21世纪高等院校教材

数值分析与实验

韩旭里　万　中　编著

科学出版社
www.sciencep.com

21世纪高等院校教材

数值分析与实验

韩旭里　万　中　编著

科学出版社

北京

内 容 简 介

本书阐述了科学与工程计算中的基本理论和方法，包括数值分析中的基本概念和基础知识、插值法、数值积分与数值微分、函数的最佳逼近、线性方程组的直接解法、解方程和方程组的迭代法、矩阵特征值与特征向量的计算、常微分方程的数值解法、偏微分方程的数值解法、MATLAB 平台上的数值实验等. 书中有丰富的例题、练习题和数值实验习题. 本书注重内容的实用性，强调数值方法的思想和原理以及在计算机上的实现. 选材深浅适度、系统性强，文字通俗易懂.

本书可作为高等学校理工科专业本科生、研究生数值分析课程的教材或教学参考书，也可供从事科学与工程计算的科技人员学习参考.

图书在版编目(CIP)数据

数值分析与实验/韩旭里，万中编著. —北京：科学出版社，2006

(21 世纪高等院校教材)

ISBN 7-03-017092-X

I. 数… II. ①韩… ②万… III. 计算机辅助计算：数值计算-高等学校-教材 IV. O241

中国版本图书馆 CIP 数据核字(2006)第 028792 号

责任编辑：李鹏奇 王 静／责任校对：钟 洋

责任印制：张克忠／封面设计：陈 敬

科学出版社 出版

北京东黄城根北街16号

邮政编码:100717

http://www.sciencep.com

双青印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2006 年 7 月第 一 版 开本：B5（720 × 1000）

2006 年 7 月第一次印刷 印张：17

印数：1—3 500 字数：320 000

定价：26.00 元

（如有印装质量问题，我社负责调换（双青））

前　言

数值分析是高等学校理工科专业本科生和研究生的重要数学基础课程. 计算科学与技术的发展对这门课程的教学提出了新的要求. 为了更好地适应数值分析课程的教学, 我们根据国内外大学数值分析课程教材的发展变化情况, 总结多年来的教学实践经验, 编写了这本旨在加强实用性和应用性的教材. 概括地说, 本书在以下几个方面做了新的尝试.

1. 本书对所精选的内容作尽量详细的论述和推导, 做到通俗易懂, 深入浅出, 例题翔实. 这是因为目前在许多高校, 数值分析课程的教学课时数都有不同程度的减少, 教师很难在课堂上完成大量数值分析经典内容的教学, 我们希望在教师有侧重性的引导下, 让学生树立起学好这门课程的信心和兴趣, 并取得成效. 我们坚信, 通过各种途径培养学生学好数学的信心, 是全面提高教学质量的前提条件.

2. 对部分结论仅作叙述, 删除了繁杂的证明过程. 我们这样做主要是想在数值分析这门课程中更加强调算法的构造思想、算法的具体构造过程、算法的评价和改进, 以及算法的具体执行等, 提高学生应用计算机进行科学与工程计算的能力.

3. 结合数值分析课程的大部分内容, 增加了该门课程的数值实验教学内容. 这主要是基于下面两点考虑：一方面因为数值分析的许多内容在理论上和实践中都已经非常成熟, 许多算法都已经被开发成了专门的数学软件包 —— 它们具有强大的数值计算功能, 易学且具有开放性; 另一方面, 对不少学生或工程技术人员来说, 数值分析这门课程他们最关心的是构造各种算法的思想和如何运用算法直接解决实际问题, 他们中的大多数没有时间或者没有兴趣了解数值分析这门课程的细节. 所以, 一种好的方式就是在他们了解这门课程的基本思想的基础上, 直接让他们掌握运用现有数学软件解决实际问题的方法. 我们把这种实践数值分析课程内容的过程, 叫做数值分析课程实验或数值实验, 这也是本教材名称的来由.

