数值分析(1)

Numerical Analysis

计算机系 软件所 喻文健

数值计算导论

- 课程简介
- 数值计算概况
- 误差分析基础
- 计算机浮点数系统

课程信息

- 授课教师：喻文健
  - **Web:** *https://numbda.cs.tsinghua.edu.cn*
  - **E-mail: yu-wj@tsinghua.edu.cn**
  - **Tel: 62773440,** 办公室**:** 东主楼**8**区**407**室
  - 助教： 高真懿, gzy22@mails.tsinghua.edu.cn
    黄杰辰, hjc22@mails.tsinghua.edu.cn
    张艺缤, zhangyb20@mails.tsinghua.edu.cn
- 答疑**:** 办公室或线上, 周三下午**2:00-3:00**

腾讯会议#: 852-1370-0268，密码: 666666

教学团队

□喻文健

课程微信群

群聊：24春-数值分析1

该二维码7天内(3月2日前)有效，重新进入将更新

□高真懿

负责计11、12、17

□黄杰辰

负责计13~16

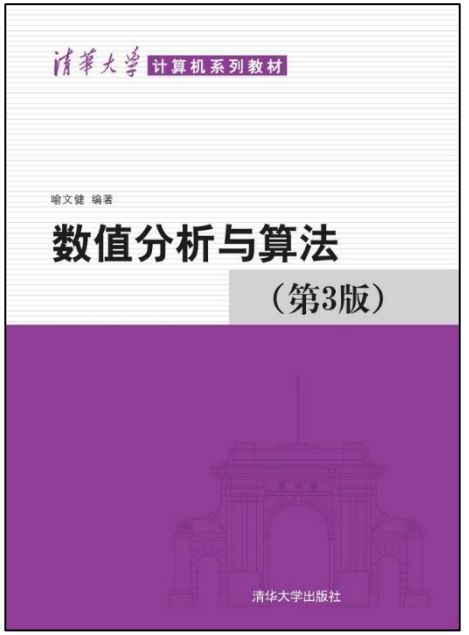
□张艺缤

负责计科2x及其他

课程简介

- 计算方法
- 数值分析与算法
- 科学计算导论（***scientific computing***）
- 数值计算基础（***numerical computing***）
- 课程目标
  - 介绍广泛应用于科学与工程领域的各种数值计算方法
  - 巩固连续数学/线性代数知识、增强实际应用能力

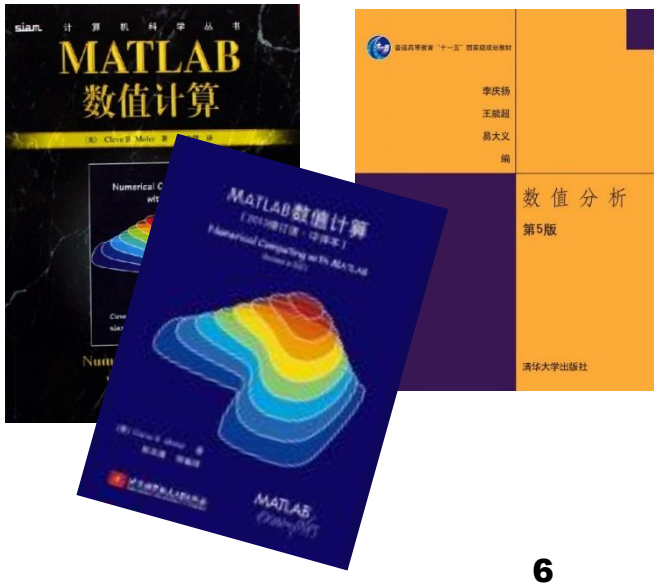
教材



- 数值分析与算法(第3版)
  - 喻文健 编著
  - 清华大学出版社, 2020年

  获2022年清华大学优秀教材特等奖
- 参考书
  - 《Matlab数值计算》, 机械工业出版社 或北航出版社

    https://www.mathworks.com/moler/index_ncm.html
  - 李庆扬 等, 数值分析(第5版), 清华大学出版社, 2009年



教学方式与考核

- 方式
  - 课堂讲授(雨课堂quiz)、作业、编程实验
  - 网络学堂：课后提供课件、作业，其他资料
    (作业、实验报告交到网络学堂上)
- 考评方法
  - 平时：每周作业 + 编程实验(15%+15%)
  - 期中：考试(30%)
  - 期末：考试/Project(40%)
  - 总成绩=max{平时+期中+期末, 平时+期末}

