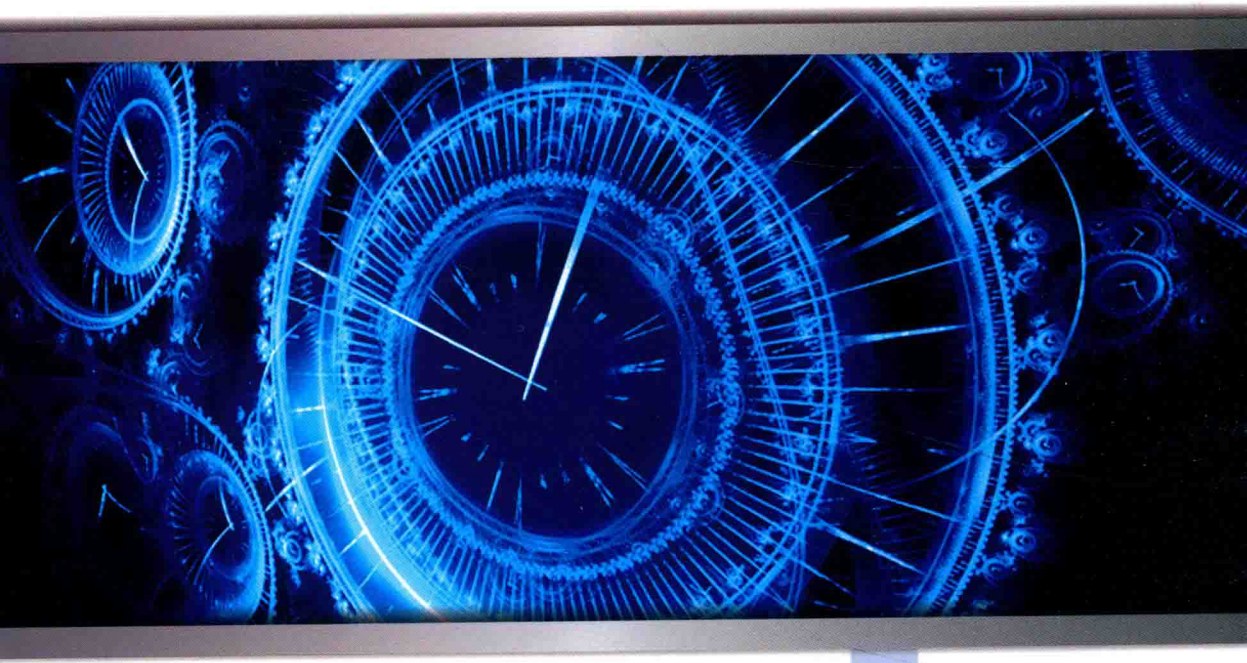
国家林业局普通高等教育“十三五”规划教材

数值计算方法



林玉蕊◎主编

中国林业出版社

国家林业局普通高等教育“十三五”规划教材

数值计算方法

主　编　林玉蕊

副主编　黄习培

中国林业出版社

内 容 简 介

本教材是国家林业局普通高等教育“十三五”规划教材。全书共9章，内容包括：数值计算中的误差、解线性方程组的直接方法、解线性方程组的迭代法、代数插值、函数逼近与曲线拟合、数值积分与数值微分、常微分方程数值解、非线性方程求解、矩阵特征值问题等。

全书按问题→基础理论→简单算法→算法改进→可用算法→应用示例的顺序编排，采用嵌入式模式结合具体应用实例来组织教材结构。着重介绍数值算法的基本思想和算法的实现。给出了大部分算法在Matlab环境下可运行的代码，小部分算法以伪代码表示，实现过程一般留作课后习题。这样，有利于提高学生科学计算的能力，从而加深对“数值分析”理论的理解。本书可作为高等学校数学与应用数学、信息与计算科学、计算机科学与技术、软件工程等理工科专业的教材，也可作为从事科学与工程计算的科技人员的参考用书。

