数值分析：绪论

刘文杰

哈尔滨工业大学数学学院

http://homepage.hit.edu.cn/LiuWenjie

计算数学

1947 年 Von Neumann 和 Goldstine 在《美国数学会通报》发表了题为“高阶矩阵的数值求逆”的著名论文，开启了**现代计算数学**的研究。
一般来说，计算数学主要研究如何求出数学问题的**近似解（数值解）**，包括算法的设计、分析与计算机实现。

计算数学主要研究内容

数值代数（线性方程组求解和矩阵特征值计算），非线性方程和方程组求解，数值逼近，数值微积分，微分方程数值解（常微分方程、偏微分方程），数值优化，反问题计算，等等

为什么计算数学

计算科学是 21 世纪确保国家核心竞争能力的战略技术之一。

——计算科学：确保美国竞争力，2005 年总统信息技术咨询委员会报告

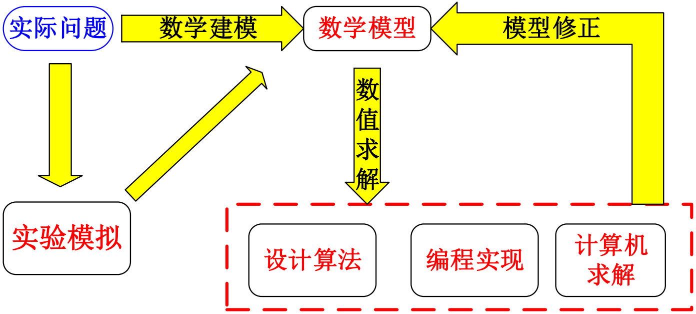
科学计算是 20 世纪重要科学技术进步之一，已与**理论研究**和**实验研究**相并列成为科学研究的第三种方法. 现今科学计算已是体现国家科学技术核心竞争力的重要标志，是国家科学技术创新发展的关键要素

——国家自然科学基金•重大项目指南，2014

数值分析的研究对象

数值分析

借助计算机的高速计算能力，解决现代科学、工程和经济等领域中的各类复杂（数学）问题，是数学与计算机的有机结合



(a) **运用科学计算解决实际问题**

应用举例

例: 非线性方程的求解

怎样计算平方根? 如 $\sqrt{2}, \sqrt{21}, \sqrt{201}, \sqrt{n}$

求解方程: $f(x) = x^2 - 2$

Newton 迭代法: $x_{k+1} = \frac{1}{2}\left(x_k + \frac{2}{x_k}\right)$

得到迭代序列: $x_0 = 1, x_1 = 1.500000, x_2 = 1.416667,$
$x_3 = 1.414216, x_4 = 1.414214$

例: 矩阵特征值/特征向量计算

Google 搜索引擎 1998 年创立，市值超 1.88 万亿美元 (2022/01/12)

$$\boldsymbol{G}\boldsymbol{x} = \lambda \boldsymbol{x}, \quad \boldsymbol{x}^t \boldsymbol{x} = 1$$

$\boldsymbol{G}$: Google Matrix, “the world’ s largest matrix computation”

$\boldsymbol{x}$: PageRank vector, “The 25,000,000,000 Eigenvector” ——**SIAM Review, 2006**

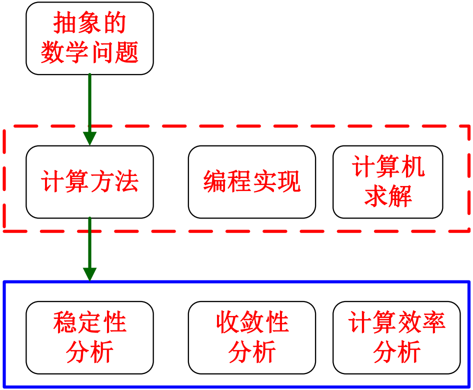
计算数学的主要任务

设计求解各种实际问题的高效可靠的数值方法

- **算法设计**: 构造求解各种数学问题的数值方法
- **算法分析**: 收敛性、稳定性、复杂性、计算精度等
- **算法实现**: 编程实现、软件开发

对于同一个问题，不同的算法在计算性能上可能相差很大!

