高等学校教材

数值分析

■ 徐明华　张燕新　李志林　编

SHUZHI FENXI

高等教育出版社
HIGHER EDUCATION PRESS

高等学校教材

数值分析

Shuzhi Fenxi

徐明华 张燕新 李志林 编

高等教育出版社·北京
HIGHER EDUCATION PRESS BEIJING

内容提要

本书主要介绍线性方程组与非线性方程的数值解法、插值法与曲线拟合、数值积分与数值微分、常微分方程数值解法、方阵的特征值和特征向量的数值解法。全书注重通过数值方法的比较来培养学生的数值计算思想，注重通过算法程序的介绍来帮助学生掌握算法的实现技巧，注重培养学生应用数值方法解决问题的能力。内容条理清楚，通俗易懂，既可作为理工科硕士研究生和部分理工科本科生数值分析课程教材，也可供从事数值计算的科技工作者参考。

图书在版编目(CIP)数据

数值分析/徐明华,张燕新,李志林编.--北京：高等教育出版社,2013.2

ISBN 978-7-04-036845-1

Ⅰ.①数… Ⅱ.①徐…②张…③李… Ⅲ.①数值分析-高等学校-教材 Ⅳ.①O241

中国版本图书馆CIP数据核字(2013)第009014号

策划编辑	李　蕊	责任编辑	李　蕊	特约编辑	师钦贤	封面设计	李卫青
版式设计	马敬茹	插图绘制	邓　超	责任校对	杨凤玲	责任印制	韩　刚

出版发行	高等教育出版社	咨询电话	400-810-0598
社　　址	北京市西城区德外大街4号	网　　址	http://www.hep.edu.cn
邮政编码	100120		http://www.hep.com.cn
印　　刷	北京鑫丰华彩印有限公司	网上订购	http://www.landraco.com
开　　本	787mm×960mm　1/16		http://www.landraco.com.cn
印　　张	15.5	版　　次	2013年2月第1版
字　　数	270千字	印　　次	2013年2月第1次印刷
购书热线	010-58581118	定　　价	26.10元

物 料 号　36845-00

前　言

随着计算机技术的发展和应用的深入, 科学计算已经在许多工程应用领域得到广泛应用。与科学实验一样, 科学计算已经成为人们进行科学活动的一个重要方法, 利用计算机进行科学计算已经成为许多工程科技人员的一项基本能力。

数值分析是从事科学计算的理论基础和方法基础, 是联系工程计算问题与计算机的桥梁。通过对数值分析的系统学习可以获得科学计算的基本素质。

本书是编者在积累多年讲课经验的基础上编写而成的, 主要适合理工科硕士研究生学习使用, 兼顾部分理工科本科专业教学使用。本书介绍了处理工程计算中常见数学问题的数值计算方法。在内容的组织上, 编者比较注意各种方法引入背景的介绍, 注重方法思想的介绍。同时, 编者还注重方法的评价以及方法的实现。这样可以逐步帮助学生形成好的科学计算思想和良好的计算习惯, 最终达到培养学生运用数值分析的方法思想处理实际问题的能力。考虑到数值计算软件 Matlab 在工程计算方面的广泛应用, 编者将部分常用算法的 Matlab 程序放在书中供读者学习参考。相信通过对这些程序的了解掌握, 读者也能够利用 Matlab 编写自己需要的程序。

