

计算方法

Computational Methods

Lecture 1: 误差、条件性与牛顿法

Error, Conditioning, and Newton's Method

刘景钺

南京大学 · 计算机学院

课程信息

计算方法 Fall 2026

授课教师 刘景铎 liu@nju.edu.cn

上课 周四 5-6 节 · 1-18 周 · 仙 I-319

答疑 周四下午 4 点

助教 周宇恒 · 侯哲

作业 nm_nju_2026@163.com

课程主页 TCS Wiki · Fall 2026

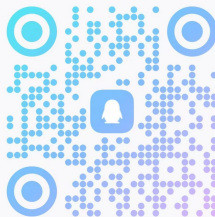
QQ 群 1054896504 加入时注明姓名、专业、
学号

教材: Timothy Sauer, 《数值分析》(原书第 2 版)。 最新公告以
课程主页为准。



计算方法 Fall 2026

群号: 1054896504



扫一扫二维码, 加入群聊



作业与合作

- ▶ 可以讨论：学习小组最多 **3 人**；
- ▶ 必须独立写作：注明合作者、重要帮助和参考资料；
- ▶ 不得公开自己的作业解答；
- ▶ 迟交需要**提前申请**。

Homework 1 9 月 10 日 23:59 前

学号 _ 姓名 _ A1.pdf → nm_nju_2026@163.com

先自己想，再讨论；如果自己几乎没有贡献，这种合作既损失学习机会，也会损害之后的自信。

作业节奏：预计约每两周一次（以主页为准） • **学术不诚信零容忍** • 学术规范参照 ACM 学术抄袭政策

浮点数误差：先猜一猜

```
double x = 0.3 * 3 + 0.1;  
double y = 1.0;  
  
if (x == y) cout << "Equal";  
else cout << "Not equal";
```

会输出什么？

浮点数误差：先猜一猜

```
double x = 0.3 * 3 + 0.1;
double y = 1.0;

if (x == y) cout << "Equal";
else cout << "Not equal";
```

Not equal

(常见 IEEE 754 binary64 实现)

数学公式上的准确，在有限精度计算中未必准确。

常见修补：用 $|x - y| < \varepsilon$ 判断“接近”。但 ε 应该多大？

浮点数：两个典型问题

数值范围：溢出 / 下溢

直接计算

$$\|x\|_2 = \sqrt{\sum_i x_i^2}$$

可能在平方时超出可表示范围。

先令 $m = \max_i |x_i|$ ：

$$\|x\|_2 = m \sqrt{\sum_i (x_i/m)^2}$$

可显著降低风险。

同一个数学量，可以有数值上完全不同的计算方式。

William Kahan • IEEE 754 • 1989 图灵奖

长计算：舍入误差累积

反复求和时：

就近取整 vs. 截断

误差的偏向与累积方式并不相同。

Kahan 补偿求和

Python 3.12 起 `sum()` 也改进了浮点求和精度。

$$\text{softmax}(z)_i = \frac{e^{z_i}}{\sum_j e^{z_j}}.$$

若 $z = (1000, 999, 998)$ ，直接形成 e^{1000} 会溢出。

令 $m = \max_j z_j$ 。整体平移不改变 softmax：

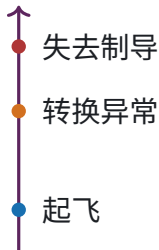
$$\text{softmax}(z)_i = \frac{e^{z_i - m}}{\sum_j e^{z_j - m}}$$

数学上完全等价；数值范围却从 1000 附近变成 $0, -1, -2$ 。

同样的平移也给出稳定的 log-sum-exp 计算。

37

秒



1996 年 Ariane 501 首飞:

64 位浮点数转为 16 位有符号整数时超出可表示范围, 触发转换异常, 惯性参考系统失效。

在 Ariane 4 上没出错, 不代表这个假设是正确的。

数值范围、异常处理、输入域和测试覆盖面, 都是计算正确性的一部分。

另一个经典案例: Patriot 防空导弹

来源: Ariane 501 Inquiry Board, 1996.

```
uint32_t bytes = width * height * 4;  
void *buf = malloc(bytes);
```

如果乘法超出表示范围而**回绕**：

缓冲区过小 → 越界访问.