只取10次

再转换为等级制

主要教学内容

- 一.数值计算导论 (4.5课时)
  - 概述、误差分析基础、浮点算术系统与舍入误差
- 二.非线性方程解法 (4课时)
- 三.线性方程组的直接解法 (5.5课时)
- 四.线性方程组的迭代解法 (4课时)
- 五.矩阵特征值计算 (6.5课时)

(期中考试, 第8周)

- 六.函数逼近与函数插值 (7课时)

主要教学内容

- 七.数值积分与微分 (4.5课时)
- 八.常微分方程初值问题 (4课时)
- 习题课 (~ 4次) (2课时)
- 附.Matlab数值计算与应用
  - 补充内容、非考试要求
  - 穿插在各章内容中

误差分析、线性代数相关的数值方法、微积分相关的数值方法

类似数学基础课：公式多、推导多、还有理论证明

又像工程专业课：要上机编程、实验，需要积累经验

学习建议

- 往届情况
  - 去年150→142人, 最终>30%获A-及以上
  - 百分制分数<60为不通过
- 学习建议
  - 认真对待，把握听课、作业、实验三个环节
  - 重点理解问题背景、算法思路和具体步骤(会算)
  - 适当进行公式推导、算法复杂度分析与比较
  - 多用Matlab等软件做编程实验, 提高动手能力!

数值计算的背景与概况

Wenjian Yu

Top ten algorithms of the century

"*We tried to assemble the 10 algorithms with the greatest influence on the development and practice of science and engineering in the 20th century*"

—— **Editors of IEEE *Computing in Science & Engineering*, Jan. 2000 (后被SIAM转载)**

- **1.1946 美国Los Alamos国家实验室的*J. von Neumann*, *S. Ulam*和*N. Metropolis*发展的Metropolis算法（属于Monte Carlo方法；拒绝采样/MCMC, 生成给定概率分布随机点）**
- **2.1947 美国RAND公司的*G. Dantzig*提出的解线性规划的单纯形算法（simplex method）**
- **3.1950 美国UCLA大学与美国国家标准局数值分析所的*M. Hestenes*, *E. Stiefel*和*C. Lanczos*开创的Krylov子空间迭代法(CG算法、Lanczos过程)**
- **4.1950's 矩阵分解方法，由美国Oak Ridge国家实验室的*A. Householder*引入数值线性代数中 (矩阵计算研究掀起革命)**

Top ten algorithms of the century

- **5.1957 美国IBM的*J. Backus*领导开发出的Fortran编译器**
- **6.1959-61 英国Ferranti Ltd.的*John G.F. Francis*发明的QR算法，能稳定地计算矩阵的所有特征值/向量**
- **7.1962 英国Elliot Brothers, Ltd.的*Tony Hoare*提出快速排序算法（Quicksort）**
- **8.1965 美国IBM Watson研究中心的*J. Cooley*与普林斯顿大学及AT&T Bell实验室的*J. Turkey*共同提出了的FFT算法**
- **9.1977 美国Brigham Young大学的*H. Ferguson*和*R. Forcede*提出的整数关系侦察算法(实验数学/简化量子场理论计算)**
- **10.1987 美国Yale大学的*L. Greengard*和*V. Rokhlin*发明的快速多极算法(fast multipole algorithm, 多体物理仿真)**

大多属于/涉及数值计算!

数值分析、科学计算、数值计算

数值计算，也称为科学计算，已成为当今科学研究的三种基本手段之一。它是计算数学、计算机科学和其他工程学科相结合的产物，并随着计算机的普及和各门类科学技术的迅速发展日益受到人们的重视。

科学计算的发展涉及硬件和软件两个方面，这里我们只考虑软件方面，即数值计算的有关算法与程序实现

“数值分析”、“数值计算”是研究求解连续数学问题的算法的学科（而不仅仅局限于计算误差的研究）

对象

核心

数值计算与数值算法

- 数值计算的特点
  - 处理连续数学的量(实数量)，问题中还可能涉及微分、积分和非线性。被求解的问题一般没有解析解、或理论上无法通过有限步四则运算求解
    - 无解析解： $33x^5+3x^4-17x^3+2x^2+4x-39=0$
    - 有解析解，但需无限步计算：**sin(*x*)**
    - 更多的实际应用问题通过数值仿真(simulation)来解决
  - 目标：寻找迅速完成的(迭代)算法，评估结果的准确度
- 好数值算法的特点
  - 计算效率高、计算复杂度低
  - 可靠性好：在考虑实际计算的各种误差情况下，结果尽可能地准确

