里存放 a, b 的单元并没有被破坏, 按 M-3 的结构 64 为 $A_1 \Rightarrow I$, $B \Rightarrow A_2$, 并印出, 37 为停机指令, 停机后启动寄存器 I , I 中出现