实用数值分析教程

A Course in Practical Numerical Analysis

刘春凤 米翠兰 等 编著

刘保相 审校

冶金工业出版社

实用数值分析教程

A Course in Practical Numerical Analysis

编著　刘春凤　米翠兰　何亚丽

马醒花　杨爱民

审校　刘保相

北　京

冶　金　工　业　出　版　社

2006

内容简介

本书共分6章：第一章绪论；第二章插值与拟合；第三章线性方程组的解法；第四章数值微积分；第五章非线性方程的数值解法；第六章常微分方程数值解法。

本书适合大专院校以及科研院所的理工科学生和研究人员学习、参考。

图书在版编目（CIP）数据

实用数值分析教程/刘春凤等编著．—北京：冶金工业出版社，2006.4

ISBN 7-5024-3928-5

Ⅰ．实…　Ⅱ．刘…　Ⅲ．数值计算-教材　Ⅳ．O241

中国版本图书馆 CIP 数据核字（2006）第013705号

出 版 人　曹胜利（北京沙滩嵩祝院北巷39号，邮编100009）
责任编辑　杨盈园　美术编辑　李心
责任校对　王永欣　李文彦　责任印制　牛晓波
北京百善印刷厂印刷；冶金工业出版社发行；各地新华书店经销
2006年4月第1版，2006年4月第1次印刷
787mm×1092mm　1/16；11.75印张；280千字；175页；1—3000册
26.00元
冶金工业出版社发行部　电话：（010）64044283　传真：（010）64027893
冶金书店　地址：北京东四西大街46号（100711）　电话：（010）65289081

前　言

数值分析是数学科学与计算机技术结合的一门学科。该学科重点介绍计算机上常用的基本计算方法的构造和使用，同时对计算方法的工作量、稳定性、收敛性、误差估计、适用范围及优缺点等也作适当的分析。多年来《数值分析》课程主要在数学系计算数学专业和计算机软件专业开设。进入 21 世纪，随着现代科学技术的发展和计算机的广泛使用，科学计算已经成为平行于理论分析和科学实验的第三种科学手段。在诸多科学领域里，无论是数学工作者、数值计算专家，还是工程技术人员，数值计算已经成为必须掌握的知识和工具。

目前，研究数学问题的数值解且利用算法和计算机求解数学问题的数值计算方法，主要以高等数学、线性代数为数学基础，既有数值分析理论上的抽象性和严谨性，又注重实用性和实验性，应用领域日益广泛。在理工科院校除部分特殊专业以外，大部分学生应当说具有比较扎实的高等数学和工程数学的功底，然而在应用数学知识解决实际问题时往往不知所措，这是数学从学到用缺乏衔接所致。而《实用数值分析教程》恰是在“学数学”和“用数学”之间搭建了一座桥梁。所以在理工科院校尽可能多的专业开设《数值分析》课程对培养应用型人才是很有必要的。本书就是为理工科学生而编写的。

本书内容包括误差概念、插值与拟合、数值微积分、非线性方程及方程组的数值解法、线性方程（组）的直接解法和迭代法、常微分方程的数值解法。

本书吸取了国内教材的新理念，力求处理好数学基础和应用方法之间的衔接和关系，既不以严谨理论为主导，也不是全篇数据的数值计算，而是以方法为中心，以例题为载体，围绕方法给出简单的、典型的数值例题，通过例题进一步理解计算对象、计算公式、计算的限定条件和计算步骤。本书突出的特点是：紧密围绕数值分析的内容主线，引入了 Mathematica 的相关内容，并集作者多年的教学经验，编制了大量的 Mathematica 程序，配置了适量

的应用范例。使数值分析中的诸多方法计算快捷、结果直观。力求不但能使学生掌握常用的数值计算方法，而且还能培养学生的数学综合分析能力和科学计算的能力，为他们深入学习打好基础。

讲授本书全部内容大约需要40～50学时，其中上机应不少于10学时。

感谢我校领导对本书的出版给予的大力支持。

限于水平，且编写时间仓促，书中不当之处难免，恳请读者批评指正。

编 者

2005年8月

目　　录

第一章 绪 论

第一节 数值分析的研究对象与特点

数值分析（Numerical Analysis）是寻求数学问题近似解的方法、过程及其理论的一个数学分支，亦称计算方法（Computational Method）. 主要研究适合于在计算机上使用的数值计算方法及其理论与软件实现. 鉴于此，本书主要介绍各种常见数值问题的常用方法及其算法设计和编程，即注重实际应用. 内容包括解线性方程组的直接法、函数近似计算的插值法与最小二乘法、数值积分和数值微分、常微分方程的数值解法、解线性方程组和非线性方程的迭代法，并注意与计算机和相关软件的密切结合，培养学生的应用能力.

