SHUZHI JISUAN FANGFA

China University of Mining and Technology Press

中国矿业大学研究生教育专项资金资助出版教材

数值计算方法

Shuzhi Jisuan Fangfa

主编 胡建华 陈兴同 曹德欣

中国矿业大学出版社
China University of Mining and Technology Press

中国矿业大学研究生教育专项资金资助出版教材

数值计算方法

主编　胡建华　陈兴同　曹德欣

中国矿业大学出版社

内 容 提 要

本书是为普通院校工科硕士研究生编写的教材。全书共分 10 章，内容包括：绪论、线性方程组的直接解法、函数插值、函数逼近、数值积分法、线性方程组的迭代解法、非线性方程(组)的数值解法、数值最优化、常微分方程的数值解法、矩阵特征值问题的数值解法。

本书系统介绍各种算法的数学原理，注重算法的实现和应用。书中各种算法都是用 Matlab 语言格式来描述的，有的直接给出源代码，便于学生上机编程实现。每种算法都附有数值算例，同时每章后还配备大量的数值实验和习题。

本书还可作为理工类院校本科计算专业的教材，也可供从事数值计算的科技人员参考。

图书在版编目(CIP)数据

数值计算方法/胡建华，陈兴同，曹德欣主编．—徐州：
中国矿业大学出版社，2008.9
ISBN 978-7-5646-0065-5

Ⅰ.数… Ⅱ.①胡…②陈…③曹… Ⅲ.数值计算—计算方法—研究生—教材 Ⅳ.O241

中国版本图书馆 CIP 数据核字(2008)第 137341 号

书　　名　数值计算方法
主　　编　胡建华　陈兴同　曹德欣
责任编辑　王江涛
出版发行　中国矿业大学出版社
（江苏省徐州市中国矿业大学内　邮编 221008）
网　　址　http://www.cumtp.com　**E-mail**:cumtpvip@cumtp.com
排　　版　中国矿业大学出版社排版中心
印　　刷　徐州中矿大印发科技有限公司
经　　销　新华书店
开　　本　787×960　1/16　**印张** 15　**字数** 277 千字
版次印次　2008 年 9 月第 1 版　2008 年 9 月第 1 次印刷
定　　价　25.00 元

前　言

数值计算方法又称数值分析，是研究用计算机求解各种数学问题的一个数学分支。它研究的对象都是科学与工程中经常遇到的数学问题，数值计算往往成为问题解决的关键。因此，数值计算方法是进行科学研究与工程设计的基础。数值计算方法不仅涉及算法构造的数学原理和算法的理论分析，而且涉及算法的计算机实现和实际应用。掌握数值计算方法的基本原理，熟练地利用计算机解决科学与工程中的计算问题，已成为高等院校理工科大学生、研究生必备的基本技能。本教材就是基于这样的考虑而编写的。

本教材具有下列特点：

(1) 门槛低，内容广。只要具有高等数学与线性代数的基本知识就可以学习本书，内容包括线性代数、微积分、函数逼近、微分方程、最优化等数学问题的数值解法。

(2) 注重算法，兼顾理论。重点讲述算法的构造原理，强化数学软件 Matlab 的应用。所有算法都是用 Matlab 语言格式来描述的，有的直接给出源代码。对于成熟的算法还给出了 Matlab 函数，简要介绍这些函数的调用方法。与此同时，兼顾理论知识，着重介绍方法的数学原理，弱化了繁琐的理论推导和证明。

(3) 理论联系实际。注重介绍数学问题的应用背景，对于主要算法给出数值算例。每章后面都配备了大量的数值实验和习题。它们大多都具有直接应用价值。

按照我们的设想，对工科研究生教学，完成本教材的教学大约需要 60 学时，另外上机实验大约需要 12 学时；对理工科本科生教学，可以根据学时数有选择地讲授部分内容。

本教材是编者在多年教学工作的经验基础上编写的，并参考了许多国内外教材。本书主要由胡建华、陈兴同、曹德欣编写，由胡建华统编。王海军、马国选、韩超、芮文娟、范胜君等老师也参与了部分编写工作。