数值计算的步骤

- 建立数学模型（需要相关学科背景）
- 研究数值计算、求解方程的算法
- 通过计算机语言编程实现算法

} 重点

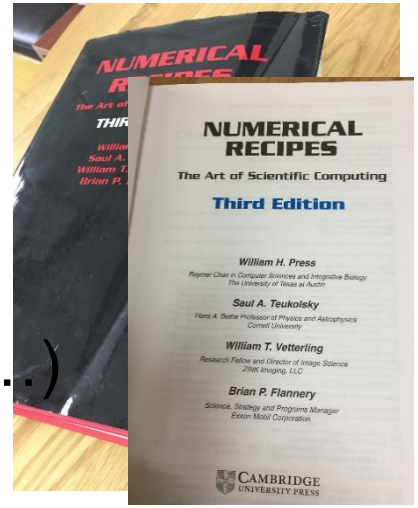
- 在计算机上运行程序进行数值实验、仿真
- 将计算结果用较直观的方式输出，如图形可视化方法
- 解释和验证计算结果，如果需要重复上面的某些步骤

上述各步骤相互间紧密地关联，影响着最终的计算结果和效率（问题的实际背景和要求也左右着方法的选择）

- 设计数值方法(算法)的关键：将问题简化或加以近似(估计带来的误差)，然后求解简化后的问题

数值软件/程序包

- 用途与学习的意义
  - 解决共性数学问题，促进领域的发展及应用
  - 了解、学习算法设计和实现技巧
  - 成为聪明的软件/程序包使用者
- 存在形式和资源
  - 免费(netlib, GitHub, TOMS…)，付费(NR, …)
  - Fortran, C, C++, Matlab, Python, Julia, …
  - 源代码，或API调用，或集成开发环境 www.mathworks.com



https://blogs.mathworks.com/cleve/

MathWorks

- 数值计算相关研究的好帮手：**Matlab软件**
  - 集成环境：交互式计算系统，高级编程语言

	Matlab(作为编程语言)	C, C++, Fortran
	第四代编程语言	第三代编程语言
编译方式	解释器，或JIT加速器	编译器
申明变量？	不需要	需要
开发时间	较快	较慢
运行时间	较慢	较快
开发环境	集成环境(编辑器、调试器、命令历史、变量空间、profiler、编译器)	--

  - 数值计算、矩阵计算功能强大(包含很多先进算法)
  - 大量专题工具箱(**Toolbox**)，为专业应用提供便利
  - 开发环境、可视化功能等方面比较强

>>help svdsketch

MATLAB 2020b的函数手册

References

[1] Yu, Wenjian Yu Gu, and Yaohang Li. "Efficient Randomized Algorithms for the Fixed-Precision Low-Rank Matrix Approximation." *SIAM Journal on Matrix Analysis and Applications* 39, no. 3 (August 2018): 1339–1359. https://doi.org/10.1137/17M1141977.

See Also

数值软件/程序包

2016 Dense Linear Algebra Software Packages Survey (J. Dongarra)

http://www.netlib.org/lapack/lawnspdf/lawn290.pdf

- 矩阵计算有关程序包
  - LAPACK, www.netlib.org/lapack/index.html (LAPACKE) 87%

"as much as possible computation is performed by calls to BLAS" "exploit Level 3 BLAS"

    - Basic linear algebra subprogram (BLAS) www.netlib.org/blas/
    - 厂家提供的BLAS; ATLAS, GotoBLAS, OpenBLAS*
  - ScaLAPACK: Distributed-memory variant 40%
  - MAGMA: Matrix Algebra for GPU & Multicore Arch. 22%

https://icl.cs.utk.edu/magma/

- 更高层的程序库与环境
  - Intel公司Math Kernel Library (in C) **No. 1**
  - Eigen (a C++ template library): eigen.tuxfamily.org **No. 2**
  - Matlab, Octave, Python (Numpy), …

Linux: 95%, Mac: 32%, Win: 22%

(有**20%**被调查者要自己写线性代数子程序)

- 数值计算知识应用广泛 (以计算机相关方向为例)
  - 人工智能、机器人、数据挖掘、多媒体：矩阵计算、奇异值分解、数据拟合(回归) 、常微分方程数值解
  - 计算机图形学CAD：函数插值、逼近、微分方程数值解
  - 电子设计自动化（EDA）：大规模线性方程组求解、常微分方程、偏微分方程、非线性方程、函数逼近
  - 高性能计算（性能评测、算法实现与优化）、电力系统仿真、大气仿真、。。。。。。
- 数值分析的后继课：机器学习、最优化方法、数学建模、图谱理论、凸优化、微分方程数值解法、高等数值算法、统计机器学习、数据挖掘、有限元分析、电磁场计算等