本书内容可以根据学时要求适当选学, 重点是要让学生形成科学的计算观, 能够针对实际问题提出合理的数值方法, 并且能够应用计算机和相关软件实现。

本书在编写过程中力求做到通俗易懂, 方便学生学习。学习本书需要具备高等数学、线性代数的基础知识。

本书第二章由张燕新、李志林负责编写, 其他各章由徐明华负责编写, 最后由徐明华统稿。由于编者水平有限, 稿中不足之处难免, 欢迎读者指正。

编　者

2012 年 8 月

目　录

第一章　绪论 …… 1
1.1　引言 …… 1
1.2　误差 …… 5
1.3　减少误差危害的几个手段 …… 11
习题一 …… 17
第二章　线性方程组的数值解法 …… 20
2.1　引言 …… 20
2.2　Gauss 消元法及其变形 …… 22
2.2.1　三角形方程组的解法 …… 22
2.2.2　Gauss 消元法 …… 23
2.2.3　列主元 Gauss 消元法 …… 28
2.3　矩阵分解法 …… 35
2.3.1　LU 分解法 …… 35
2.3.2　列主元 LU 分解法 …… 38
2.3.3　平方根法 …… 41
2.3.4　改进平方根法 …… 43
2.4　追赶法 …… 44
2.5　向量范数与矩阵范数 …… 49
2.5.1　向量范数 …… 49
2.5.2　矩阵范数 …… 53
2.5.3　线性方程组的扰动分析 …… 60
2.6　线性方程组的迭代解法 …… 66
2.6.1　迭代法的基本思想 …… 66
2.6.2　迭代法的收敛性 …… 66
2.6.3　Jacobi 迭代法 …… 71
2.6.4　Gauss−Seidel 迭代法 …… 74
2.6.5　SOR 迭代法 …… 76

习题二 ······ 78
第三章 插值法 ······ 83
3.1 问题提出 ······ 83
3.2 Lagrange 插值法 ······ 83
3.3 逐次线性插值法 ······ 90
3.4 Newton 插值法 ······ 93
3.5 Hermite 插值法 ······ 100
3.6 分段插值法 ······ 106
3.7 三次样条插值法 ······ 109
习题三 ······ 119
第四章 曲线拟合 ······ 123
4.1 引言 ······ 123
4.2 最小二乘法 ······ 124
习题四 ······ 131
第五章 数值积分与数值微分 ······ 133
5.1 引言 ······ 133
5.2 Newton–Cotes 公式 ······ 135
5.2.1 插值求积公式 ······ 135
5.2.2 Newton–Cotes 公式 ······ 136
5.2.3 复合求积公式 ······ 139
5.3 Romberg 方法 ······ 142
5.3.1 复合梯形公式的递推公式 ······ 142
5.3.2 Romberg 公式 ······ 143
5.4 Gauss 公式 ······ 150
5.5 数值微分 ······ 158
习题五 ······ 163
第六章 方程求根 ······ 166
6.1 引言 ······ 166
6.2 二分法 ······ 167
6.3 迭代法 ······ 169
6.4 Newton 法 ······ 176
6.5 割线法与抛物线法 ······ 182
习题六 ······ 185

第七章　常微分方程数值解法 …… 187
7.1　引言 …… 187
7.2　Euler 方法与梯形方法 …… 190
7.3　Runge–Kutta 方法 …… 196
7.4　显式单步法的收敛性和稳定性 …… 199
7.5　线性多步法 …… 203
7.6　一阶方程组及高阶方程初值问题的数值解法 …… 209
习题七 …… 211
第八章　方阵特征值和特征向量的数值解法 …… 214
8.1　引言 …… 214
8.2　幂法与反幂法 …… 215
8.2.1　幂法 …… 216
8.2.2　幂法的加速 …… 219
8.2.3　带原点平移的反幂法 …… 221
8.3　QR 方法 …… 222
8.3.1　Householder 变换 …… 223
8.3.2　Givens 变换 …… 223
8.3.3　QR 分解 …… 224
8.3.4　QR 方法 …… 227
8.3.5　带原点平移的 QR 方法 …… 232
习题八 …… 236
参考文献 …… 239

第一章 绪　论

1.1 引言

如大家所知, 数学在许多领域都有广泛的应用, 科学与工程领域中的许多问题经过抽象后均可以用一个数学模型来表示. 这些来自工程实际的数学模型通常都很复杂, 很难解析求解, 求解这类问题要依靠计算机强大的存储能力及运算能力才能完成. 这门课程正是为计算机提供求解这些数学模型所需方法的重要基础.

数值分析是研究用计算机解决各种数学模型的数值计算方法 (算法) 及其理论. 它是与计算机密切相关的一门数学课程, 也是目前许多学科, 如: 计算数学、计算物理学、计算力学、计算化学、计算生物学、计算地质学、计算经济学等的理论与方法基础. 根据不同的数学模型, 数值分析的研究内容又大致分为数值逼近、方程求根、数值线性代数、矩阵特征值与特征向量、数值积分、求微分方程数值解、数值最优化等. 随着科学水平的提高和计算机的发展, 各种基于人们认识能力和计算机以及网络能力的算法层出不穷, 极大地丰富并发展了数值分析这门课程, 同时数值分析的发展也进一步增强了应用计算机处理实际问题的能力, 拓宽了计算机的应用范围. 目前科学计算已经与理论研究、科学试验一起成为科学研究的三大方法与手段.

