Chapter 1 绪论

张亚楠*

February 24, 2024

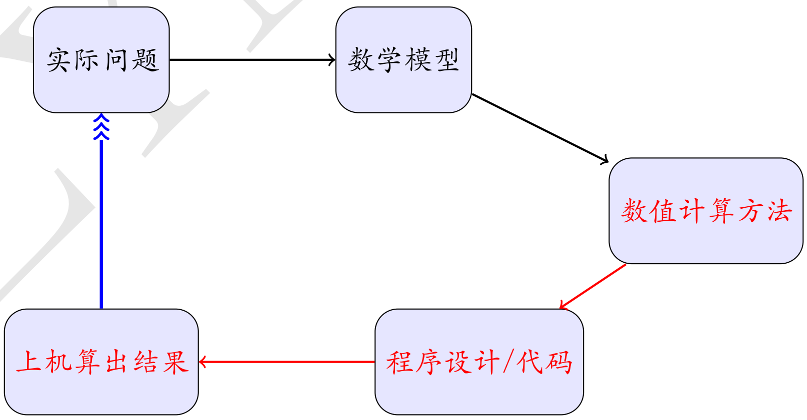
1 数值分析的对象与特点

1.1 什么是数值分析

数值分析[1]是计算数学的一个主要部分，计算数学是数学的一个分支，它研究用 计算机求解各种数学问题的数值计算方法及其理论与软件实现。

> 《普林斯顿数学指南》(该书有中译本，齐民友，科学出版社)一书中，Trefethen 给"数值分析"条目下的定义 Numerical analysis is the study of algorithms for solving the problems of continuous mathematics, by which we mean problems involving real or complex variables.

1.2 科学研究的大致流程



2 数值分析的基本内容

1. 数值逼近

*ynzhang@suda.edu.cn 苏州大学数学科学学院

[1]国内老教材常用定义.

- 插值与逼近
- 数值积分与数值微分

2. 数值代数

   - 线性方程组求解(特征值问题)
   - 非线性方程组数值解法

3. 微分方程数值解

   - ODEs数值解
   - PDEs数值解

IV.22. Set Theory 615

Table 1 Some algorithmic developments in the history of numerical analysis.

Year	Development	Key early figures
263	Gaussian elimination	Liu, Lagrange, Gauss, Jacobi
1671	Newton's method	Newton, Raphson, Simpson
1795	Least-squares fitting	Gauss, Legendre
1814	Gauss quadrature	Gauss, Jacobi, Christoffel, Stieltjes
1855	Adams ODE formulas	Euler, Adams, Bashforth
1895	Runge–Kutta ODE formulas	Runge, Heun, Kutta
1910	Finite differences for PDE	Richardson, Southwell, Courant, von Neumann, Lax
1936	Floating-point arithmetic	Torres y Quevedo, Zuse, Turing
1943	Finite elements for PDE	Courant, Feng, Argyris, Clough
1946	Splines	Schoenberg, de Casteljau, Bezier, de Boor
1947	Monte Carlo simulation	Ulam, von Neumann, Metropolis
1947	Simplex algorithm	Kantorovich, Dantzig
1952	Lanczos and conjugate gradient iterations	Lanczos, Hestenes, Stiefel
1952	Stiff ODE solvers	Curtiss, Hirschfelder, Dahlquist, Gear
1954	Fortran	Backus
1958	Orthogonal linear algebra	Aitken, Givens, Householder, Wilkinson, Golub
1959	Quasi-Newton iterations	Davidon, Fletcher, Powell, Broyden
1961	QR algorithm for eigenvalues	Rutishauser, Kublanovskaya, Francis, Wilkinson
1965	Fast Fourier transform	Gauss, Cooley, Tukey, Sande
1971	Spectral methods for PDE	Chebyshev, Lanczos, Clenshaw, Orszag, Gottlieb
1971	Radial basis functions	Hardy, Askey, Duchon, Micchelli
1973	Multigrid iterations	Fedorenko, Bakhvalov, Brandt, Hackbusch
1976	EISPACK, LINPACK, LAPACK	Moler, Stewart, Smith, Dongarra, Demmel, Bai
1976	Nonsymmetric Krylov iterations	Vinsome, Saad, van der Vorst, Sorensen
1977	Preconditioned matrix iterations	van der Vorst, Meijerink
1977	MATLAB	Moler
1977	IEEE arithmetic	Kahan
1982	Wavelets	Morlet, Grossmann, Meyer, Daubechies
1984	Interior methods in optimization	Fiacco, McCormick, Karmarkar, Megiddo
1987	Fast multipole method	Rokhlin, Greengard
1991	Automatic differentiation	Iri, Bischof, Carle, Griewank