数值分析研究内容



(b) **数值分析研究内容**

例

求解一个 n 阶线性方程组，若使用克莱姆法则，需要计算 $n+1$ 个 n 阶行列式，在不计加减运算情况下，需要 $n!(n^2-1)$ 次加减乘除运算。而使用高斯消去法，只需约 $2n^3/3$ 次加减乘除运算。

当 $n=20$ 时

$$20! \times (20^2 - 1) \approx 9.7 \times 10^{20}$$

$$2 \times 20^3/3 \approx 5.3 \times 10^3$$

如果用每秒运算 30 亿次（主频 3.0G）的计算机求解，克莱姆法则大约需要 10000 年！但如果使用高斯消去法，不到一秒钟就能完成！

数值分析基本概念

- 解析解、精确解、真解、真值; 数值解、近似解
- 数值分析的特点
  - 求的是近似解，即求出的解是有误差的
  - 与计算机紧密结合，易于上机实现
- 算法的评价
  - 时间复杂度（计算机运行所需的时间）
  - 空间复杂度（所占用的计算机存储空间）
  - 逻辑复杂度（影响程序开发的周期以及后续维护的难易程度）

好的数值算法

好的数值算法

- 有可靠的理论分析，即收敛性、稳定性等有数学理论保证
- 有良好的计算复杂性 (时间和空间)
- 易于在计算机上实现
- 要有具体的数值试验来证明是行之有效的

时间复杂度（计算机运行所需的时间）；空间复杂度（所占用的计算机存储空间）

课程主要内容

- 非线性方程与方程组的数值解法
- 线性方程组的数值求解：直接法和迭代法
- 插值法与数值逼近
- 数值积分
- 矩阵特征值（**自学，幂法和反幂法**）
- 常微分方程数值解法

预备知识

- 工科数学分析（高等数学）
- 线性代数
- 复变函数
- 计算机编程（推荐 C、C++ 或 OCTAVE）

成绩评定

平时成绩（上机报告 + 课堂表现）20%，期末成绩（笔试）80%

教材

- 吴勃英、孙杰宝主编. 数值分析原理（第二版）. 科学出版社. 2023

数值计算的误差

数值计算的特点之一就是所求得的解是近似解，总是存在一定的误差。

- **模型误差**： 在建立数学模型时，往往是抓住主要因素，忽视很多次要因素，把模型“简单化”，“理想化”，因此，数学模型与实际问题之间总会存在一定的误差。
- **观测误差**： 模型中往往包含各种数据或参量，这些数据一般都是通过测量和实验得到的，也会存在一定的误差。
- **截断误差**：也称**方法误差**，是指对数学模型进行数值求解时产生的误差。
- **舍入误差**： 由于计算机的机器字长有限，做算术运算时存在一定的精度限制，也会产生误差。

在数值计算中，我们总假定数学模型是准确的，因而不考虑模型误差和观测误差，主要研究截断误差和舍入误差对计算结果的影响。

误差的概念

- **绝对误差** 设 x 某一量的精确值，x^* 为其近似值，称

$$e^* = x - x^*$$

为近似数 x^* 的绝对误差。

- **绝对误差限** 如果

$$|x - x^*| \le \varepsilon$$

则称 ε 为近似值 x^* 的绝对误差限。

- **相对误差** 称

$$\delta = \frac{x - x^*}{x}$$

为近似值 x^* 的相对误差。

在实际问题中常取

$$\frac{x - x^*}{x^*}$$

为近似值 x^* 的相对误差。

- **相对误差限** 如果 $\left|\frac{x-x^*}{x}\right| \le \Delta$, 则称 Δ 为近似值 x^* 的相对误差限。

关于误差和误差限的几点说明

- 绝对误差不是误差的绝对值，可能是正的，也可能是负的
- 由于精确值通常是不知道的，因此绝对误差一般也是不可知的
- 在做误差估计时，我们所求的通常是误差限
- 误差限不唯一，越小越好，一般是指所能找到的最小上界
- 近似值的精确程度不能仅仅看绝对误差，更重要的是看相对误差