数值分析是研究解决各种数学模型并且能够在计算机上实现的算法, 这就决定了这门课程是一个应用性很强的数学课程, 既具有数学学科的严谨性, 同时又与计算机技术发展相互依赖、相互促进、相互制约. 因此, 这门课程具备如下特点:

- 面向计算机　数值分析所提供的数值方法必须是能够在计算机上实现的算法, 算法必须适应并且能充分利用计算机的计算能力. 面向计算机这一特点, 要求我们的算法既要考虑计算机的字长、存储量以及计算机只能进行四则运算和逻辑运算, 不能够处理无穷过程运算等特点, 又要考虑尽可能充分利用计算机的计算能力, 如并行机的并行处理能力等.

例如, 对于 $\forall x \in (-\infty, +\infty)$, 理论上有

$$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!} = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!} + \cdots,$$

这种计算 e^x 的方法不是面向计算机的方法, 因为计算机不能够处理无穷过程的运算. 但

$$e^x \approx \sum_{k=0}^{n} \frac{x^k}{k!} = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!}$$

是面向计算机的计算 e^x 的一种算法.

- 有可靠的理论分析　数值分析所提供的算法通常都是求解数学模型的近似方法. 一方面, 我们要考虑算法本身能否给出所求数学模型的解. 另一方面, 由于算法运行需要的初始数据往往有误差, 加上计算机字长有限, 算法在运行过程中不可避免地有舍入误差. 因此我们还要考虑上述误差对算法结果的影响如何. 所有这些问题都需要有相应的理论分析来回答, 理论分析是一个算法是否可行的基础.

- 有计算复杂性分析　数值方法所需要的计算时间和存储量, 即算法的时间复杂性和空间复杂性, 必须符合计算机的计算能力, 它关系到算法能否在计算机上实现, 算法需要的计算时间和存储量越少, 则算法的时间复杂性和空间复杂性越好.

- 有数值实验　任何一个算法除了从理论上分析其可行, 还要通过数值实验证明其是行之有效的, 即要有实验基础, 这也是这门课程不同于其他数学课程的一个地方.

- 面向具体问题　算法的设计要充分利用具体问题的结构和特点, 算法运行需要的环境必须面向问题实际, 不能随意超越, 脱离实际的算法必然影响其有效性与可行性.

- 面向大众　算法的结构要尽量简单, 要方便大众使用.

根据数值分析这门课程的特点, 一个算法的好坏, 可从它符合数值分析这门课程特点的程度来评价, 因此, 算法优劣的评价标准应该包括下面几个方面:

- 收敛性与稳定性, 即算法精度与可靠性的高低;
- 收敛速度快慢, 即工作量或时间花费大小;
- 占用计算机资源多少, 如占用存储量 (空间花费) 大小;

- 能否充分利用问题本身的特点;
- 能否充分利用计算机的计算能力;
- 逻辑结构是否简单.

当然, 上述算法的评价标准不是绝对的, 不同问题对算法要求的侧重点可能不一样, 有的问题对计算结果的精度要求很高, 而有的问题特别对计算所需的时间花费要求较高等. 数值方法研究人员就是要根据具体的问题给出适合该问题的算法.

例 1 计算 $p_n(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$ 在 x_0 处的值.

解 算法一 令

$$
\begin{aligned}
& t_0 = 1;\ p_0 = a_0; \\
& for\ k = 0 : n-1 \\
& \qquad t_{k+1} = t_k x_0; \\
& \qquad p_{k+1} = p_k + a_{k+1} t_{k+1}; \\
& end
\end{aligned}
$$

显然有 $p_n = p_n(x_0)$ 为所求, 这里所需主要工作量为 $2n$ 次乘法. 另外, 由于这一算法中 t_k 与 p_k 在算出 t_{k+1} 与 p_{k+1} 之后就不再起作用, 因此在编写程序时可以将 t_{k+1} 与 p_{k+1} 分别放在 t_0 与 p_0 的存储单元中, 从而上述算法可以写成:

令

$$
\begin{aligned}
& t_0 := 1;\ p_0 := a_0; \\
& for\ k = 0 : n-1 \\
& \qquad t_0 := t_0 x_0; \\
& \qquad p_0 := p_0 + a_{k+1} t_0; \\
& end
\end{aligned}
$$

最终 p_0 为所求, 其中 “:=” 为赋值号.