此外, 本教材还吸收了同类教材的优点, 在编排体系和内容结构上尽量做到紧凑和循序渐进. 比如说, 我们把数值分析课程中要用到的最基础的知识, 如范数、向量的内积和正交概念等内容全部放到第 1 章集中介绍; 把求解方程和方程组的迭代法集中到一章介绍, 使得在内容的叙述上显得更加浑然一体; 把系列圆盘定理作为矩阵特征值的粗略估计方法单独一节叙述, 等等.

在本书的编写与出版过程中, 中南大学研究生院、湖南省教育厅和科学出版社给予了很大的支持和帮助, 在此我们表示衷心感谢. 由于教材编写中体现了新的尝试, 难免有不足和纰漏之处, 望各位读者不吝指正.

编　者

2006 年 1 月于长沙麓山

目　录

第 1 章　数值分析中基本概念和基础知识

数值分析这门课程关心的问题是如何设计有效的方法, 近似计算某个数学问题或数学模型的解. 方法的有效性取决于两个方面, 一是实际问题要求的近似解的精度能否得到满足, 二是这个方法在计算机上是否容易执行.

由于我们所研究的数学问题或数学模型一般都是工程实际问题通过一定的简化、假设得到的数学表达式, 所以该数学问题实际上是所要解决的实际问题的一个近似问题. 从这个意义上说, 设计寻找数学问题近似解的方法要比计算它的精确解更合适, 它能起到 "歪打正着" 的作用. 通常, 我们称这样一种寻找近似解的方法叫做算法, 它一般由一组有顺序的代数运算和逻辑运算构成, 目的是要计算满足指定精度的近似解.

因为方法的有效性要看它在计算机上是否容易执行, 所以合适的方法是与实际计算时采用的计算机性能及其不断发展相匹配的.

与算法紧密联系的重要问题是如何估算近似解的误差界, 即分析这个算法得到的近似解的误差不会超过多少, 这部分内容叫做误差分析.

在此后的每个章节, 我们都将围绕上述两个基本问题, 即设计算法和误差分析, 开展讨论.

数值分析所研究的典型数学问题包括: 插值与逼近, 数值积分与数值微分, 线性方程组的数值解法, 非线性方程 (组) 的数值解法, 代数特征值与特征向量的计算, 常微分方程与偏微分方程的数值解法等.

1.1　误 差 分 析

1.1.1　误差的来源

利用算法求解数学问题之所以会产生误差, 主要受下面几个因素影响.

首先是观测误差, 它是指人们利用实验方法 (或测量工具) 获得实验 (测量) 数据时, 由于不可能达到绝对准确的程度而造成的误差. 例如利用毫米刻度尺测量课桌桌面长度时, 如果读出的数据为 120 mm, 那么这个读数实际上是准确值的误差不超过半个毫米的近似数. 利用带有观测误差的近似数进行计算, 结果当然是近似数. 例如我们用测得的桌面的近似长和近似宽计算桌面面积时, 所得结果自然是面积的近似值.

其次是截断误差, 它是指算法设计中为了算法的可执行性, 用能够计算的或更容易计算的数学问题代替不易计算的问题时产生的误差. 例如要计算 $\sin 0.1276$, 方法之一是利用下面的正弦函数的泰勒展开式:

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots + (-1)^n \frac{x^{2n+1}}{(2n+1)!} + \cdots. \tag{1.1}$$

然而式 (1.1) 中含有无限多项, 计算机上无法计算. 根据微积分课程中的泰勒定理知, 如果值 $|x|$ 接近于 0, 那么可以用上式中的有限项代替无限项来近似计算 $\sin x$, 比如说可以取

$$\sin x \approx x - \frac{x^3}{3!} + \frac{x^5}{5!}, \tag{1.2}$$

这时, 其误差的绝对值不超过 $x^7/7!$.

实际上, 利用公式 (1.2) 计算 (在 MATLAB 上运行, 参考第 10 章)$\sin 0.1276$ 可得

$$\sin 0.1276 \approx 0.1276 - \frac{0.1276^3}{3!} + \frac{0.1276^5}{5!} = 0.12725402312244,$$

而直接计算得 $0.12725402301319\cdots$, 其截断误差约为 1.0925×10^{-10}, 不超过 $x^7/7! \approx 1.0928 \times 10^{-10}$.

