数 值 计 算 方 法

数值计算方法课程组

中国石油大学-华东

（课堂专属 切勿网传）

总成绩评定

1. 总成绩=50% 期末成绩+25% 平时成绩 +25% 上机成绩
2. 期末考试为闭卷纸质考试
3. 平时成绩= 50% 纸质作业成绩+ 50% 惟真学堂测试成绩
4. 上机成绩= 50% 上机作业成绩+ 50% 独立程序测试成绩

作业提交

1. 每周二交纸质作业，要求独立完成
2. 上机编程作业分组完成，每周六每组交 1 份编程作业
3. 编程作业文件为 word 或 pdf，应包含作业题目、算法设计、程序及运行结果
4. 编程语言不限制，建议用 Matlab 或 Python

使用教材及参考书

教材：

计算方法（第三版），周生田、王际朝、郭会主编，北京：石油工业出版社，2020 年.

参考书：

1. 数值计算方法算法设计与学习指导（胶印），聂立新、李维国、刘新海编著，青岛：中国石油大学（华东），2022 年.
2. 数值计算方法 (第三版)，李维国、聂立新编著，北京：石油工业出版社，2019 年.
3. 数值计算方法（第二版）上、下册，林成森编著，北京：科学出版社，2005 年.
4. 数值分析原理，封建湖、车刚明、聂玉峰编著，北京：科学出版社，2001 年.

问题：数值计算方法有什么用处？

1. 解决科学技术与工程中的复杂问题.
2. 数值计算与理论研究、物理实验，已并列成为当今世界科学活动的三种主要方式.

解决问题的步骤：

实际问题 $\longrightarrow$ 建立数学模型 $\longrightarrow$ 数值问题
数值问题 $\longrightarrow$ 研究数值计算方法（重点内容）
程序设计 $\longrightarrow$ 计算机实现 $\longrightarrow$ 近似结果

数值方法的设计原则

(1) 收敛性：方法的可行性
(2) 稳定性：初始数据等产生的误差对结果的影响
(3) 误差估计：运算结果不能产生太大的偏差且能够控制误差

(4) 便于编程实现：逻辑复杂度要小
(5) 计算量要小：时间复杂度要小，运行时间要短
(6) 存贮量要尽量小：空间复杂度要小

(1)-(3) 称为可靠性分析，(4)-(6) 称为计算复杂度.

主要内容

1. 数值逼近

   多项式插值与函数逼近, 曲线拟合与最小二乘问题 ——第 4，5 章
   数值积分与数值微分——第 6 章

2. 数值代数

   线性方程组的直接解法和迭代算法 ——第 3，7 章
   特征值问题的计算方法 ——第 8 章

3. 方程求解

   非线性方程的数值解法——第 2 章
   常微分方程的数值解法——第 9 章

目录

目 录

1. 模型误差： 实际问题 $\Leftrightarrow$ 抽象出的数学模型
2. 观测误差： 模型中某些参数（或物理量）的观测值
3. 方法误差（截断误差）： 数学模型与数值算法之间的误差
4. 舍入误差： 由于机器字长所限，原始数据在计算过程产生的误差

数值计算方法主要讨论截断误差和舍入误差所带来的影响。

§1.1.2 误差分析的基本概念

定义 (绝对误差)

设 x 为真值（精确值），x^* 为 x 的一个近似值，称 $e = x^* - x$ 为近似值 x^* 的绝对误差，简称误差。

注

1. 绝对误差有量纲的，可正可负，常常是无限位的.
2. 绝对误差无法计算时，可以估计出它的绝对上界 ε，称为绝对误差限，即 $|x^* - x| \leq \varepsilon$，如，$\pi^* = 3.14159$

$$|\pi^* - \pi| \leq 0.5 \times 10^{-5}$$

3. 绝对误差不能完全表示近似值的好坏.

