TURING 图灵数学·统计学丛书 41

PEARSON

Numerical Analysis

数值分析

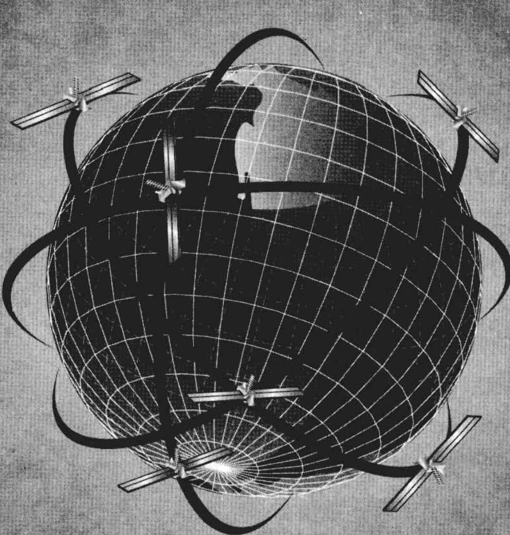
[美] Timothy Sauer 著

吴兆金 王国英 范红军 译

人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵数学·统计学丛书41

PEARSON



Numerical Analysis

数值分析

[美] Timothy Sa

吴兆金 王国英 范红军 译

人民邮电出版社

北京

图书在版编目（C I P）数据

数值分析 / (美) 索尔 (Sauer, T.) 著 ; 吴兆金, 王国英, 范红军译. -- 北京 : 人民邮电出版社, 2010.1

（图灵数学·统计学丛书）
书名原文: Numerical Analysis
ISBN 978-7-115-21759-2

Ⅰ. ①数… Ⅱ. ①索… ②吴… ③王… ④范… Ⅲ. ①数值计算 Ⅳ. ①O241

中国版本图书馆CIP数据核字(2009)第206594号

内 容 提 要

本书以收敛性、复杂性、条件作用、压缩和正交性这 5 个主要思想为核心进行展开. 内容包括求解方程组、插值、最小二乘、数值微分、数值积分、微分方程及边值问题、随机数及其应用、三角插值、压缩、最优化等. 每章都有一个实例检验, 有助于读者了解到相关应用领域. 附录中介绍了矩阵代数和 MATLAB, 并提供了部分习题的答案.

本书内容广泛, 实例丰富, 可作为自然科学、工程技术、计算机科学、数学、金融等专业人员进行教学和研究的参考书.

图灵数学·统计学丛书

数值分析

◆ 著　[美] Timothy Sauer
译　吴兆金 王国英 范红军
责任编辑　明永玲

◆ 人民邮电出版社出版发行　北京市崇文区夕照寺街 14 号
邮编 100061　电子函件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
北京铭成印刷有限公司印刷

◆ 开本：700×1000　1/16
印张：36.5
字数：736 千字　2010 年 1 月第 1 版
印数：1–3 000 册　2010 年 1 月北京第 1 次印刷

著作权合同登记号　图字：01-2006-5779 号

ISBN 978-7-115-21759-2

定价：79.00 元

读者服务热线：(010)51095186　印装质量热线：(010) 67129223

反盗版热线：(010)67171154

版权声明

译　者　序

计算机是 20 世纪以来对人类社会影响最为深刻的高新科技成果之一，而科学计算已成为当今科学研究中与理论分析和实验研究并列的三种基本手段之一. 科学计算是数学与计算机的有机结合，而且它本身也成为数学科学自身发展的源泉和途径之一. 数值分析及其有关内容在培养学生科学计算能力上具有不可替代的作用，目前许多高校已将数值分析列入自然科学、工程技术乃至社会科学的教学计划中.

由于数值分析涉及范围甚广，应用于诸多的领域，为使学习者打下较为坚实的理论基础，了解和使用相关数学软件，并在此基础上设计和编写自己的算法及程序以解决各类实际应用问题，选择一本好的教材自然是重要的. 国外最新出版的由乔治・梅森大学 Timothy Sauer 教授编著的这本《数值分析》极富特色，出版不久即广受好评.