最后一类误差是舍入误差. 它是因为计算机完成的所有计算过程中得到的任何数都只能保留有限位数字造成的. 例如, 分数 $1/3 = 0.3333\cdots$, 计算机只能取有限小数位, 从而形成了舍入误差.

既然数值计算中误差是不可避免的, 所以求解任何一个数学问题除研究各种算法外, 还必须分析该算法的计算结果是否满足精度要求, 这是误差分析的任务. 在数值分析这门课程中, 我们主要是估计截断误差和舍入误差.

1.1.2　误差分析的基本概念

设 x 是某个量的真值, x_{A} 是它的一个近似值, 我们称 $x - x_{\mathrm{A}}$ 为近似值 x_{A} 的绝对误差, 或简称误差. $(x - x_{\mathrm{A}})/x$ 称为 x_{A} 的相对误差.

实际上, 真值往往是未知的, 所以, 一个近似数的绝对误差和相对误差都是算不出来的. 不过, 解决实际问题时一般只需要估计它们的值不超过多少. 如果 $|x - x_{\mathrm{A}}| \leqslant \epsilon_{\mathrm{A}}$, 则称 ϵ_{A} 是 x_{A} 的绝对误差界 (限), 或简称误差界, $\epsilon_{\mathrm{R}} \triangleq \epsilon_{\mathrm{A}}/|x_{\mathrm{A}}|$ 为相对误差界.

例 1.1.1　已知 $\pi = 3.1415926\cdots$, 若取 π 的近似值 $x_{\mathrm{A}} = 3.14$, 则 $\pi - x_{\mathrm{A}} = 0.0015926\cdots$, 可以估计其绝对误差界为 0.002, 相对误差界为 $0.0006369426751592356 < 0.00064$.

例 1.1.2　测量一木板长是 954 cm. 问测量的相对误差界是多大?

解 因为测量中所截取的近似数, 其绝对误差一般不超过测量工具的最小刻度的半个单位, 所以测量值 x_{A}=954 cm 的绝对误差界 ϵ_{A}=0.5 cm, 从而相对误差界 $\epsilon_{\mathrm{R}} = 0.5/954 = 0.0005241090146750524 < 0.00053$.

如果 $|x - x_{\mathrm{A}}| \leqslant 0.5 \times 10^{-n}$, 即 $\epsilon_{\mathrm{A}} = 0.5 \times 10^{-n}$, 则称近似数 x_{A} 准确到了 n 位小数, 该数位到第一个非零数字的所有数位叫做该近似数的有效数位, 有效数位上的数字叫做有效数字. 例如, 用 $x_1 = 3.14$ 作为 π 的近似数时, 因为

$$|\pi - x_1| \leqslant 0.002 \leqslant 0.5 \times 10^{-2},$$

所以 x_1 准确到了 2 位小数, 且该数的所有数位都是有效数位, 各有效数位上的数字都是有效数字, 就是说, 3, 1, 4 所在数位都是准确数位, 它们都是有效数位.

如果用 $x_2 = 3.141$ 作为 π 的近似数, 因为

$$|\pi - x_2| = 0.0005926\cdots \leqslant 0.0006 \leqslant 0.5 \times 10^{-2},$$

所以 x_2 仍是准确到 2 位小数, 且该数中除最后一位数外, 其他各数位上的数字都是有效数字.

下面的定理给出了有效数字与相对误差界之间的关系.

定理 1.1.1 设 x_{A} 是 x 的近似数, 且具有如下表达式:

$$x_{\mathrm{A}} = \pm 10^k \times 0.a_1a_2\cdots a_i\cdots,$$

它可以是有限或无限小数的形式, $a_i(i = 1, 2, \cdots)$ 是 $0, 1, \cdots, 9$ 中的一个数字, $a_1 \neq 0$, k 为整数.

