杨大地　涂光裕　编

重庆大学出版社

数值分析

SHUZHI FENXI

数值分析

杨大地　涂光裕　编

重庆大学出版社

内容提要

本书系统地介绍了数值计算的基本概念，常用算法及有关的理论分析和应用．全书共分9章，主要内容包含了数值计算中的基本问题．如线性方程组的数值解法，矩阵特征值和特征向量的数值解法，非线性方程的数值解法，插值方法、数据拟合和函数逼近，数值积分以及常微分方程初值问题的数值解法等．本书基本概念叙述清晰，理论分析较为严谨，语言通俗易懂，并注重算法的实际应用．各章都给出典型例题并配有一定数量的习题．可作为理工科大学教科书，亦可供工程技术人员参考使用．

数值分析

杨大地　涂光裕　编

责任编辑　肖顺杰

*

重庆大学出版社出版发行

新　华　书　店　经　销

四川外语学院印刷厂印刷

*

开本：787×1092　1/16　印张：10.25　字数：256千

1998年1月第1版　1998年1月第1次印刷

印数：1-3000

ISBN 7-5624-1601-X/O・153　定价：16.00元

前　言

随着电子计算机的迅速发展，工科院校开设数值分析课程越来越普遍. 本书是在作者编写的《数值分析讲义》的基础上修订出版的. 该讲义在重庆大学本科生和硕士研究生中使用了五届. 结合师生们提出的意见，作者进行了多次的整理、订正，在此基础上出版了现在的《数值分析》教材.

学习本书必需的数学基础是微积分、线性代数和常微分方程，这是一般理工科大学生都具备的. 全书设计讲授时数为 72 学时左右. 如学时少于 72 学时，对目录中带 * 的章节可以少讲或不讲. 本书编写时已注意到各章节的独立性，删掉带 * 的章节不至于影响后面的学习. 各章后均附有习题. 和本书配套的还有《计算实习指导》，教师可配合布置习题，安排上机实习的教学环节.

本书共分 9 章，其中第一章至第五章、第九章由杨大地同志编写；第六章、第八章由涂光裕同志编写；第七章系二人合作编写. 全书由杨大地同志统稿. 杨万年教授主持确定了本书的编写计划，段虞荣教授审阅了全稿，他们对本书提供了许多宝贵的建议和意见，这里一并表示感谢.

由于计算数学发展迅速，作者的水平有限，本书的编写又比较仓促，缺点和错误在所难免. 恳请读者提出意见和建议，以期修订时改进完善.

作者

1997 年 9 月

目　录

第一章 绪 论

运用数学方法解决科学研究或工程技术问题，一般按如下途径进行：

实际问题→模型设计→算法设计→程序设计→上机计算→问题的解.

其中算法设计是数值分析课程的主要内容.

数值分析课程研究常见的基本数学问题的数值解法. 包含了数值代数(线性方程组的解法、非线性方程的解法、矩阵求逆、矩阵特征值计算等)、数值逼近、数值微分与数值积分、常微分方程及偏微分方程的数值解法等. 它的基本理论和研究方法建立在数学理论基础之上，研究对象是数学问题，因此它是数学的分支之一.

但它又与计算机科学有密切的关系. 我们在考虑算法时，往往要同时考虑计算机的特性，如计算速度、存贮量、字长等技术指标，考虑程序设计时的可行性和复杂性. 如果我们具备了一定的计算机基础知识和程序设计方法，学习数值分析的理论和方法就会更深刻、更实际，选择或设计的算法也会更合理、更实用.

在科学研究、工程实践和经济管理等工作中，存在大量的科学计算、数据处理等问题. 应用计算机解决数值计算问题是理工科大学生应当具备的基本能力.

§1.1 算 法

解决某类数学问题的数值方法称为算法. 为使算法能在计算机上实现，它必须将一个数学问题分解为有限次的+、−、×、÷运算和一些简单的基本函数运算.

1.1.1 算法的表述形式

算法的表述形式是多种多样的.

1. 用数学公式和文字说明描述，这种方式符合人们的理解习惯，和算法的推证相衔接，易于学习接受，但离上机应用距离较大.