该书内容涵盖非常全面，其中关于边界值、随机数值分析、三角插值、压缩、特征值、优化等方面的内容是其他教材不多见的. 此外，作者把许多看似关联不大的技术融合在一本书中，并强调：不仅仅要学会如何使用 Newton、Runge-Kutta、快速 Fourier 变换等方法，还要吸收那些渗透在数值分析中并将各种不同的方法统一起来的主要思想.

本书集中讨论了收敛性、复杂性、条件作用、压缩和正交性的概念及其作用，并以此分析、评论其他相关的论题，作者以这 5 个最重要的概念为框架展开内容，不拘泥于形式的讨论，向读者传达了什么是数值分析理论中真正至关重要的主题.

作者还介绍了诸如后向误差分析、稀疏矩阵计算及信号处理等概念，融合了计算机、电子、金融等各领域最新的应用，给出了大量的实例和图片，内容生动新颖，实用性强. 实例检验提供了数值分析在各个学科中最新的应用，与 MATLAB 的紧密联系可容易地实现制图，并有效地解决工业规模化问题.

本书结构清晰，条理分明，理论描述精当，实例范围广泛. 除经典问题外，还涉及许多最新的前沿课题. 每个章节还提供了大量的概念性及计算性的练习 (习题与计算机问题)，帮助读者理解、消化、复习和巩固所学知识，并可使读者在学会解决各类问题方面逐步积累起经验. 适合于大学生、研究生以及相关人员学习和参考.

本书由吴兆金、王国英和范红军翻译，其中第 0~1 章、第 3~4 章、第 10 章及索引由吴兆金翻译，第 5~8 章、第 12 章、习题选解及附录 A 由王国英翻译，第 2 章、第 9 章、第 11 章、第 13 章及附录 B 由范红军翻译，全书由吴兆金统稿. 本书

的翻译工作得到人民邮电出版社图灵公司领导和编辑的支持和帮助, 对此向他们表示衷心的感谢.

在本书的翻译过程中, 我们力求忠实、准确地反映原著的内容和风格. 鉴于我们水平所限, 翻译错误及不妥之处在所难免, 恳请读者批评、指正.

译　者
2008 年 3 月于南京大学

译者简介

吴兆金　男, 数学编辑, 副编审. 1978 年毕业于南京大学数学系计算数学专业, 留校任教. 曾任《高等学校计算数学学报》编辑, 现为《Analysis in Theory and Applications》和《南京大学学报数学半年刊》编辑. 已参与出版教材 1 部, 发表学术论文数篇.

王国英　男, 1944 年 11 月生, 1967 年南京大学数学系毕业, 1982 年在南京大学数学系获硕士学位. 现为南京大学数学系信息与计算科学专业的教授, 研究方向是偏微分方程数值解法和奇异摄动问题的数值方法. 曾在核心一级刊物发表研究论文 20 余篇, 已出版译著《数值分析》(全美经典) 和教材《工程数学》.

范红军　男, 1966 年 9 月出生, 1989 年毕业于南京大学数学系并留校任教至今, 现为副教授. 曾编著出版《高等数学》和《大学数学典型题解析》等. 现主要从事数值代数方面的研究.

前　言

本书是为工程、科技、数学和计算机科学等专业的学生而写的入门教科书, 其目的十分明确: 描述解决科技和工程问题的算法以及讨论算法所需的数学基础, 期望适用于具有初等微积分和矩阵代数基础的学生的主修课程.

作为一门学科, 数值分析的内容极为丰富, 饱含实用思路, 要把很多灵巧但又关联不大的技术用一本书来概括是非常具有挑战性的. 要深入理解, 读者不仅必须学会如何对 Newton 方法、Runge-Kutta 方法与快速 Fourier 变换进行编程, 而且必须吸收那些渗透在数值分析中、把其他相关内容统一起来的伟大思想.