误差分析基础

Wenjian Yu

误差分析基础

- §1.2.1误差的来源
- §1.2.2误差及其分类
  - 误差与有效数字
  - 数据传递误差与计算误差
  - 截断误差与舍入误差
- §1.2.3问题的敏感性与数据传递误差
- §1.2.4算法的稳定性

误差的来源

计算前
- 模型误差　（忽略摩擦、空气阻力）
- 数据误差　常数或测量值、前一步计算的结果

计算中
- 截断误差　方法误差　例：sin(x)= …
- 舍入误差　计算时表示数的位数有限

“四舍五入”

例1.1 用球表面积公式计算地球表面积

$A = 4\pi r^2$

- 将地球近似成球体　模型误差
- 取半径$r \approx 6370km$　数据误差
- 将π的值取到有限位 (如3.14)　数据误差
- 计算$4\pi r^2$ (计算乘法)　舍入误差

误差及其分类

- **1.误差与有效数字**
- **定义1.1** x ~准确值，$\hat{x}$ ~近似值，绝对误差$e(\hat{x})=\hat{x}-x$
  - (绝对)误差往往不能反映准确程度
  - 例: 方法一测量长约1公里的距离，误差1cm; 方法二测量长约1米的物体，误差也是1cm
- **定义1.2** 相对误差$e_r(\hat{x})=\dfrac{\hat{x}-x}{x}$
  - 无论误差、相对误差，都可正可负
  - 当准确值为0时，相对误差无定义
  - 准确值未知, 估计误差绝对值上限, 误差限$\varepsilon(\hat{x})$，$\varepsilon_r(\hat{x})$
  - 误差较小时, $e_r(\hat{x})\approx\dfrac{\hat{x}-x}{\hat{x}}\approx\dfrac{\varepsilon(\hat{x})}{\hat{x}}$

误差及其分类

除非特殊说明，考虑十进制数、遵循四舍五入

- **定义1.3** 一个数的有效数字指: 从左至右第一个非零数字开始的所有数字
  - 前几位有效数字正确与相对误差有何关系？
- **定理1.2** 设对x保留p位有效数字后得到的近似值$\hat{x}$, 则$\hat{x}$的相对误差 $|e_r(\hat{x})| \le \frac{5}{d_0} \times 10^{-p}$, 其中$d_0$为$x$的第一位有效数字.

将第p+1位有效数字做四舍五入

- 证明：设$x = \pm 10^m \times (d_0 + \frac{d_1}{10} + \cdots + \frac{d_{p-1}}{10^{p-1}} + \cdots)$由于在第$p$+1位做四舍五入，$|\hat{x} - x| \le 10^m \times \frac{1}{10^p} \times 5$

而$|x| \ge 10^m \times d_0$ ⇒ $|e_r(\hat{x})| = \frac{|\hat{x}-x|}{|x|} \le \frac{5}{d_0} \times 10^{-p}$

误差及其分类

- **定理1.2** 设对x保留p位有效数字后得到的近似值$\hat{x}$, 则$\hat{x}$的相对误差 $|e_r(\hat{x})| \leq \frac{5}{d_0} \times 10^{-p}$, 其中$d_0$为$x$的第一位有效数字.
- **例1.2** $x = \pi = 3.14159265\cdots$, 保留**3**位有效数字得到 $\hat{x} = 3.14$, $|e(\hat{x})| \leq \frac{1}{2} \times 10^{-2}$, $|e_r(\hat{x})| \leq \frac{1}{2} \times \frac{1}{3} \times 10^{-2} = \frac{5}{3} \times 10^{-3}$
- **注意**：d_0取值1~9, 相对误差≤ 5×10^{-p}

以上根据近似得到的准确有效数字位数判断相对误差限

($\hat{x}$与x的首位数字大多数情况一样)

那反过来呢？

误差及其分类

- **定理1.3** 设x的第一位有效数字为d_0, 若近似值$\hat{x}$的相对误差满足: $|e_r(\hat{x})| \le \frac{5}{d_0+1} \times 10^{-p}$, 则$\hat{x}$的前p位有效数字与x的相同, 或保留p位有效数字后$\hat{x}$和x的结果相等

  □ 证明: 设$x = \pm 10^m \times (d_0 + \frac{d_1}{10} + \cdots + \frac{d_{p-1}}{10^{p-1}} + \frac{d_p}{10^p} + \cdots)$

  $\Rightarrow |x| < 10^m(d_0+1) \Rightarrow |e(\hat{x})| = |x||e_r(\hat{x})| < 10^m \times 5 \times \frac{1}{10^p}$

  $\hat{x} - x$的首位有效数字在d_p所在的数位上, 值$\in$ [-5, 5]

  若两个数的第p位有效数字均为d_{p-1}， 则...