Figure 1: 关于数值分析的一些算法历史

3 数值分析的特点

- 面向计算机: 编程对本课程很重要，各种软件均可；推荐Matlab、Python
- 可靠的理论分析：可解性、收敛性、稳定性

- 计算复杂度: 理解计算效率的重要性，提高效率就是提高生产力
- 数值试验: 检验算法的理论结果或者用于模拟科学问题

例 1 比较以下两个计算公式

$$\log 2 = 1 - \frac{1}{2} + \frac{1}{3} + \ldots + (-1)^{n-1}\frac{1}{n} + \ldots$$

$$\log 2 = 2\Big(\frac{1}{3} + \frac{1}{3}\cdot\frac{1}{3^3} + \frac{1}{5}\cdot\frac{1}{3^5} + \ldots + \frac{1}{7}\cdot\frac{1}{3^7} + \ldots\Big)$$

若截取前20项作为log2的近似值，直观感受下，第二个公式所得结果更精确!

> 冯康：同一问题的不同算法等值但往往不等效！！

冯康是中国计算数学的奠基人之一. 冯康科学计算奖是华人计算数学家的最高奖项，得奖者：舒其望、袁亚湘、许进超、候一钊、鄂维南、张平文、陈志明、汤涛、包维柱、任维清。。。

3.1 如何学好数值分析

- 掌握基本原理，方法，误差分析
- 注重作业，练习
- 上机实践

参考资料:

1. 李庆扬《数值分析》，清华大学出版社(老、旧)；主要用于作业、考试
2. 数值分析讲义（电子稿会发给大家，也会持续更新）
3. Matlab 官网文档， Cleve Moler, “Numerical Computing with MATLAB”
4. Trefethen, Chebfun—numerical computing with functions

B站 教学视频:

1. 张亚楠: 搜“zynsuda” // 适合零基础或者未入门者
2. 舒其望：搜“数值分析 中科大 ” // 适合学过一遍或者未学过但基础好的
3. Trefethen：搜“Trefethen” // 适合英文好，不注重数学原理但重实用性的

Leading journals in Numerical Analysis / Scientific Computing

ACM Transactions on Mathematical Software
Acta Numerica
Foundations of Computational Mathematics
IMA Journal of Numerical Analysis
Journal of Computational and Applied Mathematics
Journal of Computational Physics
Mathematics of Computation
Numerische Mathematik
SIAM Journal on Matrix Analysis and Applications
SIAM Journal on Numerical Analysis
SIAM Journal on Scientific Computing
SIAM Review

Some books to be aware of

E. Anderson, et al., LAPACK Users' Guide, 3rd ed., SIAM, 1999
Z. Bai, et al., Templates for the Solution of Algebraic Eigenvalue Problems, SIAM, 2000, online at http://www.cs.utk.edu/~dongarra/etemplates/.
R. Barrett, et al., Templates for the Solution of Linear Systems, SIAM, 1994, online at http://www.netlib.org/linalg/html_templates/Templates.html
Å. Björck, Numerical Methods in Matrix Computations, SIAM, 1996 (especially for lest-squares problems)
T. A. Davis, Direct Methods for Sparse Linear Systems, SIAM, 2006
T. A. Driscoll, Learning Matlab, SIAM, 2009
G. H. Golub and C. Van Loan, Matrix Computations, 4rd ed., Johns Hopkins U. Press, 2012
A. Greenbaum, Iterative Methods for Solving Linear Systems, SIAM, 1997
D. J. Higham and N. J. Higham, MATLAB Guide, 2nd ed., SIAM, 2005
MathWorks, Inc., Matlab documentation site, http://www.mathworks.com/access/helpdesk/help/helpdesk.shtml
J. Nocedal and S. J. Wright, Numerical Optimization, 2nd ed., Springer, 2006
M. Overton, Numerical Computing with IEEE Floating Point Arithmetic, SIAM, 2000
Y. Saad, Numerical Methods for Large Eigenvalue Problems, 2nd ed, SIAM, 2003, online at http://www-users.cs.umn.edu/~saad/books.html
Y. Saad, Numerical Methods for Sparse Linear Systems, online at http://www-users.cs.umn.edu/~saad/books.html
L. N. Trefethen and D. Bau, III, Numerical Linear Algebra, SIAM, 1997
H. van der Vorst, Lecture Notes on Iterative Methods and Computational Methods for Large Eigenvalue Problems, in Handbook of Numerical Analysis, Elsevier, 2002.
D. S. Watkins, Fundamentals of Matrix Computations, 3rd ed., Wiley, 2010.
D. S. Watkins, The Matrix Eigenvalue Problem, SIAM, 2007.