收敛性、复杂性、适用条件、压缩以及正交性的概念是这些思想中最重要的. 任何合适的逼近方法都必须收敛到正确的答案, 尤其是有更多计算资源提供给它时更当如此, 并且计算方法的复杂性也是由资源利用来衡量的. 一个问题的适用条件, 或者对误差放大率的敏感程度, 是了解如何求解问题的基础. 在数值分析的许多最新应用中, 目标是用更短或更浓缩的方式来表示数据. 最后, 正交性在若干领域中对效率的影响是决定性的, 并且在要考虑适用条件或者以压缩性为目标时, 它是不可替代的.

通过称为 "亮点"(Spotlight) 的主题元素, 我们强调了现代数值分析中这 5 个概念的作用. 它们评论当前的论题, 并且联系到书中其他地方出现的相同概念的其他描述. 同时, 我们希望用这种明显的方式突出这 5 个概念, 能够强调当前页面的重点知识, 起到点题之功效.

虽然公认数值分析的思想对现代科技与工程的实践来说是必需的, 但仍需不断强化这一理念. "实例检验" 就给出一些用数值分析方法解决科技问题的具体例子. 这些扩充的应用应时而选并贴近日常的经验. 虽然实例检验不可能 (甚至是不要求) 表现问题的全部细节, 但它试图从一定深度去揭示一种技术或算法可以利用少量的数学知识就在科技的设计中获得巨大的回报.

在本书中, MATLAB 既用于算法说明又用于学生作业和课题的建议平台. 本书中 MATLAB 代码的数量是经过仔细调控的, 这是因为太多的代码会适得其反. 前几章的 MATLAB 代码多一些, 可以使读者逐步熟悉程序. 当提供更加详细的代码时 (比如说, 在插值、常微分和偏微分方程的学习中), 期望读者以此作为探索和拓展的起点.

尽管本书不是非要用到一种计算平台不可, 但 MATLAB 在工程和科技部门的

广泛应用表明, 一种通用的计算机语言可以让很多工作更加顺利. 借用 MATLAB, 所有的界面问题 (数据的输入/输出与绘图等) 都能被一举解决. 数据结构问题 (比如, 研究稀疏矩阵方法时产生的问题) 可以利用适当的命令进行标准化. MATLAB 有专门设施来处理声音和图像文件的输入和输出. 微分方程的模拟是容易实现的, 因为 MATLAB 中有相应的动画命令. 这些目标虽说也可以用其他方式去实现, 但是拥有一个可以运行在几乎所有的操作系统上并简化了细节的软件包, 以便学生能集中精力处理实际的数学问题, 何乐而不为呢? 附录 B 是简短的 MATLAB 指南, 它可以用作学生的 MATLAB 入门介绍或作为已经熟悉该软件的读者的一个参考.

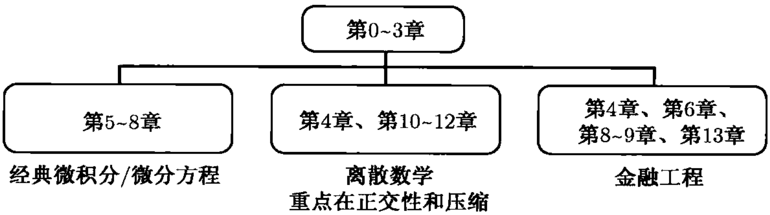
本书附 CD[1]一张, 内容包括直接取自书中的 MATLAB 程序. 这些程序在网站 www.aw-bc.com/sauer 上也可以找到, 该站点上还有新的资源和更新供使用者下载.

本书的结构是先介绍基本思想, 然后描述更复杂的概念. 第 0 章给出了后面要用到的基本知识. 一些教师喜欢从头讲起; 另一些教师 (包括作者本人) 喜欢从第 1 章开始并在需要时折回到第 0 章的一些主题. 第 1 章与第 2 章给出了解方程的各种形式. 第 3 章讲用插值法处理数据拟合, 第 4 章介绍最小二乘法拟合. 在接下来的第 5~8 章, 我们回到连续数学中的经典数值分析领域: 数值微积分, 带初值与边值条件的常微分方程与偏微分方程的数值解.