中国矿业大学研究生院、出版社对本书的编写出版给予了大力支持。对此，编者表示衷心感谢。

由于成书时间仓促，作者水平有限，取材和叙述方式难免有不妥之处，恳请读者和专家们指正。

编　者

2008 年 7 月

目　录

第一章 绪 论

§1 课程研究的内容和构造算法的主要途径

1.1 课程研究的内容

数值计算方法是研究用计算机求解各种数学问题的计算方法及其理论的一门学科。数值计算方法也称计算方法、数值分析或科学计算等。本书所涉及的数学问题主要有线性方程组求解、非线性方程求解、函数插值、数据拟合、微积分、常微分方程、矩阵特征值和最优化等。

利用计算机解决实际问题一般需要经过以下几个环节：

建立数学模型→设计计算方法→编制程序→上机运行输出结果→检验结果给出答案

其中，从第二个环节到最后一个环节是本课程的主要任务。在这个过程中还要对算法进行理论分析，如收敛性分析、误差分析、数值稳定性分析和计算复杂性分析等。

下面通过两个例子来初步了解一下本课程所研究的内容及其重要性。

例 1.1 求方程 $x^2-2=0$ 的正根。

这是方程求根问题，显然其精确解（也称解析解）为 $x^*=\sqrt{2}$。但是$\sqrt{2}$只是表示一个实数的符号，在实际应用中，我们更关心它的近似值（也称数值解）是多少。为求得$\sqrt{2}$的近似值，至少要做以下四个方面的工作：

第一，要根据一定的理论，设计求数值解的计算方法（简称算法）。所谓算法，是指用计算机能够直接处理的加、减、乘、除运算和逻辑运算写成的一系列的计算公式。

比如，用 Newton 迭代法求解（详见第七章）：

任取初值 $x_0>0$，有

$$x_{k+1}=\frac{1}{2}\left(x_k+\frac{2}{x_k}\right) \quad (k=0,1,2,\cdots) \tag{1.1}$$

算法(1.1)是一个迭代算法，如果能证明 $x_k\to\sqrt{2}\,(k\to\infty)$，则当 k 较大时，有理由把 x_k 作为$\sqrt{2}$的近似值。

第二，要进行收敛性分析，即要证明 $x_k \to \sqrt{2}\ (k \to \infty)$。由迭代公式(1.1)得

$$x_{k+1} - \sqrt{2} = \frac{1}{2x_k}(x_k^2 - 2x_k\sqrt{2} + 2) = \frac{1}{2x_k}(x_k - \sqrt{2})^2$$

因此，对任意 $x_0 > 0$，都有 $x_k > \sqrt{2}\ (k = 1, 2, \cdots)$。另外又有

$$x_k - x_{k+1} = \frac{1}{2x_k}(x_k^2 - 2) > 0 \quad (k = 1, 2, \cdots)$$

因此 $\{x_k\}_{k=1}^{\infty}$ 是单调递减的数列，且有下界 $\sqrt{2}$。根据单调有界定理，$\{x_k\}$ 必有极限(设极限为 A)。在式(1.1)两边取极限，即得 $A = \sqrt{2}$。这说明，迭代算法(1.1)对任意初值 $x_0 > 0$ 都是收敛的。

第三，要编程上机计算，输出结果。下面是用 Matlab 编写的程序：

```
format long
x=1;
for k=1:5
  k
  x=(x+2/x)/2
end
```

输出结果为：

k	x_k
1	1.500 000 000 000 00
2	1.416 666 666 666 67
3	1.414 215 686 274 51
4	1.414 213 562 374 69
5	1.414 213 562 373 09

