Mc Graw Hill
Programming Tools for Scientists & Engineers

●科学家与工程师的编程工具

DISK Included

387、486与Pentium™的数值编程

NUMERICAL PROGRAMMING THE 387, 486, & PENTIUM™

〖美〗Julio Sanchez & Maria P. Canton 著
刘吉峰 李慧军 译
徐 刚 校

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

Numerical Programming the 387, 486, and Pentium™

387、486 与 Pentium™的数值编程

[美] Julio Sanchez & Maria P. Canton 著

刘吉峰　李慧军　译

徐　刚　　校

電子工業出版社

内 容 提 要

本书是一本利用微型计算机求解数值问题的最新最全最实用的工具书，是一本利用Intel 387、486和Pentium™对浮点运算单元进行数值编程的综合教程。书中包括：IEEE754和854标准概述、数值处理单元编程的各个方面、机器算术基础和Intel数学处理单元的演变过程等。对于具有极少计算机数学知识的人本书也同样适用。书中的上百个应用实例，均能由高级语言或低级语言调用，它可以引导读者顺利地完成从数字理论基础到线性方程求解的整个数值处理过程。

Numerical Programming the 387, 486, and Pentium™

Julio Sanchez & Maria P. Canton

387、486与Pentium™的数值编程

刘吉峰　李慧军　译

徐　刚　　校

责任编辑：贾　贺

*

电子工业出版社出版

北京市海淀区万寿路173信箱(100036)

电子工业出版社发行　各地新华书店经销

电子工业出版社计算机排版室排版

中国电影出版社印刷厂印刷

*

开本：787×1092毫米 1/16　印张：31　字数：794千字

1995年10月第一版　1995年10月北京第一次印刷

印数：3000册　定价：66.00元(含磁盘)

ISBN 7-5053-3038-1/TP・1055

著作权合同登记号

图字：01-1995-403

前　言

本书面向各类程序设计人员，从实用的角度出发，介绍了数值编程的原理、算法、设计及实现方法。由于英文 Mathematical Programming 是运筹学领域内的名词，其意思是“数学规划”，所以，本书的英文名字叫做 Numerical Programming，意即数值编程。不管怎样，英文 numerical 一词非常准确地表示了所处理的主要是数值。

为了使具有极少计算机数学背景知识的读者也可使用，本书从最简单的数字系统和数值数据结构开始讲起，然后讨论数学处理单元的初始化和简单编程等，最后介绍如何利用计算机使用高斯消去法对线性系统求解，对公式和方程进行计算和作图。

本书有些章节是介绍基本原理的，尽管这些内容适用于目前所有的计算机环境，但书中关于数值编程的众多实现方法和例子都是以 IBM PC 作为宿主计算机。虽然有两章内容是关于十进制和 BCD 数运算的，但书中大部分内容是关于如何对 Intel 协处理器 8087、80287、80387 及 486 DX、Pentium 内的数学处理单元进行数值编程的。为了运行书中及附盘内的所有程序，读者应拥有一台配有 486DX 或 Pentium 处理器的计算机，安装有 80386 和 80387 这两个处理器的计算机也能运行其中的大部分软件。

Intel 公司的数学协处理器使微型机也具有了以前只有主机(mainframe machines)才拥有的数值处理能力。因此，该书讨论了基于这些器件及其标准的数值编程。由于在理解和使用数学协处理器及运算单元时遇到的许多困难可以说都是因为缺乏一般的背景知识造成的，因此作为一本教程，书中讨论的内容并不局限于 FPU。它还包括如下一些预备知识：计算机数制系统、数值数据的存储，用于二进制运算的数字逻辑和电子线路，十进制数和二进制编码的十进制数(BCD)，ASCII 码，目前的二进制浮点运算标准和与基数无关的浮点运算标准等。

尽管 FPU 功能强大且使用方便，但在一些技术和商业领域(如粒子物理、天文学和应用数学)中仍有一些数值计算问题，这些问题所要求的精度远远超过了数学协处理器所能提供的 18 或 19 位有效数字。因此，本书还包括对十进制数、ASCII 码数的求补运算和将 BCD 数运算的精度扩展到无限位的方法。这些实用程序可用主 CPU 的指令集以软件方式实现，书中及附盘内提供了相应的算法和代码。