  否则, 两者第p位为相邻数字, 且$\hat{x}$与x保留p位后相同

- **例1.3** $x = 9.1423$, $\hat{x}_1 = 9.1428$, $\hat{x}_2 = 9.1419$. 均满足 $|e_r(\hat{x}_i)| \le \frac{5}{9+1} \times 10^{-4}$. $\hat{x}_1$和$\hat{x}_2$都是前4位“比较准确”

误差及其分类

- **定理1.3** 设x的第一位有效数字为d_0,若近似值$\hat{x}$的相对误差满足: $|e_r(\hat{x})| \leq \frac{5}{d_0+1} \times 10^{-p}$, 则$\hat{x}$的前p位有效数字与x的相同, 或保留p位有效数字后$\hat{x}$和x的结果相等
- 说明：结论的两种情形可粗略地认为都是表示“前p位有效数字正确”
- 考虑 d_0 取值1~9, 若相对误差$\leq \frac{1}{2} \times 10^{-p}$, 则... ... (**定理1.4**)

更粗略一点：

相对误差10^{-p} ~ p位有效数字正确

- 区分两个词
  - 准确度: 反映误差大小
  - 精度: 与表示数的有效数字位数有关 (单/双精度)

3.111111有7位十进制精度, 但它近似π的准确度不高

误差及其分类

- **2.数据传递误差与计算误差**
- 以简单的函数求值问题为例
  - $x \to f(x)$, $\hat{x} \to \hat{f}(\hat{x})$
  - 误差 $\hat{f}(\hat{x}) - f(x) = \underbrace{\left[\hat{f}(\hat{x}) - f(\hat{x})\right]}_{\text{单纯的计算误差}} + \underbrace{\left[f(\hat{x}) - f(x)\right]}_{\text{数据误差传递到结果}}$

    称之为: 数据传递误差
- 说明：这里的数据传递误差与具体的计算方法无关, 分析它时考虑精确计算过程，它仅受问题本身影响

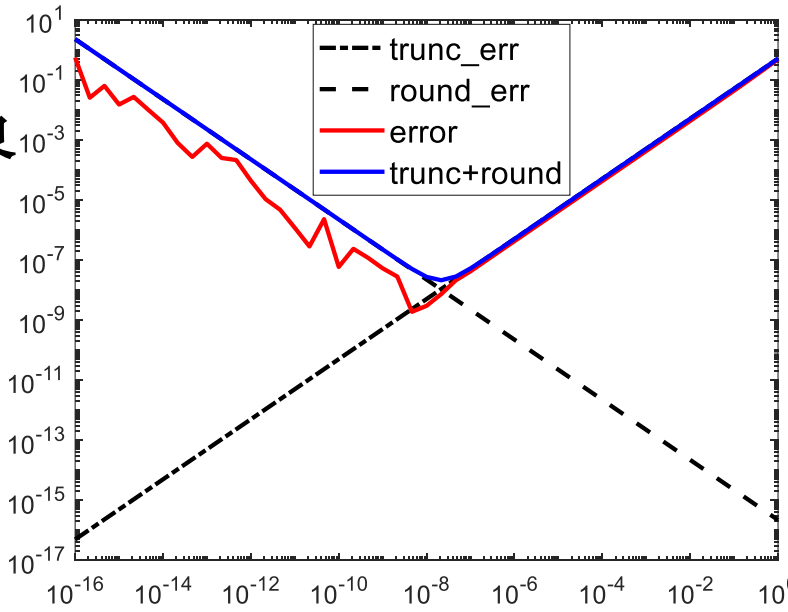
误差及其分类

- **3.截断误差与舍入误差** $\hat{f}(\hat{x}) - f(\hat{x})$
  - 数值方法近似、有限精度运算 (计算误差的两部分)
- **例1.4** 用差商近似一阶导数 $f'(x) \approx \dfrac{f(x+h)-f(x)}{h}$
  - h为步长, 分析两种误差与h关系
  - 截断误差$e_T = hf''(\xi)/2$
  - $\varepsilon_T = Mh/2$, M是$|f''(\xi)|$上界
  - 设计算$f(x)$误差限为ϵ，则

