

第一章 课程引论

殷东生

dyin@math.tsinghua.edu.cn

清华大学数学科学系

2018年秋季学期

课程基本介绍

- 课程名称：高等数值分析
- 参考书：数值分析基础（第二版） 关治、陆金甫著，高等教育出版社，
Iterative Methods for Sparse Linear Systmes, Youself Saad, second edition.
- 任课教师：殷东生，理科楼A408，Tel: 62772871，
dyin@math.tsinghua.edu.cn
- 助教：张旭，1910866793@qq.com
余业锋，yuyf13@tsinghua.org.cn
- 考核方式：平时作业：15%，程序及大作业：15%，期中考试：35%，期末考试：35%。
- 课程所需准备知识：微积分，高等代数和计算机编程。

20世纪40年代，电子计算机的发明为计算成为第三种科学研究手段提供了可能。半个多世纪以来，随着计算机技术的飞速发展，计算作为科学研究方法的地位不断上升。现在试验、理论分析和计算“三足鼎立”，已成为科学活动的主要方式。

计算科学是通过计算的手段来解决实际问题的一门科学，其处理问题的过程主要有如下三个环节：

- 建立数学模型；
- 设计计算方法—编制程序—用计算机计算—得到数值结果；
- 将计算结果与理论分析和实际得到的数据结合给出实际问题的答案，或提出对模型的修正。

上述第二个环节的核心是算法的分析和设计，也是这门课的主要内容。

各系开设的计算课程

- 1 工物系：电磁场数值计算；反应堆物理与数值计算；计算机模拟物理。
- 2 水利系：计算流体力学；弹性力学与有限元，计算河流及河口海岸动力学，渗流力学与计算分析。
- 3 化学系：计算化学导论；计算化学实验。
- 4 航天航空学院：计算流体力学；计算固体力学；有限元法基础；热物理数值计算。
- 5 热能系：数值传热学（英）；弹性力学与有限元。
- 6 电机系：高电压工程与数值计算。物理系：计算物理。
- 7 土木系：弹性力学及有限元基础；计算结构力学概论。
- 8 交叉信息院：计算生物学；计算理论。
- 9 生医系：系统与计算神经科学。材料学院：计算材料学（II）。
- 10 机械系：机械工程数值计算；计算机分子模拟：基础及应用；有限元分析（基础）

高等数值分析的基本内容

本课程主要讲述以下内容

- ① 线性方程组的直接法;
- ② 求解非线性方程组的不动点迭代法;
- ③ 求解线性方程组的经典迭代法;
- ④ 求解大型稀疏方程组的Krylov子空间方法;
- ⑤ 矩阵特征值问题;
- ⑥ 函数的插值、数值积分与微分;
- ⑦ 函数的最佳平方逼近、有理逼近与最小二乘法, SVD。

数值线性代数的基本问题

数值线性代数主要包括如下三大矩阵计算问题

- ① 求解线性代数方程组，即给定 $A \in \mathbb{R}^{n \times n}$, $\det A \neq 0$ 和 $\mathbf{b} \in \mathbb{R}^n$ ，求 $\mathbf{x} \in \mathbb{R}^n$ ，使得

$$A\mathbf{x} = \mathbf{b};$$

- ② 线性最小二乘问题，即给定 $A \in \mathbb{R}^{m \times n}$ 和 $\mathbf{b} \in \mathbb{R}^m$ ，求 $\mathbf{x} \in \mathbb{R}^n$ 使得

$$\|A\mathbf{x} - \mathbf{b}\|_2 = \min\{\|A\mathbf{y} - \mathbf{b}\|_2 : \mathbf{y} \in \mathbb{R}^n\};$$

- ③ 矩阵特征值问题，即给定一个 $A \in \mathbb{R}^{n \times n}$ ，求它的部分或全部特征值以及对应的特征向量。

$$A\mathbf{x} = \lambda\mathbf{x}.$$

数值代数的重要性

GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems

Y Saad, MH Schultz - SIAM Journal on scientific and statistical computing, 1986 - SIAM

We present an iterative method for solving linear systems, which has the property of minimizing at every step the norm of the residual vector over a Krylov subspace. The algorithm is derived from the Arnoldi process for constructing an l_2 -orthogonal basis of ...

被引用次数: 9694 相关文章 所有 42 个版本 引用 保存

[图书] Iterative methods for sparse linear systems

Y Saad - 2003 - books.google.com

Tremendous progress has been made in the scientific and engineering disciplines regarding the use of iterative methods for linear systems. The size and complexity of linear and nonlinear systems arising in typical applications has grown, meaning that using direct ...

被引用次数: 11955 相关文章 所有 83 个版本 引用 保存

[图书] Matrix computations

GH Golub, CF Van Loan - 2012 - books.google.com

The fourth edition of Gene H. Golub and Charles F. Van Loan's classic is an essential reference for computational scientists and engineers in addition to researchers in the numerical linear algebra community. Anyone whose work requires the solution to a matrix ...

被引用次数: 55381 相关文章 所有 20 个版本 引用 保存

Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems

HA Van der Vorst - SIAM Journal on scientific and Statistical Computing, 1992 - SIAM

Recently the Conjugate Gradients-Squared (CG-S) method has been proposed as an attractive variant of the Bi-Conjugate Gradients (Bi-CG) method. However, it has been observed that CG-S may lead to a rather irregular convergence behaviour, so that in some ...

被引用次数: 4850 相关文章 所有 7 个版本 引用 保存

[图书] The algebraic eigenvalue problem

JH Wilkinson, JH Wilkinson - 1965 - tocs.ulb.tu-darmstadt.de

Necessary and sufficient conditions for positive definiteness 28 Differential equations with constant coefficients 30 Solutions corresponding to non-linear elementary divisors 31 Differential equations of higher order 32 Second-order equations of special form 34 ...