数值计算方法是近代数学的一个重要分支，它既有数学类课程中理论上的抽象性和严谨性，又有实用性和实验性的技术特征，是一门理论性和实践性都很强的学科. 其特点概括起来有以下四点：

第一，面向计算机，要根据计算机的特点提供切实可行的有效算法.

第二，有可靠的理论分析，能任意逼近并达到精度要求，对近似算法要保证收敛性和数值稳定性，还要对误差进行分析.

第三，要有数值实验.

第四，要有好的计算复杂性. 计算复杂性包括时间复杂性和空间复杂性（时间复杂性好指节省时间，空间复杂性好是指节省存储量）.

第二节 数值计算的误差

一、误差的来源与分类

数值方法得到的是解的近似. 真解与近似解之间的偏差就是误差（Error）. 按照误差的来源，可将其分为以下四种：

（一）模型误差（Model Error）

从实际问题转化为数学问题，即建立数学模型时，对被描述的实际问题进行了抽象和简化，这种数学模型与实际问题之间出现的误差称为模型误差.

（二）观测误差（Measurement Error）

数学问题中总包括一些参量，它们的值往往是由观测得到的. 而观测不可能绝对准确，由此产生的误差称为观测误差.

（三）截断误差（Truncation Error）

一般数学问题常常难以求出精确解，需要简化为较易求解的问题，以简化问题的解作为原问题解的近似. 如求一个收敛的无穷级数之和，总是用它的部分和作为近似值，也就

是截去该级数后面的无穷多项. 这样由于简化问题所引起的误差称为截断误差或方法误差.

(四) 舍入误差 (Round off Error)

用计算机作数值计算时，由于计算机的字长有限，原始数据在计算机上表示会产生误差，计算过程又可能产生新的误差，这种误差称为舍入误差. 例如，用 3.14159 近似代替 π，产生的误差 $R=\pi-3.14159=0.0000026\cdots$ 就是舍入误差. 少量舍入误差是微不足道的，但在电子计算机上完成了千百万次运算后，舍入误差的积累有时可能是十分惊人的.

由以上误差来源的分析可以看到：误差是不可避免的，要求绝对准确，绝对严格实际上是办不到的. 既然描述问题的方法都是近似的，那么要求解的绝对准确也就没有意义了. 因此在数值分析里讨论的都是近似解，那种认为近似解是不可靠的、不准确的看法是错误的，应该认为求近似解是正常的，问题是怎样尽量设法减少误差，提高精度. 在四种误差来源的分析中，前两种误差是客观存在的，后两种是由计算方法引起的. 本课程是研究数学问题的数值解法，因此只涉及后两种误差.

二、绝对误差与相对误差

定义 1.1 设 x^* 为准确值 x 的一个近似值，称

$$e=x-x^*$$

为近似值 x^* 的**绝对误差** (Absolute Error)，简称误差.

误差是有量纲的，可正可负. 误差是无法计算的，但可估计出它的一个上界 ε，即

$$|e|=|x-x^*|\leqslant\varepsilon$$

称 ε 为近似值 x^* 的绝对误差限，简称误差限. 显然误差限是不唯一的. ε 越小，表示近似值 x^* 的精度越高. 显然有

$$x^*-\varepsilon\leqslant x\leqslant x^*+\varepsilon$$

即准确值 x 必定在区间 $[x^*-\varepsilon,\ x^*+\varepsilon]$ 内，也常记作 $x=x^*\pm\varepsilon$.

绝对误差并不足以表示近似值的好坏. 例如，设 $x_1=100\pm1$，$x_2=1000\pm1$，近似值 $x_1^*=100$ 的绝对误差限与 $x_2^*=1000$ 的绝对误差限相同. 不过在 100 之内差 1 和 1000 之内差 1 比较，后者比前者精确.