第 9 章研究随机数, 以便为第 5~8 章提供补充的方法: 可替换标准数值积分格式的蒙特卡罗法以及随机微分方程, 当模型中出现不确定性时这些方法是必要的.

压缩是数值分析的核心主题, 尽管它经常隐藏在插值、最小二乘与 Fourier 分析中. 第 10 章和第 11 章特别讲述了现代压缩技术. 在第 10 章中, 快速 Fourier 变换是作为进行三角插值的一种工具, 包括精确插值和最小二乘近似. 第 11 章介绍离散余弦变换与 Hoffman 编码, 重点并充分讨论了音频压缩的诸多联系, 这两种方法也是现代音频与图像压缩的标准工具. 第 12 章讲述特征值与奇异值, 也强调它们与数据压缩之间的联系, 这在当前的应用中变得越来越重要. 第 13 章是关于优化技术的简短介绍.

如果审慎地选讲一些主题, 本书也可用于一学期的课程. 第 0~3 章是必备的基础. 作为一学期授课的话, 可做如下设计:



① 中文版将原英文书所附 CD 的内容放在了图灵网站 (www.turingbook.com) 上供读者免费注册下载. ——编者注

特别要感谢在撰写本书过程中给予我帮助的许多人, 包括使用本书前几个版本并提出有用建议的学生们. 感谢以下人员, 他们让我避免了一些难堪的错误: Frank Purcell, Paul Lorczak, Steve Whalen, Diana Watson, Joan Saniuk, Robert Sachs, David Walnut, Stephen Saperstone, Tom Wegleitner, Tjalling Ypma. 感谢友好、机智的 Addison Wesley 员工, 包括 William Hoffman, Joanne Ha, Peggy McMahon, Joe Vetere, Emily Portwood, Barbara Atkinson, Beth Anderson 以及在 Westwords 的 Melena Fenn, 正是他们使得本书得以出版. 另外, 还要感谢来自其他大学的下面这些读者, 他们给予我鼓励, 并对改进早期版本提出了宝贵的建议.

Eugene Allgower, 科罗拉多州立大学
Jerry Bona, 伊利诺伊大学芝加哥分校
George Davis, 佐治亚州立大学
Alberto Delgado, 布莱德利大学
Robert Dillon, 华盛顿州立大学
Gregory Goeckel, 长老会学院
Herman Gollwitzer, 德雷塞尔大学
Don Hardcastle, 贝勒大学
David R. Hill, 坦普尔大学
Daniel Kaplan, 麦卡利斯特学院
Lucia M. Kimball, 本特利学院
Seppo Korpela, 俄亥俄州立大学
William Layton, 匹兹堡大学
Doron Levy, 斯坦福大学
Shankar Mahalingam, 加州大学河滨分校
Amnon Meir, 奥本大学
Peter Monk, 特拉华大学
Joseph E. Pasciak, 得克萨斯 A&M 大学
Steven Pav, 加州大学圣地亚哥分校
Jacek Polewczak, 加州州立大学
Jorge Rebaza, 西南密苏里州立大学
Jeffrey Scroggs, 北卡罗来纳州立大学
Sergei Suslov, 亚利桑那州立大学
Daniel Szyld, 坦普尔大学
Ahlam Tannouri, 摩根州立大学
Bruno Welfert, 亚利桑那州立大学

T.S.

目　录

第0章 基　础

本书的主要目的是介绍和讨论用计算机解数学问题的方法. 最基本的算术运算是加法和乘法. 它们也是计算多项式 $P(x)$ 在特定点 x 处的值所必需的运算. 多项式是我们将要构造的许多计算技术的基石, 这并非巧合.