2. 用框图描述，这种方式描述计算过程流向清楚，易于编制程序，但对初学者有一个习惯过程. 此外框图描述格式不很统一，详略难以掌握.

3. 算法程序，即用计算机语言描述的算法，它是面对计算机的算法，可以是印刷文本，也可以是磁介质上存贮的文件. 实际上我们以后讨论的算法，都有现成的程序文本和软件可资利用. 一个合格的计算工作者，应当能熟练地应用这些已有的软件工具. 但从学习算法的角度看，这种描述方式并不有利.

本教材将采用第一种方式表述各种算法.

1.1.2 算法的基本特点

1. 算法常表现为一个无穷过程的截断：

例 1 计算 $\sin x$ 的值 $x\in\left(0,\frac{\pi}{4}\right)$

根据泰勒公式：

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

这是一个无穷级数，我们只能在适当的地方“截断”，使计算量不太大，而精度又能满足要求.

如计算 $\sin 0.5$，取 $n=3$

$$\sin 0.5 \approx 0.5 - \frac{0.5^3}{3!} + \frac{0.5^5}{5!} - \frac{0.5^7}{7!} = 0.479\ 426$$

据泰勒余项公式，它的误差应为

$$R = (-1)^4 \frac{\xi^9}{9!} \quad \xi \in \left(0,\frac{\pi}{4}\right) \tag{1.2}$$

$$|R| \leqslant \frac{\left(\frac{\pi}{4}\right)^9}{362\ 880} = 3.13 \times 10^{-7}$$

可见结果是相当精确的. 实际上结果的六位数字都是正确的.

2. 算法常表现为一个连续过程的离散化

例 2 计算积分值.

$$I = \int_0^1 \frac{1}{1+x}\mathrm{d}x$$

将[0,1]分为 4 等分，分别计算 4 个小曲边梯形的面积的近似值，然后加起来作为积分的近似值. 记被积函数为 $f(x)$，即 $f(x)=\frac{1}{1+x}$

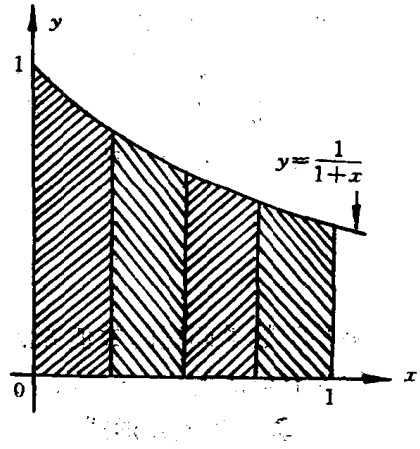

图 1-1

$h=\frac{1}{4}$，$x_i=ih$，

$$T_i=\frac{f(x_i)+f(x_{i+1})}{2}h, \quad i=0,1,2,3.$$

$$I\approx\sum_{i=0}^{3}T_i \tag{1.3}$$

计算有：$I\approx 0.697\ 024$，与精确值 0.693 147 比较，可知结果不够精确，如进一步细分区间，精度可以提高.

3. 算法常表现为“迭代”形式. 迭代是指某一简单算法的多次重复，后一次使用前一次的结果. 这种形式易于在计算程序中实现，在程序中表现为“循环”过程.

例 3 多项式求值.

$$P_n(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n \tag{1.4}$$

用 t_k 表示 x^k，u_k 表示(1.4)式前 $k+1$ 项之和. 作为初值令：

$$\begin{cases} t_0 = 1 \\ u_0 = a_0 \end{cases} \tag{1.5}$$

对 $k=1,2,\cdots,n$，反复执行：

$$\begin{cases} t_k = xt_{k-1} \\ u_k = u_{k-1} + a_kt_k \end{cases} \tag{1.6}$$

显然 $P_n(x)=u_n$，而(1.6)式是一种简单算法的多次循环.

