数值计算基础+线性代数回顾

计算机表达的整数

机器上能表达的最小正实数：机器精度

$$\epsilon_m = \min\{g \in A | 1 \oplus g > 1, g > 0\}$$

机器精度的数值依赖于我们在一个实数上愿意投入多少内存。
float：10^{-7}
double：10^{-16}

浮点数：β, t, L, U

一个一般的浮点数系统F可以用基底β,尾数位数t,指数e的上下限L, U表示：

$$F(\beta, t, L, U) = \{0\} \cup \left[x \in R : x = (-1)^s \beta^e \sum_{k=1}^{t} a_k \beta^{-k} \right]$$

$$F(\beta, t, L, u) = \{0\} + \cup \left[x \in R : x = (-1)^s \beta^e \sum_{k=1}^{t} a_k \beta^{-k} \right]$$

为保证唯一性，一般假设$a_1 ! = 0$，否则总可移动满足条件。这样的浮点数表示称为规范化的浮点数表达。

特殊字符与存储

Inf, NaN
big endian与little endian

浮点数范围

$e = U, \quad x_{max} = \beta^U (1 - \beta^{-t})$
$e = L, \quad x_{min} = \beta^{L-1}$

函数求值误差分析：

$$\text{所求}: f(x)$$
$$\text{实际}: \hat{f}(\hat{x})$$
$$\hat{f}(\hat{x}) - f(x) = [\hat{f}(\hat{x}) - f(\hat{x})] + [f(\hat{x}) - f(x)]$$

- 第一项为输入同时为$\hat{x}$时计算过程的误差。包括截断误差和舍入误差。
- 第二项是数据传递误差。它是由问题敏感性决定的

1. 计算误差

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$
$$f(x+h) = f(x) + h \cdot f'(x) + \frac{h^2}{2} f''(\xi), \quad \xi \in [x, x+h]$$

设M是$|f''(x)|$上限，阶段误差$\frac{Mh^2}{2}$. 假设计算$f(x)$ 过程中舍入误差ϵ,在差分中被算了两次，所以舍入误差进入$f'(x)$中大小$\frac{2\epsilon}{h}$，总误差

$$\epsilon_{tot} = \frac{Mh^2}{2} + \frac{2\epsilon}{h}$$

敏感性：问题的变态性，一定的输入相对变化下输出的相对变化率。

$$\text{cond} = \left| \frac{[f(\hat{x}) - f(x))]/f(x)}{(\hat{x} - x)/x} \right|$$

$$f(\hat{x}) - f(x) \approx f'(x)(\hat{x} - x)$$
$$\text{cond} \approx \left| \frac{xf'(x)}{f(x)} \right|$$

例子（Wilkinson,1960s）

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

its roots are $x_i = i \quad i = 1, 2, \cdots, 20$
consider the root of the polynomial $P(x) + Yx^{19} = 0$. Its roots are $x_i = x_i(Y) = i + \epsilon_i$
if $Y \to 0$, $x_i(y) \to x_i(0) = i$
After the addition of Yx^{19}:

$$P(i + \epsilon_i) + Y(i + \epsilon_i)^{19} + P(i) + P'(i)\epsilon_i + Yi^19 = 0$$
$$\text{so} \quad \frac{\epsilon_i}{Y} = -\frac{i^{19}}{P'(i)}$$

for small$i$, i^{19} is way smaller than $P'(i)$. However, as i get bigger, i^{19} grow up quickly. And the solution x_i become unstable as Y varies.

find roots for polynomial

$$C = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -6 & 4 & 2 \end{bmatrix}$$

$$|C - \lambda i| = \begin{vmatrix} -\lambda & 1 & 0 \\ 0 & -\lambda & 1 \\ -6 & 5 & 2-\lambda \end{vmatrix} = -\lambda^3 + 2\lambda^2 + 5\lambda - 6 = 0$$

The rootfinding problem is converted to finding the eigenvalues for a specific matrix(companion matrix).

舍入误差与算法的稳定性
既然计算机中的实数都是利用不精确的浮点数来表达的，我们需要讨论算法的稳定性，也就是初始值中可能的误差，随着算法的运行时如何传递到最终的结果中去的。既然每次计算机的操作往往伴随着舍入误差，我们就需要分析这个误差时如何随着算法的发展而传递的。

算法复杂度

> 十把完全相同的钥匙但对应不同的锁，如果对应关系完全未知，只通过试开来配对，最好用多少次？最差要用多少次？平均要开多少次？

解：最好10次，最差45次，期望值为29.571，被定义为**平均计算复杂度**。不是几何平均值或算数平均值20.125

矩阵与线性代数

方阵的trace与determinant

$$\mathrm{tr}(A) = \sum_{i=1}^{n} a_{ii}$$
$$\det(A) = \sum_{\pi \in P} \mathrm{sign}(\pi) a_{1\pi 1} a_{2\pi 2} \cdots a_{n\pi n}$$
$$A^{-1} = \frac{1}{\det(A)} \Delta^T$$

方阵的逆存在的充要条件为行列式不为零。

Cramer's rule

$$A\vec{x}=b$$
$$x_j=\Delta_j/det(A), j=1,\cdots,n$$

其中Δ_j 是将原矩阵第j列换位矢量b所得到矩阵的行列式的值

矩阵的rank与kernel

rank(A):从矩阵A中抽取的非奇异子矩阵的最大阶数。
值域range(A)

$$range(A)=\{y\in K^m|y=A\cdot x,x\in k^n\}$$
$$rank(A)=dim(range(A))$$

kernel:

$$\ker(A)=\{x\in K^n|A\cdot x=0\}$$

性质

- $rank(A)+dim(ker(A))=n$

矢量与矩阵的模

Holder 模：$||\vec{x}||_p=\left(\sum_{i=1}^{n}|x_i|^p\right)^{1/p}$

模的等价性

两个模$||\cdot||_p,||\cdot||_q$的等价：
$\exists$不依赖x的$c,C>0\quad\text{such that}\quad c||\vec{x}||_q\leq||\vec{x}||_p\leq C||\vec{x}||_q,\forall\vec{x}\in V$

证明

introduce a unit circle:

$$S=\{\vec{x}|||\vec{x}||_\infty=1\}$$
$$\cdots$$

得证

可以证明，有限维矢量空间中的任何模都是等价的。