除了尽量为 Intel 浮点运算单元的编程提供适当的背景知识外，书中还包括了数学协处理器的发展历史、数值数据格式、FPU 结构及其指令集等内容，同时还考虑了硬件兼容问题。书中考虑的主要问题包括：FPU 的标识和初始化、实用转换程序、超越原语、异常处理、用于一般数值编程的实用程序和过程、线性系统的求解、函数与方程的计算和图示等。

本书附有一张 3 英寸的软盘，盘内含有一个名叫 MATHUNIT.LIB 的数值程序库和一些关于库中过程使用方法的演示程序。考虑到其目的是为了学习和应用，库中的实用程序只实现了基本的处理操作。也就是说，没有对它们进行优化，如错误捕捉、兼容性测试等商业化数值应用程序应拥有的精度与舍入控制等。进一步讲，它们没有经过商品化产品所必须

的严格测试和调试。

在为本书收集资料时曾得到 Intel 公司的帮助，为此我们感谢 Intel 公司协作出版项目经理 Janet Brownstone 和媒体关系部的 Nadine Nunes。

McGraw-Hill 公司的专业文献部与我们合作出版了许多书，对此我们不尽感谢。我们特别要感谢 Gerald T. Papke 主编，他使我们共享在计算机书籍出版方面的丰富经验。作为我们的朋友，Jay Ranade 编辑几年来向我们提出了许多好的建议。还要感谢 David Fogarty，Thomas Kowalczyk，Rechel Hirschfied，Eileen Kramer，以及 McGraw-Hill 分公司的 Cathy Gilligan，并对 Barbara Yanucil 致以特别的感谢。

在写作和研究的过程中，我们不断得到北蒙大拿州立大学的鼓励和支持。我们特别感谢主管业务的副校长 Karen LaRoe 博士，还有 Great Falls 的法律指导 Wes Tucker，部门主管 Virgil Hawkinson 以及我们的同事 Kevin Carlson，Roger Stone，Jay Howland 和 Sharon Lcwman 等。

Julio Sanchez
Maria P. Canton

目　录

第 1 章　计算机与数制

本章概要

对数值数据进行处理和存储是计算机系统的基本用途之一。为了对数值数据进行更有效的操作，计算机科学家及电子设计专家对传统的计算器系统结构作了调整。本章介绍一些必要的背景知识，这些知识有助于理解和使用数字计算机内采用的数制和存储结构。

1.0　计数

计数也许是数字的最初的、最基本的应用。可以想象，在原始社会中猎人可以用手指向其他部落成员说明最近一次捕猎共捕获多少头大象。猎人可用这种方式传递单一的信息，这种信息不是由动物的种类、大小或颜色构成，而是由它们的数目构成的。最原始的计数方法是用一种简单方便的物体表示要计数的对象，如前面所说的猎人是用手指来表示大象的。同样，猎人也可以用小石块、小树枝、在地上划线、在洞中石壁上刻下痕迹等方法来表示物体的数目。这些都与被记数对象的特有属性无关。

1.0.1　计数系统

不难理解，计数系统起源于刻在木棒上的记号或刻在洞中石壁上的刻痕。最简单的计数方式，叫做逐点计数系统，是用一条线或一个刻痕来表示一个物体的。我们目前还偶而使用这种方法来记录物体逐个增加的过程。逐点计数系统的基础是一个记号对应一个物体，猎人通过对每一头隐藏在山谷的大象在木棒上做一个记号的方法来记录个数。后来改为在洞中墙壁上划线的方式以向部落中的其他成员传递数字信息。这种方式不需要有量的概念，也不需要有特殊含义的符号。如在峡谷中有 38 头大象，则在洞中墙壁上可用如下方式表示：

||||||||||||||||||||||||||||||||
||||||||

逐点计数系统的进一步发展是使用一组符号以实现表达方式的可视化。例如，可以用如下分组形式逐个对 38 头大象进行计数：

||||| ||||| ||||| ||||| ||||| |||||
||||| |||

也可使用如下熟悉的方式：

1.0.2 罗马数字

从早期的罗马数字可以看到数值从图形表示到符号表示的演化过程。在早期的罗马数字表示中,符号的位置并不重要,前五个数字的编码方式如下:

I, II,III,IIII 和 V

表 1.1 列出了罗马数字对应的十进制。

表 1.1 罗马记数系统中的符号

罗马数字	十进制
I	1
V	5
X	10
L	50
C	100
D	500
M	1000

罗马计数系统以加 - 减规则为基础,读的时候是从左向右的,其理解方法如下:

1.如果某元素右边的值大于等于其左边的值,则该值加到原来的总值上。

2. 如果该元素右边的值小于其左边的值,则从总值中减去该值。

例如:

III = I + I + I = 3

IV = V - I = 4

XL = L - X = 40

十进制数 1994 可用罗马数字表示如下:

$$\begin{aligned} \text{MCMXCIV} &= \text{M} + (\text{M} - \text{C}) + (\text{C} - \text{X}) + (\text{V} - \text{I}) \\ &= 1000 + (1000 - 100) + (100 - 10) + (5 - 1) \\ &= 1000 - 900 + 90 + 4 \\ &= 1994 \end{aligned}$$

由于每位数字的位置权值不定,以及缺少可以表示零值的符号,因此即使较小的数值也必须要用一个或两个符号(如Ⅰ,Ⅳ,Ⅴ,Ⅵ和Ⅹ)来表示,因此,罗马数字运算的规则比较复杂。

1.1 现行数制的由来

当前采用的计数系统是一种超越所有文化的人类文明,这是一个非常有趣的现象。当人类还在对适于大多数人的政治体制、共同的道德行为准则及共同的语言进行讨论不止时,世界上所有的文化、所有的国家都采用了印度 - 阿拉伯数字。它能如此广泛地被接受,说明它有一些卓越的优点。

目前所用的计数系统由 10 个符号组成,它们是 0,1,2,3,4,5,6,7,8,9。这些符号在大约公元 300 年前起源于印度,其最突出的特点使用了一个特殊符号"0",这个符号表示的量是零。可将这一特殊符号与其他符号混合使用,即当表示某个数值时可重复使用这些符号。十进制系统的另一特点是每位数字所表示的值取决于它在数字串中所处的位置。这种与位置有关的特点加上特殊符号"0"也占一数位,决定了这一数制有如下特征:

1: 代表一
10: 代表十
100: 代表一百
1000: 代表一千

在公元 800 年左右,阿拉伯人就已经采用了十个符号的与位置有关的计数系统,其中就使用了特殊符号"0"。这种数制(后来就叫印度 - 阿拉伯计数系统)在八世纪被介绍到欧洲,可能是通过西班牙。在西班牙研究印度 - 阿拉伯数字的 Pope Sylvester II 是第一位采用和教授这种计数系统的欧洲学者。第一本关于此计数系统的书的拉丁名字是 *Liber de Numero Indorum*,其作者是阿拉伯数学家 al - Khowarizmi。

尽管这种计数系统具有很多明显的优点,但还是经过许多激烈的讨论之后才在欧洲被采用。那时许多学者仍认为罗马数字最易学习、最适于在算盘上进行运算。罗马数字的支持者与印度 - 阿拉伯数字的支持者之间的争论由 al - Khowarizmi 作了记录。由于 Catholic 主教经常站在支持采用罗马数字人的一边,双方争论了几个世纪。直到十六世纪印度 - 阿拉伯计数系统才在欧洲被普遍采用。

人们常说印度 - 阿拉伯数字有十个符号是因为人有十个手指。但如果将这十个符号与手指一一对应就会发现有一个手指对应于"10",它由两个符号组成,而符号"0"不与任何手指对应。事实上,正是由于它采用了零数字且与位置有关,因此可以说十进制计数系统是人类智慧的结晶。我们不能确定印度 - 阿拉伯计数系统与我们有十个手指是否有关系,但此数制的实用性及其表示的深刻性明显地超出了这一生理现象。

十进制的基本特点表现在它的计数方式,每个符号所代表的值由其所处的位置决定。我们已经看到这种位置特征是以特殊符号"0"的使用为基础的,这一符号本身不代表任何数值量,但它在多位数值的表示中可占用一位。人们一定会对这种深度抽象的表达能力惊叹不已。很难想象如果没有发明这种计数系统,数学、科学及技术的发展将会怎样。另一方面,按照我们的数学思维方法,一个含有特殊符号"0"的具有位置特征的计数系统是否是自然要产生的,或者是否说该计数系统的发明是人类智能的一个飞跃,而这个飞跃在其后的