定义 (相对误差)

设 x 为真值（精确值），x^* 为 x 的一个近似值，称 $\frac{e}{x} = \frac{x^*-x}{x}$ 为近似值 x^* 的相对误差，记着 e_r。

注

1. 相对误差是一个相对数，没有量纲的，可正可负.
2. 相对误差的绝对值的上界 ε_r，称为相对误差限，即 $|\frac{e}{x}| \leq \varepsilon_r$.
3. 实际计算时，x 是未知的，相对误差通常可取 $e_r = \frac{e}{x^*}$, 原因是

$$\frac{e}{x} - \frac{e}{x^*} = \frac{e(x^*-x)}{xx^*} = \frac{e^2}{(x^*-e)x^*} = \frac{(e/x^*)^2}{1-e/x^*} = \mathcal{O}(e_r^2)$$

§1.1.3 有效数字

定义 (有效数字)

若近似值 x^* 与准确值 x 的误差绝对值不超过某一位的半个单位，该位到 x^* 的第一位非零数字共有 n 位, 则称 x^* 有 n 位有效数字.

例

$$\begin{aligned}
&\pi = 3.1415926535\ldots \\
&\pi^* = 3.14, \quad |e| \leq \frac{1}{2} \times 10^{-2}, \quad 3\text{位有效数字} \\
&\pi^* = 3.141592, \quad |e| \leq \frac{1}{2} \times 10^{-5}, \quad 6\text{位有效数字}
\end{aligned}$$

用科学计数法，记 $x^* = \pm 0.a_1 a_2 \cdots a_n \cdots a_k \times 10^m$, $a_1 \neq 0$, $a_1, a_2, \ldots, a_k$ 为 0 到 9 中任一整数，如果 $|x^* - x| \leq \frac{1}{2} \times 10^{m-n}$, 则 x^* 至少有 n 位有效数字.
有效位数越多，绝对误差限越小.

例

记 $\pi^* = 3.1415$, 问 π^* 有几位有效数字?
解:

$$\begin{aligned} \pi^* &= 0.31415 \times 10^1 \\ |\pi^* - \pi| &< 0.5 \times 10^{-3} = 0.5 \times 10^{1-4} \end{aligned}$$

所以知，π^* 有 4 位有效数字，精确到小数点后第 3 位.

注

若 x^* 的每一位都是有效数字，则 x^* 称是有效数. 特别，经“四舍五入”得到的数均为有效数.

例

按四舍五入原则写出下列各数具有五位有效数字的近似数:

$$187.9325, 0.03785551, 8.000033, 2.7182818$$

解：按定义，上述各数具有五位有效数字的近似数分别是

$$187.93\,, 0.037856\,, 8.0000\,, 2.7183.$$

注意 8.0000 有 5 位有效数字，精确到小数点后第 4 位，而 8 只有一位有效数字.

数值运算的误差估计

注（1）

两个近似数 x_1^*, x_2^*, 其误差限分别为 $\epsilon(x_1^*), \epsilon(x_2^*)$, 四则运算得到误差限分别为

$$\begin{aligned}
\epsilon(x_1^* \pm x_2^*) &= \epsilon(x_1^*) + \epsilon(x_2^*),\\
\epsilon(x_1^* \cdot x_2^*) &\approx |x_1^*| \cdot \epsilon(x_2^*) + |x_2^*| \cdot \epsilon(x_1^*),\\
\epsilon(\frac{x_1^*}{x_2^*}) &\approx \frac{|x_1^*|\epsilon(x_2^*) + |x_2^*|\epsilon(x_1^*)}{|x_2^*|^2}, x_2^* \neq 0.
\end{aligned}$$

设真实值分别为 x_1, x_2,

$$\begin{aligned}
e(x_1^* - x_2^*) = |x_1 - x_2 - (x_1^* - x_2^*)| &= |(x_1 - x_1^*) - (x_2 - x_2^*)|\\
&\leq |(x_1 - x_1^*)| + |(x_2 - x_2^*)| = \epsilon(x_1^*) + \epsilon(x_2^*)
\end{aligned}$$

注（2）

对于一元函数 $y=f(x)$, 若用近似数 x^* 取代 x, 则

$$e(y)=f(x^*)-f(x)=f'(\xi)(x^*-x)=f'(\xi)e(x^*)$$

x^* 与 x 非常接近时，可认为 $f'(\xi)=f'(x^*)$, 从而有

$$|e(y)|\approx|f'(x^*)|\cdot|e(x^*)|\Rightarrow|\epsilon(y)|\approx|f'(x^*)|\cdot|\epsilon(x^*)|$$

若函数为多元函数 $y=f(x_1,x_2,\ldots,x_n)$，则有

$$\begin{aligned}&f(x_1^*,x_2^*,\ldots,x_n^*)-f(x_1,x_2,\ldots,x_n)\\\approx&\sum_{i=1}^{n}\frac{\partial f(x_1^*,x_2^*,\ldots,x_n^*)}{\partial x_i}(x_i^*-x_i)\end{aligned}$$

目 录

注意事项

计算机中两数相加时，要先对阶，即把两数都写成绝对值小于 1 而阶码相同数.