“舍入误差” $\varepsilon_R = 2\epsilon/h \Rightarrow \varepsilon_{tot} = \dfrac{Mh}{2} + \dfrac{2\epsilon}{h}$

  - 实验: 看h取何值使ε_{tot}最小
  - $\epsilon \approx 10^{-16}$, 最佳$h = 2\times10^{-8}$

$f(x)=\sin x,\ x=1$



问题的敏感性 (数据传递误差)

- **定义1.8** 问题的敏感性: 输入数据扰动对问题解的影响程度. 不敏感(良态) vs 敏感(病态)
- **定义1.9** 用**条件数**反映问题的敏感性　　*也叫“相对条件数”*

$$\text{cond} = \frac{\left\| \text{问题的解的相对变化量} \right\|}{\left\| \text{输入数据的相对变化量} \right\|}$$ → 范数

即问题对数据误差的“放大因子”. cond越大问题越病态

- 例如函数求值问题: $x \to f(x)$, $\hat{x} \to f(\hat{x})$
- 结果相对误差 $\frac{f(\hat{x})-f(x)}{f(x)}$, 数据相对变化 $\frac{\hat{x}-x}{x}$

$$\Rightarrow \text{cond} = \left| \frac{[f(\hat{x})-f(x)]/f(x)}{(\hat{x}-x)/x} \right| \approx \left| \frac{xf'(x)}{f(x)} \right|$$ (近似公式)

输入扰动无限小的情况

问题的敏感性 (数据传递误差)

- 类似地, 也可定义绝对条件数
- 例如对函数求值问题，绝对条件数$\mathrm{cond}_A = \left|\frac{f(\hat{x})-f(x)}{\hat{x}-x}\right|$
- 说明: 1.条件数反映问题的特性, 与计算方法无关. 但它会受输入数据(及扰动量)的影响, 因而常考虑其上限

  2.对简单运算, 可直接用微积分知识推数据传递误差限

  例: $\hat{y} = f(\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_n)$, $y = f(x_1,\ x_2, \cdots, x_n)$

多元Taylor展开取线性项, $y-\hat{y} \approx \sum_{i=1}^{n} \frac{\partial f}{\partial x_i}(\hat{x}_1, \cdots, \hat{x}_n)(x_i - \hat{x}_i)$

$$\Rightarrow \varepsilon(\hat{y}) = \sum_{i=1}^{n} \left|\frac{\partial f}{\partial x_i}(\hat{x}_1, \cdots, \hat{x}_n)\right| \varepsilon(\hat{x}_i)$$

应用于+, –, ×, /运算, 见课本(1.5)~(1.7)式, 如

$$\varepsilon(\hat{x}_1\hat{x}_2) = |\hat{x}_2|\varepsilon_1 + |\hat{x}_1|\varepsilon_2$$

算法的稳定性

有限字长的数的四舍五入, 或精度

- 与问题的敏感性相对应的一个概念 (也叫数值稳定性)

舍入误差

- [1]结果对计算过程中的扰动不敏感的算法更稳定
- 例1.7 对长度100的数组求和
- 算法1: 按存储顺序对这100个数直接累加
- 算法2: 先按元素绝对值递增的顺序排序, 再依次累加
- 若数的表示精度很低, 只有2位, 则:
- 算法1, 对于数据为2.0, 0.01, …, 0.01 (99个), 则结果? sum=2.0
- 算法2, 对上述数据, sum=0.99+2.0= 3.0, 更准确!

算法2比算法1更稳定!

误差传递

- [2]对包含一系列计算步的过程, 若中间步结果的(相对)误差不放大或放大不严重, 则该过程对应的算法更稳定

算法的稳定性

- **例1.8** 计算黄金分割比$\phi = \frac{\sqrt{5}-1}{2}$的前n次幂(n= 20)
- 算法1: 直接乘法,$f(x) = x^n$, x=0.618034 (ϕ的近似值)
- 算法2: 利用递推式 $\begin{cases} \phi^{n+1} = \phi^{n-1} - \phi^n \\ \phi^0 = 1,\ \phi^1 = x \end{cases}$ 每步仅做一次减法
- 算法2的效果

n	ϕ^n的计算值
2	0.381966
3	0.236068
…	…
18	0.000144
19	0.000154
20	-0.000010

双精度浮点运算

- 分析误差传播趋势 (算ϕ^n误差为e_n)

$$\begin{cases} e_{n+1} = e_{n-1} - e_n \\ e_0 = 0, e_1 = x - \phi \end{cases}$$

$e_2 = -e_1,\ e_3 = e_1 - e_2 = 2e_1,\ e_4 = -3e_1,$

$e_5 = 5e_1,\quad \ldots,\quad |e_{20}| = c|e_1|, c$是?