2000 年中被人类忽视了。

1.1.1 数字电子计算机使用的数制

早在 1940 年,美国生产的计算机是对十进制数进行运算的。但在 1946 年, John Von Neumann、Burks 和 Goldstine 发表了一篇名为"Preliminary Discussion of the Logical Design of an Electronic Computing Instrument"的论文,他们指出:

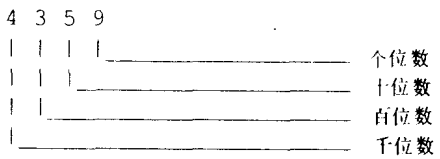
"在谈到计算机的计算部件时,人们自然会考虑要采用哪种数制。尽管以往都认为应该建立十进制的数字计算机,但我们强烈地感到,在我们的计算机部件中应采用二进制"。

尽管在这篇论文中作者认为计算部件应该采用 BCD 数值,但还是采用了纯二进制编码,这是因为二进制数值要比 BCD 紧凑。

数字计算机中采用的是二进制,对该数制的研究始于 1703 年 G. W. Leibnitz 在巴黎发表的一篇论文。数学研究说明可使用两个符号来进行计数和算术运算。许多性能可靠的电子元件具有两个稳定的状态,可利用二进制计数系统将这两个状态编码为"0"和"1"。一种对应方法是 1 对应电子元件的开"ON", 0 对应电子元件的关"OFF"。二进制的这两个符号也可用来表示导通或不导通、正或负以及其他的二值情况。600 年来印度 - 阿拉伯的数字首次遇到二进制数字的挑战,这是由于具有两个稳定状态的数字电子器件更易于实现。

1.1.2 位置特征

目前使用的所有数制,包括十进制、十六进制及二进制都具有位置特征。利用这一特征可以确定多位数的值。例如,十进制数 4359 中各位的权如下所示:



通过将各数字位代表的值相加可求出总值,如下所示:

```
4000 ———— 4 千位
 300 ———— 3 百位
  50 ———— 5 十位
   9 ———— 9 个位
-------------------
4359
```

也可用如下方式求得:

$$4 * 10^3 + 3 * 10^2 + 5 * 10^1 + 9 * 10^0 = 4359$$

注意:$10^1 = 10, 10^0 = 1$。

1.1.3 基数

在任何具有位置特征的数制中,各数字位代表的值是由该数制中符号的个数(包括特殊

符号 0)来决定的。数制中所用的符号个数叫做该数制的基数。因此,十进制的基数是 10,二进制的基数是 2,十六进制的基数是 16。如果一个数的位数大于 1,则该数表示的值可用如下方式求得:

$$P = d * B^c$$

其中 d 是某位的值,B 是基数,c 是该数字位的位序号(从零开始自右向左数)。需要注意的是各数字位从右向左权值依次增大纯粹是约定,当然也可作相反的约定,事实上最初印度人表示数值时是把最重要的数字位放在最右边的。

一个十进制数可通过如下公式表示为各数字位值与其相应权之积的和:

$$\sum_{i=-m}^{n} d_i * 10^i \qquad 0 \leqslant d_i \leqslant 9 \quad (d_i \text{ 为整数})$$

同样,二进制数也可作如下表示:

$$\sum_{i=-m}^{n} b_i * 2^i \qquad b_i = 0 \text{ 或 } b_i = 1$$

1.2 数据类型

由于不同类型的数据具有不同的表示方法,其作用也不仅限于简单的计数功能。

1.2.1 整数

整数是用来表示事物个数的有序符号集。所谓有序符号集意思是为它们指定了顺序。我们已经提到具有位置特性的数制要求有一个特殊的数字"0",它本身不表示任何量,但与其他数字混合使用时却起着重要作用,比如,10 或 30。整数是包括 0 的全体自然数的集合。

1.2.2 带符号整数

数制还可以有方向。常使用"+"和"-"来表示相反的数值方向。带符号数的典型表示方法如下:

-9 -8 -7 -6 -5 -4 -3 -2 -1 0 +1 +2 +3 +4 +5 +6 +7 +8 +9

负数 ←零→ 正数

注意将正数和负数分开的数值 0,其本身是没有符号的。

1.2.3 有理数与无理数

数制还可用来表示整体的一部分。例如,当木匠将一木板分成等长度的两半时,可以用分数 1/2 表示其中的一半。1/2 说明构成物体的两部分中的一部分。有理数是所有可以表