应用数值分析

Applied Numerical Analysis

刘春凤 何亚丽 等 编著

刘保相 审校

冶金工业出版社

应用数值分析

Applied Numerical Analysis

编著　刘春凤　何亚丽
　　　米翠兰　马醒花　杨爱民
审校　刘保相

北京
冶金工业出版社
2005

内 容 简 介

本书共分为九章，主要内容包括绪论、插值法、函数逼近、解线性方程组的直接方法、解线性方程组的迭代法、数值微分与数值积分、非线性方程求根、常微分方程数值解法、矩阵特征值与特征向量的计算以及附录和习题参考答案。本书主要是为工科硕士生及高年级本科生提供适用的教科书，也可供有关专业工程技术人员阅读和参考。

图书在版编目(CIP)数据

应用数值分析/刘春凤等编著. —北京：冶金工业出版社，2005.2

ISBN 7-5024-3755-X

Ⅰ.应… Ⅱ.刘… Ⅲ.数值计算 Ⅳ.0241

中国版本图书馆 CIP 数据核字(2005)第 046950 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 王贺兰 李文彦 责任印制 牛晓波
北京市铁成印刷厂印刷；冶金工业出版社发行；各地新华书店经销
2005 年 2 月第 1 版，2005 年 2 月第 1 次印刷
787mm × 1092mm 1/16；16 印张；383 千字；244 页
36.00 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
（本社图书如有印装质量问题，本社发行部负责退换）

前　　言

随着现代科学技术的发展和计算机的广泛使用，在各门自然科学和工程技术的发展中，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。目前，在诸多科学领域里，无论是数学工作者、数值计算专家，还是一般的工程技术人员，数值计算已经成为必须掌握的知识和工具。

数值分析是高等工科院校日益普遍开设的一门课程。本书根据教育部关于“数值计算方法”课程的基本要求，力求满足“重概念、重方法、重应用、重能力”的培养要求，为工科硕士生及高年级本科生提供适用的教科书，也可供有关专业工程技术人员参考。

本书介绍的数值方法大多是基础性和应用较广的方法，涉及数值计算的基本问题、函数的插值和逼近、数值微积分、常微分方程初值问题的数值解、线性代数方程组和特征值问题的数值解法、非线性方程的数值解法等。阅读本书只要具备一般理工科大学的《高等数学》、《线性代数》的基础，而且粗懂 Mathematica 软件即可。

由于篇幅所限，也考虑到一般工科院校的人才培养目标，本书力求通俗易懂，简洁实用，一些有效但需要用到较多其他数学工具的方法未列入本书范围。本书以能正确选择计算对象的计算方法为前提，领会计算原理和掌握计算步骤为主线，注重如何在计算机上实现数值计算。

Mathematica 是一个符号计算系统，是一个做数学的软件系统，对于应用数学的科技工作者来说，是博学多才而又极其友好的“帮手”。目前已在诸多领域得到成功应用。本书突出的特点是：紧密围绕数值分析的内容主线，有机地引入 Mathematica 的相关内容，并配置了适量的应用范例。力求内容简明，计算快捷，结果直观，以提高读者科学计算的能力。

讲授本书全部内容大约需要 56 ~ 70 学时，可以根据不同的教学对象和要求选择其中的某些章、节。

刘保相教授对本书的编写提出了宝贵的意见，并审校了全稿。

感谢我校领导对本书的出版给予了大力支持；感谢冶金工业出版社给予的帮助。

由于水平有限，且编写时间仓促，书中有不妥之处，恳请读者批评指正。

编著者

2005 年 1 月

目　　录

第一章　绪　　论

第一节　数值分析方法的内容

数值分析（Numerical Analysis）方法是一种研究并解决数学问题的数值近似解方法，是在计算机上使用的解数学问题的方法之一，简称计算方法（Computational Method）。它的计算对象是那些在理论上有解，而无求解公式或计算量过大难以用手工实现的数学问题。

数值分析的内容包括函数的数值逼近、数值微分与数值积分、非线性方程数值解、数值线性代数、微分方程数值解等。

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。即算法只能包括加、减、乘、除运算和逻辑运算。

第二，要有可靠的理论分析，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，要有好的计算复杂性。时间复杂性好是指节省时间，空间复杂性好是指节省存储量。

第四，要有数值实验。

过去，大多数高校仅在数学系的计算数学专业和计算机系开设此课程。随着计算机技术的迅速发展和普及，现在很多专业已将本课程作为必修课程。数值分析方法与计算机技术相结合已深入到计算物理、计算化学、计算生物学、计算经济学等各个领域，计算机上使用的数值计算方法浩如烟海，本书只限于介绍最基本的数值分析方法。