图书在版编目(CIP)数据

数值计算方法 / 林玉蕊主编. —北京：中国林业出版社，2017.4
国家林业局普通高等教育“十三五”规划教材
ISBN 978-7-5038-7888-6

Ⅰ. ①数… Ⅱ. ①林… Ⅲ. 数值计算－计算方法－高等学校－教材 Ⅳ. ①O241

中国版本图书馆CIP数据核字(2017)第042123号

国家林业局生态文明教材及林业高校教材建设项目
福建农林大学出版基金资助

中国林业出版社·教育出版分社

策划、责任编辑：张东晓

电话：(010)83143560　　**传真：**(010)83143516

出版发行　中国林业出版社(100009　北京市西城区德内大街刘海胡同7号)
E-mail：jiaocaipublic@163.com　　电话：(010)83143500
http：//lycb.forestry.gov.cn
经　　销　新华书店
印　　刷　北京市昌平百善印刷厂
版　　次　2017年4月第1版
印　　次　2017年4月第1次印刷
开　　本　787mm×1092mm　1/16
印　　张　13.75
字　　数　326千字
定　　价　29.00元

前　言

本书针对高等学校数学与应用数学、信息与计算科学、计算机科学与技术、软件工程等理工科专业的教学需要而编写。在本书编写过程中，我们力求学生能学到本领域的基础知识，学会数值算法设计的一般过程，并掌握有效地解决实际问题的实用知识。

全书按问题→基础理论→简单算法→算法改进→可用算法→应用示例的顺序编排，采用嵌入式模式结合具体应用实例来组织教材结构。着重介绍了数值算法的基本思想和数值算法的实现。我们给出了大部分算法在 Matlab 环境下可运行的代码，但可以很容易地将其修改为在其他语言环境可运行的代码，小部分算法以伪代码表示，实现过程一般留作课后习题。这样，可以让学生体会枯燥无味的理论如何被用于指导算法实现。有利于提高学生科学计算能力，从而加深对《数值分析》理论的理解。

全书共有 9 章，系统介绍了数值计算中的误差、解线性方程组的直接方法、解线性方程组的迭代法、代数插值、函数逼近与曲线拟合、数值积分与数值微分、常微分方程数值解、非线性方程求解、矩阵特征值问题等内容。本书可作为高校数学与应用数学、信息与计算科学、计算机科学与技术、软件工程等工科专业的教材用书，也可作为从事科学与工程计算的科技人员的参考用书。

在本书的编写过程中，参阅了许多教材、论文和网页，引用了部分论点，限于篇幅，仅列出主要参考文献，在此向所有参考文献的作者致以诚挚的谢意！

限于编者的学识水平，加之时间仓促，书中难免存在疏漏、不妥之处，敬请学界同行、师生批评指正。

林玉蕊

2017 年 1 月

目　录

第 1 章　数值计算中的误差

对误差的研究是数值分析的主要课题，大多数数值方法给出的答案仅仅是近似解，因此了解并估计所引起的误差是重要的。本章介绍可能产生的各种误差，介绍有效数字、函数值误差、计算方法的数值稳定性等概念，以递推法为例指出设计数值稳定性好的计算方法对减少舍入误差影响的重要性。本章还介绍了一种高效率计算方法——秦九韶算法。