算法二 由于 $p_n(x_0)$ 可以改写成如下等价形式

$$p_n(x_0) = (\cdots((a_n x_0 + a_{n-1})x_0 + a_{n-2})x_0 + \cdots + a_1)x_0 + a_0,$$

设

$$t_n = a_n;\ t_k = t_{k+1}x_0 + a_k; \quad k = n-1, \cdots, 1, 0.$$

显然 t_0 为 $p_n(x_0)$, 而这里的主要计算量仅需 n 次乘法, 比算法一节省一半工作量. 该算法称为秦九韶方法或 Horner(霍纳) 算法. 类似地, 在编程时可以节省存储单元, 采用如下算法:

$$
\begin{aligned}
& t_0 := a_n; \\
& for\ i = n : 1 \\
& \qquad t_0 := t_0 x_0 + a_{i-1}; \\
& end
\end{aligned}
$$

例 2 在字长为 7 的十进制计算机 (即该计算机所表示的十进制数的小数部分的位数与整数部分的位数之和不超过 7, 对位数超过 7 的数存放到该计算机, 则需通过“四舍五入”方式或截断方式舍去多余的位数. 为方便起见, 本书均假设用“四舍五入”的方式舍去多余的位数) 上计算 $f(x) = \mathrm{e}^x$ 在 $x = 1$ 处的导数 $f'(1)$.

解 由导数定义

$$f'(x_0) = \lim_{h \to 0} \frac{f(x_0 + h) - f(x_0)}{h}$$

可知 $f'(x_0)$ 可以通过下述差商

$$f[x_0 + h, x_0] = \frac{f(x_0 + h) - f(x_0)}{h}$$

近似得到. 从导数定义来看, h 越小, 差商近似导数的误差似乎应该也越小, 下面我们在字长为 7 的十进制计算机上观测随着 h 变化, $f'(1)$ 与 $f[1 + h, 1]$ 的差异, 计算结果见表 1–1.

表 1–1

h	$f[1+h,1]$	$f'(1)$	$\lvert f[1+h,1] - f'(1)\rvert$
1.0×10^{-1}	2.858 840	2.718 282	$1.405\ 580 \times 10^{-1}$
1.0×10^{-2}	2.731 900	2.718 282	$1.361\ 800 \times 10^{-2}$
1.0×10^{-3}	2.719 000	2.718 282	$7.180\ 000 \times 10^{-4}$
1.0×10^{-4}	2.720 000	2.718 282	$1.718\ 000 \times 10^{-3}$
1.0×10^{-5}	2.700 000	2.718 282	$1.828\ 200 \times 10^{-2}$
1.0×10^{-6}	3.000 000	2.718 282	$2.817\ 180 \times 10^{-1}$
1.0×10^{-7}	0.000 000	2.718 282	$2.718\ 282 \times 10^{0}$

数值计算结果表明 $|f[1+h,1]-f'(1)|$ 不总是随着 h 的减小而减小. 产生这一现象的关键是在上述计算机上通常得不到 $f(x)$ 在所求点的精确值. 因此, 在设计或使用算法时, 还要考虑计算机的特点.

例 3 分别利用下列理论上与 $(\sqrt{2}-1)^6$ 等价的表达式计算 $(\sqrt{2}-1)^6$, 取 $\sqrt{2}\approx 1.4$, 试比较计算结果.

$$\frac{1}{(\sqrt{2}+1)^6}\ ,\ (3-2\sqrt{2})^3\ ,\ \frac{1}{(3+2\sqrt{2})^3}\ ,\ 99-70\sqrt{2}.$$

解 若取 $\sqrt{2}\approx 1.4$, 则

$$\begin{aligned}
&\frac{1}{(\sqrt{2}+1)^6}\approx\frac{1}{(1.4+1)^6}=\frac{1}{2.4^6}=0.005\ 232\ 78\cdots,\\
&(3-2\sqrt{2})^3\approx(3-2\times 1.4)^3=0.008,\\
&\frac{1}{(3+2\sqrt{2})^3}\approx\frac{1}{(3+2.8)^3}=0.005\ 125\ 26\cdots,\\
&99-70\sqrt{2}\approx 99-70\times 1.4=1,
\end{aligned}$$

与 $(\sqrt{2}-1)^6=0.005\ 050\ 63\cdots$ 比较, 用 $\dfrac{1}{(3+2\sqrt{2})^3}$ 所得数值结果最好.