1. 如果 x_{A} 有 n 位有效数字, 则

$$\epsilon_{\mathrm{R}} \leqslant \frac{1}{2a_1} \times 10^{1-n};$$

2. 如果

$$\epsilon_{\mathrm{R}} \leqslant \frac{1}{2(a_1+1)} \times 10^{1-n},$$

则 x_{A} 至少具有 n 位有效数字.

证明留作习题 (见习题 6).

例 1.1.3 已知近似数 x_{A} 的相对误差界为 0.003. 问 x_{A} 至少有几位有效数字?

解 设 x_{A} 至少有 n 位有效数字.

尽管 x_{A} 的第一个有效数字 a_1 没有具体给出, 但 a_1 一定是 $1, 2, \cdots, 9$ 中的一个. 由于

$$\epsilon_{\mathrm{R}} \leqslant 0.003 < 0.5 \times 10^{-2} = \frac{1}{2 \times 10^2} = \frac{1}{2(9+1)} \times 10^{-1},$$

根据定理 1.1.1 的结论 2 知, $1 - n = -1$, 即 $n = 2$.

1.1.3　基本误差估计公式

设 $y=f(x)$ 为一元二次连续可微函数, 即二阶导数连续的函数. 要计算函数在定义域内任意一点 x_0 处的函数值 $f(x_0)$, 实际中常用 "更好的" 点 x_{A} 近似 x_0, 并计算 $f(x_{\mathrm{A}})$ 以作为 $f(x_0)$ 的近似值. 例如, 要计算 $\sin\dfrac{\pi}{5}$, 常用计算 $\dfrac{\pi}{4}$ 处的函数值来近似. 如果其误差界记作 $\epsilon(f(x_{\mathrm{A}}))$, 则根据泰勒定理知

$$|f(x_0)-f(x_{\mathrm{A}})|\approx|f'(x_{\mathrm{A}})||x_0-x_{\mathrm{A}}|\leqslant|f'(x_{\mathrm{A}})|\epsilon_{\mathrm{A}}(x_{\mathrm{A}})\triangleq\epsilon(f(x_{\mathrm{A}})),$$

所以

$$f(x_0)\approx f(x_{\mathrm{A}})\pm\epsilon(f(x_{\mathrm{A}})).$$

同样地, 对 n 元二次连续可微的函数 $u=f(x_1,x_2,\cdots,x_n)$, 如果各自变量的误差界分别为 ϵ_1, ϵ_2, $\cdots$, ϵ_n, 那么函数值的误差界

$$\epsilon(u)\approx\sum_{k=1}^{n}\left|\frac{\partial f}{\partial x_k}\right|\Bigg|_{\boldsymbol{x}=\boldsymbol{x}^{(\mathrm{A})}}\epsilon_k,$$

其中, $\boldsymbol{x}^{(\mathrm{A})}=(x_1^{(\mathrm{A})},x_2^{(\mathrm{A})},\cdots,x_n^{(\mathrm{A})})^{\mathrm{T}}$ 为已知近似自变量的取值, 因此

$$f(\boldsymbol{x}^{(0)})\approx f(\boldsymbol{x}^{(\mathrm{A})})\pm\epsilon(u),$$

式中, $\boldsymbol{x}^{(0)}=(x_1^{(0)},x_2^{(0)},\cdots,x_n^{(0)})^{\mathrm{T}}$ 为待计算函数值的点.

特别地, 如果 $u=f(x_1,x_2)=x_1\pm x_2$, 那么

$$\epsilon(u)=\epsilon(x_1\pm x_2)\approx\epsilon_1+\epsilon_2.$$

如果 $u=f(x_1,x_2)=x_1x_2$, 那么

$$\epsilon(u)=\epsilon(x_1x_2)\approx|x_2^{(\mathrm{A})}|\epsilon_1+|x_1^{(\mathrm{A})}|\epsilon_2.$$

如果 $u=f(x_1,x_2)=x_1/x_2$, 那么

$$\epsilon(u)=\epsilon\left(\frac{x_1}{x_2}\right)\approx\frac{|x_2^{(\mathrm{A})}|\epsilon_1+|x_1^{(\mathrm{A})}|\epsilon_2}{\left(x_2^{(\mathrm{A})}\right)^2},\quad x_2^{(\mathrm{A})}\neq0.$$

例 1.1.4　设有长为 l, 宽为 d 的某场地. 经测量得 $l_{\mathrm{A}}\approx$120 m, $d_{\mathrm{A}}\approx$90 m. 已知它们的测量误差界为 0.2 m. 试估计场地面积的大小和它的误差界.