被引用次数: 9936 相关文章 所有 7 个版本 引用 保存 更多

研究数值算法的必要性

线性代数方程组的求解

$$Ax = b,$$

其中 $A = (a_{ij})_{n \times n}$ 是 n 阶方阵,

$$b = (b_1, b_2, \dots, b_n)^T, x = (x_1, x_2, \dots, x_n)^T \in \mathbb{R}^n.$$

已知矩阵 A , 向量 b , 求解未知量 x 。若行列式

$$d = \det A = |A| \neq 0$$

则方程有唯一解:

$$x = A^{-1}b, \quad A^{-1} \text{ 为 } A \text{ 的逆矩阵}$$

其中逆矩阵 A^{-1} 可以通过 Laplace 展开求得。

而行列式可以如下计算

$$d = \sum_{j_1 j_2 \cdots j_n} \tau(j_1 j_2 \cdots j_n) a_{1j_1} a_{2j_2} \cdots a_{nj_n},$$

这里 $j_1 \cdots j_n$ 是 $1, 2, \cdots, n$ 的一个排列, 而

$$\tau(j_1 j_2 \cdots j_n) = \begin{cases} 1, & j_1 \cdots j_n \text{ 偶排列} \\ -1, & j_1 \cdots j_n \text{ 奇排列} \end{cases}$$

d 中共有 $n!$ 项求和, 每次要做 $n-1$ 次乘法运算, 因而计算 d 需要 $n!(n-1)$ 次乘法和 $n!-1$ 次加法, 总计算量

$$\times \div \quad \mathcal{O}((n+1)!)$$

$$+- \quad \mathcal{O}((n+1)!)$$

Cramer法则

定理

令 $A = (\mathbf{a}_1 | \cdots | \mathbf{a}_n)$ 是列向量为 $\mathbf{a}_i \in \mathbb{R}^n$ 的非奇异矩阵, 则 $A\mathbf{x} = \mathbf{b}$ 的解为

$$x_i = \frac{\det(\mathbf{a}_1 | \cdots | \mathbf{a}_{i-1} | \mathbf{b} | \mathbf{a}_{i+1} | \cdots | \mathbf{a}_n)}{\det A}, \quad 1 \leq i \leq n.$$

计算量

当 $n = 25$ 时大约需要 10^{26} 次运算。

世界最快的计算机美国“Sum-mit”，每秒钟运算速度为 1.87×10^{17} 次，求解一个25阶的代数方程组需要 10^9 秒。

$$10^9 \text{秒} \approx 32 \text{年}$$

这说明不能通过求逆矩阵在计算机上求解线性代数方程组。

而在实际问题中方程组的规模大都为

$$n = 10^6 \sim 10^8$$

必须有数值上有效的算法来求解这些方程组。

在线性代数中，用Gauss 消去法求解线性方程组，其计算量为 $O(n^3)$ 。

九章算术中的线性方程组

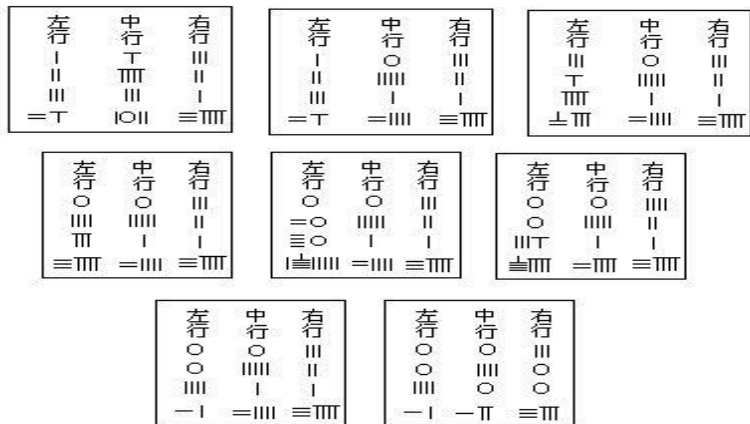


图 1-29

Hilbert矩阵

求解线性方程组

$$\begin{cases} x_1 + \frac{1}{2}x_2 + \frac{1}{3}x_3 = \frac{11}{6} \\ \frac{1}{2}x_1 + \frac{1}{3}x_2 + \frac{1}{4}x_3 = \frac{13}{12} \\ \frac{1}{3}x_1 + \frac{1}{4}x_2 + \frac{1}{5}x_3 = \frac{47}{60} \end{cases}$$

精确解:

$$x_1 = x_2 = x_3 = 1$$

取三位十进制有效数字在计算机用消去法计算

$$x_1 = 1.089, x_2 = 0.488, x_3 = 0.491$$

条件数

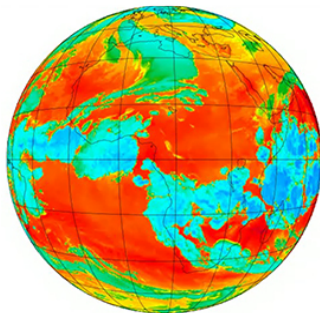
$$\text{cond}(A) = \|A\| \|A^{-1}\| = 748.$$

微分方程与人类文明

是微分方程引领着人类文明进程：

- Newton运动方程
- Laplace 方程
- Maxwell电磁学方程
- Fourier热传导方程
- Einstein广义相对论方程
- Yang-Mills规范场方程
- 量子力学的Schrödinger方程
- Navier-Stokes方程
- 弹塑性力学方程
- Black-Scholes方程（期权定价理论）

神威·太湖之光连获戈登·贝尔奖-2016

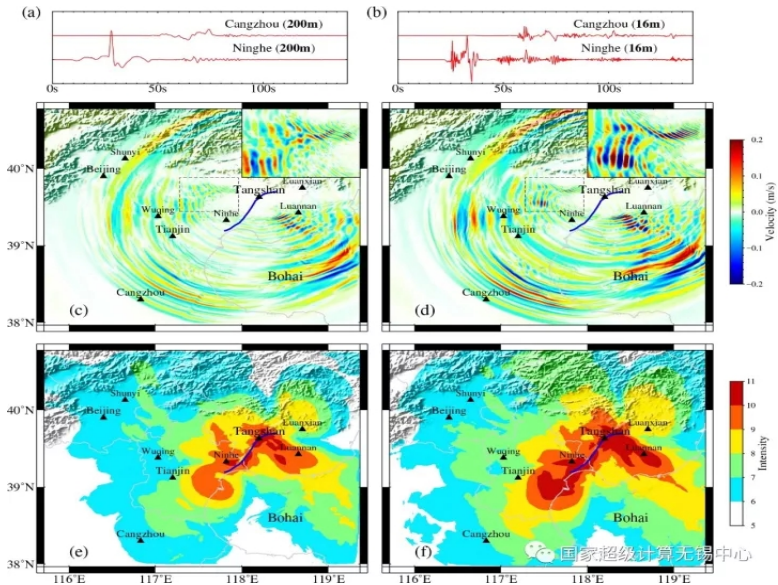


74 100 126 152 178 204 230 256 282 308 334

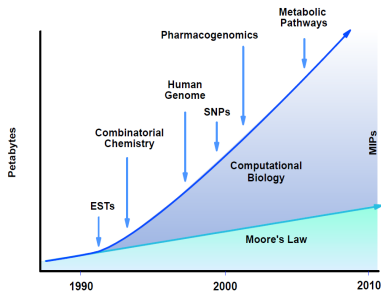
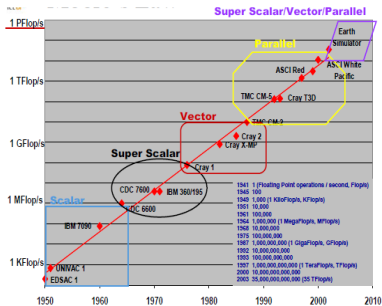