1、避免绝对值小的分母

如计算 $\frac{x}{y}$，若 $0 < |y| \le |x|$，则可能对计算结果带来严重影响，应尽量避免.

例

线性方程组

$$\begin{cases} 0.00001x_1 + x_2 = 1, \\ 2x_1 + x_2 = 2, \end{cases}$$

的准确解为 $x_1 \approx 0.5000025$，$x_2 \approx 0.999995$.

解：现在仿机器实际计算用四位浮点十进制数和消元法求解上述方程组

$$
\begin{aligned}
10^{-4} \times 0.1000 x_1 + 10^1 \times 0.1000 x_2 &= 10^1 \times 0.1000, \\
10^1 \times 0.2000 x_1 + 10^1 \times 0.1000 x_2 &= 10^1 \times 0.2000.
\end{aligned}
$$

若用 (1) 式消去 (2) 式，需要计算乘数 $l = \frac{10^1 \times 0.2000}{10^{-4} \times 0.1000} = 10^6 \times 0.2000$. 然后用式 (2)− 式 (1) $\times\, l$ 得到解为

$$
x_2 = 1, x_1 = 0.
$$

显然结果失真.

2、注意避免两个相近数的相减

两个相近的数相减，有效数字会大大损失。例如，

$a_1 = 0.12345, a_2 = 0.12346$，各有 5 位有效数字。而 $a_2 - a_1 = 0.00001$，只剩下 1 位有效数字。

避免办法：进行变换。几种经验性避免方法:

$$\sqrt{x+\varepsilon} - \sqrt{x} = \frac{\varepsilon}{\sqrt{x+\varepsilon} + \sqrt{x}},$$

$$\ln(x+\varepsilon) - \ln(x) = \ln(1 + \frac{\varepsilon}{x})$$

$$\text{当}|x| << 1\text{时}, 1 - \cos x = 2\sin^2\frac{x}{2}$$

$$e^x - 1 = x(1 + \frac{1}{2}x + \frac{1}{6}x^2 + \cdots)$$

3、避免大数吃小数

例

在五位十进制计算机上，计算 $A = 51234 + \sum_{i=1}^{1000} \delta_i$, 其中 $0.1 \le \delta_i \le 0.9$.

解：若取 $\delta_i = 0.9$, 对阶时 $\delta_i = 0.000009 \times 10^5$，由于计算机只能表示五位小数，所以

$$A = 0.51234 \times 10^5 + 0.000009 \times 10^5 + \cdots + 0.000009 \times 10^5 \triangleq 0.51234 \times 10^5.$$

对阶时出现了大数 51234 吃掉小数 δ_i 的结果.
如果计算时先把数量级相同的一千个 δ_i 相加，最后再加 51234，就不会出现大数吃掉小数的现象.
这时 $\sum_{i=1}^{1000} \delta_i = 0.9 \times 10^3$, 于是

$$A = 0.51234 \times 10^5 + 0.00900 \times 10^5 = 52134.$$

所以在数值计算中，应先分析计算的数量量级，编程序时加以合理安排, 使重要的物理量不至在计算过程中被“吃掉”.

4、先化简再计算，减少步骤，避免误差积累

同样一个计算问题，如果能减少运算次数，不但可以节省计算时间，还能减少舍入误差，数值计算中需要遵循的原则.

一般来说，计算机处理下列运算的速度为:

$$(+,-) > (\times,\div) > (\exp)$$

例

计算 x^{255} 的值.

解: 如果逐个相乘要用 254 次乘法，但若写成

$$x^{255} = x \cdot x^2 \cdot x^4 \cdot x^8 \cdot x^{16} \cdot x^{32} \cdot x^{64} \cdot x^{128}$$

只需做 14 次乘法运算即可.

又如计算多项式

$$p_n(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$$

的值，若直接计算 $a_k x^k$ 再逐项相加，一共需做

$$n + (n-1) + \cdots + 2 + 1 = \frac{n(n+1)}{2}$$

次乘法和 n 次加法.
若采用秦九韶算法

$$b_0 = a_0,\ \ b_i = a_i + b_{i-1}x,\ \ i = 1, 2, \cdots, n,$$

则 $b_n = p_n(x)$, 即要 n 次乘法和 n 次加法就可算出 $p_n(x)$ 的值.