解　因为场地面积 $S=ld$, 所以

$$\epsilon(S)\approx|l_{\mathrm{A}}|\epsilon(d_{\mathrm{A}})+|d_{\mathrm{A}}|\epsilon(l_{\mathrm{A}}).$$

已知 $\epsilon(l_{\mathrm{A}})=\epsilon(d_{\mathrm{A}})=0.2$, $l_{\mathrm{A}}=120$, $d_{\mathrm{A}}=90$, 代入上式得面积的误差界

$$\epsilon(S)\approx 42\ \mathrm{m}^2,$$

因此场地面积 $S=(10800\pm 42)\ \mathrm{m}^2$.

例 1.1.5 设有三个近似数

$$a=2.31,\quad b=1.93,\quad c=2.24,$$

它们都有三位有效数字. 试计算 $p=a+bc$ 的误差界和相对误差界, 并问 p 的计算结果能有几位有效数字?

解 因为 a, b, c 均有三位有效数字, 所以它们的误差界 $\epsilon(a)=\epsilon(b)=\epsilon(c)=0.5\times 10^{-2}$, 又因为

$$\epsilon(p)\approx\epsilon(a)+\epsilon(bc)=\epsilon(a)+|b|\epsilon(c)+|c|\epsilon(b)=0.02585\leqslant 0.5\times 10^{-1},$$

所以 p 的相对误差界

$$\epsilon_r(p)=\frac{\epsilon(p)}{|p|}\approx\frac{0.02585}{6.6332}\approx 0.0039,$$

且 p 的近似值准确到了 1 位小数, 从而有两位有效数字 (个位和小数点后第一位小数).

1.1.4 计算机中的数的表示

任意给定的实数 x, 都可以唯一写成如下形式:

$$x=\pm 10^{k}\times 0.a_1a_2\cdots a_i\cdots,$$

其中 $a_i(i=1,2,\cdots)$ 是 $0,1,\cdots,9$ 中的一个数字, 且 $a_1\neq 0$, k 为整数. 这种标准化的记数方法叫做十进制科学计数方法. 不过, 计算机中的数字表示通常采用二进制 (或其变形的十六进制), 即任何一个实数 x, 都可以表示成如下浮点形式:

$$x=\pm 2^{m}\times 0.\beta_1\beta_2\cdots\beta_t,$$

其中整数 m 称为阶码, 小数 $0.\beta_1\beta_2\cdots\beta_t$ 称为尾数, t 为尾数部位的位数, $\beta_1=1$, $\beta_j=0$ 或 $1(j=2,3,\cdots,t)$. 如果整数 m 的二进制表示式为

$$m=\pm\alpha_1\alpha_2\cdots\alpha_s,$$

其中 $\alpha_j=0$ 或 $1(j=2,3,\cdots,s)$, 那么称 s 为阶码的位数.

由于在任何一台计算机中,s, t 都是确定的有限数, 所以计算机所能表示的数系不是整个实数数系, 而是其特殊的子集. 该子集中的数称为机器数. 显然, 机器数一般都有舍入误差.

为了适应人们的习惯, 今后我们对算法作误差分析时, 仍采用十进制.

1.2　算法的数值稳定性和减少误差的措施

1.2.1　数值稳定性

对某个给定的算法, 如果输入数据的误差随着算法的运算过程不断增长而得不到控制, 那么我们就说该算法是数值不稳定的; 否则就是数值稳定的.