第二节　误差与有效数字

一、误差的来源

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：

1. 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，忽略了一些次要因素，这样建立的数学模型虽然具有“精确”、“完美”的外衣，其实只是客观现象的一种近似。这种数学模型与实际问题之间出现的误差称为模型误差。

2. 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。

3. 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。例如

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots + \frac{(-1)^n x^{2n}}{(2n)!} + \cdots$$

当 $|x|$ 很小时，可以用 $1 - \frac{x^2}{2!}$ 作为 $\cos x$ 近似值，由交错级数判断的莱布尼兹（Leibniz）准则，它的截断误差的绝对值不超过 $\frac{x^4}{24}$。

4. 舍入误差（Roundoff 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 $x = x^* \pm \varepsilon$。

例如：用毫米刻度的直尺测得某一物体的长度为 123mm，由于直尺以毫米为刻度，所以其误差不超过 0.5mm，即 $|x - x^*| \leqslant 0.5$。从这个不等式我们不能得出准确值，但却知道其范围为 $122.5 \leqslant x \leqslant 123.5$。

绝对误差并不足以表示近似值的好坏。例如，设 $x_1 = 100 \pm 1$，$x_2 = 1000 \pm 1$，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。可见决定一个量的近似值的精确度除了要看绝对误差的大小之外，还必须考虑到该量本身的大小。这就有必要引入相对误差的概念。

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）

如果数值运算取有限位进行，则当数值位超过运算位长的部分将按四舍五入的原则舍入。例如，我们取 4 位位长运算，则 $x = 15.26784$ 按取值 $x^* = 15.27$ 实施运算，而

$$|x - x^*| \leqslant \frac{1}{2} \times 10^{-2} = 0.005$$

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

$$\left|\sqrt{3} - 1.73\right| < 0.5 \times 10^{-2}, \left|\sqrt{3} - 1.7321\right| < 0.5 \times 10^{-4}$$

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。

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取位。

例1　对下列各数写出具有五位有效数字的近似值：

$$236.48, 0.00234711, 9.000024, 9.000034\times10^3$$

解　按定义，上述各数具有五位有效数字的近似值分别是

$$236.48, 0.0023471, 9.0000, 9.0000\times10^3$$

注意： $x^*=9.000024$ 的五位有效数字的近似值是9.0000，而不是9，因为9只有一位有效数字。

第三节　误差的传播

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。由微分学可知，当自变量改变量（误差）很小时，函数的微分作为函数改变量的主要线性部分可以近似函数的改变量，故利用微分运算公式可导出误差运算公式。

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

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

例如 $\sqrt[3]{x^*}$ 的相对误差约为 x^* 的相对误差的 $\frac{1}{3}$ 倍是因为

$$e_r(\sqrt[3]{x^*})\approx\frac{\mathrm{d}(\sqrt[3]{x^*})}{\sqrt[3]{x^*}}=\frac{\frac{1}{3}(x^*)^{-\frac{2}{3}}\mathrm{d}(x^*)}{\sqrt[3]{x^*}}=\frac{1}{3}\frac{\mathrm{d}(x^*)}{x^*}=\frac{1}{3}e_r(x^*)$$

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（$x_1^*,x_2^*,\cdots,x_n^*$）可微分，则当数据误差较小时，绝对误差为

$$e(y^*)=y-y^*=f(x_1,\cdots,x_n)-f(x_1^*,\cdots,x_n^*)\approx \mathrm{d}f(x_1^*,x_2^*,\cdots,x_n^*)$$

$$=\sum_{k=1}^{n}\left(\frac{\partial f(x_1^*,\cdots,x_n^*)}{\partial x_k}\right)(x_k-x_k^*)=\sum_{k=1}^{n}\left(\frac{\partial f(x_1^*,\cdots,x_n^*)}{\partial x_k}\right)e(x_k^*)$$

相对误差为

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

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

特别地，由上述结论可推出常用的两数之和、差、积、商的误差公式。

$$\begin{cases}e(x_1\pm x_2)=e(x_1)\pm e(x_2)\\ e_r(x_1\pm x_2)=\dfrac{x_1}{x_1\pm x_2}e_r(x_1)+\dfrac{x_2}{x_1\pm x_2}e_r(x_2)\end{cases}$$

$$\begin{cases}e(x_1x_2)\approx x_2e(x_1)+x_1e(x_2)\\ e_r(x_1x_2)\approx e_r(x_1)+e_r(x_2)\end{cases}$$

$$\begin{cases}e\left(\dfrac{x_1}{x_2}\right)\approx\dfrac{1}{x_2}e(x_1)-\dfrac{x_1}{x_2^2}e(x_2)\\ e_r\left(\dfrac{x_1}{x_2}\right)\approx e_r(x_1)-e_r(x_2)\end{cases}$$