0.001 0.008 0.064 0.512 4.096 32.768

神威·太湖之光连获戈登·贝尔奖-2017



计算机硬件发展和实际计算需求的差距

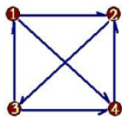


- ① 大规模复杂问题的建模：非稳态，多物理，多尺度问题；
- ② 处理不确定和随机效应；
- ③ 数据同化；
- ④ 大数据和从数据到知识；
- ⑤ 大型的优化、评估和设计问题。

Google的网页排序

■ 如何度量网页本身的重要性呢？

- 互联网上的每一篇html文档除了包含文本、图片、视频等信息外，还包含了大量的链接关系，利用这些链接关系，能够发现某些重要的网页



网页是节点，网页间的链接关系是边

- 直观地看，某网页A链向网页B，则可以认为网页A觉得网页B有链接价值，是比较重要的网页。
- 某网页被指向的次数越多，则它的重要性越高；越是重要的网页，所链接的网页的重要性也越高。

Google的网页排序

■ 如何度量网页本身的重要性呢？

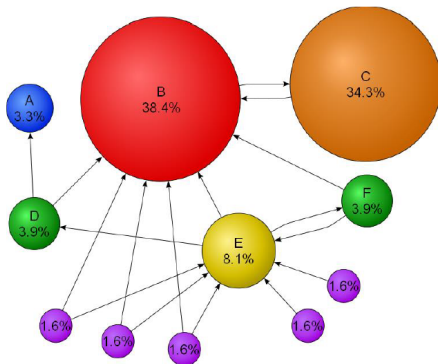
- 比如，新华网体育在其首页中对新浪体育做了链接，人民网体育同样在其首页中对新浪体育做了链接



- 可见，新浪体育被链接的次数较多；同时，人民网体育和新华网体育也都是比较“重要”的网页，因此新浪体育也应该是比较“重要”的网页。

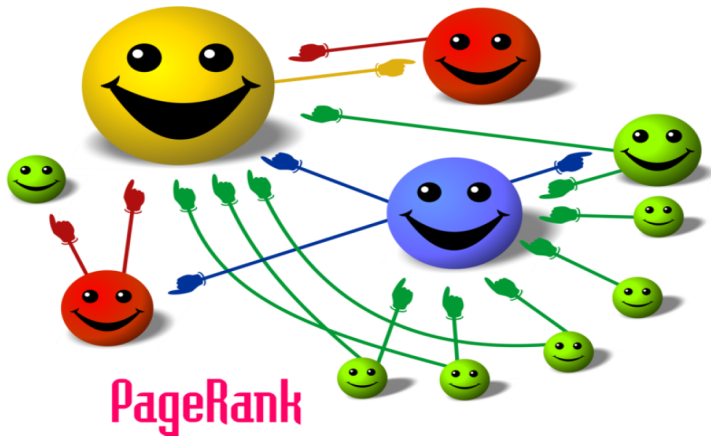
Google的网页排序

■ 一个更加形象的图



链向网页E的链接远远大于链向网页C的链接，但是网页C的重要性却大于网页E。这是因为因为网页C被网页B所链接，而网页B有很高的重要性。

Pagerank 算法原理:



PageRank的简单计算

- 一个由只有4个页面组成的集合：A，B，C和D，若所有页面都链向A，那么A的PR（PageRank）值将是B，C及D的和

$$PR(A) = PR(B) + PR(C) + PR(D).$$

- 若B也有链接到C，且D也有链接到包括A的3个页面。一个页面不能投票2次，所以B给每个页面半票。同样，D投出的票只有三分之一算到了A上。

$$PR(A) = \frac{PR(B)}{2} + \frac{PR(C)}{1} + \frac{PR(D)}{3}.$$

- 换言之，根据链出总数平分一个页面的PR值。

$$PR(A) = \frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)}.$$

一般情形，网页 u 的PR值可表示为

$$PR(u) = \sum_{v \in B_u} \frac{PR(v)}{L(v)},$$

亦即 u 的PR值依赖于所有链接到 u 的网页集合 B_u 。

实际的网络链接环境没有这么理想化，PageRank有两个问题

- Rank leak, 一个独立的网页如果没有外出的链接就会产生等级泄露；
- Rank sink, 整个网页中紧密链接成环的网页没有外出的链接；

解决方案：随机浏览模型。假定用户从一个随机的网页开始浏览，不断点击当前网页的链接开始下一次浏览，直到其厌倦，开始了随机浏览。

随机上网者用以上方式访问一个新网页的概率就是该网页的PageRank值。这种方式更接近与用户浏览网页的行为，且可以保证PageRank的具有唯一值。

简言之，其计算原理如下：

- 互联网是一个有向图；
- 每一个网页是图的一个顶点；
- 网页间的每一个链接是图的一个有向边；
- 用邻接矩阵来表示图，定义邻接矩阵为 G ，若网页 j 到 i 有超链接，则 $g_{ij} = 1$ ，反之 $g_{ij} = 0$ ；
- N 个网页的邻接矩阵为 $N \times N$ 的0,1 方阵；
- 记矩阵 G 的行、列和分别是

$$c_j = \sum_i g_{ij} \quad \text{链出链接数；}$$

$$r_i = \sum_j g_{ij} \quad \text{链入链接数}$$

- 假定在上网浏览网页并选择下一个网页的过程与过去浏览的网页无关，而仅依赖于当前的网页。则这一选择过程可视为一个有限状态、离散时间的随机过程，其状态转移概率用Markov链描述。
- 定义转移概率矩阵 $A = (a_{ij})$:

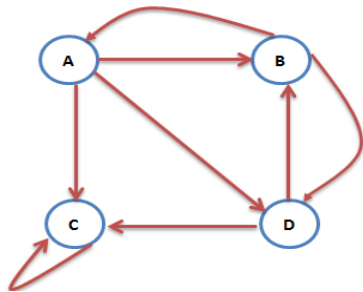
$$a_{ij} = \frac{g_{ij}}{c_j}, \quad i, j = 1, \dots, n$$