该计算结果表明理论上恒等的表达式, 数值计算结果并不一定恒等. 处理同一问题的不同算法, 所得结果并不一定相同, 而且可能相差很大. 正因为如此, 探索解决问题的算法, 并不断追求更好的算法是这门学科不断向前发展的动力所在.

1.2 误差

数值分析为各类数学模型提供的数值解一般都是近似解, 都存在误差. 导致数值解有误差的原因主要有三方面: 一是数学模型中的原始数据, 即运行算法需要的初始数据, 有误差或不能够被精确地使用. 二是由于计算机只能做有限运算, 因此用数值方法解数学模型时, 通常需要先离散化原问题或化原问题的无穷过程为有限过程, 然后再求解一个相比原问题容易的、近似的问题, 这就会产生误差, 这种误差是由所选择的方法或化无穷过程为有限过程引起的, 因而称为方法误差或截断误差. 例如, 在计算函数 e^x 在某一点 x 处的函数值时, 我们可以使用公式

$$e^x = 1 + x + \frac{1}{2}x^2 + \cdots + \frac{1}{n!}x^n + \cdots,$$

但实际计算时, 我们只能取上述级数的前面有限项作为 e^x 的近似, 即

$$e^x \approx 1 + x + \frac{1}{2}x^2 + \cdots + \frac{1}{n!}x^n,$$

由此产生的误差就是截断误差. 再如, 我们有

$$f'(x_0) = \lim_{h\to 0} \frac{f(x_0+h) - f(x_0)}{h},$$

但实际计算时我们并不能够将 h 取得任意小, 只能够取适当的 h, 用

$$\frac{f(x_0+h) - f(x_0)}{h}$$

近似 $f'(x_0)$, 由此产生的误差也是截断误差. 三是由于计算机的字长有限, 利用计算机计算就不可避免地产生误差, 这种误差称为舍入误差. 例如, 在字长为 7 的十进制计算机上计算 $1.234\ 567^2$ 只能得到 1.524 156, 而不是理论上的 1.524 155 677 489.

研究算法的一个主要目的就是要控制或减少上述误差对计算结果的影响. 下面我们先给出与误差有关的基本概念, 然后对误差进行简单的分析.

定义 1.1　设某量的准确值为 x^*, x 为 x^* 的一个近似值, 称 $\varepsilon(x) = x^* - x$ 为 x 的绝对误差, 简称误差. 如果存在数 $\bar{\varepsilon}(x)$ 使得 $|\varepsilon(x)| = |x^* - x| \leqslant \bar{\varepsilon}(x)$ 成立, 则称 $\bar{\varepsilon}(x)$ 为 x 的绝对误差限. 进一步, 把近似值的误差 $\varepsilon(x)$ 与准确值 x^* 的比值称为近似值 x 的相对误差, 记为 $\varepsilon_r(x)$, 即

$$\varepsilon_r(x) = \frac{\varepsilon(x)}{x^*} = \frac{x^* - x}{x^*}.$$

如果存在数 $\bar{\varepsilon}_r(x)$, 使得 $|\varepsilon_r(x)| \leqslant \bar{\varepsilon}_r(x)$ 成立, 则称 $\bar{\varepsilon}_r(x)$ 为 x 的相对误差限.

需要说明的是, 尽管由于 x^* 未知, $\varepsilon(x)$ 往往未知, 但我们一般能够根据观测数据的设备及处理数据的工具或方法估计出 x 的误差限. 例如: 用毫米刻度尺测量物体所获长度的误差限为 0.5 个毫米. 无理数 π 在字长为 m 的十进制计算机上表示, 引起的误差不超过 $0.5 \times 10^{1-m}$. 另外, 在实际计算中, 由于 x^* 未知, 通常用 $\dfrac{\varepsilon(x)}{x}$ 近似表示 $\varepsilon_r(x)$.

下面我们给出一个工程上经常用来度量误差大小的概念: 有效数字.

定义 1.2 设 x^* 的近似数 x 的标准形式为

$$x = \pm 10^m \times 0.a_1a_2a_3\cdots a_n\cdots,$$

其中 $a_1 \neq 0, 0 \leqslant a_i \leqslant 9(i=2,3,\cdots,n,\cdots)$, m 为整数. 如果 $\frac{1}{2}\times 10^{m-n-1} < |x^*-x| \leqslant \frac{1}{2}\times 10^{m-n}$, 则称 x 是 x^* 的具有 n 位有效数字的近似数.