1.1　误差来源

利用计算机解决生产活动和科学实践中提出的问题的一般步骤可用图 1-1 表示。

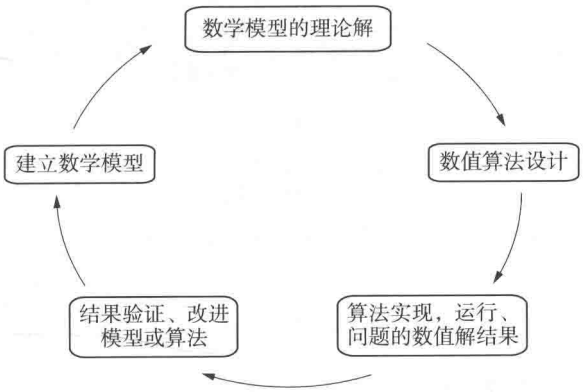


图 1-1　利用计算机解决科学问题的一般步骤

在生产实践或科学实践中，物理现实是用观测得到的数据集表示，建立数学模型的工作就是根据所给的数据集找到一个能比较真实地反映物理现实的数学模型。一个可用的数学模型必须在理论上是可解的，其解应对应于实际问题的解。根据不同的问题其建模方法也不同，这并不在本教材的讨论范围之内。对于一个可用的数学模型，利用计算机给出该模型的满足精度的数值解，是本教材的核心内容。简单地说，就是只用计算机所能完成的运算(加、减、乘、除四则运算以及逻辑运算)，根据给定的数据集，给出问题的数值解。

在图 1-1 所示中的各个步骤都有可能产生误差，归结起来主要有以下四种。

1. 模型误差

建立数学模型时，需对实际问题进行抽象和简化，忽略一些次要因素，这样建立的数学模型只是实际问题的一种近似，它们之间的误差称为模型误差。

例 1.1　物体在重力作用下自由下落，其下落距离 d 和时间 t 的关系，在不考虑阻力的影响下可表示为 $d=\frac{1}{2}gt^2$，其中 $g=9.81\text{m/s}^2$ 是重力加速度。

这是一个经典的数学模型，由于忽略了空气阻力这个因素，从而求出的 d 只是近似的。它与物体实际下落距离之差就是模型误差。

2. 观测误差

在数学模型中往往有若干常数和参数，它们多半是观测得来的，受测量仪器和操作人员的素质等因数的限制必然导致观测结果带有误差，称这种误差为观测误差。例如，例 1.1 中的重力加速度 $g=9.81\text{m/s}^2$ 就是观测来的，它与实际重力加速度之差就是观测误差。

3. 截断误差

在建立数学模型时，其目标是用数学语言准确表达物理现实，为此，往往数学家会使用复杂的数学工具。比如，Fourier 为表示热的传播，就用到了无穷级数。这种含无限个计算项的工作对于计算机来说是不可实现的。事实上数值分析的基本原理是用有限逼近无限，离散逼近连续，把无限的计算过程用有限步计算代替，由此所产生的误差称为截断误差或方法误差。

例 1.2 用只含加、减、乘、除的运算计算 $e^{0.5}$ 的近似值。

解： $e^{0.5}$ 是指数函数 e^x 在 $x=0.5$ 处的值，用 Taylor 级数法求和。e^x 的 Taylor 级数为

$$e^x = 1 + x + \frac{1}{2}x^2 + \cdots = \sum_{k=0}^{\infty} \frac{x^k}{k!} \quad (x < \infty)$$

取 $x=0.5$ 得

$$e^{0.5} = \sum_{k=0}^{\infty} \frac{0.5^k}{k!}$$

上式右端有无限多个计算项，在计算机里是不可实现的，为此取其有限项作为它的近似，即

$$e^{0.5} \approx \sum_{k=0}^{n} \frac{0.5^k}{k!}$$

用这个公式算，把无限的计算过程用有限步计算代替，例如取 $n=3$ 计算得

$$e^{0.5} \approx 1 + \frac{0.5}{1!} + \frac{0.5^2}{2!} + \frac{0.5^3}{3!} = 1.645833$$