CWE-190

Integer Overflow or
Wraparound

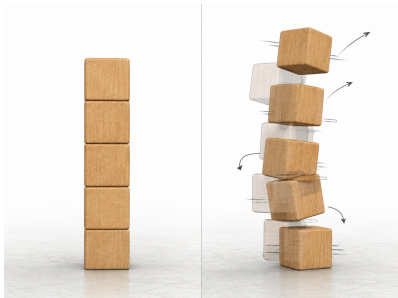
真实案例

CVE-2023-32434

Operation Triangulation

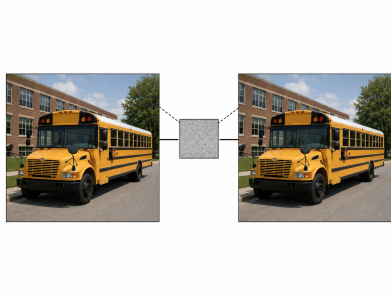
Apple XNU 内核整数溢出漏洞

实时物理 / 机器人仿真



每一步都只有很小的数值误差，
为什么长时间以后会**抖动、穿透甚至
崩坏**？

模型鲁棒性



输入几乎没有变化，
为什么模型输出可能**完全改变**？

共同问题：**小的扰动，经过计算以后会变成多大的变化？**

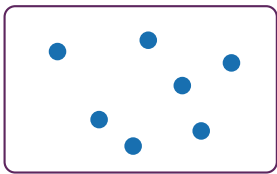
计算方法

计算方法并不仅仅是一些“工程小技巧”。

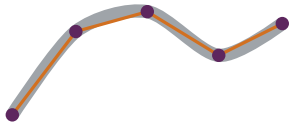
作为一门学科，它反复研究：

- ▶ **收敛性**：随着计算资源或近似层次增加，结果是否逼近目标；
- ▶ **复杂性**；
- ▶ **条件性（鲁棒性）**：问题本身对扰动有多敏感；
- ▶ **压缩性**；
- ▶ **正交性**；
- ▶ ...

误差与偏差，也不仅仅来源于浮点数或离散化处理。



理想气体



光滑表面 $\rightarrow$ 离散近似

“All models are wrong, but some are useful.”

—George E. P. Box

输入扰动如何传播到输出？

无论扰动来自测量、建模还是计算，先看最基本的一维问题：

输入从 x 变成 $x + \Delta x$ ：

$$f(x + \Delta x) - f(x) \approx \Delta x f'(x)$$

中值定理给出严格版本：若 $f \in C^1$ ，则存在 ξ 位于 x 与 $x + \Delta x$ 之间，使得

$$f(x + \Delta x) - f(x) = \Delta x f'(\xi).$$

绝对误差：误差被放大多少？

输入与输出的绝对误差分别是

$$|\Delta x|, \quad |\Delta f| = |f(x + \Delta x) - f(x)|.$$

由局部线性化，

$$\boxed{\frac{|\Delta f|}{|\Delta x|} \approx |f'(x)|}$$

$$\begin{array}{ll} |f'(x)| > 1 & \text{误差被放大} \\ |f'(x)| < 1 & \text{误差被收缩} \end{array}$$

比较 $|f'(x)|$ 与 1 依赖所选尺度/单位——这正是下一页引入**相对误差**的原因。

相对误差与条件数

相对误差比较的是

$$\frac{|\Delta x|}{|x|} \quad \text{与} \quad \frac{|\Delta f|}{|f(x)|}.$$

由局部线性化，

$$\boxed{\text{cond}_f(x) := \left| \frac{x f'(x)}{f(x)} \right| \approx \frac{|\Delta f|/|f(x)|}{|\Delta x|/|x|}}.$$

条件数 (Condition number): 相对误差的局部放大倍数。

无量纲；简单的单位换算不改变它。

如果 f 不可导，或 $f'(x)$ 难以直接控制呢？

条件性与数值稳定性

算法输出 $\hat{y}$ 。若它可以解释成某个**附近输入**的精确答案：

$$\hat{y} = f(\tilde{x}),$$

那么在相对尺度下比较：

后向误差

$$\frac{|\tilde{x} - x|}{|x|}$$

前向误差

$$\frac{|\hat{y} - f(x)|}{|f(x)|}$$

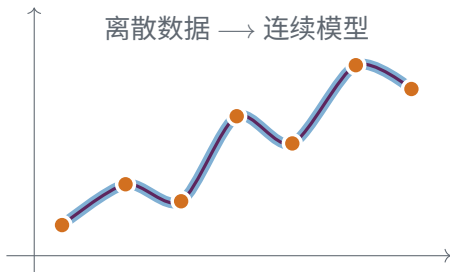
$$\text{前向误差} \lesssim \text{cond}_f(x) \times \text{后向误差}$$