由定义可知, 若近似数 x 的误差限不超过 x 的某一位的半个单位 (小数点后第 k 位的半个单位为 0.5×10^{-k}, 小数点前第 k 位的半个单位为 $0.5\times 10^{k-1}$), 且从该位直到 x 的第一位非零数字一共有 n 位, 则近似数 x 至少有 n 位有效数字.

例 4 设 $x^* = 1.234\ 567$ 为精确数, 则

$$1.23,\ 1.234, 1.234\ 5, 1.234\ 6, 1.234\ 567, 1.234\ 567\ 0$$

作为 x^* 的近似数分别具有 3, 3, 4, 5, 7, 8 位有效数字.

定理 1.1 设近似数 x 由下式

$$\pm 10^m \times 0.a_1a_2a_3\cdots a_n\cdots \tag{1.2.1}$$

表示, 其中 $a_1 \neq 0, 0 \leqslant a_i \leqslant 9, i=2,3,\cdots$. 若 x 具有 n 位有效数字, 则其相对误差限为 $\frac{1}{2a_1}\times 10^{-n+1}$; 反之, 若 x 的相对误差限为 $\frac{1}{2(a_1+1)}\times 10^{-n+1}$, 则 x 至少具有 n 位有效数字.

证 由表达式 (1.2.1) 可得 $a_1\times 10^{m-1} \leqslant |x| \leqslant (a_1+1)\times 10^{m-1}$, 当 x 有 n 位有效数字时

$$\varepsilon_r(x) = \frac{|x^*-x|}{|x|} \leqslant \frac{0.5\times 10^{m-n}}{a_1\times 10^{m-1}} = \frac{1}{2a_1}\times 10^{-n+1}.$$

反之, 由

$$|x^*-x| = |x|\varepsilon_r(x) \leqslant (a_1+1)\times 10^{m-1}\times \frac{1}{2(a_1+1)}\times 10^{-n+1} = 0.5\times 10^{m-n},$$

故 x 至少有 n 位有效数字. $\square$

定理 1.1表明有效数字越多, 相对误差限越小; 反之, 相对误差限越小, 有效数字越多.

下面简单介绍一下误差在数值计算过程中的传播过程. 由于数值分析所研究的算法都可以通过有限次四则运算或逻辑运算表示. 假设原问题的初始数据 $\boldsymbol{x}^{(0)}$ 经过 k 次运算得到结果 $\boldsymbol{x}^{(k)}$, 设

$$\boldsymbol{x}^{(k)}=[x_1^{(k)},x_2^{(k)},\cdots,x_{n_k}^{(k)}]^{\mathrm{T}},\quad k=0,1,2,\cdots,$$

并设 $\boldsymbol{x}^{(k-1)}$ 经过第 k 次运算后得到 $\boldsymbol{x}^{(k)}$, 且

$$\begin{cases} x_1^{(k)}=f_1\left(x_1^{(k-1)},x_2^{(k-1)},\cdots,x_{n_{k-1}}^{(k-1)}\right), \\ x_2^{(k)}=f_2\left(x_1^{(k-1)},x_2^{(k-1)},\cdots,x_{n_{k-1}}^{(k-1)}\right), \\ \qquad\cdots\cdots\cdots\cdots \\ x_{n_k}^{(k)}=f_{n_k}\left(x_1^{(k-1)},x_2^{(k-1)},\cdots,x_{n_{k-1}}^{(k-1)}\right), \end{cases}$$

其中 $f_j\left(x_1^{(k-1)},x_2^{(k-1)},\cdots,x_{n_{k-1}}^{(k-1)}\right)(j=1,2,\cdots,n_k)$ 有一阶连续偏导数, 则 $\boldsymbol{x}^{(k)}$ 的误差 $\varepsilon\left(\boldsymbol{x}^{(k)}\right)$ 与 $\boldsymbol{x}^{(k-1)}$ 的误差 $\varepsilon\left(\boldsymbol{x}^{(k-1)}\right)$ 有如下近似关系

$$\begin{aligned}\varepsilon\left(\boldsymbol{x}^{(k)}\right) &\approx \begin{bmatrix} \dfrac{\partial f_1}{\partial x_1^{(k-1)}} & \cdots & \dfrac{\partial f_1}{\partial x_{n_{k-1}}^{(k-1)}} \\ \vdots & & \vdots \\ \dfrac{\partial f_{n_k}}{\partial x_1^{(k-1)}} & \cdots & \dfrac{\partial f_{n_k}}{\partial x_{n_{k-1}}^{(k-1)}} \end{bmatrix} \varepsilon\left(\boldsymbol{x}^{(k-1)}\right)+\delta\left(\boldsymbol{x}^{(k)}\right) \\ &= \boldsymbol{J}_k\varepsilon\left(\boldsymbol{x}^{(k-1)}\right)+\delta\left(\boldsymbol{x}^{(k)}\right),\quad k=1,2,\cdots,\end{aligned}\tag{1.2.2}$$