关于误差和误差限的几点说明

泰勒中值定理

设函数 $f(x)$ 在含有 x_0 的开区间 (a,b) 内具有直到 $(n+1)$ 阶的导数, 则当 $x\in(a,b)$ 时, $f(x)$ 可以表示为 $(x-x_0)$ 的一个 n 次多项式与一个余项 $R_n(x)$ 之和:

$$\begin{aligned} f(x) = & f\left(x_0\right)+f'\left(x_0\right)\left(x-x_0\right)+\frac{f''\left(x_0\right)}{2!}\left(x-x_0\right)^2 \\ & +\cdots+\frac{f^{(n)}\left(x_0\right)}{n!}\left(x-x_0\right)^n+R_n(x), \end{aligned}$$

其中 $R_n(x)=\frac{f^{(n+1)}(\xi)}{(n+1)!}\left(x-x_0\right)^{n+1}$, 这里 ξ 介于 x_0, x.

$$e^x = 1+x+\frac{x^2}{2!}+\cdots+\frac{x^7}{7!}+\frac{e^{\xi}}{8!}, \quad x_0=0$$

$$e^{-1} \approx 1-1+\frac{1}{2!}+\cdots-\frac{1}{7!}, \quad 0<\frac{e^{\xi}}{8!}\leq\frac{1}{8!}$$

有效数字

定义

设准确值 x 的近似值 x^* 可表示为

$$x^* = \pm 0.a_1 a_2 \dots a_n \dots \times 10^m$$

其中 m 是整数，a_i 是 0 到 9 之间的一个数字且 $a_1 \neq 0$, 如果

$$|x - x^*| \leq \frac{1}{2} \times 10^{m-n}$$

则称近似值 x^* 具有 n 位有效数字.

有效数字

例

已知精确值 $\pi = 3.14159265\cdots$, 则近似值 $x_1 = 3.14$ 有 3 位有效数字, 近似值 $x_2 = 3.1416$ 有 5 位有效数字.

从上面的例子中可以看出, 我们在计算有效数字的个数时, 是**从最小的有效数字位开始往前数, 直至第一个非零数字为止**, 如果总共有 n 个数字, 那么我们就称其有 n 位有效数字.

有效数字

定理

若准确值 x 的近似值

$$x^* = \pm 0.a_1 a_2 \ldots a_n \times 10^m$$

的相对误差满足

$$\left| \frac{x - x^*}{x^*} \right| \leq \frac{1}{2(a_1 + 1)} \times 10^{-n+1}$$

则 x^* 至少有 n 位有效数字.

有效数字越多, 相对误差越小。同样, 相对误差越小, 则有效数字越多。

数值计算中应注意的若干问题

- 要使用数值稳定的算法
- 减小误差危害
  - 要避免两个相近的数相减
  - 要避免除数的绝对值远小于被除数的绝对值
    可能会产生溢出，即超出计算机所能表示的数的范围
  - 要防止大数“吃掉”小数的现象
  - 注意简化运算步骤，减少运算次数

数值稳定性

定义：数学问题的适定性

如果数学问题满足

(1) 对任意满足一定条件的输入数据，存在一个解，

(2) 对任意满足一定条件的输入数据，解是唯一的，

(3) 问题的解关于输入数据是连续的，

则称该数学问题是**适定的** (well-posed)，否则就称为**不适定的**(ill-posed).