第四，要进行误差估计，即估计 x_k 与 $\sqrt{2}$ 的误差有多大，是否达到要求的精度。记

$$\varphi(x) = \frac{1}{2}\left(x + \frac{2}{x}\right)$$

则迭代公式(1.1)为：$x_{k+1} = \varphi(x_k)$ 且 $\varphi(\sqrt{2}) = \sqrt{2}$。由于 $x_0 = 1, x_1 = 1.5$，由第二步的收敛性分析知，$x_k \in [\sqrt{2}, 1.5]\ (k = 1, 2, \cdots)$。又 $\varphi'(x) = \frac{1}{2} - \frac{1}{x^2}$，所以当 $x \in [\sqrt{2}, 1.5]$ 时，有

$$|\varphi'(x)|\leqslant\frac{1}{2}-\frac{1}{1.5^2}<0.06\equiv L$$

再由 Lagrange 中值定理，得

$$|x_{k+1}-\sqrt{2}|=|\varphi(x_k)-\varphi(\sqrt{2})|=|\varphi'(\xi)||x_k-\sqrt{2}|<L|x_k-\sqrt{2}|$$

根据上式递推并注意到 $x_1=1.5$，有

$$|x_{k+1}-\sqrt{2}|<L^k|x_1-\sqrt{2}|<0.1\times L^k$$

对于所求的 x_5 有如下误差估计

$$|x_5-\sqrt{2}|<0.1\times L^4<0.13\times 10^{-5}$$

这只是一个非常保守的估计，实际上 $|x_5-\sqrt{2}|<0.52\times 10^{-14}$。

另外，由于我们在计算 x_k 时不是精确运算，而是用计算机进行的带有舍入的近似运算，因此还要对计算结果进行舍入误差分析等工作。

例 1.2 考察常微分方程的初值问题：

$$\begin{cases}y'=f(x,y)=1-2xy\\ y(0)=0\end{cases}\tag{1.2}$$

容易验证问题(1.2)的解析解为

$$y(x)=\mathrm{e}^{-x^2}\int_0^x\mathrm{e}^{t^2}\,\mathrm{d}t\tag{1.3}$$

由于解的表达式(1.3)中函数 $f(t)=\mathrm{e}^{t^2}$ 的原函数 $F(t)$ 不能用初等函数表示，因此无法用 Newton-Leibniz 公式计算定积分 $\int_0^x\mathrm{e}^{t^2}\,\mathrm{d}t$ 的值(实际上，e^{-x^2} 的值也无法直接得到)。在实际应用中，往往不需要知道解的精确表达式，只要知道解 $y(x)$ 在一些离散点上的近似值(即数值解)就够了。例如，我们需要知道 $y(0.1)$，$y(0.2)$，…，$y(1)$ 的近似值。

采用二阶 Runge-Kutta 算法(详见第九章)：

$x_0=0,y_0=0,h=0.01$

for $n=0:99$

$k_1=1-2x_ny_n$

$k_2=1-2(x_n+h)(y_n+hk_1)$

$y_{n+1}=y_n+\frac{h}{2}(k_1+k_2)$

$x_{n+1}=x_n+h$

end

即可算得(具体程序略)

$y(0.1)\approx y_{10}=0.099\ 3,y(0.2)\approx y_{20}=0.194\ 7,\cdots,y(1)\approx y_{100}=0.538\ 1$

理论分析表明，精确值 $y(x_n)$ 与计算值 y_n 的误差为

$$|y_n-y(x_n)|\leqslant Ch^2(C\text{ 为常数}) \tag{1.4}$$

以后常把式(1.4)记为 $|y_n-y(x_n)|=O(h^2)$。

1.2 构造算法的主要途径

构造算法的主要途径大体有以下六种：

(1) 迭代技术；

(2) 解析问题离散化技术；

(3) 离散问题解析化技术；

(4) 非线性问题线性化技术；

(5) 优化技术；

(6) 加速技术。

例 1.1 采用的是迭代技术，例 1.2 采用的是解析问题离散化技术。第六章(线性方程组的迭代解法)、第七章(非线性方程(组)的数值解法)、第十章(矩阵特征值问题的数值解法)主要采用的是迭代技术；第五章(数值积分法)和第九章(常微分方程的数值解法)主要采用的是解析问题离散化技术；第三章(函数插值)和第四章(函数逼近)主要采用的是离散问题解析化技术。其它技术在以后各章中均有体现。