由以上各式还可得出：和、差的绝对误差限不超过各数的绝对误差限之和；积、商的相对误差限不超过各数的相对误差限之和，即

$$|e(x_1\pm x_2)|=|e(x_1)\pm e(x_2)|\leqslant|e(x_1)|+|e(x_2)|$$

$$|e_r(x_1x_2)|\approx|e_r(x_1)+e_r(x_2)|\leqslant|e_r(x_1)|+|e_r(x_2)|$$

$$\left|e_r\left(\frac{x_1}{x_2}\right)\right|\approx|e_r(x_1)-e_r(x_2)|\leqslant|e_r(x_1)|+|e_r(x_2)|$$

例2　已测得某物体行程s的近似值$s^*=800\text{m}$，所需时间t的近似值$t^*=35\text{s}$。若已知$|t-t^*|\leqslant0.05s$，$|s-s^*|\leqslant0.5\text{m}$，试求平均速度$v$的绝对误差限和相对误差限。

解　因为$v=\frac{s}{t}$，由商的误差估计式有

$$e(v)=e\left(\frac{s}{t}\right)\approx\frac{1}{t}e(s)-\frac{s}{t^2}e(t),e_r(v)=e_r\left(\frac{s}{t}\right)\approx e_r(s)-e_r(t)$$

故

$$|e(v)| \approx \left|\frac{1}{t}e(s) - \frac{s}{t^2}e(t)\right| \leqslant \frac{1}{t}\varepsilon(s) + \frac{s}{t^2}\varepsilon(t) = \frac{1}{35}\times 0.5 + \frac{800}{35^2}\times 0.05$$

$$\approx 0.0469 \leqslant 0.05$$

$$|e_r(v)| \approx |e_r(s) - e_r(t)| \leqslant |e_r(s)| + |e_r(t)| \leqslant \frac{0.5}{800} + \frac{0.05}{35} \approx 0.00205$$

所以平均速度 v 的绝对误差限和相对误差限分别为 0.05 和 0.00205。

第四节 误差的改善

在实际计算中误差几乎是不可避免的，定量地分析误差的积累过程往往都是非常繁杂的。况且，一个工程或科学计算问题通常要算千万次，如果每步运算都分析误差，这是不可能的，也是不必要的。为了减少误差的影响，设计算法时应注意以下几个方面，以防止误差危害现象的产生。

一、选择数值稳定的算法

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 3 建立积分 $I_n = \int_0^1 \frac{x^n}{x+5}\mathrm{d}x \quad n = 0,1,\cdots,20$ 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

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

计算出 I_0 后，由递推关系式可逐次求出 $I_1, I_2, \cdots, I_{20}$ 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^* $(n = 1,2,\cdots,20)$。即

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

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 1-1 可得

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

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 1-1 是数值不稳定的。

现在从另一方向使用这一公式

$$I_{n-1} = \frac{1}{5}\left(\frac{1}{n} - I_n\right) \qquad (n = 20,19,\cdots,1) \tag{1-2}$$

只要给出 I_{20}的一个近似值 I_{20}^*，即可递推得到 $I_{19}^*,I_{18}^*,\cdots,I_0^*$ ，类似于上面的推导可得

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

每递推一步误差缩小到原值的$\frac{1}{5}$，所以递推公式 1-2 是数值稳定的。

由于

$$x \in [0,1] \text{ 时}, \frac{x^n}{6} < \frac{x^n}{5+x} < \frac{x^n}{5}$$

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

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

二、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，最好是改变计算方法，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。

三、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

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

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

例4 在五位十进制计算机上，计算

$$A = 52492 + \sum_{i=1}^{1000} \delta_i, \quad 其中 0.1 \leqslant \delta_i \leqslant 0.9$$

把运算的数写成规格化形式

$$A = 0.52492 \times 10^5 + \sum_{i=1}^{1000} \delta_i$$

由于在计算机内计算时要对阶，若取 $\delta_i = 0.9$，对阶时，$\delta_i = 0.000009 \times 10^5$，在五位计算机中表示为0，因此

$$A = 0.52492 \times 10^5 + 0.000009 \times 10^5 + \cdots + 0.000009 \times 10^5 = 0.52492 \times 10^5$$

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