5、使用稳定的算法

一个算法如果输入数据有扰动（即误差），而计算过程中舍入误差不增长，则称此算法是数值稳定的，否则此算法就称为不稳定的。

例

计算 $I_n = \int_0^1 x^n e^{x-1} dx, \quad n = 0, 1, 2, \ldots$
易知，$x \in [0,1]$ 时，$e^{-1} < e^{x-1} < e^0$，从而

$$0 < \frac{1}{e(n+1)} < I_n < \frac{1}{n+1}$$

算法 1：
由分部积分法可得递推公式

$$I_0 = 1 - e^{-1}, \quad I_n = 1 - nI_{n-1}$$

用 4 位有效数字计算可知：$I_0^* = 0.6321$,

n	1	2	3	4	5	6	7	8
I_n^*	0.3679	0.2642	0.2074	0.1704	0.1480	0.1120	0.2160	-0.7280

注意到算法第 n 步的误差 e_n

$$|e_n| = |I_n - I_n^*| = |1 - nI_{n-1} - 1 + nI_{n-1}^*| = n|e_{n-1}| = n!|e_0|$$

由于 $|e_0| \leq \frac{1}{2} \times 10^{-4}$, $|e_8| = 8!|e_0| \leq 40320|e_0|$, 误差扩大了 4 万倍，故算法 1 不稳定.

算法 2:

$$I_n = 1 - nI_{n-1} \Longrightarrow I_{n-1} = \frac{1}{n}(1 - I_n)$$

则可以先估计一个 $I_N, N >> n$. 注意到

$$\frac{1}{e(N+1)} < I_N < \frac{1}{N+1}$$

可取 $I_N^* = \frac{1}{2}(\frac{1}{e(N+1)} + \frac{1}{N+1}) \approx I_N$, 显然当 $N \longrightarrow +\infty$, $e_N = I_N - I_N^* \longrightarrow 0$.
此时相对误差

$$|e_n| = |\frac{1}{n+1}(1 - I_{n+1}) - \frac{1}{n+1}(1 - I_{n+1}^*)| = \frac{|e_{n+1}|}{n+1} = \frac{|e_N|}{(n+1)\cdot(n+1)\cdots N}$$

故算法 2 稳定.

目 录

计算机的数系结构

计算机的数系是一个不完整的数系。计算机只能表示有限个数，即计算机的精度是有限的.

每种计算机内部运算是按固定的有限位数进行的，也就是按固定位数的有限位浮点数进行运算的.

浮点数系统由四个整数表征：基 β ，精度（尾数）位数 t，下溢界 L 和上溢界 U .

规格化的浮点数可以表示为

$$x = \pm 0.d_1 d_2 \cdots d_t \times \beta^j = (d_1 * \beta^{-1} + d_2 * \beta^{-2} + \cdots + d_t * \beta^{-t}) \times \beta^j$$

这里 $0 \leq d_i < \beta, i = 1, 2, \ldots t$, $d_1 > 0$,$0.d_1 d_2 \cdots d_t$ 称为尾数，$L \leq j \leq U$ 为阶码.

从而可知，计算机所表示的数的集合为

$$\begin{aligned} F = & \{f = \pm 0.d_1 d_2 \cdots d_t \times \beta^j | 0 \le d_i < \beta, i = 1, 2, \ldots t, \\ & d_1 > 0, L \le j \le U\} \cup \{0\}. \\ = & F(\beta, t, L, U) \end{aligned}$$

注

1. 若 $f \neq 0$, 则 $m \le |f| \le M$, $m = \beta^{L-1}, M = \beta^U(1 - \beta^{-t})$.
2. 当 $f > M$ 时，计算机就出现溢出而中断运算.
3. 当 $f < m$ 时，称为下溢，计算机自动作零处理.

若 $x \in [-M, M]$，当 $x \neq 0$ 且不能用浮点规格化表示时，在计算机中就不能精确表示. 这时需要按舍入规则或截断规则得到浮点数表示.