4 数值计算的误差

4.1 误差来源

- 模型误差
- 观测误差
- 截断误差: 精确公式用近似公式代替所产生的误差; 例如: Tayler 级数
- 舍入误差： 在数值计算中只能对有限位字长的数值进行计算；利用有限位数字代替精确数产生误差； 例如：MATLAB 命令窗口输入：pi sin(pi)

例 2 用Matlab计算 $1-3*(4/3-1)$, 并观察计算结果. ！！ 不等于零！！

标注 1 计算机普遍采用浮点数（floating-point number 不是实数）计算；有兴趣的同学可查阅浮点数的制定标准 和运算规则; 本课程只提醒读者注意两点: (1) 误差不可避免; （2）尽可能避免误差的过分传播（关注舍入误差）；

记所有浮点数集合为F, 则F是有理数集Q的有限真子集，即 $F\subset Q$.

- 浮点数是有限位的(双精度64位)， 符号位置，小数位，指数位; 无限小数都不是浮点数
- 有限位的小数有些也不是浮点数：例如: 0.1不是浮点数
- 浮点数有最大，最小值，matlab命令：realmax, realmin,
- 浮点数之间有间隙，以零为中心，远离零的浮点数之间的间隔越大.

浮点数对应实数轴的一个有限区间[-realmax, realmax]上零星分布的一些点!!

定义 1 绝对误差，简称误差：

$$e^\star = x^\star - x$$

其中$x^\star$是准确值x的近似值. $|e^\star|$ 的任一个上界ε称为误差限。 相对误差：

$$e_r^\star = \frac{e^\star}{x};\quad e_r^\star = \frac{e^\star}{x^\star}$$

相对误差限：$\varepsilon_r = |e_r^\star|$的一个上界。

例如： $x = 10\pm 1, y = 1000\pm 5$

定义 2 近似值$x^\star$的误差限是某一位数字的半个单位，从该位开始到$x^\star$的 第一位非零数字共有n位，则称：$x^\star$具有n位有效数字.

例如： pi = 3.141592653589793 取五位有效数字?

4.2 数值运算的误差估计

四则运算：设x_1, x_2 为准确值，$x_1^\star, x_2^\star$为近似值，则 误差限：

$$
\begin{aligned}
&\varepsilon(x_1^\star \pm x_2^\star) = \varepsilon(x_1^\star) + \varepsilon(x_2^\star)\\
&\varepsilon(x_1^\star * x_2^\star) = |x_1^\star| * \varepsilon(x_2^\star) + |x_2^\star| * \varepsilon(x_1^\star)\\
&\varepsilon(x_1^\star / x_2^\star) = \frac{|x_1^\star| * \varepsilon(x_2^\star) + |x_2^\star| * \varepsilon(x_1^\star)}{|x_2^\star|^2}
\end{aligned}
$$

一元函数的误差限：

$$
f(x) - f(x^\star) = f^{'}(x^\star)(x - x^\star) + f^{''}(x^\star)(x - x^\star)^2 + o(x - x^\star)^2
$$

则：$\varepsilon(f(x^\star) \approx |f^{'}(x^\star)| * \varepsilon(x^\star)$.

思考： 多元函数的误差限如何刻画?