§2 误 差

2.1 误差的来源

在科学计算中，估计计算结果的精确度是十分重要的，而影响精确度的是各种各样的误差。误差按其来源大体可分为四种，即模型误差、观测误差、截断误差和舍入误差。

1. 模型误差

由实际问题建立数学模型(如微分方程等)，要忽略一些次要因素，简化许多条件。数学模型是对实际问题近似的反映。实际问题的解与数学模型的解之间的误差称为模型误差。

2. 观测误差

数学模型中包括的一些参数(主要是物理量，如时间、长度、电压等)往往是通过观察、测量得到的。由于受到测量工具的限制，测量值只能是近似的。称测量值与真值之间的误差为观测误差。

本课程不对模型误差与观测误差进行研究，主要研究截断误差和舍入误差。

3. 截断误差

求解数学模型(即数学问题)所用的计算方法如果是一种近似的方法，那么只能得到数学问题的近似解。由此产生的误差称为截断误差，也称方法误差。

例 2.1 计算 e^x 的近似值。

由 Taylor 展开公式

$$e^x=1+x+\frac{x^2}{2!}+\cdots+\frac{x^n}{n!}+R_n(x)$$

其中，$R_n(x)=\frac{e^{\xi}}{(n+1)!}x^{n+1}$ 为 Lagrange 余项。如果把余项 $R_n(x)$ 舍掉，就得到计算 e^x 值的近似公式(计算方法)：

$$e^x\approx P_n(x)=1+x+\frac{x^2}{2!}+\cdots+\frac{x^n}{n!}$$

该计算方法的截断误差为

$$e^x-P_n(x)=R_n(x)=\frac{e^{\xi}}{(n+1)!}x^{n+1}$$

例 2.2 计算导数 $f'(x_0)$ 的近似值。

由于 $f'(x_0)=\lim\limits_{h\to 0}\frac{f(x_0+h)-f(x_0)}{h}$，故当 $|h|$ 很小时，得计算导数 $f'(x_0)$ 的近似方法

$$f'(x_0)\approx\frac{f(x_0+h)-f(x_0)}{h}\equiv f[x_0,x_0+h]$$

由 Taylor 展开公式：$f(x_0+h)=f(x_0)+hf'(x_0)+\frac{h^2}{2}f''(\xi)$，得

$$f'(x_0)=\frac{f(x_0+h)-f(x_0)}{h}-\frac{h}{2}f''(\xi)$$

从而该方法的截断误差为

$$f'(x_0)-f[x_0,x_0+h]=-\frac{h}{2}f''(\xi)$$

截断误差反映的是计算方法(一般是近似的)的解与原问题的解之间的近似程度，与算法的收敛性有密切关系。

4. 舍入误差

有了求解数学问题的计算方法(公式)后，用计算机进行数值计算时，由于计算机的字长有限，参与计算的数据以及计算结果只能用有限位数字存放，要进行“四舍五入”，这时产生的误差称为舍入误差。进一步说明如下：

计算机采用的是二进制，为更符合习惯，我们采用十进制模拟计算机来阐述

问题。

计算机通常使用规格化浮点数系统来近似表示实数系。所谓规格化浮点数系统为

$$F=\{\pm 0.d_1d_2\cdots d_t\times 10^m\}\cup\{0\} \tag{2.1}$$

其中，$d_1,d_2,\cdots,d_t$ 是 0,1,2,…,9 中的数字且 $d_1\neq 0$，$0.d_1d_2\cdots d_t$ 称为尾数，尾数位数 t（也称字长）总是固定的，m 为整数，满足 $L\leqslant m\leqslant U$，其中 L,U 是给定的整数，m 称为阶数。F 中的数称为机器数。

显然机器数全体 F 是一个离散的有限集合。记机器数中的最小正数为 realmin，最大正数为 realmax，则任何一个机器数 $x(x\neq 0)$ 满足：realmin$\leqslant|x|\leqslant$realmax。如果一个数（除零外）不在这个范围内，则产生溢出。以后假设所讨论的数均不产生溢出。