更准确地说, 假设输入数据的误差为 ϵ, 经 n 次运算后计算结果的误差为 E_n. 当 $E_n \approx Cn\epsilon$ (C 为与 n 无关的常数) 时, 就说误差是线性增长, 当 $E_n \approx k^n\epsilon$ ($k>1$ 为与 n 无关的常数) 时, 就说误差是指数增长. 如果算法的误差增长是线性的, 则该算法是数值稳定的, 如果算法的误差是指数增长的, 则该算法是数值不稳定的. 显然, 误差的线性增长是不可避免的, 而指数增长是必须避免的.

例 1.2.1　计算数列 $p_n=(1/3)^n$.

解　第一种方法是令 $p_0=1$, 利用递推公式 $p_n=(1/3)p_{n-1}$, 在五位数字计算机上可算出

$$p_1=0.33333\times 10^0,\quad p_2=0.11111\times 10^0,$$

$$p_3=0.37036\times 10^{-1},\quad p_4=0.12345\times 10^{-1},\ \cdots$$

这里, $1/3$ 用 0.33333 近似, 误差限为 $\epsilon=0.5\times 10^{-5}$. 经 n 步运算后的近似结果的误差限 $E_n=n(0.33333)^{n-1}\times 0.5\times 10^{-5}<n\epsilon$, 所以误差是线性增长的, 从而算法是稳定的.

如果令 $p_0=1$, $p_1=1/3$, 并利用递推公式 $p_n=(10/3)p_{n-1}-p_{n-2}$ 在五位数字计算机上计算, 则前八步的近似结果如表 1.1 所示.

表 1.1

n	p_n 的近似值	p_n 的准确值
1	0.33333×10^0	0.33333×10^0
2	0.11110×10^0	0.11111×10^0
3	0.37000×10^{-1}	0.37037×10^{-1}
4	0.12230×10^{-1}	0.12346×10^{-1}
5	0.37660×10^{-2}	0.41152×10^{-2}
6	0.32300×10^{-3}	0.13717×10^{-2}
7	-0.26893×10^{-2}	0.45725×10^{-3}
8	-0.92872×10^{-2}	0.15242×10^{-3}

从表 1.1 可以看出, 后一种算法不稳定. 实际上, 这里的输入数据 1/3 也是用 0.33333 近似, 其误差限为 $\epsilon = 0.5 \times 10^{-5}$. 但可以证明经 n 步运算后近似结果的误差限 $E_n = -0.125 \times 3^n \times 10^{-5}$. 因为误差是指数增长的, 所以算法不稳定.

当然, 数值不稳定的算法在实际计算中不能采用. 下面详细讨论如何避免这种算法不稳定现象.

1.2.2 减少误差的措施

在数值计算中, 参加运算的数有时数量级相差很大, 由于计算机可保留的数位有限, 所以若不注意, 运算中 "小数" 的作用可能消失, 即出现 "大数" 吃 "小数" 的现象.

例 1.2.2 用三位十进制数字计算

$$x = 101 + \delta_1 + \delta_2 + \cdots + \delta_{100},$$

其中 $0.1 \leqslant \delta_i \leqslant 0.4$, $i = 1, 2, \cdots, 100$.

解 一种方法是自左至右逐个相加. 这时因为在三位数字计算机上计算, 所以后面所有的 δ_i 都会被舍掉, 得 $x \approx 101$.

另一种方法是先把所有的 δ_i 加起来, 再与 101 相加. 这时 $111 \leqslant x \leqslant 141$, 且可以精确算出 x 的值.

由此可见, 计算的次序有时会大大影响计算结果.

数值计算中, 相近数相减有时也会造成有效数字的严重损失.

例 1.2.3 求实系数二次方程 $ax^2 + bx + c = 0$ 的根, 其中 $b^2 - 4ac > 0$, $ab \neq 0$.