定义：病态问题

如果输入数据的微小扰动会引起输出数据（即计算结果）的很大变化（误差），则称该数学问题是**病态**的，否则就是**良态**的。

数值稳定性

定义：稳定性

如果误差不增长或能得到有效控制，则称该算法是**稳定**的，否则为**不稳定**的。

要使用数值稳定的算法

例 计算 $I_n = \mathrm{e}^{-1}\int_0^1 x^n \mathrm{e}^x\ \mathrm{d}x (n = 0, 1, \cdots)$ 并估计误差.
解 由分部积分可得计算 I_n 的递推公式

$$\begin{cases} I_n = 1 - nI_{n-1}, & n = 1, 2, \cdots \\ I_0 = \mathrm{e}^{-1}\int_0^1 \mathrm{e}^x\ \mathrm{d}x = 1 - \mathrm{e}^{-1} \end{cases} \tag{1}$$

若计算出 I_0, 代入(1)式, 可逐次求出 $I_1, I_2, \cdots$ 的值. 要算出 I_0 就要先计算 e^{-1}, 若用泰勒多项式展开部分和

$$\mathrm{e}^{-1} \approx 1 + (-1) + \frac{(-1)^2}{2!} + \cdots + \frac{(-1)^k}{k!}$$

并取 $k = 7$, 用 4 位小数计算, 则得 $\mathrm{e}^{-1} \approx 0.3679$, 截断误差

$$R_7 = \left|\mathrm{e}^{-1} - 0.3679\right| \leqslant \frac{1}{8!} < \frac{1}{4} \times 10^{-4}.$$

计算过程中小数点后第 5 位的数字按四舍五入原则舍入.

要使用数值稳定的算法

当初值取为 $I_0 \approx 0.6321 = \widetilde{I}_0$ 时，用(1)式递推的计算公式为

$$\text{(A)} \begin{cases} \widetilde{I}_0 = 0.6321 \\ \widetilde{I}_n = 1 - n\widetilde{I}_{n-1}, \quad n = 1, 2, \cdots \end{cases} \tag{2}$$

用上公式计算得到

$$\widetilde{I}_1 = 0.3679, \widetilde{I}_2 = 0.2642, \widetilde{I}_3 = 0.2074, \widetilde{I}_4 = 0.1704$$
$$\widetilde{I}_5 = 0.1480, \widetilde{I}_6 = 0.1120, \widetilde{I}_7 = 0.2160, \widetilde{I}_8 = -0.7280$$

看到 $\widetilde{I}_8$ 出现负值，这与一切 $I_n > 0$ 相矛盾. 实际上，由积分估值得

$$\frac{\mathrm{e}^{-1}}{n+1} = \mathrm{e}^{-1}\left(\min_{0\leqslant x\leqslant 1} \mathrm{e}^x\right)\int_0^1 x^n \,\mathrm{d}x < I_n < \mathrm{e}^{-1}\left(\max_{0\leqslant x\leq 1} \mathrm{e}^x\right)\int_0^1 x^n \,\mathrm{d}x = \frac{1}{n+1}$$

因此，当 n 较大时，用 $\widetilde{I}_n$ 近似 I_n 显然是不正确的. 这里计算公式与每步计算都是正确的。

要使用数值稳定的算法

是什么原因使计算结果出现错误?

主要就是初值 $\widetilde{I}_0$ 有误差 $E_0 = I_0 - \widetilde{I}_0$, 由此引起以后各步计算的误差 $E_n = I_n - \widetilde{I}_n$ 满足关系

$$E_n = -nE_{n-1}, \quad n = 1, 2, \cdots$$

由此容易推得

$$E_n = (-1)^n n! E_0$$

这说明 $\widetilde{I}_0$ 有误差 E_0, 则 $\widetilde{I}_n$ 就是 E_0 的 $n!$ 倍误差. 例如 , $n = 8$, 若

$$|E_0| = \frac{1}{2} \times 10^{-4},$$

则

$$|E_8| = 8! \times |E_0| > 2.$$

这就说明 $\widetilde{I}_8$ 完全不能近似 I_8 了. 它表明计算公式 (A) 是数值不稳定的.