Matlab 中，尾数长在十进制下大约为 15 位，最小正数与最大正数分别为

$$\text{realmin}\approx 2.225\ 1\times 10^{-308},\ \text{realmax}\approx 1.797\ 7\times 10^{308} \tag{2.2}$$

当一个实数 x 不能被机器精确表示时，一般采用四舍五入原则进行舍入，记为 $fl(x)$ 为 x 经过舍入后的数。显然，$fl(x)$ 是 F 中最接近 x 的数。称 $x-fl(x)$ 为舍入误差。

例如，假设机器的尾数长度为 5 位，$\pi=0.314\ 159\ 26\cdots\times 10^1$，则 $fl(\pi)=0.314\ 16\times 10^1$，舍入误差 $\pi-fl(\pi)=-0.73\cdots\times 10^{-5}$。

约定：以后我们说“用 5 位浮点数计算”是指在尾数为 5 位的十进制浮点数系下计算。这只是为了方便来模拟计算机讨论问题而已。

为了刻画一个近似数的精确程度，常使用绝对误差、相对误差和有效数字这三个量。

2.2 绝对误差与相对误差

定义 2.1 设 $\hat{x}$ 是准确值 x 的一个近似值，则称

$$e(\hat{x})=x-\hat{x} \tag{2.3}$$

为 $\hat{x}$ 的绝对误差，简称误差。一般 x 是未知的，只能估计误差的大小。如果

$$|e(\hat{x})|=|x-\hat{x}|\leqslant\varepsilon(\hat{x}) \tag{2.4}$$

则称 $\varepsilon(\hat{x})$ 为 $\hat{x}$ 的一个绝对误差限，也称绝对误差界，简称误差限或误差界。

习惯上常把误差限说成误差。

例 2.3 假设计算机的尾数为 t 位，$fl(x)=\pm 0.d_1d_2\cdots d_t\times 10^m$（规格化），则

$$|x-fl(x)|\leqslant 0.5\times 10^{-t}\times 10^m\equiv\varepsilon \tag{2.5}$$

ε 就是任何一个数的舍入误差限。

用绝对误差来刻画近似值的精确程度是有局限性的，因为它没有考虑原数的大小。

例如，$x=100$ cm，$\hat{x}=99$ cm，绝对误差 $e(\hat{x})=1$ cm，而 $y=10\ 000$ cm，$\hat{y}=9\ 950$ cm，绝对误差 $e(\hat{y})=50$ cm，从表面上看，后者的误差是前者的 50 倍，但后者每厘米产生的误差是 0.005 cm，而前者每厘米产生的误差是 0.01 cm。

定义 2.2 设 $\hat{x}$ 是准确值 x 的一个近似值，则称

$$e_r(\hat{x})=\frac{x-\hat{x}}{x} \quad (x\neq 0) \tag{2.6}$$

为 $\hat{x}$ 的相对误差。如果

$$|e_r(\hat{x})|\leqslant \varepsilon_r(\hat{x}) \tag{2.7}$$

则称 $\varepsilon_r(\hat{x})$ 为 $\hat{x}$ 的一个相对误差限。

设 $\hat{x}$ 的绝对误差限为 $\varepsilon(\hat{x})$，即 $|e(\hat{x})|=|x-\hat{x}|\leqslant\varepsilon(\hat{x})$，通常把

$$\varepsilon_r(\hat{x})=\frac{\varepsilon(\hat{x})}{|\hat{x}|} \tag{2.8}$$

作为 $\hat{x}$ 的一个相对误差限。

要确定一个近似数的精确程度，除了要看绝对误差的大小，还要看相对误差的大小。使用较多的是相对误差。通常所说的“精度”一般指相对误差。

例 2.4 承例 2.3，由式(2.5)有

$$|e_r|=\frac{|x-fl(x)|}{|x|}\leqslant\frac{0.5\times 10^{-t}\times 10^{m}}{0.1\times 10^{m}}=0.5\times 10^{-t+1}\equiv eps \tag{2.9}$$