其中

$$\varepsilon\left(\boldsymbol{x}^{(k)}\right)=\begin{bmatrix}\varepsilon\left(x_1^{(k)}\right)\\ \vdots \\ \varepsilon\left(x_{n_k}^{(k)}\right)\end{bmatrix},\quad \delta(\boldsymbol{x}^{(k)})=\begin{bmatrix}\delta\left(x_1^{(k)}\right)\\ \vdots \\ \delta\left(x_{n_k}^{(k)}\right)\end{bmatrix},$$

$\boldsymbol{J}_k$ 为 Jacobi(雅可比) 矩阵, 称 $\boldsymbol{J}_k\varepsilon\left(\boldsymbol{x}^{(k-1)}\right)$ 为 $\varepsilon\left(\boldsymbol{x}^{(k-1)}\right)$ 经第 k 次运算所得的

传播误差, $\delta\left(\boldsymbol{x}^{(k)}\right)$ 为第 k 次运算新产生的舍入误差. 若 $\delta\left(\boldsymbol{x}^{(k)}\right)$ 相比 $\varepsilon\left(\boldsymbol{x}^{(k)}\right)$ 很小, 则可舍去. 表达式 (1.2.2) 大致描述了在上述算法中误差的传播及产生的过程. 类似地, $\boldsymbol{x}^{(k)}$ 的相对误差 $\varepsilon_r\left(\boldsymbol{x}^{(k)}\right)$ 与 $\boldsymbol{x}^{(k-1)}$ 的相对误差 $\varepsilon_r\left(\boldsymbol{x}^{(k-1)}\right)$ 之间有如下近似关系

$$
\begin{aligned}
\varepsilon_r\left(\boldsymbol{x}^{(k)}\right) &\approx \begin{bmatrix} \dfrac{x_1^{(k-1)}}{f_1}\dfrac{\partial f_1}{\partial x_1^{(k-1)}} & \cdots & \dfrac{x_{n_{k-1}}^{(k-1)}}{f_1}\dfrac{\partial f_1}{\partial x_{n_{k-1}}^{(k-1)}} \\ \vdots & & \vdots \\ \dfrac{x_1^{(k-1)}}{f_{n_k}}\dfrac{\partial f_{n_k}}{\partial x_1^{(k-1)}} & \cdots & \dfrac{x_{n_{k-1}}^{(k-1)}}{f_{n_k}}\dfrac{\partial f_{n_k}}{\partial x_{n_{k-1}}^{(k-1)}} \end{bmatrix} \varepsilon_r\left(\boldsymbol{x}^{(k-1)}\right) + \delta_r\left(\boldsymbol{x}^{(k)}\right) \\
&= \boldsymbol{R}_k \varepsilon_r\left(\boldsymbol{x}^{(k-1)}\right) + \delta_r\left(\boldsymbol{x}^{(k)}\right), \quad k = 1, 2, \cdots,
\end{aligned} \tag{1.2.3}
$$

其中 $\boldsymbol{R}_k\varepsilon_r\left(\boldsymbol{x}^{(k-1)}\right)$ 是 $\varepsilon_r\left(\boldsymbol{x}^{(k-1)}\right)$ 经第 k 次运算传播所得相对误差, $\delta_r\left(\boldsymbol{x}^{(k)}\right)$ 是第 k 次运算中的舍入误差所产生的新的相对误差. 特别地, 由 (1.2.3) 有

$$
\begin{aligned}
\varepsilon_r(x_i^{(k)}) \approx & \left(\frac{x_1^{(k-1)}}{f_i}\frac{\partial f_i}{\partial x_1^{(k-1)}}\right)\varepsilon_r\left(x_1^{(k-1)}\right) + \cdots + \\
& \left(\frac{x_{n_{k-1}}^{(k-1)}}{f_i}\frac{\partial f_i}{\partial x_{n_{k-1}}^{(k-1)}}\right)\varepsilon_r\left(x_{n_{k-1}}^{(k-1)}\right) + \delta_r\left(x_i^{(k)}\right),
\end{aligned}
$$

其中因子

$$
\frac{x_j^{(k-1)}}{f_i}\frac{\partial f_i}{\partial x_j^{(k-1)}}
$$

反映了 $x_j^{(k-1)}$ 的相对误差 $\varepsilon_r(x_j^{(k-1)})$ 对相对误差 $\varepsilon_r(x_i^{(k)})$ 的影响程度.