解 计算该方程的根有两种方法. 算法 1 是直接利用公式

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a},$$

算法 2 是根据 $x_1 x_2 = c/a$, 利用公式

$$x_1 = \frac{-b - \operatorname{sgn}(b)\sqrt{b^2 - 4ac}}{2a}, \quad x_2 = \frac{c}{ax_1}$$

计算, 其中 sgn 表示符号函数:

$$\operatorname{sgn}(x) = \begin{cases} 1, & x > 0, \\ 0, & x = 0, \\ -1, & x < 0. \end{cases}$$

对算法 1 来说, 如果 $b^2 \gg 4ac$, 则算法不稳定, 否则是稳定的. 这是因为在算法 1 中, 分子是相近数相减, 会造成有效数字的严重损失, 从而结果的误差很大. 对算法 2 来

说, 不会出现上述现象, 算法总是稳定的. 例如, 我们在四位数字计算机上解方程

$$x^2 + 62.10x + 1.00 = 0.$$

结果如下:

算法 1　$x_1 = -62.08, x_2 = -0.0200.$

算法 2　$x_1 = -62.08, x_2 = -0.0161.$

同方程的准确解 $x_1 = -62.083892\cdots, x_2 = -0.016107237\cdots$ 相比, 算法 1 误差太大. 这是因为 $b^2 \gg 4ac$ 造成的.

数值计算时如果遇到相近数相减的情形, 可通过变换计算公式来避免或减少有效数字的损失. 例如, 如果 $|x| \approx 0$, 要计算

$$\frac{1-\cos x}{\sin x},$$

可把它变换成公式

$$\frac{\sin x}{1+\cos x}.$$

如果 $x_1 \approx x_2$, 要计算

$$\lg x_1 - \lg x_2,$$

可把它变换成公式

$$\lg \frac{x_1}{x_2}.$$

如果 $x \gg 1$, 要计算

$$\sqrt{x+1} - \sqrt{x},$$

可把它变换成公式

$$\frac{1}{\sqrt{x+1}+\sqrt{x}}.$$

如果无法改变算法, 则只有通过增加有效数位的方法进行计算, 或在计算机上采用双精度运算, 但这样做会增加机器计算时间和多占内存单元.

数值计算中注意简化计算步骤, 减少运算次数是保证算法稳定, 减少误差积累的另一重要措施. 下面来看一个例子.

例 1.2.4　对给定的 x, 计算多项式

$$P_n(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_0$$

的值.

解 一种算法是先求 a_kx^k, $k=1,2,\cdots,n$, 这需要进行

$$\sum_{k=1}^{n} k=\frac{n(1+n)}{2}$$

次乘法, 再把各项加起来便得到多项式的值, 这需要进行 n 次加法运算.

另一种算法是先把多项式 $P_n(x)$ 改写成

$$P_n(x)=x\{x\cdots[x(a_nx+a_{n-1})+a_{n-2}]+\cdots+a_1\}+a_0,$$

再从上面算式的内层逐步往外层计算 (即先算圆括弧, 再算方括弧, $\cdots$). 不难统计, 这时共进行了 n 次乘法和 n 次加法.

显然, 第二种算法比第一种算法的计算次数大大减少, 它不仅能够节省机器计算时间, 而且有利于防止误差积累. 该种算法称为秦九韶算法. 其实际计算机程序可描述为

$$\begin{cases} u_n=a_n, \\ u_k=u_{k+1}x+a_k, \end{cases} \qquad k=n-1,n-2,\cdots,0.$$

算法最后的输出结果 u_0 即为要计算的多项式的值.

最后, 数值计算中为减少误差积累, 要避免小除数、大乘数. 这是因为

$$\epsilon(x_1x_2)=\epsilon(x_1x_2)\approx|x_2^{(\mathrm{A})}|\epsilon_1+|x_1^{(\mathrm{A})}|\epsilon_2,$$

$$\epsilon\left(\frac{x_1}{x_2}\right)=\epsilon\left(\frac{x_1}{x_2}\right)\approx\frac{|x_2^{(\mathrm{A})}|\epsilon_1+|x_1^{(\mathrm{A})}|\epsilon_2}{\left(x_2^{(\mathrm{A})}\right)^2}, \qquad x_2^{(\mathrm{A})}\neq 0.$$

如果前式中 $|x_1|$ 很大或后式中 $|x_2|\approx 0$, 计算结果的绝对误差可能很大.