eps 就是任何一个数的机器舍入的相对误差限，称之为机器精度。

Matlab：$eps\approx 2.220\ 4\times 10^{-16}$。

2.3 有效数字

为了反映一个近似数的精确程度，除使用绝对误差和相对误差外，还经常使用有效数字。

式(2.5)给出了一个数的舍入误差限，现在倒过来看给出下面定义：

定义 2.3 假设机器尾数为 t 位，x 的近似值 $\hat{x}=\pm 0.d_1d_2\cdots d_t\times 10^{m}$（规格化）。如果

$$|x-\hat{x}|\leqslant 0.5\times 10^{-n}\times 10^{m} \tag{2.10}$$

则称 $\hat{x}$（至少）具有 n 位有效数字。

例如，$x=0.509\ 966\ 6\times 10^{-1}$，机器尾数为 4 位，

$\hat{x}_1=0.500\ 0\times 10^{-1}$，$|x-\hat{x}_1|=0.99\cdots\times 10^{-3}\leqslant 0.5\times 10^{-1}\times 10^{-1}$，$\hat{x}_1$ 具有 1 位有效数字；

$\hat{x}_2=0.509\ 0\times 10^{-1}$，$|x-\hat{x}_2|=0.96\cdots\times 10^{-4}\leqslant 0.5\times 10^{-2}\times 10^{-1}$，$\hat{x}_2$ 具有

2 位有效数字；

$\hat{x}_3=0.509\ 9\times10^{-1}$，$|x-\hat{x}_3|=0.66\cdots\times10^{-5}\leqslant0.5\times10^{-3}\times10^{-1}$，$\hat{x}_3$ 具有 3 位有效数字；

$\hat{x}_4=0.510\ 0\times10^{-1}$，$|x-\hat{x}_4|=0.33\cdots\times10^{-5}\leqslant0.5\times10^{-4}\times10^{-1}$，$\hat{x}_4$ 具有 4 位有效数字。

有效数字是用绝对误差限定义的，它不仅反映了绝对误差的大小，还反映了相对误差的大小。见下面定理。

定理 2.1 设 x 的近似值 $\hat{x}=\pm0.d_1d_2\cdots d_t\times10^m$（规格化），如果 $\hat{x}$ 具有 n 位有效数字，则 $\hat{x}$ 的相对误差限为

$$|e_r(\hat{x})|=\left|\frac{x-\hat{x}}{x}\right|\leqslant\frac{1}{2d_1}\times10^{-n+1}\leqslant\frac{1}{2}\times10^{-n+1} \tag{2.11}$$

反之，如果 $\hat{x}$ 的相对误差满足

$$|e_r(\hat{x})|\leqslant\frac{1}{2(d_1+1)}\times10^{-n+1}\left(\text{特别地，}|e_r(\hat{x})|\leqslant\frac{1}{2}\times10^{-n}\right) \tag{2.12}$$

则 $\hat{x}$ 至少具有 n 位有效数字。

定理的证明是容易的，留给读者完成。

定理 2.1 说明，如果 $\hat{x}$ 具有 n 位有效数字，则 $\hat{x}$ 的相对误差大约为 10^{-n} 量级；反之，如果 $\hat{x}$ 的相对误差为 10^{-n} 量级，则 $\hat{x}$ 大约有 n 位有效数字。

2.4 舍入误差分析

由于计算机每秒做几亿次运算，几乎每次运算都会产生舍入误差，因此对舍入误差的跟踪和估计是非常困难的。目前已提出的舍入误差分析方法主要有向前误差分析法、向后误差分析法、区间分析法和概率分析法。我们只简要介绍前两种误差分析法。