定义 1.2 设 x^* 为准确值 x 的近似值，称绝对误差与准确值之比为近似值 x^* 的相对误差 (Relative Error)，记为 e_r，即

$$e_r=\frac{e}{x}=\frac{x-x^*}{x}$$

显然，上例中 x_2^* 的相对误差为 0.001，x_1^* 的相对误差为 0.01. 由于在计算过程中准确值 x 总是未知的，故一般取相对误差为

$$e_r=\frac{e}{x^*}$$

可以证明，当 $|e_r|$ 很小时，$\frac{e}{x}-\frac{e}{x^*}$ 是 e_r 的高阶无穷小，可以忽略不计. 所以，取绝对误差与近似值之比为相对误差是合理的.

类似于绝对误差的情况，如果存在正数 ε_r，使得

$$|e_r|=\left|\frac{e}{x^*}\right|\leqslant\varepsilon_r$$

则称 ε_r 为 x^* 的相对误差限. 可以看出，相对误差是个相对数，是无量纲的，也可正可负. 而相对误差限也是不唯一的. 显然，误差限与近似值绝对值之比 $\frac{\varepsilon}{|x^*|}$ 就是 x^* 的一个相对误差限.

三、有效数字（Significant Figure）

在计算中，当准确值 x 有很多位数时，常常按“四舍五入”原则得到 x 的近似值 x^*. 例如取 4 位小数运算，则 $x=15.26784$ 按取值 $x^*=15.2678$ 实施运算. 无论取几位小数所得到的近似值，其绝对误差都不会超过其末位数的半个单位，如

$x=\sqrt{3}=1.732050808\cdots$ 按取值 $x_1^*=1.73$，有 $\varepsilon_1<0.005$；按取值 $x_2^*=1.7321$，有 $\varepsilon_2<0.00005$. 它们的误差都不超过末位的半个单位，即

$$|\sqrt{3}-1.73|<0.5\times10^{-2},\ |\sqrt{3}-1.7321|<0.5\times10^{-4}$$

定义 1.3　如果近似值 x^* 的误差限是 0.5×10^{-n}，则称 x^* 准确到小数点后第 n 位，并从第一个非零数字到这一位的所有数字均称为有效数字.

如 $x^*=15.27$ 和 $x^*=0.005246$ 均有四位有效数字. 而 $x^*=0.0005$ 只有一位有效数字，$x^*=0.000500$ 却有三位有效数字. 0.342×10^{-2} 和 0.3420×10^{-2} 的绝对误差限分别为 0.5×10^{-5}，0.5×10^{-6}，前者具有三位有效数字，而后者具有四位有效数字.

有效数字也称有效数位. 有效数字概念虽然简单，但经常会引起某些错误的理解. 稍有疏忽，也许就会把 3.0 与 3.0000 视为毫无区别的值.

另外，若 $x^*=3.0000$，$y^*=5.0003$，则

$$|(x\pm y)-(x^*\pm y^*)|\leqslant\frac{1}{2}\times10^{-4}+\frac{1}{2}\times10^{-4}=10^{-4}$$

而若 $x^*=3.0, y^*=0.2$，则 $|(x\pm y)-(x^*\pm y^*)|\leqslant\frac{1}{2}\times10^{-1}+\frac{1}{2}\times10^{-1}=10^{-1}$，

可见，有效数字的多少直接影响到近似值的误差，因此要想缩小误差的最直接而有效的办法是增加运算中的有效数位.

例 1.1　按四舍五入原则，求下列各数的具有四位有效数字的近似值

168.957，3.0004，73.2250，0.00152632

解　按定义，上述各数具有四位有效数字的近似值分别是

169.0，3.000，73.23，0.001526

注意：$x^*=3.0004$ 的四位有效数字的近似值是 3.000，而不是 3，因为 3 只有一位有效数字.

四、基本运算中的误差估计

数值运算中由于所给数据的误差必然引起函数值的误差，这种数据误差的影响较为复杂，通常用微分学的相关知识解决.