准确值为 $e^{0.5}=1.648721\cdots$，出现了误差，这个误差就是截断误差。

上述 Taylor 级数法是计算函数值的一种基本方法，这方法舍弃了无穷级数的后半段，因而出现了误差，这种误差就是截断误差，又称为方法误差。

4. 舍入误差

虽然现在的计算机能表示的数的范围很大，但受限于数在计算机存储单元中存储方式与存储位数(也称为字长)，计算机并不能准确表示所有的实数。

数 x 在计算机中被表示为规格化的浮点形式

$$x = \pm 0.a_1a_2\cdots a_r \times 10^p \tag{1.1}$$

其中，$a_1 \neq 0$，$a_i \in \{0, 1, \cdots, 9\}(i=2, 3, \cdots, r)$，$p$ 称为阶码，r 称为字长。

计算机字长有限，因此计算机上只能表示出有限位数。当一个数位太多或为无理数时，计算机就要进行四舍五入，从而产生误差，这种误差称为舍入误差。

例 1.3 试在 7 位十进制计算机上，给出 π，$-\sqrt{200}$ 和 $\frac{1}{3}$ 的表示。

解：按式(1.1)有

$$\pi = 3.1415926\cdots = 0.3141593 \times 10^{1}$$

$$-\sqrt{200} = -14.142135\cdots = -0.1414214 \times 10^{2}$$

$$\frac{1}{3} = 0.33333333\cdots = 0.3333333 \times 10^{0}$$

它们的误差就是舍入误差。

从总体上来说，数在计算机里的存储总是存在四舍五入过程，而且每次计算过程也会涉及四舍五入，为了减少舍入误差的影响，计算机总是将参与运算的两个数进行“对阶转换”，即把两个参与运算的数转化为相同的阶码。比如，若要计算 $\pi - \sqrt{200}$，在七位计算机中先取出两个数，分别为 0.3141593×10^{1} 与 0.1414214×10^{2} 由于后者的阶高于前者，所以把前者表示为 0.0314159×10^{2} 然后再作运算 $0.0314159 \times 10^{2} - 0.1414214 \times 10^{2}$。由此可见，当数量级相差很大的数作加减运算时，会出现“大数吃掉小数”的现象，从而产生误差，这种误差也称为舍入误差。

例 1.4　在 7 位十进制计算机上

$$\begin{aligned} 10^{7} + 1 &= 0.1000000 \times 10^{8} + 0.1000000 \times 10^{1} \\ &= 0.1000000 \times 10^{8} + 0.0000000 \times 10^{8} \\ &= 0.1000000 \times 10^{8} \\ &= 10^{7} \end{aligned}$$

从计算结果看 1 被 10^{7}“吃掉”了，导致计算结果有舍入误差。

为了避免大数吃掉小数这种现象发生，多个符号相同的数求和时，从绝对值最小的数到绝对值最大的数依次相加。

上面四种误差中，前两种属应用数学范畴，是不可避免的，所以一般情形而言，数学模型总是近似的，既然这样，硬是要求数学问题的准确解也就没有意义了。后两种属计算数学范畴，计算数学的基本任务是分析数值方法的截断误差和舍入误差，并把它们控制在允许范围内。

1.2　误差、误差限及有效数字

设 x^{*} 是准确值，x 是 x^{*} 的近似值，怎样衡量 x 的精度？

定义 1.1　称 $e(x) = x^{*} - x$ 为 x 的绝对误差，简称误差。

注意，绝对误差不是误差的绝对值，$e(x) > 0$ 意味着 x 为 x^{*} 的不足近似值，$e(x) < 0$ 意味着 x 为 x^{*} 的过量近似值。

定义 1.2　设 $\varepsilon > 0$，若 $|x^{*} - x| \leqslant \varepsilon$，则称 ε 是 x 的一个误差上限，简单称为 x 误差限。

显然误差限不唯一，有实际意义的是最小的误差限 ε。

在衡量一个近似值的优劣时，最直观的是用绝对误差描述，但由定义 1.1 可以看出，如果能获得绝对误差，那也就获得了精确值了，而这是不太可能的。不过在实践中，通常可以根据问题的背景、测量工具的精度等，估计出近似值 x 的误差限是有可能的。