1. *向前误差分析法*

假设 y 由原始数据 $x_1,x_2,\cdots,x_n$ 经过基本的算术运算得到，记为 $y=\varphi(x_1,x_2,\cdots,x_n)$。若 x_i 的近似值为 $\hat{x}_i$，误差限为 $\varepsilon(\hat{x}_i)$，计算得到的 y 的近似值为 $\hat{y}$，向前误差分析法就是用 $\hat{x}_i$ 的误差 $\varepsilon(\hat{x}_i)$ 直接估计 $\hat{y}$ 的误差 $\varepsilon(\hat{y})$。下面以二元函数求值为例，说明向前误差分析的方法。

设 $z=f(x,y)$，假设 $\hat{x}$ 是 x 的近似值，误差限为 $\varepsilon(\hat{x})$，$\hat{y}$ 是 y 的近似值，误差限为 $\varepsilon(\hat{y})$，$\hat{z}=f(\hat{x},\hat{y})$，下面估计 $z-\hat{z}$ 的大小。由 Taylor 展开公式

$$\begin{aligned}z-\hat{z}&=f(x,y)-f(\hat{x},\hat{y})\\&=\frac{\partial f(\hat{x},\hat{y})}{\partial x}(x-\hat{x})+\frac{\partial f(\hat{x},\hat{y})}{\partial y}(y-\hat{y})+R\end{aligned} \tag{2.13}$$

其中，$R=\frac{1}{2}\left[\frac{\partial^2 f(\xi,\eta)}{\partial x^2}(x-\hat{x})^2+2\frac{\partial^2 f(\xi,\eta)}{\partial x\partial y}(x-\hat{x})(y-\hat{y})+\frac{\partial^2 f(\xi,\eta)}{\partial y^2}(y-\hat{y})^2\right]$。

假设 R 中二阶偏导数有界，则可以忽略 R，把式(2.13)写为

$$|z-\hat{z}|\approx\left|\frac{\partial f(\hat{x},\hat{y})}{\partial x}(x-\hat{x})+\frac{\partial f(\hat{x},\hat{y})}{\partial y}(y-\hat{y})\right|\leqslant\left|\frac{\partial f(\hat{x},\hat{y})}{\partial x}\right|\varepsilon(\hat{x})+\left|\frac{\partial f(\hat{x},\hat{y})}{\partial y}\right|\varepsilon(\hat{y})$$

从而

$$\varepsilon(\hat{z})\approx\left|\frac{\partial f(\hat{x},\hat{y})}{\partial x}\right|\varepsilon(\hat{x})+\left|\frac{\partial f(\hat{x},\hat{y})}{\partial y}\right|\varepsilon(\hat{y}) \tag{2.14}$$

特别地，如果 $z=f(x,y)$ 为 x 与 y 的加、减、乘、除四则运算时，有(以后常把“≈”写成“=”)

$$\varepsilon(\hat{x}\pm\hat{y})=\varepsilon(\hat{x})+\varepsilon(\hat{y}) \tag{2.15}$$

$$\varepsilon(\hat{x}\cdot\hat{y})=|\hat{y}|\varepsilon(\hat{x})+|\hat{x}|\varepsilon(\hat{y}) \tag{2.16}$$

$$\varepsilon(\hat{x}/\hat{y})=\frac{|\hat{y}|\varepsilon(\hat{x})+|\hat{x}|\varepsilon(\hat{y})}{|\hat{y}|^2} \tag{2.17}$$

由式(2.15)又得相对误差

$$\varepsilon_r(\hat{x}\pm\hat{y})=\frac{\varepsilon(\hat{x})+\varepsilon(\hat{y})}{|x\pm y|}=\frac{|x|\varepsilon_r(\hat{x})+|y|\varepsilon_r(\hat{y})}{|x\pm y|} \tag{2.18}$$

式(2.16)表明，两个数在作乘法时，如果有一个绝对值很大，则绝对误差会很大。

式(2.17)表明，两个数在作除法时，如果分母的绝对值很小，则绝对误差会很大。

式(2.18)表明，两个数在作减法时，如果这两个数很接近，则相对误差会很大，也就是会严重丢失有效数字。

例 2.5 测得某矩形场地的长 l 的值为 $\hat{l}=110$ m，宽 d 的值为 $\hat{d}=80$ m，已知 $|l-\hat{l}|\leqslant 0.2$ m，$|d-\hat{d}|\leqslant 0.1$ m，求面积 $\hat{S}=\hat{l}\hat{d}$ 的绝对误差限与相对误差限。