概括地说, 设计算法时减少误差积累可采用如下措施:

a. 若干数相加, 为避免大数吃小数, 应该先加绝对值较小的数;

b. 尽量避免相近数相减;

c. 尽量减少运算次数;

d. 避免小除数、大乘数.

1.3 线性空间和赋范线性空间

作为普通向量空间 $\mathbf{R}^n$ 概念的抽象和推广, 线性空间和赋范线性空间是学习数值分析时常用到的两个基本概念, 它们是研究物理、力学中满足叠加原理的系统的数学模型. 线性空间是对集合的元素在线性运算 (向量的加法运算和数乘运算) 方

面所表现的共性加以概括而形成的概念, 赋范线性空间是对集合的元素的大小或长度方面所表现的共性加以概括而形成的概念. 这些概念在讨论函数的数值逼近理论, 算法的收敛性、稳定性及误差分析等问题时, 具有重要作用.

1.3.1　线性空间的概念

为了叙述线性空间的概念, 我们先介绍什么是数环和数域. 一个非空数集 E 叫做数环, 如果其中任何两个数的和、差与积还属于 E, 即关于和、差及积运算封闭. 如数集 $A=\{0\}$, 整数集 $\mathbf{Z}$, 有理数集 $\mathbf{Q}$ 和实数集 $\mathbf{R}$ 等都是数环.

一个至少含有两个元素的数集 F 称为数域, 如果其中任何两个数的和、差、积与商仍属于 F, 即关于和、差、积及商运算封闭. 如有理数集 $\mathbf{Q}$、实数集 $\mathbf{R}$ 和复数集 $\mathbf{C}$ 都是数域, 而数集 $A=\{0\}$ 和整数集 $\mathbf{Z}$ 不是数域.

定义 1.3.1　设 V 是一个非空集合, P 是一个数域. 如果 V 满足如下两个条件.

1. 在 V 中定义了封闭的加法运算 "+", 即当 $x,y\in V$ 时, 有唯一的和 $x+y\in V$. 这种加法运算还具有如下四条性质:

(1) $x+y=y+x$ (交换律);

(2) $x+(y+z)=(x+y)+z$ (结合律);

(3) 存在零元素 $0\in V$, 所谓零元素是指对 V 中的任何元素 x, 都有 $x+0=x$;

(4) 存在负元素, 即对 V 中的任何元素 x, 都存在一个元素 $y\in V$ 使得 $x+y=0$. 我们称 y 为 x 的负元素, 并记为 $-x$.

2. 在 V 中定义了封闭的数乘运算, 即当 $x\in V$, $\lambda\in P$ 时, 有唯一的元素 $\lambda x\in V$. 这种数乘运算要求具有以下四条性质:

(1) $(\lambda+\mu)x=\lambda x+\mu x$ (分配律);

(2) $\lambda(x+y)=\lambda x+\lambda y$ (数因子的分配律);

(3) $\lambda(\mu x)=(\lambda\mu)x$ (结合律);

(4) $1x=x$.

其中 x,y,z 表示 V 中的任意元素; λ,μ 是数域 P 中的任意数.

这时, 我们称集合 V 是数域 P 上的线性空间. P 为实数域 $\mathbf{R}$ 时, V 称为实线性空间; P 为复数域 $\mathbf{C}$ 时, V 称为复线性空间. 满足上述性质的加法和数乘运算, 统称为线性运算. 因此, 线性空间是指定义了线性运算的集合.

上述定义很抽象, 但线性空间的例子随处可见.

例 1.3.1　集合 $\mathbf{R}^n=\{\boldsymbol{\alpha}=(a_1,a_2,\cdots,a_n):a_i\in\mathbf{R}\}$ 关于普通意义下向量的加法和数乘运算构成了任何数域 K 上的线性空间.

例 1.3.2　集合 $M_{m,n}[\mathbf{R}]=\{A=(a_{ij})_{m\times n}:a_{ij}\in\mathbf{R}\}$ 关于普通意义下矩阵的加法和数乘运算构成了任何数域 K 上的线性空间.