对此问题还有一种更好的迭代算法.

$$\begin{aligned}P_n(x)&=a_nx^n+a_{n-1}x^{n-1}+\cdots+a_1x+a_0\\&=(a_nx^{n-1}+a_{n-1}x^{n-2}+\cdots+a_1)x+a_0\\&=((a_nx^{n-2}+a_{n-1}x^{n-3}+\cdots+a_2)x+a_1)x+a_0\\&=(\cdots(a_nx+a_{n-1})x+\cdots+a_1)x+a_0\end{aligned}$$

令
$$\begin{cases}v_0=a_n\\v_k=xv_{k-1}+a_{n-k}\qquad k=1,2,\cdots,n\end{cases}\tag{1.7}$$

显然 $P_n(x)=v_n$.

这两种算法都是将 n 次多项式化为 n 个一次多项式来计算，这种化繁为简的方法在数值分析中经常使用.

下面估计一下以上两种算法的计算量：

第一法：执行 n 次(1.6)式，每次 2 次乘法，一次加法，共计 $2n$ 次乘法，n 次加法；

第二法：执行 n 次(1.7)式，每次 1 次乘法，一次加法，共计 n 次乘法，n 次加法.

显然第二种方法运算量小，它是我国宋代数学家秦九韶最先提出的，被称为“秦九韶算法”.

例 4 不用开平方计算 $\sqrt{a}$ $(a>0)$的值.

假定 x_0 是 $\sqrt{a}$ 的一个近似值，$x_0>0$，则 $\frac{a}{x_0}\approx\sqrt{a}$ 也是 $\sqrt{a}$ 的一个近似值，且 x_0 和 $\frac{a}{x_0}$ 两个近似值必有一个大于 $\sqrt{a}$，另一个小于 $\sqrt{a}$，可以设想它们的平均值应为 $\sqrt{a}$ 的更好的平均值，于是设计一种算法：

$$x_{k+1}=\frac{1}{2}\left(x_k+\frac{a}{x_k}\right)\quad(k=0,1,2,\cdots)\tag{1.8}$$

如计算 $\sqrt{3}$，取 $x_0=2$，有

$$x_{k+1}=\frac{1}{2}\left(x_k+\frac{3}{x_k}\right)\quad(k=0,1,2,\cdots)$$

计算有：$x_0=2$

$x_1=1.75$

$x_2=1.732\ 142\ 9$

$x_3=1.732\ 050\ 8$

…

可见此法收敛速度很快，只算三次得到 8 位精确数字.

迭代法应用时要考虑是否收敛、收敛条件及收敛速度等问题，今后课程将进一步讨论.

§1.2 误 差

1.2.1 误差的来源

在运用数学方法解决实际问题的过程中，每一步都可能带来误差.

1. 模型误差 在建立数学模型时，往往要忽视很多次要因素，把模型“简单化”，“理想化”，这时模型就与真实背景有了差距，即带入了误差.

2. 测量误差　数学模型中的已知参数，多数是通过测量得到. 而测量过程受工具、方法、观察者的主观因素、不可预料的随机干扰等影响必然带入误差.

3. 截断误差　数学模型常难于直接求解，往往要近似替代，简化为易于求解的问题，这种简化带入误差称为方法误差或截断误差.

4. 舍入误差　计算机只能处理有限数位的小数运算，初始参数或中间结果都必须进行四舍五入运算，这必然产生舍入误差.

在数值分析课程中不分析讨论模型误差；截断误差是数值分析课程的主要讨论对象，它往往是计算中误差的主要部分，在讲到各种算法时，通过数学方法可推导出截断误差限的公式(如(1.2)式)；舍入误差的产生往往带有很大的随机性，讨论比较困难，在问题本身呈病态或算法稳定性不好时，它可能成为计算中误差的主要部分；至于测量误差，我们把它作为初始的舍入误差看待.

误差分析是一门比较艰深的专门学科. 在数值分析中主要讨论截断误差及舍入误差. 但一个训练有素的计算工作者，当发现计算结果与实际不符时，应当能诊断出误差的来源，并采取相应的措施加以改进，直至建议对模型进行修改.

1.2.2　误差的基本概念

1. 误差与误差限

定义 1.1　设 x^* 是准确值，x 是它的一个近似值，称 $e=x-x^*$ 为近似值 x 的绝对误差，简称误差.