- 正则Markov链存在平稳分布 α 满足

$$A\alpha = \alpha, \quad \sum_{i=1}^n x_i = 1$$

- α 表示在极限状态（转移次数趋于无穷）下各网页被访问的概率分布； α 定义为网页的PageRank向量， x_i 表示第 i 个网页的PR值。

转移矩阵



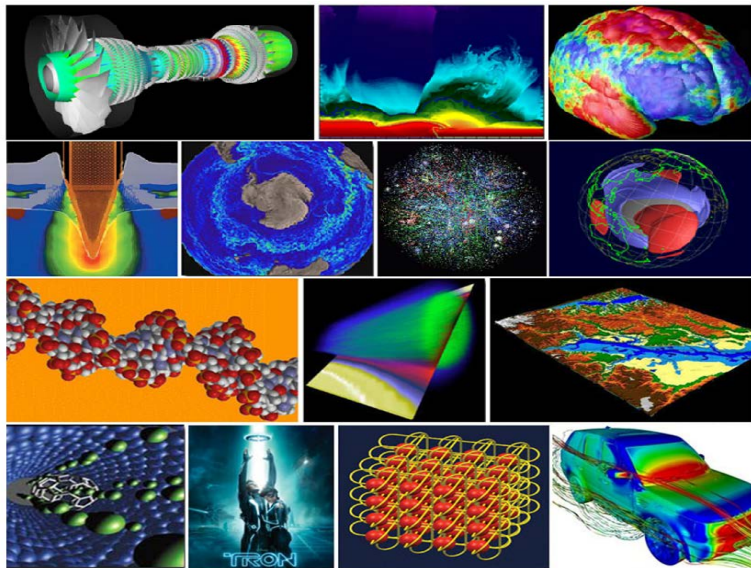
$$S = \begin{pmatrix} 0 & 1/2 & 0 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 1 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{pmatrix}$$

20世纪十大算法

Computing in Science and Engineering 杂志评出了10大算法：

- 1 Metropolis算法 (Metropolis Algorithm) ;
- 2 单纯型法 (Simplex method) ;
- 3 Krylov 子空间迭代法(Krylov subspace iteration methods)
- 4 矩阵分解技术 (Matrix decomposition approach) ;
- 5 Fortran 编译器 (Fortran Compiler) ;
- 6 QR 算法 (QR algorithm) ;
- 7 快速排序算法 (Quicksort algorithm) ;
- 8 快速Fourier 变换 (FFT) ;
- 9 整数关系检测 (Integer relation detector) ;
- 10 快速多极子算法 (Fast multipole method) .

数值模拟



算法的重要性

许多科学与工程计算问题都有如下特点：

- ① 高维数、多尺度、非线性、不适定、大时间、奇异性、复杂区域、高度病态；
- ② 计算规模大；
- ③ 精度高。

人类的计算能力主要依赖两个方面：

- ① 计算机的性能，
- ② 计算方法的效率。

计算方法的发展对于计算能力的提高和计算机的进步是同等重要的。

好的算法

设计好的算法尤为重要。公认的好的算法的标准：

- 运算次数少；
- 运算过程具有规律性，便于编制程序；
- 要记录的中间结果少；
- 能够控制误差的传播和积累，以保证精度。

误差来源

通过数值计算求出的解通常都是近似解，原问题的准确解和近似解之间的偏差就是误差，误差的来源有如下四个方面：

- ❶ **模型误差** 从实际问题得到数学模型时往往忽略了许多次要因素，因而即使数学模型能够求出精确解，也与实际问题的真解不同，它们之间的偏差就是模型误差。
- ❷ **观测误差** 由观测、试验等方法得到的数值必然带有误差，这种误差称作观测误差。
- ❸ **截断误差** 理论上的精确值往往要用无限次的运算才能达到，而实际计算只能用有限次运算的结果来近似，由此产生的误差叫做截断误差。
- ❹ **舍入误差** 计算机只能对有限数字进行运算，每个超出计算机字长的数据都要经过取舍或截断方法处理，这样引起的误差即为舍入误差。

误差的例子

想知道地球的表面积 A 是多少，近似的把地球看成球体：

$$A = 4\pi r^2, \quad r \text{ 是地球半径}$$

- ① 即使这一公式可以精确计算，所得值也和地球表面积的真值有一定的误差，这就是模型误差。
- ② 而在实际计算中如果取 $r = 6370\text{km}$ ，这一观测值具有一定的误差，这就是观测误差。
- ③ 公式中的 π 是圆周率，它是一个无理数，必须取它的有限位，例如取 $\pi = 3.141$ ，此处就产生了截断误差。
- ④ 最后将 $4 \times 3.141 \times 6370^2$ 的计算结果进行四舍五入得 $A = 5.0981 \times 10^8$ ，这一步就产生了舍入误差。

定义 (绝对误差和相对误差)

设 x 是精确值, x_A 是一个近似值,

$$e(x) = |x - x_A| \quad \text{称为 } x_A \text{ 的绝对误差,}$$

而

$$e_r(x) = \frac{|x - x_A|}{|x|} \quad \text{称为 } x_A \text{ 的相对误差。}$$

因为 x 通常是未知的, 一般只能给出 $|x - x_A|$ 的上界 ϵ_A , 即

$$|x - x_A| \leq \epsilon_A, \quad \text{称 } \epsilon_A \text{ 是 } x_A \text{ 的绝对误差界。}$$

类似的

$$\epsilon_r = \frac{\epsilon_A}{|x|} \quad \text{是 } x_A \text{ 的相对误差界。}$$

误差的例子

绝对误差和相对误差那个更能刻画误差的本质？

- ① $x = 0.3000 \times 10, x_A = 0.3100 \times 10,$
 $e(x) = 0.1, e_r(x) = 0.3333 \times 10^{-1}.$
- ② $x = 0.3000 \times 10^{-3}, x_A = 0.3100 \times 10^{-3},$
 $e(x) = 0.1 \times 10^{-4}, e_r(x) = 0.3333 \times 10^{-1}.$
- ③ $x = 0.3000 \times 10^4, x_A = 0.3100 \times 10^4,$
 $e(x) = 0.1 \times 10^3, e_r(x) = 0.3333 \times 10^{-1}.$

相对误差更能刻画出误差的本质。

浮点数系统

在计算机上, 实数系 \mathbb{R} 是用浮点数系统(floating point) \mathbb{F} 来近似的。实数 x 在 β 进制浮点数系统下的表示为:

$$x = (-1)^s \cdot (0.a_1a_2 \cdots a_t) \cdot \beta^e = (-1)^s \cdot m \cdot \beta^{e-t}$$

其中

- s : 符号位, 1 为负数, 0 为正数;
- $m = a_1a_2 \cdots a_t, 0 \leq a_i \leq \beta - 1$; 尾数 (有效数字部分), t 称为字长;
- $e, L \leq e \leq U$: 指数。

浮点数系统一般可表示为

$$\mathbb{F} = \{x \in \mathbb{R} : x = (-1)^s \beta^e \sum_{i=1}^t a_i \beta^{-i}\} \cup \{0\},$$