因此, 了解如何计算多项式是很重要的. 读者可能已经知道如何计算多项式, 并且可能会认为对这种容易的问题花时间近乎荒谬. 但是越是基本的运算, 我们越要把它做得正确. 因此, 我们将考虑如何尽可能高效地进行多项式计算.

0.1　多项式计算

计算在 $x=\frac{1}{2}$ 处的多项式 $P(x)=2x^4+3x^3-3x^2+5x-1$, 最好的方法是什么? 假设多项式的系数和数 $\frac{1}{2}$ 存储在存储器里, 设法使求 $P(\frac{1}{2})$ 所需要的加法和乘法的次数最少. 为了简单, 我们将不考虑把数从存储器存入和取出所花的时间.

方法 1　首先最直接的方法是

$$P\left(\frac{1}{2}\right)=2*\frac{1}{2}*\frac{1}{2}*\frac{1}{2}*\frac{1}{2}+3*\frac{1}{2}*\frac{1}{2}*\frac{1}{2}-3*\frac{1}{2}*\frac{1}{2}+5*\frac{1}{2}-1=\frac{5}{4}. \tag{0.1}$$

它需要 10 次乘法和 4 次加法. 其中两次加法实际上是减法, 但是, 因为减法可以看作是加上一个负的存储数, 所以我们不必担心这种区别.

当然还有比 (0.1) 更好的方法. 工作量是重复的, 通过消除对 1/2 的重复相乘, 可以减少一些运算. 更好的策略是: 首先计算 $(\frac{1}{2})^4$, 同时存储计算过程中的部分积. 这样就导出了以下方法.

方法 2　首先求出输入数 $x=\frac{1}{2}$ 的各次幂, 并把它们存储起来备用:

$$\frac{1}{2}*\frac{1}{2}=\left(\frac{1}{2}\right)^2,\quad \left(\frac{1}{2}\right)^2*\frac{1}{2}=\left(\frac{1}{2}\right)^3,\quad \left(\frac{1}{2}\right)^3*\frac{1}{2}=\left(\frac{1}{2}\right)^4.$$

现在我们就可以把这些项加起来:

$$P\left(\frac{1}{2}\right)=2*\left(\frac{1}{2}\right)^4+3*\left(\frac{1}{2}\right)^3-3*\left(\frac{1}{2}\right)^2+5*\frac{1}{2}-1=\frac{5}{4}.$$

上式中有 3 个 $\frac{1}{2}$ 的乘积以及 4 个相关乘积. 总计我们已减少至 7 次乘法以及 4 次加法. 将运算次数从 14 次减少到 11 次是否是重大的改进? 如果仅是要做一次计

算, 则答案也许是否定的. 不论使用方法 1 或者方法 2, 在你的手指离开计算机键盘之前就可得到答案. 然而, 假如每秒钟需要多项式对不同的输入 x 进行几次计算, 那么能否及时地得到信息, 其间的差别就可能是关键的.

对于 4 次多项式来说, 第 2 种方法是否做得最好呢? 可能很难想象我们还能再减少 3 次运算, 但是我们的确能够做到. 最好的初等方法如下.

方法 3(嵌套乘法) 把多项式改写为下面的形式以便能依括号从内到外进行计算:

$$\begin{aligned} P(x) &= -1 + x(5 - 3x + 3x^2 + 2x^3) = -1 + x(5 + x(-3 + 3x + 2x^2)) \\ &= -1 + x(5 + x(-3 + x(3 + 2x))) = -1 + x * (5 + x * (-3 + x * (3 + x * 2))), \end{aligned} \tag{0.2}$$