错误! $|e_r(\hat{\phi}^n)|$>100%

算法的稳定性

- **例1.8** 计算黄金分割比$\phi = \frac{\sqrt{5}-1}{2}$的前n次幂(n= 20)
- 算法1: 直接乘法,$f(x) = x^n$, x=0.618034 (ϕ的近似值)
- 算法2: 利用递推式 $\begin{cases} \phi^{n+1} = \phi^{n-1} - \phi^n \\ \phi^0 = 1,\ \phi^1 = x \end{cases}$ 每步仅做一次减法

n	ϕ^n的计算值
2	0.381966
3	0.236068
…	…
18	0.000144
19	0.000154
20	-0.000010

$e_{n+1} = e_{n-1} - e_n$

$|e_{20}| = c|e_1|$, c是Fibonacci序列的第20项 $\sim 6.7 \times 10^3$

相对误差的放大更严重!

- 反过来看算法1

由于x<1, $n\nearrow$, 误差$\searrow$ $|e_{20}| \approx 20\phi^{19}|e_1|$ $\sim 10^{-3}$

- 注意: 可忽略计算过程的舍入误差

(也可从条件数角度分析)

算法的稳定性

- 一般单次四则运算的舍入误差很小, 但一个算法含很多步, 从输入量开始“向前”做舍入误差分析很难
- 向后误差分析是分析算法稳定性的另一个思路
  - 以函数求值为例

    $y = f(x)$，计算结果为$\hat{y} = \hat{f}(x)$
  - 求$\hat{x}$使其满足$f(\hat{x}) = \hat{y}$, 则$\Delta x = \hat{x} - x$称为向后误差, 通过考察它的大小来分析算法过程的稳定性
- 对一些问题, 可通过向后误差分析来研究算法的稳定性, 例如对于求解线性方程组的高斯消去法 (第3章会提到)

计算机浮点数系统

Wenjian Yu

计算机浮点数系统与舍入误差

- 浮点数的表示
- 机器精度（$\varepsilon_{\text{mach}}$）
- 抵消现象 (cancellation)

(课本1.3节的主要内容)

计算机中的浮点数

- 实数x在计算机中的表示即浮点数$\mathrm{fl}(x)$ (**2进制**)

$$\mathrm{fl}(x) = \pm\left(d_0 + \frac{d_1}{2} + \frac{d_2}{2^2} + \cdots + \frac{d_{p-1}}{2^{p-1}}\right) \times 2^E$$

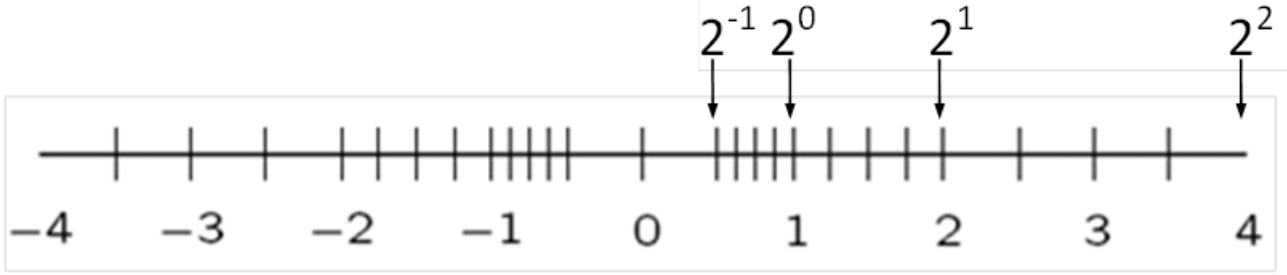
- 基数: β进制 (β=2); 指数E: 上限值U, 下限值L
- p位β进制数, 也称p为精度位数 p位有效数字
- IEEE浮点数系统已成为标准, 分单精度和双精度
- 规范化的规则要求$d_0 = 1$
- 好处：数的表示唯一、尾数都是有效数字、d_0不用存储 (该位表示±的信息)

计算机中的浮点数

IEEE双精度数的表示：

$$[b_1 \underbrace{b_2 b_3 \cdots b_{12}}_{E+1023} b_{13} \cdots b_{64}]_2$$

- 浮点数为有限个，且非均匀地分布在实数轴上



例：一个简单浮点数系统，　p=3, L= -1, U=1　(5个bit)

- 机器精度$\varepsilon_{\text{mach}} = 2^{-p}$　(1与右边相邻数间隔的一半)
- 下溢值：2^L　~2.2×10^{-308}　上溢值：$(2 - 2^{-p+1}) \times 2^U$