误差是有量纲的量，量纲同 x，它可正可负.

误差一般无法准确计算，只能根据测量或计算情况估计出它的绝对值的一个上限，这个上界称为近似值 x 的误差限，记为 ε

$$|x-x^*|\leqslant\varepsilon,\text{其意义是：}x-\varepsilon\leqslant x^*\leqslant x+\varepsilon$$

在工程中常记为：

$$x^*=x\pm\varepsilon.$$

如　$l=10.2\pm0.05\text{mm}$

$R=1\ 500\pm100\Omega$

2. 相对误差与相对误差限　误差不能完全刻画近似值的精度. 如测量百米跑道产生10cm 的误差与测量一个课桌长度产生 1cm 的误差，我们不能简单地认为后者更精确，还应考虑被测值的大小. 下面给出定义：

定义 1.2　误差与精确值的比值

$\dfrac{e}{x^*}=\dfrac{x-x^*}{x^*}$ 称 x 的相对误差，记为 e_r.

相对误差是无量纲的量，常用百分比表示，它也可正可负.

相对误差也常不能准确计算，而是用相对误差限来估计.

相对误差限：

$$\varepsilon_r=\frac{\varepsilon}{|x^*|}\geqslant\frac{|x-x^*|}{|x^*|}=|e_r|.$$

实际上由于 x^* 不知道，用上式无法确定 ε_r，常用 x 代 x^* 作分母，此时：

$$\left|\frac{\varepsilon}{|x^*|}-\frac{\varepsilon}{|x|}\right|=\frac{|\varepsilon(|x|-|x^*|)|}{|x^*x|}\leqslant\frac{\varepsilon^2}{|x^*x|}$$

$$= \frac{\left(\frac{\varepsilon}{x^*}\right)^2}{\left|\frac{x}{x^*}\right|} = O(\varepsilon_r^2).$$

可见此时产生的影响是 ε_r^2 量级，当 ε_r 较小时，可以忽略不计，以后我们就用 $\frac{\varepsilon}{|x|}$ 表示相对误差限.

例 5　在刚才测量的例子中，若测得跑道长为 100±0.1m，课桌长为 120±1cm，则

$$\varepsilon_r^{(1)} = \frac{0.1}{100} = 0.1\%, \varepsilon_r^{(2)} = \frac{1}{120} = 0.83\%$$

显然后者比前者相对误差大.

1.2.3　**有效数字**

定义 1.3　如果近似值 x 的误差限 ε 是它某一数位的半个单位，我们就说 x 准确到该位，从这一位起直到前面第一个非零数字为止的所有数字称 x 的有效数字.

如：$x = \pm 0.a_1a_2\cdots a_n \times 10^m$，其中 $a_1, a_2, \cdots, a_n$ 是 0～9 之中的自然数，且 $a_1 \neq 0$，如 $e = |x - x^*| \leqslant \varepsilon = 0.5 \times 10^{m-l}$　$1 \leqslant l \leqslant n$，则称 x 有 l 位有效数字.

如：$\pi = 3.141\ 592\ 65\cdots$

则 3.14 和 3.141 6 分别有 3 位和 5 位有效数字. 而 3.143 相对于 π 也只能有 3 位有效数字.

在更多的情况，我们不知道准确值 x^*. 如果我们认为计算结果各数位可靠，将它四舍五入到某一位，这时从这一位起到前面第一个非零数字共 l 位，它与计算结果之差必小于该位的半个单位. 我们习惯上说将计算结果保留 l 位有效数字.

如计算机上得到方程 $x^3 - x - 1 = 0$ 的一个正根为 1.324 72，保留 4 位有效数字的结果为 1.325，保留 5 位有效数字的结果为 1.324 7.

相对误差与有效数位的关系十分密切. 定性地讲，相对误差越小，有效数位越多，反之亦正确. 定量地讲，有如下两个定理.

定理 1.1　设近似值 $x = 0.a_1a_2\cdots a_n \times 10^m$ 有 n 位有效数字，则其相对误差限

$$\varepsilon_r \leqslant \frac{1}{2a_1} \times 10^{-n+1}$$

此定理的证明不难，可作为习题完成.