这里的多项式是倒过来写的, 而且 x 的乘幂是作为多项式余下部分的因子. 一旦你能看懂这种写法, 就会明白这种改写并不需要计算——系数没有改变. 现在我们从里往外进行计算:

$$\begin{array}{llll} \text{乘}\ \frac{1}{2} * 2, & \text{加}\ +3 \to 4; & \text{乘}\ \frac{1}{2} * 4, & \text{加}\ -3 \to -1; \\ \text{乘}\ \frac{1}{2} * (-1), & \text{加}\ +5 \to \frac{9}{2}; & \text{乘}\ \frac{1}{2} * \frac{9}{2}, & \text{加}\ -1 \to \frac{5}{4}. \end{array} \tag{0.3}$$

这种方法称为**嵌套乘法**(nested multiplication) 或 **Horner方法**. 计算该多项式仅用了 4 次乘法和 4 次加法. 通常一个 d 次多项式能用 d 次乘法和 d 次加法进行计算. 嵌套乘法与多项式运算的综合除法密切相关.

这个多项式计算的例子具有科学计算中计算方法所研究的所有问题的特征. 首先, 计算机能够十分迅速地进行非常简单的工作. 其次, 重要的是即使对简单的任务也要尽可能提高工作效率, 因为它们可能要执行许多次. 第三, 最好的方法可能不是显而易见的. 在过去半个多世纪里, 数值分析和科学计算领域与计算机硬件技术密切相关, 对于一些常规的问题已经开发出有效的解题技术.

虽然多项式 $c_1 + c_2x + c_3x^2 + c_4x^3 + c_5x^4$ 的标准形式能写成

$$c_1 + x(c_2 + x(c_3 + x(c_4 + x(c_5)))) \tag{0.4}$$

这种嵌套形式, 但是某些应用要求更一般的形式. 特别地, 第 3 章里的插值计算将需要

$$c_1 + (x - r_1)(c_2 + (x - r_2)(c_3 + (x - r_3)(c_4 + (x - r_4)(c_5)))) \tag{0.5}$$

这种形式, 这里 r_1, r_2, r_3, r_4 称为**基点**(base point). 注意, 在式 (0.5) 中取 $r_1 = r_2 = r_3 = r_4 = 0$ 便恢复到原来的嵌套形式 (0.4).

以下 MATLAB 代码提供了嵌套乘法的一般形式 (与式 (0.3) 比较):

```
%Program 0.1 Nested multiplication
%Evaluates polynomial from nested form using Horner's Method
%Input: degree d of polynomial,
%       array of d+1 coefficients c (constant term first),
%       x-coordinate x at which to evaluate, and
%       array of d base points b, if needed
%Output: value y of polynomial at x
function y=nest(d,c,x,b) if nargin<4, b=zeros(d,1); end y=c(d+1);
for i=d:-1:1
  y = y.*(x-b(i))+c(i);
end
```

运行这个 MATLAB 函数只是置换包括次数、系数、求值点及基点等输入数据. 例如可以用 MATLAB 命令

```
>> nest(4,[-1  5  -3  3  2],1/2,[0  0  0  0])
ans=
     1.2500
```

来计算多项式 (0.2) 在 $x=\frac{1}{2}$ 处的值, 就像我们以前用手算求得的一样. 在执行指令时必须经由 MATLAB 路径 (或在当前目录中) 使用文件`nest.m`. 本书中给出的其余 MATLAB 代码的使用方法与此相同.

若`nest`指令用于如 (0.2) 中所有基点为 0 的情形, 那么使用其简化形式

```
>> nest(4,[-1  5  -3  3  2],1/2)
```

可以得到同样的结果. 这是由于`nest.m`中的`nargin`语句. 假如输入参数的数量少于 4, 那么就自动将基点设为 0.