例 5 设 $x_1^{(0)} = 1.21, x_2^{(0)} = 3.65, x_3^{(0)} = 9.81$ 的误差均为 0.5×10^{-2}, 在字长为 3 的十进制计算机上计算 $x_1^{(0)}x_2^{(0)} + x_3^{(0)}$, 并分析计算结果的误差及相对误差.

解 设

$$
\boldsymbol{x}^{(1)} = \begin{bmatrix} x_1^{(1)} \\ x_2^{(1)} \end{bmatrix} = \begin{bmatrix} fl(x_1^{(0)}x_2^{(0)}) \\ fl(x_3^{(0)}) \end{bmatrix}, \quad x_1^{(2)} = fl(x_1^{(1)} + x_2^{(1)}),
$$

其中 $fl(x)$ 表示 x 在给定计算机上的大小, 则 $x_1^{(2)}$ 为所求结果, 由假设及 (1.2.2)

式知

$$\begin{aligned}
\begin{bmatrix} x_1^{(1)} \\ x_2^{(1)} \end{bmatrix} &= \begin{bmatrix} 4.42 \\ 9.81 \end{bmatrix}, \\
\delta\left(\boldsymbol{x}^{(1)}\right) &= \begin{bmatrix} 4.416\,5 - 4.42 \\ 9.81 - 9.81 \end{bmatrix} = \begin{bmatrix} -0.35 \times 10^{-2} \\ 0.00 \end{bmatrix}, \\
\varepsilon\left(\boldsymbol{x}^{(1)}\right) &\approx \begin{bmatrix} x_2^{(0)} & x_1^{(0)} & 0 \\ 0 & 0 & 1 \end{bmatrix} \varepsilon\left(\boldsymbol{x}^{(0)}\right) + \delta\left(\boldsymbol{x}^{(1)}\right) \\
&= \begin{bmatrix} 3.65 \times 0.5 \times 10^{-2} + 1.21 \times 0.5 \times 10^{-2} - 0.35 \times 10^{-2} \\ 0.5 \times 10^{-2} \end{bmatrix} \\
&= \begin{bmatrix} 2.08 \times 10^{-2} \\ 0.5 \times 10^{-2} \end{bmatrix}; \\
x_1^{(2)} &= 14.2, \delta(x_1^{(2)}) = 0.03, \\
\varepsilon(x_1^{(2)}) &\approx \varepsilon(x_1^{(1)}) + \varepsilon(x_2^{(1)}) + \delta(x_1^{(2)}) = 5.58 \times 10^{-2},
\end{aligned}$$

因此, 所求结果的相对误差为

$$\varepsilon_r(x_1^{(2)}) \approx 5.58 \times 10^{-2}/14.2 \approx 0.393 \times 10^{-2}.$$

例 6 利用下列等价表达式分别计算 $(\sqrt{2}-1)^6$, 试分析当 $\sqrt{2} \approx 1.4$ 时各计算结果的误差.

$$\frac{1}{(\sqrt{2}+1)^6}, \ (3-2\sqrt{2})^3, \ \frac{1}{(3+2\sqrt{2})^3}, \ 99-70\sqrt{2}.$$

解 设 $x^* = \sqrt{2}$, $x = 1.4$, $|x^* - x| = |\sqrt{2} - 1.4| \leqslant \bar{\varepsilon}(x)$, 其中 $\bar{\varepsilon}(x)$ 为 x 近似 x^* 的误差限. 令

$$\begin{aligned}
&f_1(t) = (t+1)^{-6}, \ f_2(t) = (3-2t)^3, \\
&f_3(t) = (3+2t)^{-3}, \ f_4(t) = 99 - 70t.
\end{aligned}$$

则本题仅考虑 x 的误差对计算结果的影响, 即考虑 $f_i(x)$ 近似 $f_i(x^*)(i = 1,2,3,4)$ 的误差. 由误差估计式 (1.2.2) 知