定理 1.2　设近似值 $x = \pm 0.a_1a_2\cdots a_n\cdots \times 10^m$ 的相对误差限不大于 $\frac{1}{2(a_1+1)} \times 10^{-n+1}$，则它至少有 n 位有效数字.

证　$|x| \leqslant (a_1+1) \times 10^{m-1}$

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

$$= 0.5 \times 10^{m-n}$$

由定义 1.3 知 x 有 n 位有效数字.

例 5　计算 sin1.2，问要取几位有效数字才能保证相对误差限不大于 0.01%.

解　$\sin 1.2 = 0.93\cdots$，故 $a_1 = 9, m = 0$

$$\varepsilon_r = \frac{1}{2a_1} \times 10^{-n+1} \leqslant 0.01\% = 10^{-4}.$$

解关于 n 的不等式.

$10^{-n}\leqslant 18\times 10^{-5}=1.8\times 10^{-4}$.

所以取 $n=4$,即可满足要求.

对有效数字的观察比估计相对误差容易得多,故监视有效数字是否损失,常可发现相对误差的突然扩大.

例 6 计算 $\frac{1}{759}-\frac{1}{760}$,视已知数为精确值,用 4 位浮点数计算

解 原式 $=0.131\ 8\times 10^{-2}-0.131\ 6\times 10^{-2}$

$=0.2\times 10^{-5}$.

结果只剩一位有效数字,有效数字大量损失,造成相对误差的扩大. 若通分后再计算:

原式 $=\frac{1}{759\times 760}=\frac{1}{0.576\ 8\times 10^{6}}=0.173\ 4\times 10^{-5}$

就得到 4 位有效数字的结果. 下文将会提到相近数字相减会扩大相对误差.

§1.3 设计算法时应注意的原则

1.3.1 数值运算时误差的传播

当参与运算的数值带有误差时,结果也必然带有误差,问题是结果的误差与原始误差相比是否扩大.

1. 对函数 $f(x)$的计算 设 x 是 x^*的近似值,则结果误差

$$e(f(x))=f(x)-f(x^*)$$

用泰勒展式分析

$$f(x^*)=f(x)+f'(x)(x^*-x)+f''(\xi)\frac{(x^*-x)^2}{2}$$

$$e(f(x))=f'(x)(x-x^*)-\frac{f''(\xi)}{2}(x-x^*)^2\quad \xi\in(x,x^*)$$

$$|e(f(x))|\leqslant|f'(x)|\varepsilon(x)+|\frac{f''(\xi)}{2}|\varepsilon^2(x)$$

忽略第二项高阶无穷小之后,可得函数 $f(x)$计算后的误差限估计式

$$\varepsilon(f(x))\approx|f'(x)|\varepsilon(x) \tag{1.9}$$

2. 对多元函数 $f(x_1^*,x_2^*,\cdots,x_n^*)=A^*$,若 $x_1^*,x_2^*,\cdots,x_n^*$ 的近似值分别是 $x_1,x_2,\cdots,x_n$,则 $A=f(x_1,x_2,\cdots,x_n)$是结果的近似值.

$$e(A)=A-A^*=f(x_1,x_2,\cdots,x_n)-f(x_1^*,x_2^*,\cdots,x_n^*)$$

$$|A-A^*|=|f(x_1^*,x_2^*,\cdots,x_n^*)-f(x_1,x_2,\cdots,x_n)|$$

$$\leqslant\sum_{k=1}^{n}\left|\frac{\partial f(x_1,\cdots,x_n)}{\partial x_k}\right||x_k-x_k^*|+O((\Delta x)^2)$$

其中 $\Delta x=\max\limits_{k}|x_k-x_k^*|$

略去高阶项后

$$\varepsilon(A)\approx\sum_{k=1}^{n}\left|\frac{\partial f(x_1^*,\cdots,x_n^*)}{\partial x_k}\right|\varepsilon(x_k) \tag{1.10}$$