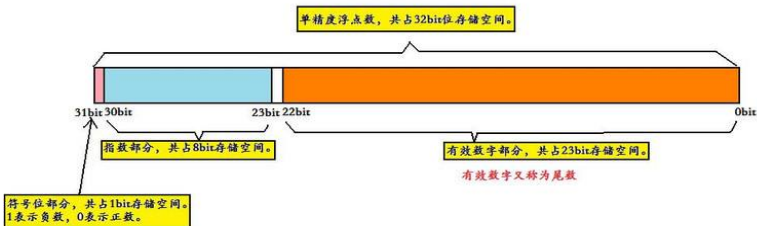
常用的浮点数系统

系统	进制 β	t	L	U
IEEE 单精度	2	23	-126	127
IEEE 双精度	2	52	-1022	1023
Cray	2	48	-16383	16384
HP calculator	10	12	-499	499
IBM mainframe	16	6	-64	63

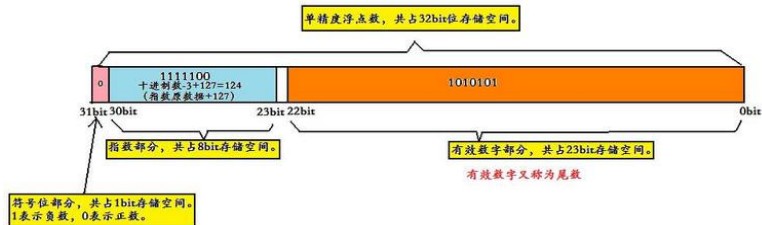
如2进制的浮点数系统 $\mathbb{F}(2, t, L, U)$ 定义为：

$$\mathbb{F} = \{\pm 0.d_1 d_2 \cdots d_t \times 2^k\} \cup \{0\},$$

其中 $d_1 = 1, d_j (2 \leq j \leq t)$ 为0 或 1。



十进制数10.625转为二进制数1010.101b
 十进制数10.625的科学计数法表示为 10625×10^{-3}
 二进制数1010.101b的科学计数法表示为 1010101×2^{-3}



二进制表示

$\forall x \in \mathbb{R}$ 总可以写成

$$\begin{aligned}x &= \pm 0.1d_2 \cdots d_t d_{t+1} \cdots \times 2^k \\&= \pm (1 \times 2^{-1} + d_2 2^{-2} + \cdots + d_t 2^{-t} + \cdots) 2^k\end{aligned}$$

Q: 请写出81.625的二进制表达式?

记 x 在一个浮点数系统 \mathbb{F} 中的表示为 $fl(x)$, 一般说来, $fl(x)$ 只是 x 的某种近似。常用的近似方法有两种

- 截断 (chop), $x = \pm 0.1d_2 \cdots d_t \times 2^k$;
- 舍入 (rounding), $x = \pm 0.1d_2 \cdots \delta_t \times 2^k$, $\delta_t = d_t$, if $d_{t+1} = 0$,
 $\delta_t = d_t + 1$, if $d_{t+1} = 1$ (0舍1入)。

IEEE 采用的是舍入, 则

$$\begin{aligned}\left| \frac{x - fl(x)}{x} \right| &= \left| \frac{0.1d_2 \cdots d_t d_{t+1} \cdots \times 2^k - 0.1d_2 \cdots d_t \times 2^k}{0.d_1 d_2 \cdots \times 2^k} \right| \\ &= \left| \frac{0.d_{t+1} d_{t+2} \cdots}{0.d_1 d_2 \cdots} \right| \times 2^{-t} \leq \frac{2^{k-t-1}}{2^{k-1}} = 2^{-t}\end{aligned}$$

这就是实数的浮点数表示产生的误差, 称为**机器精度**, 记为 ϵ_{mach} 。

① 单精度: 符号1位, 阶8位, $t = 23$ 。

$$\epsilon_{mach} = 2^{-23} \approx 10^{-7}$$

② 双精度: 符号1位, 阶11位, $t = 52$ 。

$$\epsilon_{mach} = 2^{-52} \approx 10^{-16}$$

因此在浮点数系统中有

$$fl(x) = x(1 + \epsilon_{mach})$$

浮点数系统 \mathbb{F} 具有下列值得注意的特点:

- 它是实数系 \mathbb{R} 的一个只包含 $\beta^t(U - L + 1) + 1$ 个数的有限集, 这些数对称地离散分布在区间 $[UFL, OFL]$ 和 $[-OFL, -UFL]$ 中, 这里 UFL 是下溢限(underflow limit), OFL 是上溢限(overflow limit)。在2进制下,

$$UFL = 2^L \times 0.1, \quad OFL = 0.11 \cdots 1 \times 2^U$$

- 具有最小正数 $UFL = 2^L \times 0.1$;
- 具有最大正数 $OFL = 0.11 \cdots 1 \times \beta^U$;
- 具有机器精度 $\epsilon_{mach} = \beta^{-t}$;
- 其中的四则运算并不满足实数系中的运算规律。

在一个浮点数系统中, 绝对值超过 OFL 的实数值就被认为是无穷大, 绝对值小于 UFL 的实数就被认为是零。

Matlab默认为双精度:

- `>> Xmax=realmax`
`Xmax =1.7977e+308`
- `>> Emin=realmin`
`Emin =2.2251e-308`
- `>> eps`
`ans =2.2204e-16`
- `>> log2(Xmax)`
`ans =1024`
- `>> log2(Emin)`
`ans =-1022`
- `>> log2(eps)`
`ans =-52`

定义 (有效数字)

设 x_A 是 x 的一个近似值,

$$x_A = \pm 0.a_1a_2 \cdots a_i \cdots \times 10^k,$$

其中 k 为整数, $a_i \in \{0, 1, 2, \cdots, 9\}, a_1 \neq 0$ 。如果有

$$|x - x_A| \leq \frac{1}{2} \times 10^{k-n},$$

则称 x_A 具有 n 位有效数字: a_1, a_2, \cdots, a_n 。

Q:

$$\pi_A = 3.1416 = 0.31416 \times 10$$

是 π 的近似值, 具有几位有效数字?