例如，用一把毫米刻度的尺子测量桌子的长度 x^*，读出桌子的近似长度 $x=1200\text{mm}$，它是 x^* 的近似值。x 的绝对误差无法求出，但从尺子的刻度可知 $|x^*-x|=|x^*-1200|\leqslant 0.5$，即 x 的误差限 $\varepsilon=0.5\text{mm}$。

绝对误差并不能完全刻画一个近似值的优劣程度。例如有甲、乙两个打字员，甲输入 100 个字中有 2 个错别字，乙方输入 1000 个字中有 5 个错别字。如何评价两位打字员的水平呢？如果按绝对误差的角度来衡量，那么甲比乙更优秀一点，但这会导致人们最不希望看到的：多做多错少做少错不做就不会错。如果从错误占总体工作的比例来看，甲错误率为 2%，而乙错误率为 0.5%，显然乙优秀于甲。

定义 1.3 称单位量上的误差 $e_r(x)=\dfrac{x^*-x}{x}$ 为 x 的相对误差。

定义 1.4 设 $\varepsilon_r>0$，若 $|e_r(x)|\leqslant\varepsilon_r$，则称 ε_r 是 x 的相对误差限。

显然，如果 ε 是 x 的一个误差限则

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

这说明 $\varepsilon_r=\dfrac{\varepsilon}{|x|}$ 是 x 的一个相对误差限。

定义 1.5 设近似值 x 以式(1.1)表示，若 x 的一个误差限是它的第 n 位的半个单位，即

$$\varepsilon=|e(x)|=|x^*-x|\leqslant 0.\underbrace{0\cdots}_{n-1}01\times 10^p\times\frac{1}{2}=\frac{1}{2}\times 10^{p-n}$$

则称 x 有 n 位有效数字。

定义 1.5 刻画了误差和有效数字的关系。例如，若近似值 $x=11.55$，它的误差限为 $|e(x)|\leqslant 0.5$，即

$$x=11.55=0.1155\times 10^2\Rightarrow p=2$$

$$\varepsilon=|e(x)|=|x^*-x|\leqslant\frac{1}{2}\times 10^{p-n}=\frac{1}{2}\Rightarrow n=p=2$$

即 x 有两位有效数字。

若已知 $x=17.390$ 有 3 位有效数字，则先把 $x=17.390$ 先写为式(1.1)形式，即

$$x=17.390=0.17390\times 10^2\Rightarrow p=2$$

由定义 1.5 知道它的一个误差限为

$$\varepsilon=|e(x)|=|x^*-x|\leqslant\frac{1}{2}\times 10^{2-3}=0.05$$

如果近似值 x 用式(1.1)表示，则有效数字位与相对误差限之间有十分重要的关系，下述定理告诉我们有效数字损失会导致相对误差增大，换句话说，有效数字损失会使计算结果的可靠性降低。

定理 1.1 近似值 x 用式(1.1)表示，若 x 具有 n 位有效数字，则其相对误差限为

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

反之，若 x 的相对误差限 $\varepsilon_r\leqslant\dfrac{1}{2(a_1+1)}\times 10^{-(n-1)}$，则 x 至少具有 n 位有效数字。

例 1.5　下列各数是按四舍五入原则得到的近似数，它们各有几位有效数字？

$$82.769,\ 0.00724,\ 1.3200$$

解： 82.769 是四舍五入得到的，故误差不超过末位(千分位)的半个单位，所以按定义它准确到千分位，有 5 位有效数字。同理 0.00724 和 1.3200 分别有 3 位和 5 位有效数字。

例 1.6　近似数 3.1 与 3.10，2.5 万和 2 万 5 千，3.1×10^3 与 3100 是否是相同的数？为什么？

解： 不是同一个数。因为按近似数计算规定原始数据要用有效数字写，凡不标明误差限的原始数据都被认为是有效数，于是近似数 3.1 与 3.10，2.5 万与 2 万 5 千，3.1×10^3 与 3100 就有不同的含义，它们的有效数字位数分别是 2，3；2，5；2，4。