3. 四则运算中误差的传播 按公式(1.10)易得近似数作四则运算后的误差限公式:

$$\varepsilon(x_1 \pm x_2) = \varepsilon(x_1) + \varepsilon(x_2) \tag{1.11}$$

$$\varepsilon(x_1 x_2) \approx |x_1|\varepsilon(x_2) + |x_2|\varepsilon(x_1) \tag{1.12}$$

$$\varepsilon\left(\frac{x_1}{x_2}\right) \approx \frac{|x_1|\varepsilon(x_2) + |x_2|\varepsilon(x_1)}{|x_2|^2} \quad (x_2 \neq 0), \tag{1.13}$$

其中公式(1.11)取等号，是因为作为多元函数，加减运算是一次函数，泰勒展开式没有二次余项.

例 7 若电压 $V=220\pm5$(V)，电阻 $R=300\pm10(\Omega)$，求电流 I 并计算其误差限及相对误差限.

解 $I=\dfrac{220}{300}=0.733\ 3$(A)

$$\varepsilon(I)\approx\frac{|V|\varepsilon(R)+|R|\varepsilon(V)}{R^2}=\frac{220\times10+300\times5}{90\ 000}$$

$$=0.041\ 1\text{(A)}$$

所以 $I=0.733\ 3\pm0.041\ 1$(A)　　$\varepsilon_r(I)=\dfrac{0.041\ 1}{0.733\ 3}=0.6\%$

1.3.2 算法中应避免的问题

1. 避免相近数相减　由公式(1.11)

$$\varepsilon(x_1 - x_2) = \varepsilon(x_1) + \varepsilon(x_2),$$

推出

$$\varepsilon_r(x_1 - x_2)=\frac{\varepsilon(x_1 - x_2)}{|x_1 - x_2|} = \frac{|x_1|}{|x_1 - x_2|} \times \frac{\varepsilon(x_1)}{|x_1|} + \frac{|x_2|}{|x_1 - x_2|} \times \frac{\varepsilon(x_2)}{|x_2|}$$

$$= \frac{|x_1|}{|x_1 - x_2|}\varepsilon_r(x_1) + \frac{|x_2|}{|x_1 - x_2|}\varepsilon_r(x_2)$$

当 x_1, x_2 十分相近时，$|x_1-x_2|$ 接近零，$\dfrac{|x_1|}{|x_1-x_2|}$ 和 $\dfrac{|x_2|}{|x_1-x_2|}$ 将很大，所以 $\varepsilon_r(x_1-x_2)$ 将比 $\varepsilon_r(x_1)$ 和 $\varepsilon_r(x_2)$ 大很多，即相对误差将显著扩大.

从直观上看，相近数相减会造成有效数位的减少，本章例 6 就是一个例子. 有时，通过改变算法可以避免相近数相减.

例 8 解方程 $x^2-18x+1=0$，假定用 4 位浮点计算.

解 用公式解法

$$x_1 = \frac{18 + \sqrt{18^2 - 4}}{2} = 9 + \sqrt{80} \approx 17.94$$

$$x_2 = 9 - \sqrt{80} \approx 9.000 - 8.944 = 0.056$$

可见到第二个根只有两位有效数字，精度较差，若第二个根改为用韦达定理计算

$x_2=1/x_1\approx0.055\ 74$

可以得到较好的结果.

如 $\sqrt{x+1}-\sqrt{x}$　$(x>>1)$

可改为 $\dfrac{1}{\sqrt{x+1}+\sqrt{x}}$，

如 $1-\cos x$　　$(|x|<<1)$

可改为 $2\sin^2\left(\frac{x}{2}\right)$ 等等，都可以得到比直接计算好的结果.

2. 避免除法中除数的数量级远小于被除数　由公式(1.12)

$$\varepsilon\left(\frac{x_1}{x_2}\right)\approx\frac{|x_1|}{|x_2|^2}\varepsilon(x_2)+\frac{1}{|x_2|}\varepsilon(x_1)$$

若 $|x_2|<<|x_1|$，则 $\frac{|x_1|}{|x_2|^2}>>1$，这时 $\varepsilon\left(\frac{x_1}{x_2}\right)$ 将比 $\varepsilon(x_2)$ 扩大很多.