由于 MATLAB 中向量记法的无缝处理, 这种`nest`指令可以立即对 x 的一组数值进行计算. 以下代码便可说明这一点:

```
>> nest(4,[-1  5  -3  3  2],[-2  -1  0  1  2])
ans=
     -15  -10  -1  6  53
```

最后, 第 3 章中的 3 次插值多项式

$$P(x)=1+x\left(\frac{1}{2}+(x-2)\left(\frac{1}{2}+(x-3)\left(-\frac{1}{2}\right)\right)\right)$$

有基点 $r_1=0, r_2=2, r_3=3$, 可以通过以下代码计算出它在 $x=1$ 处的值:

```
>>n est(3[1,1/2  1/2  -1/2],1,[0  2  3])
ans=
    0
```

例 0.1 找出一种高效的方法来计算多项式

$$P(x)=4x^5+7x^8-3x^{11}+2x^{14}.$$

改写多项式可以帮助减少所需要的计算次数. 一种想法是从各项中提出因子 x^5, 并把其余部分写成 x^3 的多项式

$$P(x)=x^5(4+7x^3-3x^6+2x^9)=x^5*(4+x^3*(7+x^3*(-3+x^3*(2)))).$$

首先, 对每一个输入 x, 我们需要计算 $x*x=x^2, x*x^2=x^3$ 以及 $x^2*x^3=x^5$. 这 3 次乘法连同与 x^5 的乘法, 再加上关于 x^3 的 3 次多项式的 3 次乘法和 3 次加法, 就给出了: 每次计算总共需要 7 次乘法和 3 次加法运算. ◀

习题 0.1

1. 改写下列多项式为嵌套形式. 在 $x=\frac{1}{3}$ 时, 分别用嵌套形式和不用嵌套形式进行计算:
   (a) $P(x)=6x^4+x^3+5x^2+x+1$;
   (b) $P(x)=-3x^4+4x^3+5x^2-5x+1$;
   (c) $P(x)=2x^4+x^3-x^2+1$.
2. 改写下列多项式为嵌套形式, 并在 $x=-\frac{1}{2}$ 时计算:
   (a) $P(x)=6x^3-2x^2-3x+7$;
   (b) $P(x)=8x^5-x^4-3x^3+x^2-3x+1$;
   (c) $P(x)=4x^6-2x^4-2x+4$.
3. 把 $P(x)$ 看成 x^2 的多项式, 并用嵌套乘法计算当 $x=\frac{1}{2}$ 时 $P(x)=x^6-4x^4+2x^2+1$ 的值.
4. 计算带基点的嵌套多项式 $P(x)=1+x\left(\frac{1}{2}+(x-2)\left(\frac{1}{2}+(x-3)\left(-\frac{1}{2}\right)\right)\right)$ 在 (a)$x=5$ 及 (b)$x=-1$ 处的值.
5. 计算带基点的嵌套多项式 $P(x)=4+x(4+(x-1)(1+(x-2)(3+(x-3)(2))))$ 在 (a)$a=\frac{1}{2}$ 及 (b)$x=-\frac{1}{2}$ 处的值.
6. 说明在给定的输入 x 处, 如何用尽可能少的运算计算多项式. 需要多少次乘法和加法?
   (a) $P(x)=a_0+a_5x^5+a_{10}x^{10}+a_{15}x^{15}$;
   (b) $P(x)=a_7x^7+a_{12}x^{12}+a_{17}x^{17}+a_{22}x^{22}+a_{27}x^{27}$.
7. 用一般的嵌套乘法算法, 计算带基点的 n 次多项式需要多少次加法和乘法?

计算机问题 0.1

1. 用函数`nest`计算 $P(x)=1+x+\cdots+x^{50}$ 在 $x=1.000\,01$ 处的值. (用 MATLAB 的`ones`命令省去录入.) 通过与等价表达式 $Q(x)=(x^{51}-1)/(x-1)$ 比较, 求计算误差.
2. 用`nest.m`计算 $P(x)=1-x+x^2-x^3+\cdots+x^{98}-x^{99}$ 在 $x=1.000\,01$ 处的值. 找出一个更简单的等价表达式, 并用它来估计嵌套乘法的误差.

0.2 二 进 制 数

为了在 0.3 节中深入研究计算机算术, 我们需要了解二进制数系. 为了在计算机上存储数并且简化如加法和乘法这样的计算机运算, 我们把十进制数从以 10 为