	浮点数系统	β	p	L	U	$\varepsilon_{\text{mach}}$
32 bits	IEEE单精度	2	24	-126	127	5.960×10^{-8}
64 bits	IEEE双精度	2	53	-1022	1023	1.110×10^{-16}

计算机中的浮点数

+–*/运算, 以及简单函数的误差与$\varepsilon_{\text{mach}}$同级别(sin, tan, atan, exp)
--C. Moler's blog, 2017.1

- **定理1.5**: 设实数x在浮点数系统中的表示为浮点数$\text{fl}(x)$，则$\left|\frac{\text{fl}(x)-x}{x}\right| \leq \varepsilon_{\text{mach}}$ 应用定理1.2可证明! (2进制)
  (默认四舍五入，或"最近舍入")
- **定理1.6**: $x_1, x_2 \in \mathbb{R}$, 若$\left|\frac{x_2}{x_1}\right| \leq \frac{1}{2}\varepsilon_{\text{mach}}$, 则$x_2$的值对浮点运算$x_1 + x_2$的结果毫无影响 被称为"大数吃掉小数"
- 证明: $|x_2| < 2^E \varepsilon_{\text{mach}}$, 所以$x_1 + x_2 = x_1$.
  (类似定理1.4)

若$\left|\frac{x_2}{x_1}\right| > \varepsilon_{\text{mach}}$，一定不"吃小数"

十进制系统，这两个定理如何?

$$\left|\frac{\text{fl}(x)-x}{x}\right| \leq \frac{\varepsilon_{\text{mach}}}{d_0} = \frac{1}{2d_0}10^{-p+1}$$

若$\left|\frac{x_2}{x_1}\right| \leq \frac{1}{2}10^{-p} = \frac{1}{10}\varepsilon_{\text{mach}}$，一定会"吃小数"，若$>\varepsilon_{\text{mach}}$一定不"吃小数"

抵消现象

- 两个符号相同、值相近的p位数相减使结果的有效数字远少于p位, 称之为抵消(cancellation)
- 例： $x = 1.92305 \times 10^3$， $y = 1.92137 \times 10^3$， 则 $x - y = 1.68$
- 减法计算未发生误差, 但其结果仅有三位有效数字
- 结果的有效数字位数的减少, 意味着相对误差的放大, 往往会影响后续计算的准确度 (操作数有误差!)
- 抵消现象是发生信息丢失、误差变大的信号！

抵消现象

- 一元二次方程求根公式的例子

$$ax^2 + bx + c = 0$$

解为：

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

若$b > 0$，且$|4ac| \ll b^2$，计算x_1时出现抵消现象

如何避免？ (算法的调整)

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \cdot \frac{-b - \sqrt{b^2 - 4ac}}{-b - \sqrt{b^2 - 4ac}} = \frac{2c}{-b - \sqrt{b^2 - 4ac}}$$

计算x_2可能出现的问题也类似地解决

计算结果的准确性

减小舍入误差的几条建议/认识

- 采用双精度浮点数　　(提高精度, 增大p)
- 对包含大量计算的算法，分析舍入误差很难
- 应遵循如下几条建议
  - 避免中间计算结果出现上(下)溢出

    例：计算$\dfrac{x_1}{x_2 \cdot x_3 \cdots x_n}$，　　若$x_2 \ll x_1, \ldots$
  - 避免“大数吃掉小数”(加、减法)

    例：计算 $1+\varepsilon+\varepsilon$,　$\varepsilon \approx 1\times 10^{-16}$　　(调整计算顺序)
  - 避免符号相同的两相近数相减 (抵消现象)
  - 注意简化步骤，减少运算次数

总结

	总误差		
	计算误差		数据传递误差
	截断误差	舍入误差	
如何评估大小?	分析具体的计算方法	向后误差分析; 区间分析法; 一般很难定量分析	问题敏感性(条件数); 直接近似分析
如何减小误差?	选择截断误差小的算法	选稳定的算法; 减小舍入误差的建议; 采用更高精度浮点数	变换问题形式(计算过程); 改善敏感性

(更多例子和讨论，自学课本1.4节)

演示程序与Matlab

- 算法演示程序NumDemo

  放在网络学堂-教学资源

- **Matlab的简单演示**
  - 简单的操作说明
  - 浮点数系统有关的参数
  - 例1.4: 截断误差与舍入误差的实验
- 课程的编程实验
  - 共7次实验，选做4次，交报告并检查验收