3. 防止小数被大数“吃掉”　在大量数据的累加运算中，由于加法必须进行对位，有可能出现小数被大数“吃掉”.

例如用六位浮点数计算某市的工业总产值，原始数据是各企业的工业产值，当加法进行到一定程度，部分和超过 100 亿元(0.1×10^{11})，再加产值不足 10 万元的小企业产值，将再也加不进去. 而这部分企业可能为数不少，合计产值相当大. 这种情况应将小数先分别加成大数，然后相加，结果才比较正确.

这个例子告诉我们，在计算机数系中，加法的交换律和结合律可能不成立，这是在大规模数据处理时应注意的问题.

习题一

1. 用例 4 的算法计算 $\sqrt{10}$，迭代 3 次，计算结果保留 4 位有效数字.

2. 推导开平方运算的误差限公式，并说明什么情况下结果误差不大于自变量误差.

3. 以下各数都是对准确值进行四舍五入得到的近似数，指出它们的有效数位、误差限和相对误差限.

$x_1=0.3040$，　$x_2=5.1\times10^9$，　$x_3=400$，　$x_4=0.003346$，　$x_5=0.875\times10^{-5}$.

4. 证明 1.2.3 之定理 1.1.

5. 为使球体积 V 的相对误差小于 0.1%，要求它的半径 R 的相对误差为多少?

6. 若跑道长的测量有 0.1%的误差，对 400m 成绩为 60s 的运动员的成绩将会带来多大的误差和相对误差.

7. 为使 $\sqrt{20}$的近似数相对误差小于 0.05%，试问该保留几位有效数字.

8. 下列各式应如何改进，使计算更准确：

1)已知 $|x|<<1$

$$y=\frac{1}{1+2x}-\frac{1-x}{1+x},$$

2)已知 $|x|>>1$

$$y=\sqrt{x+\frac{1}{x}}-\sqrt{x-\frac{1}{x}}.$$

3)已知 $|x|<<1$

$$y=\frac{1-\cos2x}{x}.$$

4)已知 $p>0,q>0,p>>q$

$$y=\sqrt{p^2+q^2}-p.$$

第二章　线性方程组的直接解法

§2.1　引　言

在自然科学和工程技术中，很多问题归结为解线性方程组. 有的问题的数学模型中虽不直接表现为含线性方程组，但它的数值解法中将问题“离散化”或“线性化”为线性方程组. 因此线性方程组的求解是数值分析课程中最基本的内容之一.

线性方程组：

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{cases} \tag{2.1}$$

常记为矩阵形式

$$Ax = b \tag{2.2}$$

此时 A 是一个 $n\times n$ 方阵，x 和 b 是 n 维列向量.

根据线性代数知识若 $|A|\neq 0$，(2.2)的解存在且唯一.

关于线性方程组的解法一般分为两大类，一类是直接法，即经过有限次的算术运算，可以求得(2.1)的精确解(假定计算过程没有舍入误差). 如线性代数课程中提到的克莱姆算法就是一种直接法. 但该法对高阶方程组计算量太大，不是一种实用的算法[1]. 实用的直接法中具有代表性的算法是高斯消元法，其它算法都是它的变形和应用.

另一类是迭代法，它将(2.1)变形为某种迭代公式，给出初始解 x_0，用迭代公式得到近似解的序列 $\{x_k\}$，$k=0,1,2,\cdots$，在一定的条件下 $x_k\to x^*$(精确解). 迭代法显然有一个收敛条件和收敛速度问题.

这两种解法都有广泛的应用，我们将分别讨论，本章介绍直接法.

§2.2　高斯(Gauss)消元法

高斯消元法是一种古老的方法. 我们在中学学过消元法，高斯消元法就是它的标准化的、适合在计算机上自动计算的一种方法.

2.2.1　高斯消元法的基本思想

例1　解方程组

$$\begin{cases} x_1 + 2x_2 + 3x_3 = 1 & (2.3) \\ 2x_1 + 7x_2 + 5x_3 = 6 & (2.4) \\ x_1 + 4x_2 + 9x_3 = -3 & (2.5) \end{cases}$$