解 由式(2.16)得 $\hat{S}$ 的绝对误差限为

$$\varepsilon(\hat{S})=|\hat{l}|\varepsilon(\hat{d})+|\hat{d}|\varepsilon(\hat{l})=110\times 0.1+80\times 0.2=27\ \text{m}^2$$

从而 $\hat{S}$ 的相对误差限为

$$\varepsilon_r(\hat{S})=\frac{\varepsilon(\hat{S})}{|\hat{S}|}=\frac{27}{110\times 80}=0.31\%$$

2. *向后误差分析法*

假设 y 由原始数据 $x_1,x_2,\cdots,x_n$ 经过基本的算术运算得到，记为 $y=\varphi(x_1,x_2,\cdots,x_n)$。把计算得到的 y 的近似值 $\hat{y}$ 归结为有扰动的原始数据 $x_i+\Delta x_i$ 精确运算的结果，即

$$\hat{y}=\varphi(x_1+\Delta x_1,x_2+\Delta x_2,\cdots,x_n+\Delta x_n)$$

然后对 $|\Delta x_i|$ 进行估计：$|\Delta x_i|\leqslant|x_i|\varepsilon$，$\varepsilon$ 越小，说明舍入误差对算法的影响越小。

这种误差分析的方法称为向后误差分析法。

定理 2.2 设机器精度为 eps，则

$$fl(x)=x(1+\varepsilon),|\varepsilon|\leqslant eps \tag{2.19}$$

进而，设 x,y 为机器数，用$\circ$来表示$+,-,\times,/$中的任一种运算，则

$$fl(x\circ y)=(x\circ y)(1+\varepsilon),|\varepsilon|\leqslant eps \tag{2.20}$$

证明 由例 2.4，记$\dfrac{fl(x)-x}{x}=\varepsilon$，则$|\varepsilon|\leqslant eps$，所以 $fl(x)=x(1+\varepsilon),|\varepsilon|\leqslant eps$。证毕。

对式(2.20)中的加法运算，有

$$fl(x+y)=(x+y)(1+\varepsilon)=x(1+\varepsilon)+y(1+\varepsilon)\equiv(x+\Delta x)+(y+\Delta y)$$

上式可以把 x 加 y 浮点运算的结果 $fl(x+y)$看做有扰动的原始数据 $x+\Delta x$ 与 $y+\Delta y$ 精确运算的结果，而且$|\Delta x|\leqslant|x|eps,|\Delta y|\leqslant|y|eps$。这就是一个向后误差分析的结果。

关于舍入误差的分析，本书不作深入的讨论，必要时给出有关结果。

§3 有效算法要具备的条件

有效算法(也称实用算法)一般要具备下面三个条件：

(1)“快”——指计算机耗时少，计算速度快。包括计算步骤少，收敛速度快。

(2)“准”——指计算结果准确、可靠。即算法的数值稳定性好。

(3)“省”——指占用计算机内存少。

下面作进一步解释。

3.1 计算速度要快

为提高计算速度，固然可以通过改善计算机硬件来实现，但这不是本课程所要研究的。本课程研究的重点是通过“软”的方法来提高计算速度，这主要包括构造计算量少的方法和构造收敛速度快的方法。

例 3.1 (Hornor 算法)考察多项式求值的问题：输入 x，计算

$$P_n(x)=a_nx^n+a_{n-1}x^{n-1}+\cdots+a_1x+a_0$$

的值。

方法一：采用自然顺序计算

$$P_n(x)=a_n\cdot\underbrace{x\cdot x\cdots x}_{n}+a_{n-1}\cdot\underbrace{x\cdot x\cdots x}_{n-1}+\cdots+a_1\cdot x+a_0$$