插值与拟合

预测天体运动轨迹

工业设计

计算机动画



稳定性与动力系统

- ▶ 线性动力系统的稳定性；
- ▶ 电力、网络调度策略的收敛性；
- ▶ 线性代数问题的稳定性。

现代算法设计

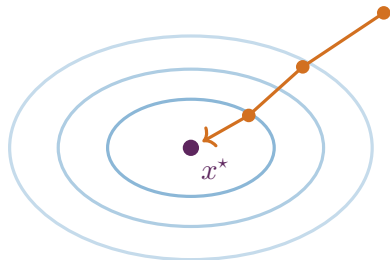
- ▶ PageRank；
- ▶ 估计互联网节点数；
- ▶ 局部聚类；
- ▶ Markov chain Monte Carlo。

最优化与线性规划

模型训练

资源 / 电力 / 物流调度

分子能量最小化、结构设计等问题也属于同一视角。



寻找使目标函数最小的点

Computational lens

- ▶ Computational biology;
- ▶ Algorithmic game theory;
- ▶ Computational phase transitions。

即使经典“组合”问题，也常需要连续视角：

- ▶ 最大流算法中会出现连续优化、电阻网络等观点；
- ▶ 一些**低影响**布尔函数问题可借助不变性原理（Invariance Principle）转到高斯空间中研究。

求解方程的根

先从一个一维模型问题，把收敛、复杂性、条件性真正算一遍。

- ▶ 输入：连续函数 f ，并承诺它至少有一个根；
- ▶ 求解：任意一个 r ，使得 $f(r) = 0$ ；
- ▶ 假设 f 以**神谕模型**给定：可以查询任意 $f(x)$ 。

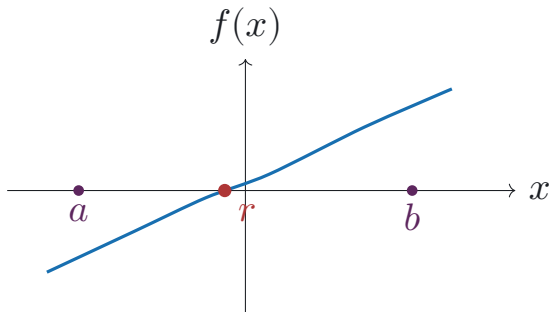
思考：为什么需要假设 f 连续？

求解方程的根——二分法

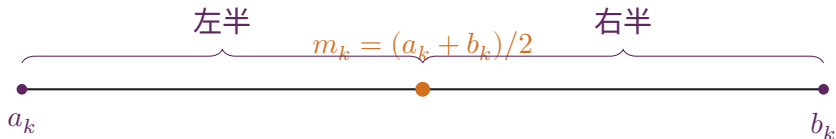
再假设已知一个**异号区间** $[a, b]$:

$$f(a)f(b) < 0.$$

由介值定理，只要 f 连续， $[a, b]$ 中就存在根 r 。



二分法：每次保留仍然异号的一半



查询 $f(m_k)$ ，保留满足端点异号的半个区间。

不变量：至少一个根始终被夹在当前区间里。

二分法的收敛速度

n 次二分后:

$$b_n - a_n = \frac{b - a}{2^n}.$$

若输出中点 x_n , 则被夹住的根 r 满足

$$|x_n - r| \leq \frac{b - a}{2^{n+1}}.$$

因此达到 $|x_n - r| \leq \varepsilon$, 只需

$$n \geq \left\lceil \log_2 \frac{b - a}{2\varepsilon} \right\rceil,$$

也就是 $O\left(\log \frac{b - a}{\varepsilon}\right)$ 次迭代。

从二分法到不动点迭代

二分法维护一个**区间**；另一条路线是只维护一个当前猜测 x_k 。
把

$$f(x) = 0$$

改写成等价的不动点方程

$$\boxed{g(x) = x},$$

然后直接迭代

$$\boxed{x_{k+1} = g(x_k)}.$$

可以利用函数的局部形状；但失去了二分法“根始终被夹住”的自动保证。

怎样选择 g ，让根成为稳定的不动点？

求解方程的根——不动点迭代法

► 若 $g(r) = r$ ，称 r 为 g 的一个**不动点**；

► 从初值 x_0 出发做不动点迭代：

$$x_{k+1} = g(x_k), \quad k = 0, 1, 2, \dots$$

► 同一个根方程也可能产生坏迭代：

$$x = 0 \quad \text{等价于} \quad x = 2x, \quad x_{k+1} = 2x_k,$$

因而 $x_k = 2^k x_0$ ；若 $x_0 \neq 0$ ，迭代发散；

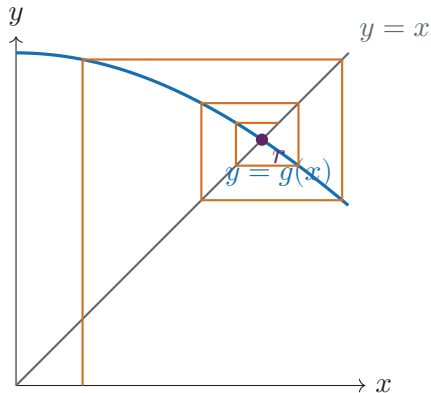
► 但对连续 g ，若 x_k 收敛，则一定收敛到不动点。

如果迭代收敛，极限一定是不动点

若 $x_{k+1} = g(x_k)$ ， g 连续，而且 $x_k \rightarrow r$ ，则

$$\begin{aligned} g(r) &= g\left(\lim_{k \rightarrow \infty} x_k\right) \\ &= \lim_{k \rightarrow \infty} g(x_k) \\ &= \lim_{k \rightarrow \infty} x_{k+1} \\ &= r. \end{aligned}$$

不动点迭代：图像上发生了什么？



从 x_0 开始：

$$x_1 = g(x_0),$$

$$x_2 = g(x_1), \dots$$

竖直到 $y = g(x)$,

水平到 $y = x$ 。

不动点迭代的收敛速度

记第 k 次迭代误差为

$$e_k = |x_k - r|.$$

若下列极限存在，称

$$S = \lim_{k \rightarrow \infty} \frac{e_{k+1}}{e_k}$$

为**收敛速度**。

$$0 < S < 1$$

线性收敛

$$S = 0$$

比线性收敛更快

(后面牛顿法会出现)

局部收敛：第一步只需要中值定理

设 g 连续可导, $g(r) = r$ 。由中值定理,

$$\begin{aligned}x_{k+1} - r &= g(x_k) - g(r) \\ &= g'(\xi_k)(x_k - r),\end{aligned}$$

其中 ξ_k 位于 x_k 与 r 之间。

因此

$$e_{k+1} = |g'(\xi_k)|e_k.$$

什么时候会收敛?

若

$$\rho := |g'(r)| < 1, \quad q = \frac{1 + \rho}{2} < 1,$$

则由 g' 连续, 在 r 的某个邻域内

$$|g'(x)| \leq q.$$

于是, 只要 x_0 足够接近 r ,

$$e_{k+1} \leq qe_k,$$

所以 $x_k \rightarrow r$; 再由 $\xi_k \rightarrow r$,

$$S := \lim_{k \rightarrow \infty} \frac{e_{k+1}}{e_k} = |g'(r)| = \rho.$$

$0 < \rho < 1$: 线性收敛; $\rho = 0$: 可能更快。

稳定与不稳定的不动点

稳定的不动点

- ▶ 附近迭代趋向 r ;
- ▶ 误差越来越小;
- ▶ $|g'(r)| < 1$ 。

不稳定的不动点

- ▶ 附近迭代远离 r ;
- ▶ 小扰动会被放大;
- ▶ $|g'(r)| > 1$;
- ▶ 与“混沌”现象相关。

例如 $g_\lambda(x) = x - \lambda f(x)$ ($\lambda \neq 0$): $g'_\lambda(r) = 1 - \lambda f'(r)$ 。

选择更新规则，就是在选择局部收缩率。

根的敏感性

设 r 是 f 的**简单根**： $f(r) = 0$ 且 $f'(r) \neq 0$ 。

假设计算函数本身有误差，记**误差函数**为 $\varepsilon h(x)$ ：

$$\tilde{f}_\varepsilon(x) = f(x) + \varepsilon h(x).$$

把附近的扰动根记为

$$r(\varepsilon) = r + \Delta r,$$

于是

$$f(r(\varepsilon)) + \varepsilon h(r(\varepsilon)) = 0.$$

一个小的函数误差，会让根移动多少？

根的敏感性：对扰动参数求导

对上一页的恒等式

$$f(r(\varepsilon)) + \varepsilon h(r(\varepsilon)) = 0$$

在 $\varepsilon = 0$ 处求导：

$$f'(r) r'(0) + h(r) = 0 \quad \text{所以} \quad \boxed{r'(0) = -\frac{h(r)}{f'(r)}}.$$