当 $y=f(x)$ 是一元函数时，设 x 的近似值为 x^*，$y=f(x)$ 在点 x^* 可微，则

$$e(y^*)=y-y^*=f(x)-f(x^*)=\Delta y\approx \mathrm{d}(y)=f'(x^*)(x-x^*)$$

于是误差限可取为

$$\varepsilon(f(x^*))\approx|f'(x^*)|\varepsilon(x^*)$$

而相对误差则为

$$e_r(y^*) = \frac{e(y^*)}{y^*} \approx \frac{\mathrm{d}y^*}{y^*} = \mathrm{d}(\ln y^*)$$

当 f 是多元函数时，例如计算 $y=f(x_1,x_2,\cdots,x_n)$，设 $x_1,x_2,\cdots,x_n$ 的近似值为 x_1^*，$x_2^*,\cdots,x_n^*$，$y=f(x_1,x_2,\cdots,x_n)$在点$(x_1^*,x_2^*,\cdots,x_n^*)$可微，则当数据误差较小时，$y$ 的绝对误差为

$$\begin{aligned} e(y^*) &= y - y^* = f(x_1,\cdots,x_n) - f(x_1^*,\cdots,x_n^*) \approx \mathrm{d}f(x_1^*,x_2^*,\cdots,x_n^*) \\ &= \sum_{k=1}^{n}\left(\frac{\partial f(x_1^*,\cdots,x_n^*)}{\partial x_k}\right)(x_k - x_k^*) = \sum_{k=1}^{n}\left(\frac{\partial f(x_1^*,\cdots,x_n^*)}{\partial x_k}\right)e(x_k^*) \end{aligned}$$

相对误差为

$$e_r(y^*) = \frac{e(y^*)}{y^*} \approx \frac{\mathrm{d}y^*}{y^*} = \mathrm{d}(\ln f) = \sum_{k=1}^{n}\left(\frac{\partial f(x_1^*,\cdots,x_n^*)}{\partial x_k}\right)\frac{e(x_k^*)}{f(x_1^*,\cdots,x_n^*)}$$

例 1.2　已知 $x_1=1.03\pm0.01$，$x_2=0.45\pm0.01$，试求用公式 $y=x_1^2+\frac{1}{2}e^{x_2}$ 计算 y 时产生的绝对误差和相对误差.

解　$e(x_1)=0.01$，$e(x_2)=0.01$，$\frac{\partial y}{\partial x_1}=2x_1$，$\frac{\partial y}{\partial x_2}=\frac{1}{2}e^{x_2}$，故绝对误差为

$$e(y) \approx \mathrm{d}(y) = \frac{\partial y}{\partial x_1}e(x_1) + \frac{\partial y}{\partial x_2}e(x_2) = 2.06\times0.01 + 0.5\times e^{0.45}\times0.01 = 0.0284416$$

相对误差为 $e_r(y) = \frac{e(y)}{y} \approx \frac{\mathrm{d}y}{y} = \frac{0.0284416}{1.03^2+0.5\times e^{0.45}} = 0.015415$

第三节　误差定性分析与避免误差危害

一、病态问题与条件数

对一个数值问题本身如果输入数据有微小扰动（即误差），引起输出数据（即问题解）相对误差很大，这就是病态问题. 它是数学问题本身性质决定的，与算法无关，也就是说对病态问题，用任何算法直接计算都将产生不稳定性.

例 1.3　解线性方程组

$$\begin{cases} x_1 + \frac{1}{2}x_2 + \frac{1}{3}x_3 = \frac{11}{6} \\ \frac{1}{2}x_1 + \frac{1}{3}x_2 + \frac{1}{4}x_3 = \frac{13}{12} \\ \frac{1}{3}x_1 + \frac{1}{4}x_2 + \frac{1}{5}x_3 = \frac{47}{60} \end{cases}$$

此方程组的精确解为 $x_1=x_2=x_3=1$. 在计算机上运算，其所有输入数据都是有限位小数，若取三位有效数字，用消元法求解得 $x_1=1.089$，$x_2=0.488$，$x_3=0.491$，误差很大，这表明输入数据有微小扰动，输出数据误差很大，故此问题是病态问题.