5 误差的定性分析，避免误差危害

5.1 算法的数值稳定性

考虑初始数据的误差在计算中的传播

例 3 计算

$$
I_n = \int_0^1 x^n e^x dx, \ n = 0, 1, 2, 3 \cdots, 20
$$

分部积分得到递推公式

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

取 e = 2.718282, 计算结果如表1： 从表中数据可知计算结果错误， 如何修正算法？ 表2给出了倒推公式

$$
I_{n-1} = (e - I_n)/n
$$

的计算结果； 结果是稳定的、高精度的. 为什么倒推公式精度高，更稳定?

定义 3 一个算法若输入数据有误差，而在计算过程中舍入误差不增长，则称此算法是数值稳定的，否则是不稳定的。

Table 1: 数值不稳定算法

I_0	1.718282	I_6	0.3446627	I_{12}	-14.35115
I_1	1	I_7	0.3056431	I_{13}	189.2833
I_2	0.7182817	I_8	0.2731371	I_{14}	-2647.248
I_3	0.5634365	I_9	0.2600479	I_{15}	39711.43
I_4	0.4645357	I_{10}	0.1178026	I_{16}	-635380.2
I_5	0.3956032	I_{11}	1.422453	I_{17}	1.080147e+07

Table 2: 数值稳定算法

I_0	1.718282	I_6	0.3446845	I_{12}	0.1950999
I_1	1	I_7	0.30549	I_{13}	0.1819828
I_2	0.7182818	I_8	0.2743615	I_{14}	0.1705232
I_3	0.5634363	I_9	0.249028	I_{15}	0.1604341
I_4	0.4645364	I_{10}	0.2280015	I_{16}	0.1513354
I_5	0.3955995	I_{11}	0.2102652	I_{17}	0.1455796

例 4 (课后作业) 计算积分

$$I_n = \int_0^1 \frac{x^n}{x+10} dx, \quad n = 0, 1, 2, \cdots, 10$$

仿照上例，给出数值稳定的递推公式。

5.2 病态问题和条件数

例 5 考虑

$$\begin{cases} x + \alpha y = r_1 \\ \alpha x + y = r_2 \end{cases}$$

当 $\alpha \approx 1$ 时，且右端项输入有误差时，会对解造成的影响。

5.3 算法优劣的标准

- 截断误差要小，收敛速度快。
- 舍入误差在计算过程中能得到控制。
- 算法实现：易于编程和上机实现。

5.4 减少运算误差原则

- 避免大数吃掉小数：相近的数相减；大数除以小数；。。。
- 简化运算步骤，减少运算次数：高次幂乘法，秦九韶算法

标注 2 误差不可避免！本课程以介绍基本算法为主，舍入误差不必过于关心！ 数值分析主要关心算法的稳定性和收敛速度！即使计算机消除了舍入误差（这不可能）， 数值分析的中9成内容依然会存在并发挥作用，因为算法的时效性一直是衡量算法优劣的重要标准！

6 几个算法例子

6.1 秦九韶算法

秦九韶（1208年- 1268年），字道古，汉族，鲁郡（今河南范县）人。 精研星象、音律、算术、诗词、弓剑、营造之学，历任琼州知府、司农丞，后遭贬，卒于梅州任所。

秦九韶其父秦季栖，进士出身，官至上部郎中、秘书少监。秦九韶聪敏勤学。宋绍定四年（1231），秦九韶考中进士，先后担任县尉、通判、参议官、州守、同农、寺丞等职。先后在湖北、安徽、江苏、浙江等地做官，1261年左右被贬至梅州，不久死于任所。他在政务之余，对数学进行潜心钻研， 并广泛搜集历学、数学、星象、音律、营造等资料，进行分析、研究。 宋淳祐四至七年（1244至1247），他在为母亲守孝时，把长期积累的数学知识和研究所得加以编辑，写成了闻名的巨著《数学九章》，并创造了“大衍求一术”。被称为“中国剩余定理”。他所论的“正负开方术”，被称为“秦九韶程序”。世界各国从小学、中学到大学的数学课程，几乎都接触到他的定理、定律和解题原则。

$$p(x) = \sum_{j=0}^{n} = a_j x^j = a_0 + a_1 x + \ldots + a_n x^n$$

直接计算需要 $\boldsymbol{O}(n^2)$次乘法；而采用如下等价形式计算

$$p(x) = (\ldots(a_n x + a_{n-1})x + a_{n-2})x + \ldots + a_1)x + a_0$$

递推公式计算$p(x^*)$

$$\begin{cases} b_0 = a_n \\ b_j = b_{j-1}x^* + a_{n-j}, \quad j = 1, \ldots, n \end{cases}$$

Matlab 用于计算多项式的函数plolyval, 采用的即是秦九韶算法(外文常称为Horner's method, 但秦九韶和波斯人更早就发现了这一计算技巧). 减少运算次数对数值计算极其重要；还比如有经典算法FFT.