- 舍入. x 的浮点数 $fl(x)$ 满足

$$|x - fl(x)| = \min_{f \in F} |x - f|,$$

若 $d_{t+1} < \beta/2$, 则 d_{t+1} 后各数舍去，若 $d_{t+1} \geq \beta/2$, 则 d_t 改为 $d_t + 1$, 之后各位舍入.

- 截断. $|fl(x)| \leq |x|$, 即舍去 d_{t+1} 及其后各数.

机器精度： 一个数 x 与计算机中能精确表示的与该数最近的一个浮点数 $fl(x)$ 的最大相对间隔，记为 u 满足

$$\left|\frac{fl(x) - x}{x}\right| \leq u = \begin{cases} \frac{1}{2}\beta^{1-t}, & \text{舍入法} \\ \beta^{1-t}, & \text{截断法} \end{cases}$$

例： $F = F(\beta, t, L, U)$, $\beta = 2, t = 3, L = -1, U = 2$, 写出 F 其所表示的所有数. 注意到：

$$F = \quad \{f = \pm 0.d_1 d_2 d_3 \times \beta^j | 0 \le d_i < 2, i = 1, 2, 3, d_1 > 0, -1 \le j \le 2\} \cup \{0\}.$$

$$\begin{array}{ccccc} d_1 & 1 & 1 & 1 & 1 \\ d_2 & 0 & 0 & 1 & 1 \\ d_3 & 0 & 1 & 0 & 1 \end{array}$$

故尾数只有四个值

$$\begin{aligned}
(0.100)_2 &= 1 \times 2^{-1} + 0 \times 2^{-2} + 0 \times 2^{-3} = \frac{1}{2} \\
(0.101)_2 &= 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} = \frac{5}{8} \\
(0.110)_2 &= 1 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3} = \frac{6}{8} \\
(0.111)_2 &= 1 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} = \frac{7}{8}
\end{aligned}$$

$F(2,3,-1,2)$ 中包含的非零数

$\pm 0.d_1 d_2 d_3 \times \beta^j$	-1	0	1	2
$(0.100)_2 = \frac{1}{2}$	$\pm\frac{1}{4}$	$\pm\frac{1}{2}$	± 1	± 2
$(0.101)_2 = \frac{5}{8}$	$\pm\frac{5}{16}$	$\pm\frac{5}{8}$	$\pm\frac{5}{4}$	$\pm\frac{5}{2}$
$(0.110)_2 = \frac{3}{8}$	$\pm\frac{3}{4}$	$\pm\frac{3}{4}$	$\pm\frac{3}{2}$	± 3
$(0.111)_2 = \frac{7}{8}$	$\pm\frac{7}{16}$	$\pm\frac{7}{8}$	$\pm\frac{7}{4}$	$\pm\frac{7}{2}$

单精度与双精度

二级制实数系统有单精度和双精度之分. 单精度 32 位和双精度 64 位，表示正负号、阶码、和尾数所占二进制的总长度，如下表所示:

单精度	阶码正负号	阶码	尾数正负号	尾数
32	1 位	7 位	1 位	23 位
双精度	阶码正负号	阶码	尾数正负号	尾数
64	1 位	10 位	1 位	52 位

单精度和双精度约相当于十进制 7 位和 15 位有效数字，其绝对值范围分布为

单精度

$$2^{-128} \sim 2^{128} (2.9 \times 10^{-39} \sim 3.4 \times 10^{38}),$$

双精度

$$2^{-1024} \sim 2^{1024} (5.56 \times 10^{-309} \sim 1.79 \times 10^{309})$$

低于该范围的值视为 0, 高于该范围的视为 ∞.
对于舍入规则产生的机器精度 ε, 单精度（32 位）$\varepsilon = 2^{-23} \approx 1.19 \times 10^{-7}$, 双精度（64 位）$\varepsilon = 2^{-52} \approx 2.22 \times 10^{-15}$, 四精度（128 位）$\varepsilon = 2^{-112} \approx 1.93 \times 10^{-34}$.

知识小结

主要内容

误差来源：模型误差、观测误差、截断误差、舍入误差
误差定义：绝对误差、相对误差、有效数字
数值算法的稳定性
数值计算的误差估计
几点注意事项：避免相近数相减、小分母、大数吃小数，简化计算步骤
计算机的数系结构

重点及难点

重点：绝对误差、相对误差、有效数字、数值算法的稳定性
难点：数值计算的误差估计

Many thanks for your attention！