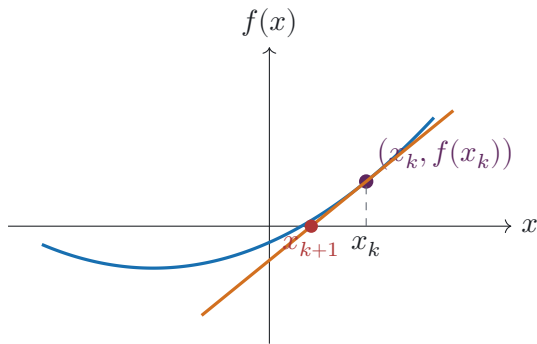
补充：隐函数定理可以严格化上述“扰动根随参数可微”的论证。因此

$$r(\varepsilon) = r - \varepsilon \frac{h(r)}{f'(r)} + o(\varepsilon).$$

简单根越“平”，根越敏感：局部尺度 $\sim 1/|f'(r)|$ 。

重根 $f'(r) = 0$ 时，一阶近似失效。 作业 1：Wilkinson 多项式。

牛顿法：解当前位置的局部线性模型



在 x_k 附近，

$$f(x) \approx f(x_k) + f'(x_k)(x - x_k).$$

令这个**局部线性模型**=0:

$$0 = f(x_k) + f'(x_k)(x_{k+1} - x_k),$$

所以

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}.$$

同一个 $1/f'$ ：前面控制根的敏感性，这里决定牛顿修正的尺度。

牛顿法的二次收敛

记 $e_k = |x_k - r|$ 。在简单根附近，牛顿法满足

$$e_{k+1} = O(e_k^2).$$

误差被“平方”：正确位数通常每步约翻倍，有时更快。

定理：若 $f \in C^2$ ， $f(r) = 0$ 且 $f'(r) \neq 0$ ，则当 x_0 足够接近 r 时牛顿法收敛，并且

$$\lim_{k \rightarrow \infty} \frac{e_{k+1}}{e_k^2} = \left| \frac{f''(r)}{2f'(r)} \right|.$$

牛顿法为什么比一般不动点迭代快？

牛顿法是不动点迭代

$$g(x) = x - \frac{f(x)}{f'(x)}.$$

求导：

$$g'(x) = \frac{f(x)f''(x)}{f'(x)^2}.$$

在根 r 处， $f(r) = 0$ ，所以

$$\boxed{g'(r) = 0.}$$

一般迭代： $e_{k+1} \approx |g'(r)|e_k$ ；**牛顿法：** $g'(r) = 0$ 。

一阶误差项被消掉；误差展开从二阶或更高阶开始。

二次收敛：一次泰勒展开

在 x_k 附近展开根 r ：

$$0 = f(r) = f(x_k) + f'(x_k)(r - x_k) + \frac{1}{2}f''(\xi_k)(r - x_k)^2.$$

牛顿步满足

$$f(x_k) + f'(x_k)(x_{k+1} - x_k) = 0.$$

两式相减：

$$e_{k+1} = \frac{|f''(\xi_k)|}{2|f'(x_k)|}e_k^2.$$

- ▶ 牛顿法每一步只需要函数值和一阶导数；二次收敛分析甚至可在 f' 局部利普希茨连续时进行；
- ▶ 使用更高阶导数，可以做更高阶局部近似（例如 Halley 方法）；
- ▶ 牛顿法很自然地推广到**多变量求根**和**最优化**。

割线法 (Secant method): 用两次函数值近似导数

$$x_{k+1} = x_k - f(x_k) \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}$$

在简单根和常规光滑条件下，其收敛阶为

$$\frac{1 + \sqrt{5}}{2} \approx 1.618.$$

高维中也有类似思想：**Broyden** 用于非线性方程组；**BFGS** 是最经典的优化拟牛顿方法之一。

二分法 / 异号区间

保持异号区间

保证根不丢

稳健但较慢

+

快速局部步

割线法 / 逆二次插值

接近根时很快

但单独使用不够稳健

Dekker / Brent：维持异号区间，同时尝试更快的插值步。

Brouwer 不动点定理

连续映射 $F : K \rightarrow K$ ，其中 $K \subset \mathbb{R}^d$ 非空、紧致、凸，则 F 有不动点。

- ▶ **Sperner 引理**：Brouwer 的组合对应物；也可用来证明 Brouwer；
- ▶ Sperner 型论证 $\rightarrow$ 公平分租；
- ▶ Brouwer / Kakutani $\rightarrow$ **纳什均衡** (Nash equilibrium) 的存在性。

下节课：插值

本讲的一条长期主线： 迭代算法 $\longleftrightarrow$ 不动点与局部稳定性

插值

在 $\mathbb{R}$ 上：不仅问“能不能插值”，还要问**误差与稳定性**。

在有限域上，同一个多项式插值思想又变成：

Shamir 秘密分享 和 Reed-Solomon 自纠错码.