6.2 迭代法与开方

计算方法需要上机实现，如果算法有递推公式，则代码简单，适合编程。

例 6 给定$a>0$, 构造求$\sqrt{a}$的迭代公式。

问题等价于求解

$$x^2 - a = 0$$

给个猜测初值x_0, 记误差$\Delta x = x - x_0$

$$(x_0+\Delta x)^2 - a = 0 \Rightarrow x_0^2 + 2\Delta x x_0 + \Delta x^2 - a = 0 \Rightarrow x_0^2 + 2\Delta x x_0 \approx a$$

$$\Delta x = \frac{a - x_0^2}{2x_0}, \quad x_1 = x_0 + \Delta x = \frac{1}{2}(x_0 + \frac{a}{x_0})$$

递推公式计算$\sqrt{a}$

$$\begin{cases} x_0 = \text{ initial guess} \\ x_j = \frac{1}{2}(x_{j-1} + \frac{a}{x_{j-1}}), \quad j = 1, \ldots, n \end{cases}$$

6.3 刘徽割圆术

刘徽（约225年—约295年），汉族，山东滨州邹平市人。他的杰作《九章算术注》和《海岛算经》，是中国最宝贵的数学遗产。刘徽思想敏捷，方法灵活，既提倡推理又主张直观。他是中国最早明确主张用逻辑推理的方式来论证数学命题的人。

阿基米德计算了96边形的周长以近似圆周长；利用递推公式求出正多边形的边长，进而计算圆的周长近似，编程时用一行代码的循环即可。 刘徽利用正多边形的面积近似圆的面积得到更为精确的圆周率3.1416. 后人猜测刘徽、祖冲之用到了外推技巧(存疑！)，如果属实， 说明古今中外的聪明人想法很相似。

小结

1）浮点数是有理数集在某个区间内的少许部分；误差不可避免

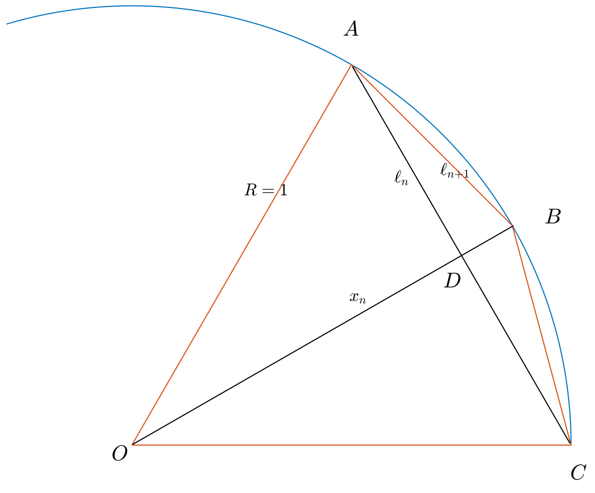


Figure 2: 割圆术：计算正多边形的边长ℓ_{n+1}与 ℓ_n之间的关系得到圆周长的迭代序列

2) 实现同样的功能，算法和代码越简单越好

3) 关注算法的计算复杂度、稳定性

4) 递推公式和迭代法: 循环代码（迭代法Matlab编程需要循环）

5) Matlab编程应当多用数组和矩阵计算，尽量避免不必要的循环

上机练习

1. 给定正实数a, 编写程序计算$\sqrt{a}$ 并与matlab开方函数"sqrt"比较计算结果。

2. 利用刘徽割圆术思想，编写计算圆周率π的程序，计算正$6*2^7$边形时的结果，并与pi比较精度。

3. 对96边形和192边形计算的近似圆周率做加权平均; 取松弛因子$\omega=1/3$, 即

$$\pi \approx S_{192} + \omega * (S_{192} - S_{96})$$

再次比较精度。

Tips: 注意程序中只允许出现四则运算和开方运算(第一题已做).