例 1.7　祖冲之在公元 480 年曾计算出圆周率 π 的值在 3.1415926 和 3.1415927 之间，问祖冲之算出的圆周率近似值有多少位有效数字？

解： 记祖冲之算出的圆周率近似值为 $\tilde{\pi}$，于是 $\tilde{\pi}\in[3.1415926,\ 3.1415927]$，这说明 $|e(\tilde{\pi})|\leqslant0.1\times10^{-6}$ 依定义 1.5 知道 $\tilde{\pi}$ 有 7 位有效数字。

1.3　误差在计算过程中的传播

1.3.1　误差在函数值计算过程中的传播

在计算函数值时，如果自变量有误差会导致求出的函数值有误差，这一小节讨论当自变量带有误差时，对函数值误差的影响。

设 $f(x)$ 是一元函数，要计算在 x^* 处的函数值 $y^*=f(x^*)$，如果仅知道 x^* 的近似值 x，则只能计算出 y^* 的近似值 $y=f(x)$。假设自变量的误差为 $e(x)=x^*-x$，所计算函数值与真实函数值之间的误差记作 $e(y)=y^*-y=f(x^*)-f(x)$，如果 f 是一个可微函数，则有

$$e(y)=f(x^*)-f(x)=f'(x)(x^*-x)+o(x^*-x)$$

这里 $o(x^*-x)$ 是 (x^*-x) 的高阶无穷小量。舍去 $o(x^*-x)$ 得

$$e(y)\approx f'(x)(x^*-x)=f'(x)e(x) \tag{1.2}$$

式(1.2)说明，当函数的自变量的近似值 x 有误差 $e(x)$ 时，在做函数值的计算过程中其误差被放大了 $f'(x)$ 倍。

根据相对误差的定义，结合(1.2)式有

$$e_r(y)=\frac{e(y)}{y}\approx\frac{f'(x)}{f(x)}e(x)=\frac{xf'(x)}{f(x)}e_r(x) \tag{1.3}$$

式(1.3)表示自变量所带的相对误差在计算函数值时，对函数值相对误差的影响会被放大 $\dfrac{xf'(x)}{f(x)}$ 倍。

例 1.8　设 $y=t^\alpha$，自变量 x 有相对误差为 $e_r(x)$，用 x 计算得函数值 $y=x^\alpha$。试求 $e_r(y)$。

解： 由(1.3)得

$$e_r(y)\approx\frac{xf'(x)}{f(x)}e_r(x)=\frac{x\alpha x^{\alpha-1}}{x^{\alpha}}e_r(x)=\alpha$$

即 x^{α} 的相对误差是 x 的相对误差 $e_r(x)$ 的 α 倍。

例 1.9 正方形边长大约为 100cm。应该怎样测量，才能使面积误差不超过 1cm^2。

解： 设测得正方形边长为 x，由此计算得正方形的面积为 $s=x^2$。由式(1.2)知道

$$e(s)\approx 2xe(x)\Rightarrow|e(x)|\approx\left|\frac{e(s)}{2x}\right|\leqslant\frac{1}{200}=0.005$$

故测量边长时的误差不得超过 0.005cm。

设 $u=f(x,y)$ 是二元函数，若已知自变量的误差分别为 $e(x)$，$e(y)$，相对误差分别为 $e_r(x)$，$e_r(y)$。若 u 是可微函数，则容易求得

$$e(u)\approx du=f_x\times e(x)+f_y\times e(y) \tag{1.4}$$

$$e_r(u)\approx f_x\times\frac{x}{u}\times e_r(x)+f_y\times\frac{y}{u}\times e_r(y) \tag{1.5}$$