病态和良态是相对的，没有严格界线，通常判断问题是否病态，可用条件数大小来衡量，条件数越大问题病态越严重. 例如计算函数值 $f(x)$的条件数 cond($f(x)$)可定义为

$$c_p = \mathrm{cond}(f(x)) = \frac{|xf'(x)|}{|f(x)|}$$

这里输入数据为 x，输出数据为 $f(x)$，当 x 有扰动 $\Delta x=x-x^*$ 时，条件数 $\mathrm{cond}(f(x))$ 为 $f(x)$ 的相对误差 $\dfrac{\Delta f(x)}{f(x)}$ 与 x 的相对误差 $\dfrac{\Delta x}{x}$ 之比，当 $\Delta x\to 0$ 时，得

$$\mathrm{cond}(f(x))=\lim_{\Delta x\to 0}\left|\frac{\Delta f(x)}{f(x)}\right|\Big/\left|\frac{\Delta x}{x}\right|=\frac{|xf'(x)|}{|f(x)|}$$

根据定义，条件数越大，$f(x)$ 的相对误差越大，当 $\mathrm{cond}(f(x))\gg 1$ 就认为问题是病态的.

二、算法的数值稳定性

一个算法，如果在一定的条件下，其舍入误差在整个运算过程中能够得到控制或者舍入误差的增长不影响产生可靠的结果，则称该算法是数值稳定的，否则称为数值不稳定的.

例 1.4　建立积分 $I_n=\int_0^1\dfrac{x^n}{x+5}\mathrm{d}x\quad n=0,1,\cdots,20$ 的递推关系式，并研究它的误差传递.

解　由

$$I_n+5I_{n-1}=\int_0^1\frac{x^n+5x^{n-1}}{5+x}\mathrm{d}x=\int_0^1 x^{n-1}\mathrm{d}x=\frac{1}{n}$$

和

$$I_0=\int_0^1\frac{1}{x+5}\mathrm{d}x=\ln 6-\ln 5$$

可建立下列递推公式

$$\begin{cases}I_0=\ln 6-\ln 5\\ I_n=\dfrac{1}{n}-5I_{n-1}\quad(n=1,2,\cdots,20)\end{cases}\tag{$*$}$$

计算出 I_0 后，由递推关系式可逐次求出 I_1，I_2，…，I_{20} 的值. 但在计算 I_0 时有舍入误差，因此在使用递推关系式中，实际算出的都是近似值 I_n^* （$n=1$，2，…，20）. 即

$$\begin{cases}I_0^*=I_0-e_0\\ I_n^*=\dfrac{1}{n}-5I_{n-1}^*\quad(n=1,2,\cdots,20)\end{cases}$$

现在来研究误差是如何传递的.

设 I_0^* 有误差 e_0，假设计算过程中不产生新的舍入误差，则由式（$*$）可得

$$e_n=I_n-I_n^*=-5I_{n-1}+5I_{n-1}^*=-5e_{n-1}\quad(n=1,2,\cdots)$$

从而有

$$e_n=(-5)^n e_0$$

即原始数据 I_0^* 的误差 e_0 对第 n 步的影响是使该误差扩大了 5^n 倍. 当 n 较大时，误差将淹没真值. 因此递推公式（$*$）是数值不稳定的.

现在从另一方向使用这一公式

$$I_{n-1}=\frac{1}{5}\left(\frac{1}{n}-I_n\right)\quad(n=20,19,\cdots,1)\tag{$**$}$$

只要给出 I_{20} 的一个近似值 I_{20}^*，即可递推得到 I_{19}^*，I_{18}^*，…，I_0^*.

类似于上面的推导可得

$$e_{n-1}=-\frac{1}{5}e_n,\quad e_0=\left(-\frac{1}{5}\right)^n e_n$$

每递推一步误差缩小到原值的$\frac{1}{5}$，所以递推公式（＊＊）是数值稳定的.

由于 $x\in[0,1]$ 时

$$\frac{x^n}{6}<\frac{x^n}{5+x}<\frac{x^n}{5}$$

所以有估计式

$$\frac{1}{6(n+1)}<I_n<\frac{1}{5(n+1)}$$

于是

$$\frac{1}{6\times 21}<I_{20}<\frac{1}{5\times 21}$$

取

$$I_{20}\approx\frac{\frac{1}{126}+\frac{1}{105}}{2}\approx 0.0087301587$$

可得另一算法

$$\begin{cases}I_{20}\approx 0.0087301587\\ I_{n-1}=\frac{1}{5}\left(\frac{1}{n}-I_n\right)\quad(n=20,19,\cdots,1)\end{cases}$$

由此可见，对于同一数学问题，使用的算法不同，效果也大不相同，只有选用数值稳定性好的算法，才能求得较准确的结果.