有效数字与相对误差

定理

设 x 的近似值 $x_A = \pm 0.a_1a_2 \cdots a_i \cdots \times 10^k$, 则

① 如果 x_A 有 n 位有效数字, 则

$$\frac{|x - x_A|}{|x_A|} \leq \frac{1}{2a_1} \times 10^{1-n}.$$

② 如果

$$\frac{|x - x_A|}{|x_A|} \leq \frac{1}{2(a_1 + 1)} \times 10^{1-n},$$

则 x_A 至少有 n 位有效数字。

计算机实行对位及有限字长的“大数吃小数”。

- 那个式子更适合在计算机上计算

$$\sin(x + \epsilon) - \sin x = 2 \cos\left(x + \frac{\epsilon}{2}\right) \sin \frac{\epsilon}{2}$$

- 在字长为8的10进制浮点数系统下算 $(x + y) + z$ 和 $x + (y + z)$, 其中

$$x = 0.23371258 \times 10^{-4},$$

$$y = 0.33678429 \times 10^2,$$

$$z = -0.33677811 \times 10^2,$$

则由浮点数的运算规则

$$(x + y) + z = 0.64100000 \times 10^{-3},$$

$$x + (y + z) = 0.64137126 \times 10^{-3},$$

$$\text{精确值 } x + y + z = 0.641371258 \times 10^{-3}.$$

向前误差分析

在浮点数系统中设 x 的浮点数表示为 x^* , 则

$$\begin{aligned}x^* &= x(1 + \delta), \\(x^*)^2 &= x^2(1 + 2\delta),\end{aligned}$$

每次乘法相对误差加倍!!!

用Gauss 消去计算

$$Ax = b, A \in \mathbb{R}^{n \times n}, x, b \in \mathbb{R}^n$$

若 $n = 10^6$, 则需进行 $O(10^{18})$ 次浮点数运算。若用双精度的浮点数运算, 最后的相对误差

$$10^{-16} \times 10^{18} = 10^2.$$

向后误差分析

设

$$f(x) = x^2, \quad (f(x))^* = x^2(1 + \rho),$$

其中 $|\rho| < \epsilon_{mach}$, 则

$$\exists \tilde{\rho}, |\tilde{\rho}| < |\rho| \text{ s.t. } (1 + \tilde{\rho})^2 = 1 + \rho$$

因此

$$[f(x)]^* = x^2(1 + \tilde{\rho})^2 = f(x(1 + \tilde{\rho})).$$

函数 $f(x)$ 即函数值 \hat{y} , 通常 $\hat{y} \neq f(x)$ 。但可类似证明: 存在 δx 满足

$$\hat{y} = f(x + \delta x), \quad |\delta x| \leq |x| \epsilon_{mach},$$

误差分析的要点

定义 (向后误差分析)

这种把计算结果归结为原始数据经扰动之后的精确的计算结果的误差分析的方法叫做 **向后误差分析法**。

- ① 向前误差分析: $(x^*)^2 = x^2(1 + 2\delta)$, 损失精度;
- ② 向后误差分析: $[f(x)]^* = f(x(1 + \tilde{\rho}))$ 。若 f 的性质很好, 则无需担心浮点数的舍入误差引起精度损失。

计算结果的精度估计:

$$\frac{|\hat{y} - f(x)|}{|f(x)|} = \frac{|f(x + \delta x) - f(x)|}{|f(x)|} \leq c(x) \frac{|\delta x|}{|x|} \leq c(x)\epsilon.$$

条件数

敏度分析是研究 x 有微小变化 δx 时函数值 f 会发生多大的变化。

定义 (条件数)

$$\frac{|f(x + \delta x) - f(x)|}{|f(x)|} \leq \kappa(x) \frac{|\delta x|}{|x|}.$$

称 $\kappa(x)$ 为 f 在 x 点的**条件数**。 $\kappa(x)$ 反映了自变量的微小变化对函数值的影响程度。

定义 (病态问题)

当 $\kappa(x)$ 很大时, 自变量的微小变化就有可能引起函数值的巨大变化, 此时称 f 在 x 点是**病态**的; 反之, 若 $\kappa(x)$ 较小, 称 f 在 x 点是**良态**的。

一个计算问题是否病态是问题本身的固有属性。

设 $f(x)$ 是一个可微函数, x_A 是 x 的近似值, 则

$$\begin{aligned}\kappa(x) &= \lim_{x_A \rightarrow x} \frac{|[f(x) - f(x_A)]/f(x)|}{|(x - x_A)/x|} \\&= \left| \frac{x}{f(x)} \right| \lim_{x_A \rightarrow x} \left| \frac{f(x) - f(x_A)}{x - x_A} \right| \\&= \left| \frac{xf'(x)}{f(x)} \right|.\end{aligned}$$

例

求下列函数的条件数, 并说明其是否病态

❶ $f(x) = \sqrt{x}$

❷ $f(x) = \frac{1}{1-x}$

Wilkinson 多项式

考虑下面的多项式：

$$w(x) = \prod_{i=1}^{20} (x - i) = (x - 1)(x - 2) \cdots (x - 20).$$

展开后 x^{19} 的系数为 -210 。

Wilkinson 发现对此系数作一个小的扰动：

$$-210 \rightarrow -210 - 2^{-23},$$

其中的根变为

$$x = 16, 17 \rightarrow 16.73 \pm 2.81i$$

而系数的扰动：

$$2^{-23} \approx 1.192 \times 10^{-7}$$

数学家简介 (James H. Wilkinson)

詹姆斯·哈迪·威尔金森(1919—1986), 英国皇家学会院士、著名的数值分析专家, 研制ACE计算机的功臣, 在建造由图灵设计的ACE计算机中扮演了举足轻重甚至最关键的角色, 获得了1970年的图灵奖。其主要学术贡献有: 向后误差分析法; 浮点测试等。1976年, 威尔金森积极参与并推动成立了一个非赢利性的名为NAG的公司 (Numerical Algorithms Group Ltd.) 以开发和推广数值分析和统计分析的软件包, 吸引了世界上许多大学和政府研究机构的专家共同合作。NAG已经为68种型号的计算机配备了Fortran库, Ada、Pascal、C的通用数学库也已上市。J. H. Wilkinson奖从1991年开始, 4年颁发一次, 颁给在数值软件领域做出突出贡献的科学家。



图灵奖 (1970年)

冯·诺伊曼奖
(1970年)

Chauvenet奖
(1987年)

矩阵特征值的计算

对于方阵 $A \in \mathbb{C}^{n \times n}$, 若有 $\lambda \in \mathbb{C}$ 和非零向量 $\mathbf{x} \in \mathbb{C}^n$, 使得

$$A\mathbf{x} = \lambda\mathbf{x}$$

则称 \mathbf{x} 为 **特征值**(eigenvalue) λ 对应的 **特征向量**(eigenvector)。

矩阵的特征值亦可通过它的特征多项式 $p(\lambda) = \det(\lambda I - A)$ 的根来定义:

$$p(\lambda) = \lambda^n - c_1\lambda^{n-1} + \cdots + (-1)^n c_n$$

其中

$$c_1 = \text{trace}(A), \quad c_n = \det(A).$$

Q: 上述特征值的定义是否可以用来数值求解特征值?