可以用类似的方式导出 n 元函数的函数值误差公式。

1.3.2 误差在四则运算中的传播

设 x 是 x^* 的近似值，误差为 $e(x)$，相对误差为 $e_r(x)$。y 是 y^* 的近似值，误差为 $e(y)$，相对误差为 $e_r(y)$。则

$$e(x\pm y)=x^*\pm y^*-(x\pm y)=e(x)\pm e(y) \tag{1.6}$$

$$e_r(x\pm y)=\frac{x^*\pm y^*-(x\pm y)}{x\pm y}=\frac{x}{x\pm y}e_r(x)\pm\frac{y}{x\pm y}e_r(y) \tag{1.7}$$

式(1.7)告诉我们，相近的数相减时，差的相对误差会很大。事实上，当 x 和 y 十分接近时，$|x-y|$就很小，$\frac{|x|}{|x-y|}$和$\frac{|y|}{|x-y|}$会很大，相对误差 $e_r(x)$和$e_r(y)$迅速扩大，导致差的相对误差会很大。还可以这样理解，相近的数前面若干位有效数字必然相同，相减后会引起有效数字丢失。例如，当 $x=123.0176$，$y=123.0131$，都有 7 位有效数字时，$x-y=0.0045$ 却只有两位有效数字。在数值分析中常通过改变计算公式以避免相近的数相减，或采用双精度(双倍位字长)计算以提高计算精度。

为讨论两数相乘时误差的传播方式，定义函数 $u=f(x,y)=xy$，由式(1.4)和式(1.5)得

$$e(xy)\approx ye(x)+xe(y) \tag{1.8}$$

$$e_r(xy)\approx e_r(x)+e_r(y) \tag{1.9}$$

用类似的方法可以得到

$$e\left(\frac{x}{y}\right)\approx\frac{1}{y}e(x)-\frac{x}{y^2}e(y) \tag{1.10}$$

$$e_r\left(\frac{x}{y}\right)\approx e(x)-e(y) \tag{1.11}$$

由式(1.8)和(1.10)知，如果 $|x|$ 很大，而 $|y|$ 很小时，计算结果的绝对值误差与相对误差都将很大。因此，实际计算中应当尽量避免用绝对值很大的数作乘数或用接近零的数作除数。简言之，避免大乘数和小除数。

例 1.10　设 x，y，z 和 w 都是近似数，$u=-\dfrac{xy}{zw}$ 已知 $e_r(x)$，$e_r(y)$，$e_r(z)$ 和 $e_r(w)$，求 $e_r(u)$ 和 u 的相对误差限。

解：由式(1.9)和式(1.11)可知

$$e_r(u)=e_r\left(\frac{xy}{zw}\right)\approx e_r(xy)-e_r(zw)\approx e_r(x)+e_r(y)-e_r(z)-e_r(w)$$

由三角形不等式得

$$|e_r(u)|\leqslant|e_r(x)|+|e_r(y)|+|e_r(z)|+|e_r(w)|$$

例 1.11　设 3.14 和 2.685 分别有 3 位和 4 位有效数字，估计函数值 $\sin(3.14\times 2.685)$ 的绝对误差和相对误差。

解：记 $x=3.14$ 和 $y=2.685$ 它们分别有 3 位和 4 位有效数字，所以

$$|e(x)|\leqslant 0.005,\quad |e(y)|\leqslant 0.0005$$

记 $u=\sin(xy)$，于是

$$|e(u)|\approx\left|\frac{\partial u}{\partial x}e(x)+\frac{\partial u}{\partial y}e(y)\right|=|y\cos(xy)e(x)+x\cos(xy)e(y)|\leqslant 8.178929\times 10^{-3}$$

以及

$$|e_r(u)|=\left|\frac{e(u)}{u}\right|\leqslant 9.758342\times 10^{-3}$$

1.4　计算方法的数值稳定性

定义 1.6　用一个计算方法进行计算时，如果初始数据误差或某一步计算的误差在以后的计算过程中不增长，就称该方法数值稳定，否则，若误差增长，就称该方法不稳定。

关于计算方法数值稳定性的定义从某种意义上看是一种定性的说法。但实际上，在讨论一个算法的优劣性的时候，误差分析是必不可少的。所以在讨论数值计算方法的数值稳定性时往往是通过误差分析完成的。这里以几个例子初探算法的数值稳定性的问题。