三、避免误差危害的若干原则

（一）避免两个相近的数相减

由两数差 $u=x-y$ 的相对误差关系式

$$e_r(u)=e_r(x-y)=\frac{e(x)-e(y)}{x-y}$$

可以看出，当 x 与 y 很接近时，u 的相对误差会很大，有效数字位数将严重丢失. 例如，$x=532.65$，$y=532.52$都具有五位有效数字，但 $x-y=0.13$ 只有两位有效数字. 这说明应当尽量避免出现这类运算，最好是改变算法，防止这种现象产生. 通常是根据不同情况分别采用因式分解、分子分母有理化、三角函数恒等式、其他恒等式等方法. 如当 x 充分大时，应作变换

$$\sqrt{x+1}-\sqrt{x}=\frac{1}{\sqrt{x+1}+\sqrt{x}},\quad \frac{1}{x}-\frac{1}{1+x}=\frac{1}{x(x+1)}$$

当 x 接近零时，应作变换

$$1-\cos x=2\sin^2\frac{x}{2},\quad \frac{1-\cos x}{\sin x}=\frac{\sin x}{1+\cos x}$$

当 x_1，x_2 很接近时，应作变换

$$\ln x_1-\ln x_2=\ln\frac{x_1}{x_2}$$

等等.

(二) 避免大数"吃"小数现象

大数"吃掉"小数，是指计算机在计算过程中，由于要把参加运算的数对阶，即把两数都写成绝对值小于 1 但阶码相同的数，而导致较小的数加不到较大的数中. 这种现象有时会影响计算结果的可靠性. 如 $a=10^9+1$，必须改写成

$$a=0.1\times10^{10}+0.0000000001\times10^{10}$$

如果计算机只能表示 8 位小数，则算出 $a=0.1\times10^{10}$，大数"吃"了小数.

例 1.5　在五位十进制计算机上，计算

$$A=52492+\sum_{i=1}^{1000}\delta_i\quad(0.1\leqslant\delta_i\leqslant0.9, i=1,2,\cdots,1000)$$

把运算的数写成规格化形式

$$A=0.52492\times10^5+\sum_{i=1}^{1000}\delta_i$$

由于在计算机内计算时要对阶，若取 $\delta_i=0.9$，对阶时，$\delta_i=0.000009\times10^5$，在五位计算机中表示为 0，因此

$$\begin{aligned}A&=0.52492\times10^5+0.000009\times10^5+\cdots+0.000009\times10^5\\&=0.52492\times10^5\end{aligned}$$

结果显然不可靠，这是由于运算中出现了大数 52492"吃掉"小数 δ_i 造成的. 如果计算时先把数量级相同的一千个 δ_i 相加，最后再加 52492，就不会出现大数"吃"小数现象，这时

$$0.1\times10^3\leqslant\sum_{i=1}^{1000}\delta_i\leqslant0.9\times10^3$$

于是

$$0.001\times10^5+0.52492\times10^5\leqslant A\leqslant0.009\times10^5+0.52492\times10^5$$

即

$$52592\leqslant A\leqslant53392$$

(三) 避免绝对值太小的数作除数

由式

$$e\left(\frac{x}{y}\right)=\frac{ye(x)-xe(y)}{y^2}$$

可知，当 $|y|\ll|x|$ 或 y 接近零时，$e\left(\frac{x}{y}\right)$可能很大. 这说明在实际计算中，应尽可能避免除数的绝对值远小于被除数的绝对值的除法或用接近于 0 的数作除数，否则会严重影响计算结果的精度，减少有效数位.

(四) 简化计算步骤，减少运算次数

求一个问题的数值解往往有多种算法，不同的算法需要不同的计算量，而计算量的大小会影响误差的积累.

计算多项式

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

的值. 若直接计算 a_ix^i 再逐项相加，一共需做