运算误差分析

设 $y = f(x_1, x_2, \dots, x_n)$, 如果参量 x_1, x_2, \dots, x_n 有误差, 则 y 也会有误差。若 x_1, x_2, \dots, x_n 的近似值为 $\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n$, 相应的解 \tilde{y} , 则 \tilde{y} 的绝对误差为

$$|y - \tilde{y}| = |f(x_1, x_2, \dots, x_n) - f(\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n)|$$

相对误差为

$$\frac{|y - \tilde{y}|}{|y|} = \frac{|y - \tilde{y}|}{|f(x_1, x_2, \dots, x_n)|}.$$

由此得到误差估计

$$\begin{aligned} |y - \tilde{y}| &\leq \sum_{i=1}^N \left| \frac{\partial f}{\partial x_i} \right| |x_i - \tilde{x}_i|, \\ \frac{|y - \tilde{y}|}{|y|} &\leq \sum_{i=1}^N \left| \frac{x_i}{y} \frac{\partial f}{\partial x_i} \right| \frac{|x_i - \tilde{x}_i|}{|x_i|}, \end{aligned}$$

把两个数 a, b 的 $+, -, \times, \div$ 看成二元函数, 则有

$$e(a \pm b) \leq e(a) + e(b), \quad e(ab) \leq |b|e(a) + |a|e(b)$$

$$e_r(a \pm b) \leq \frac{e(a) + e(b)}{|a \pm b|} = \frac{|a|e_r(a) + |b|e_r(b)}{|a \pm b|}$$

$$e_r(ab) \leq \frac{e(a)}{|a|} + \frac{e(b)}{|b|} = e_r(a) + e_r(b),$$

$$e\left(\frac{a}{b}\right) \leq \frac{e(a)}{|b|} + \left|\frac{a}{b^2}\right|e(b), \quad e_r\left(\frac{a}{b}\right) \leq e_r(a) + e_r(b),$$

运算原则:

- 避免相近的数相减, 因为相近数相减会严重丢失有效数字。
- 作乘法运算时, 避免和绝对值大的数相乘。
- 作除法运算时, 避免和绝对值小的除数相除。

避免有效数字的损失

方程 $ax^2 + 2bx + c = 0$ 的两个根的公式为

$$x_1 = \frac{-b + \sqrt{b^2 - ac}}{a}, \quad x_2 = \frac{-b - \sqrt{b^2 - ac}}{a},$$

如果 $b^2 \gg |ac|$, 则 $\sqrt{b^2 - ac} \approx |b|$, 用上面的公式计算 x_1 和 x_2 , 其中的一个将会损失有效数字。如方程 $x^2 - 62.10x + 1 = 0$ 的根分别为

$$x_1 = -0.01610723, \quad x_2 = -62.08390.$$

而 $\sqrt{b^2 - ac} = 62.06$, 则

$$f(x_1) = \frac{-62.10 + 62.06}{2.000} = \frac{-0.04000}{2.000} = -0.02000$$

而 $e_r(x_1) \approx 2.4 \times 10^{-1}$

数学上等价 \neq 计算上等价

很多算法在数学上是等价的，但是在计算机上计算时表现迥异。

例 (方程求根)

上面的二次方程的求根问题，有等价的公式：

$$x_1 = \frac{-b - \operatorname{sgn}(b)\sqrt{b^2 - 4ac}}{2a}, \quad x_2 = \frac{c}{ax_1}.$$

例

$f(x) = \sqrt{1+x^2} - 1$ ，当 $|x| \ll 1$ 时直接计算会损失精度，数学上可等价的写为

$$f(x) = \frac{x^2}{\sqrt{1+x^2} + 1}$$

这个公式可避免有效数字的损失。

- 减少运算次数;
计算

$$f(x) = x^3 - 6.1x^2 + 3.2x + 1.5$$

在 $x = 4.71$ 处的值（保留3位有效数字）， $f(4.17) = -14.263899$ ，
而 $f(f(4.17)) = -13.4$ ，相对误差为

$$\left| \frac{-14.263899 + 13.4}{-14.263899} \right| \approx 0.06$$

而 $f(x)$ 可以写成下面等价的形式

$$f(x) = x^3 - 6.1x^2 + 3.2x + 1.5 = ((x - 6.1)x + 3.2)x + 1.5$$

用上面的公式计算 $f(f(4.17)) = -14.3$ ，相对误差为0.0025，精度大大提高。

- 避免计算过程中出现上溢和下溢的问题。

一般的 n 次多项式

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

有著名的秦九韶算法

$$u_n = a_n,$$

$$u_k = u_{k+1}x + a_k, \quad k = n-1, \cdots, n-2, 1, 0$$

$$p_n(x) = u_0.$$

只需 n 次乘法和 n 次加法。

数学家简介(秦九韶)

秦九韶(1208年-1261年)，南宋官员、数学家，与李冶、杨辉、朱世杰并称宋元数学四大家。1247年完成著作《数书九章》，其中有大衍求一术、三斜求积术和秦九韶算法。

选对公式

利用公式

$$\ln(1+x) = \sum_{n=1}^{\infty} (-1)^{n+1} \frac{x^n}{n}$$

可以计算 $\ln(1+x)$ 的近似值。若求 $\ln 2$ ，则在上式中令 $x=1$ 即可求得近似值。如果精确到 10^{-5} ，则需 $n=10^5$ ！

计算量大，舍入误差累计严重！

若改用级数

$$\ln \frac{1+x}{1-x} = 2 \left(x + \frac{x^3}{3!} + \frac{x^5}{5!} + \cdots + \frac{x^{2n+1}}{(2n+1)!} + \cdots \right)$$

取 $x = \frac{1}{3}$ ，则只需 $n=9$ 即可使截断误差小于 10^{-10} 。

数值稳定性

定义 (数值稳定性)

一个数值方法，如果初始数或计算过程某一步有微小的改变，由此引起的计算结果也只是微小的变化，则称该方法是数值稳定的，否则称为数值不稳定的。

一般说来，如果具有初始误差 $\epsilon_0 > 0$ ，记计算 n 步后的误差为 ϵ_n ，若方法是数值稳定的，则存在与 n 无关的常数 C 使得

$$|\epsilon_n| \leq C\epsilon_0.$$

常见误差传播模式：

- 1 $|\epsilon_n| \approx Cn\epsilon_0$ ，线性型的误差增长；
- 2 $|\epsilon_n| \approx C^n\epsilon_0$ ，指数型的误差增长。

数学家简介 (John von Neumann)