要使用数值稳定的算法

现在换一种计算方案. 简单估计 $I_8 \approx \frac{1}{2}\left(\frac{1}{9}+\frac{\mathrm{e}^{-1}}{9}\right) = 0.0684 = I_8^*$, 然后将公式(1) 倒过来算, 即由 I_8^* 算出 $I_7^*, I_6^*, \cdots, I_0^*$, 公式为

$$
\text{(B)} \begin{cases} I_8^* = 0.07600 \\ I_{n-1}^* = \frac{1}{n}\left(1 - I_n^*\right), \quad n = 8, \cdots, 1; \end{cases}
$$

由上式计算得 (保留四位有效数字)

$$
\begin{aligned}
I_7^* &= 0.1155, I_6^* = 0.1264, I_5^* = 0.1456, I_4^* = 0.1709 \\
I_3^* &= 0.2073, I_2^* = 0.2642, I_1^* = 0.3679, I_0^* = 0.6321
\end{aligned}
$$

要使用数值稳定的算法

计算结果. 我们发现 I_0^* 与 I_0 的误差不超过 10^{-4}. 记 $E_n^* = I_n - I_n^*$, 则

$$|E_0^*| = \frac{1}{n!}\,|E_n^*|\,,$$

E_0^* 比 E_n^* 缩小了 $n!$ 倍, 因此, 尽管 E_8^* 误差较大, 但由于误差逐步缩小, 故可用 I_n^* 近似 I_n. 反之, 当用方案 (A) 计算时, 尽管初值 $\widetilde{I}_0$ 相当准确，由于误差传播是逐步扩大的, 因而计算结果不可靠。此例说明, 数值不稳定的算法是不能使用的.

减小误差危害: 要避免两个相近的数相减

例 求 $x^2 - 16x + 1 = 0$ 的小正根.

解 $x_1 = 8 + \sqrt{63}, x_2 = 8 - \sqrt{63} \approx 8.00 - 7.94 = 0.06 = x_2^*, x_2^*$ 只有一位有效数字. 若改用

$$x_2 = 8 - \sqrt{63} = \frac{1}{8 + \sqrt{63}} \approx \frac{1}{15.9} \approx 0.0629$$

具有三位有效数字.

通过各种等价公式来计算两个相近的数相减, 是避免有效数字损失的有效手段之一。
下面给出几个常用的等价公式:

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

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

$$1 - \cos(x) = 2\sin^2\frac{x}{2}, \quad |x| \ll 1$$

减小误差危害

避免数量级相差很大的数相除

可能会产生溢出，即超出计算机所能表示的数的范围. 特别需要注意的是，尽量不要用很小的数作为除数，否则为放大分子的误差。

如果两个数相除，一般情况下建议把绝对值小的数作为分子，这在后面的算法中会经常遇到。

避免大数吃小数

如 $(10^9 + 10^{-9} - 10^9)/10^{-9}$，直接计算的话，结果为 0. 另外，在对一组数求和时，建议按照绝对值从小到大求和。

减小误差危害：注意简化运算步骤，减少运算次数

尽量减少运算次数，从而减少误差的积累。
在计算多项式的值时，将多项式改写成

$$
\begin{aligned}
p(x) &= a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \cdots + a_1 x + a_0 \\
&= ((\cdots((a_n x + a_{n-1})\,x + a_{n-2})\,x + \cdots)\,x + a_1)\,x + a_0
\end{aligned}
$$

这种方法就是有名的**秦九韶算法**，五百多年后，英国数学家 Horner (1819) 重新发现了该公式，因此西方也称为**Horner 算法**. 如果直接计算的话，需要 $\frac{n(n+1)}{2}$ 次乘法和 n 次加法. 但如果采用秦九韶方法，只需做 n 次乘法和 n 次加法.