一元二次方程的求根公式是每一个中学生都必须熟练掌握的基础知识，在计算机上用求根公式解方程时出现了数值稳定性问题。

例 1.12　在 7 位十进制计算机上求 $x^2-26x+1=0$ 的根。

解：它的两个精确根为 $x_1^*=13+\sqrt{168}$，$x_2^*=13-\sqrt{168}$。

方法 1　用求公式计算，把参与运算的数都表达成式(1.1)的形式

$$\sqrt{168}\approx 0.1296148\times 10^2,\quad 13=0.1300000\times 10^2$$

于是

$$x_1=0.1300000\times 10^2+0.1296148\times 10^2=0.2596148\times 10^2$$

$$x_2=0.1300000\times 10^2-0.1296148\times 10^2=0.0003852\times 10^2$$

在算法 1 的情况下，x_1 有 7 位有效数字，x_2 有 4 位有效数字。

方法 2　x_1 的计算方法同算法 1，而对于 x_2 的计算采用如下形式

$$x_2^*=13-\sqrt{168}=\frac{13^2-168}{13+\sqrt{168}}=\frac{1}{x_1}\approx 0.0385186=x_2$$

它具有六位有效数字。

对于本例，算法 2 的数值稳定性优于算法 1。

在数值算法的设计中，从某个初始值开始，依一定的规则依序算出各中间结果及最后结果的方法称为递推法(也称为迭代法)。其中初始条件或问题本身已经给定，或是通过对问题的分析与化简后确定。递推法是数值计算中极为重要的算法构造方法，其主要优点是算法结构比较简单，容易在计算机上实现，但稳定性问题不容忽视。

例 1.13 取 7 位数字计算积分

$$I_n = \int_0^1 \frac{x^n}{x+5}\,\mathrm{d}x \quad n = 0,1,\cdots,14$$

(所谓取 7 位数字计算，指计算过程中从左边第一位非零数字起第 8 位的数值按四舍五入的原则舍入)

解： 对被积函数作适当变形，就有

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

方法 1 构造递推算法

$$\begin{aligned} I_0 &= \int_0^1 \frac{1}{x+5}\mathrm{d}x = \ln\frac{6}{5} = 0.1823216 \\ I_n &= \frac{1}{n} - 5I_{n-1} \quad n = 1,2,\cdots,14 \end{aligned} \tag{1.12}$$

I_0 有 7 位有效数字。由此算法计算得到的 I_n 列于表 1-1。

表 1-1 例 1.13 算法 1 计算结果

n	I_n	n	I_n	n	I_n
0	0.1823216	5	0.0284583	10	0.0466767
1	0.0883922	6	0.0243750	11	-0.1424738
2	0.0580390	7	0.0209821	12	0.7957026
3	0.0431383	8	0.0200893	13	-3.901590
4	0.0343083	9	0.0106647	14	19.57938

从表 1-1 中可以发现这种算法的计算结果是很不可靠的。事实上，由于被积函数总是非负的，所以积分结果也应该是非负的，而计算结果中有 $I_{11} = -0.1424738 < 0$。

方法 2 由于 $0 < x < 1$ 时，x^n 收敛于零($n \to \infty$)。这使我们有理由相信：当 n 很大时将有 $I_n = 0$。为此，我们构造迭代过程

$$\begin{cases} I_{100} = 0 \\ I_{n-1} = \dfrac{1}{5n} - \dfrac{I_n}{5} \quad n = 100,\ 99,\ \cdots \end{cases} \tag{1.13}$$

由此算法计算得到的 I_n 列于表 1-2。