20世纪最重要的数学家之一，在现代计算机、博弈论、核武器和生化武器等领域内的科学全才之一。计算数学的缔造者之一，他首先研究线性代数和算术的数值计算，后来着重研究非线性微分方程的离散化以及稳定问题，并给出误差的估计法，发展了蒙特卡罗方法，被后人称为“计算机之父”和“博弈论之父”。

早期以算子理论、共振论、量子理论、集合论等方面的研究闻名，开创了冯·诺依曼代数。第二次世界大战期间为第一颗原子弹的研制作出了贡献。为研制电子数字计算机提供了基础性的方案。1944年与摩根斯特恩合著《博弈论与经济行为》，是博弈论学科的奠基性著作。晚年，研究自动机理论，著有对人脑和计算机系统进行精确分析的著作《计算机与人脑》。



图：冯·诺依曼

(1903–1957)，著名匈牙利裔美籍数学家，计算机科学家，物理学家和化学家。

数值稳定性示例

例

设 $x_n = \frac{1}{3^n}$, 可用下面的三种方法生成序列

- ① $x_0 = 1, x_n = \frac{1}{3}x_{n-1}, n = 1, 2, \dots;$
- ② $x_0 = 1, x_1 = \frac{1}{3}, x_n = \frac{4}{3}x_{n-1} - \frac{1}{3}x_{n-2}, n = 2, 3, \dots;$
- ③ $x_0 = 1, x_1 = \frac{1}{3}, x_n = \frac{10}{3}x_{n-1} - x_{n-2}, n = 2, 3, \dots。$

若用有限位的小数运算来实现上面的三方法:

- ① $r_0 = 0.99996, r_n = \frac{1}{3}x_{n-1}, n = 1, 2, \dots;$
- ② $p_0 = 1, p_1 = 0.33332, p_n = \frac{4}{3}p_{n-1} - \frac{1}{3}p_{n-2}, n = 2, 3, \dots;$
- ③ $q_0 = 1, q_1 = 0.33332, q_n = \frac{10}{3}q_{n-1} - q_{n-2}, n = 2, 3, \dots。$

试分析其计算结果。

- ① 理论上，为了估算计算的精度和使得数值结果可信，在计算过程中必须有条件数和精度的估计，并且这些估计必须写入计算代码中。程序的用户应该对计算中可能遇到的困难有所预判并且知道如何规避和克服这些困难。这些问题应该成为科学家和工程师使用和开发现代计算机软件的核心课程。
- ② 但是在工业的实践中，这些因素都被忽略了，甚至用某个算法求解问题时可能会出现问题的警示也没有。这种看起来很不负责的行为的原因是什么？原因之一是估计精度和条件数会降低算法的效率，但最主要的原因是用户无法正确理解这些警示。
- ③ 算法软件的开发者引入这些警示信息并不是为了骚扰用户和显摆数学，真正的目的是为了让科学家和工程师了解要问题和算法会遇到什么困难，并且教会他们如何理解和使用这些警示信息。

实践题

在Matlab中输入下列命令，输出并分析计算结果：

```
>> x = [-1:-1:-18]'; x = 10.^x
```

```
>>y = ((1 + x) - 1)./x
```

```
>>y = ((1 + x).^2 - 1 - 2*x)./x.^2
```

```
>>y = ((1+x).^3-1-3*x-3*x.^2)./x.^3
```

常用数学软件介绍

我们要讨论的算法都已经有人用各种计算机编程语言写成程序并做成软件，这类软件叫数学软件。

互联网 (Internet) 是查找和获得数学软件的主要工具。推荐的三个网站

- <http://www.netlib.org>
- <http://gmas.nist.gov>
- <http://sourceforge.net>

其中第一个网站在中科院计算数学与科学工程计算研究所有镜像：netlib.amss.ac.cn

数学软件的分类

数学软件基本上可以分成三种类型：

- 数学软件包：针对某一类数值计算问题的子程序集。
- 数学软件库：面向各类数值计算问题的子程序集。
- 交互式科学计算环境：面向各类数值计算、符号计算和图形显式问题且具有交互式用户界面的科学计算工作平台。

数学软件包

- LAPACK 是求线性代数方程组、线性最小二乘问题和矩阵特征值、奇异值问题的 FORTRAN 77 子程序包。
- ITPACK 是迭代求解大型稀疏线性代数方程组的 FORTRAN 77 子程序包。
- MINPACK 是求解非线性方程组和非线性最小二乘问题的 FORTRAN 77 子程序包。
- ODEPACK 是求解常微分方程的 FORTRAN 77 子程序包。
- ScaLAPACK 是可扩展的并行数值线性代数软件包。
- DIFFPACK 是用 Visual C++ 语言开发的求解偏微分方程的商业软件。 www.diffpack.com.

数学软件库

- IMSL 是由 Visual Numerics 公司开发的商业软件，是用FORTRAN和 C语言编写的数值 计算与统计程序库，一般与UNIX工作站捆绑在一起销售。 www.vni.com
- NGA是由 Numerical Algorithms Group (NGA)公司开发的数值计算和统计的 FORTRAN和 C 程序库。可在各种硬件平台上运行。
www.nag.com
- NR是与数值计算领域的经典工具书 《Numerical Recipes》一起发布的 FORTRAN、C和 C++语言数值计算软件库，可在
www.nr.com上免费下载。
- SLATEC 是由美国能源部发布的大型数值与统计计算 FORTRAN 程序库，可在 www.netlib.org 上免费下载。

- **MATLAB** 是 MathWorks公司开发的商业软件。它以矩阵运算为核心而且拥有数百个内部函数和各种工具箱。其内部函数用于解决基本的数值计算与图形显示问题。而工具箱分为功能性工具箱和学科工具箱。**MATLAB**被广泛的应用于科学研究和解决各种具体问题。其官方主页为 www.mathworks.com
- **MAPLE**是由加拿大 Waterloo大学计算机系开发的商业数学分析软件。与**MATLAB**相比，它的数值计算能力 较弱，但其符号计算功能较强。官方网址 www.maplesoft.com
- **MATHEMATICA**是由美国 Wolfram Research 公司拥有版权的商业数学分析软件。其功能与 **MAPLE**类似，长于符号计算。官方主页：www.mathematica.com.

- SCILAB是由法国国家信息与自动化研究院(INRIA)开发的自由数学软件。它的功能和MATLAB相似，而且和 MATLAB 是兼容的。但是它的学科工具箱不如MATLAB丰富。其最大的优点是源代码完全公开，使用者可以根据需要自行修改利用。官网：www.scilab.org
- GAUSS是由Aptech Systems 公司开发的商业软件。它有很强的统计分析能力，因而在计量经济学中应用 非常广泛。上海和深圳证券交易所每日收盘的综合股价指数以及总交易量的数据都是用GAUSS处理的。 官网：www.aptech.com