计算物理第一次作业　作业解答

隋源　2000011379

Oct 8th 2022

1. 计算 e^{-x}

直接展开法、递归法、倒数递归法的程序结果对比如下表 (计算程序见 exponent.py)

e^{-x}	e^{-10}	e^{-20}	e^{-30}	e^{-40}	e^{-50}	e^{-60}	e^{-70}	e^{-80}	e^{-90}	e^{-100}
直接展开法量级	-5	-10	-5	-1	3	9	13	17	21	26
递归法量级	-5	-9	-6	-1	3	8	12	17	21	25
倒数递归法量级	-5	-9	-14	-18	-22	-27	-31	-35	-40	-44

表中数据来自三种方法计算 200 阶级数的结果，其最终结果已不再随阶数增加而改变，且倒数递归法结果已经收敛至足够精确。

可以看出直接展开法最多算对 e^{-10} 的量级，递归法最多算对 e^{-10}, e^{-20} 的量级，通过压力测试发现直接展开法的程序速度明显慢很多。这是因为直接展开法和递归法涉及大量浮点数运算，舍入误差不可忽略；而倒数递归法只涉及整数运算，只有最后一步产生误差。

2. 矩阵的模和条件数

(a)

矩阵 A 的行列式为

$$\det(A) = \prod_{i=1}^{n} A_{ii} = 1 \neq 0 \tag{1}$$

因此 A 是非奇异矩阵。

(b)

方程组

$$Ax = e_i \tag{2}$$

的解即为 A^{-1} 的第 i 列，得到

$$(A^{-1})_{ij} \begin{cases} 2^{j-i-1} & , \quad i < j \\ \delta_{ij} & , \quad i \geq j \end{cases} \tag{3}$$

即逆矩阵的形式为

$$A^{-1} = \begin{pmatrix} 1 & 1 & 2 & \cdots & 2^{n-2} \\ & 1 & 1 & \cdots & 2^{n-3} \\ & & 1 & \cdots & 2^{n-4} \\ & & & \ddots & \vdots \\ & & & & 1 \end{pmatrix} \tag{4}$$

(c)

当 $p \to \infty$ 时

$$\begin{aligned} \|Ax\|_\infty &= \max_{i=1,\cdots,n} \left| \sum_{j=1}^{n} a_{ij} x_j \right| \\ &\leq \max_{i=1,\cdots,n} \sum_{j=1}^{n} |a_{ij}| \, |x_j| \\ &\leq \max_{i=1,\cdots,n} |x_i| \cdot \max_{i=1,\cdots,n} \sum_{j=1}^{n} |a_{ij}| \end{aligned} \tag{5}$$

因确有等号成立情形，故

$$\|A\|_\infty = \sup_{x \neq 0} \frac{\|Ax\|_\infty}{\max_{i=1,\cdots,n} |x_i|} = \max_{i=1,\cdots,n} \sum_{j=1}^{n} |a_{ij}| \tag{6}$$

证毕。

(d)

对于幺正矩阵 $U \in \mathbb{C}^{n \times n}$，有

$$\|U\|_2 = \frac{(x^* U^* U x)^{\frac{1}{2}}}{(x^* x)^{\frac{1}{2}}} = 1 \tag{7}$$

同理可得

$$\|U\|_2 = \|U^\dagger\|_2 = 1 \tag{8}$$

$\forall A \in \mathbb{C}^{n \times n}$

$$\|UA\|_2 = \frac{(x^* A^* U^* U A x)^{\frac{1}{2}}}{(x^* x)^{\frac{1}{2}}} = \frac{(x^* A^* A x)^{\frac{1}{2}}}{(x^* x)^{\frac{1}{2}}} = \|A\|_2 \tag{9}$$

(e)

根据定义

$$\|A\|_\infty = \sum_{j=1}^{n} |a_{1j}| = n, \quad \|A^{-1}\|_\infty = \sum_{j=1}^{n} |a_{1j}| = 2^{n-1} \tag{10}$$

故条件数为

$$K_\infty(A) = \|A\|_\infty \|A^{-1}\|_\infty = n \cdot 2^{n-1} \tag{11}$$

3.Hilbert 矩阵

(a)

对积分取极值，有

$$\frac{\partial D}{\partial c_i} = \int_0^1 \left(\sum_{j=1}^{n} c_i x^{i-1} - f(x) \right) x^{i-1} \mathrm{d}x = 0 \tag{12}$$

整理得到

$$\sum_{j=1}^{n} \frac{c_j}{i+j-1} = \int_0^1 f(x)\, x^{j-1} \mathrm{d}x \tag{13}$$

这等价于矩阵乘法 $H_n c = b$，其中

$$(H_n)_{ij} = \frac{1}{i+j-1}, \quad b_j = \int_0^1 f(x)\, x^{j-1} \mathrm{d}x \tag{14}$$

(b)

由式 ()，显然有

$$(H_n)_{ij} = (H_n)_{ji} \tag{15}$$

由于

$$\begin{aligned}
\int_0^1 \left(\sum_{i=0}^{n} c_i x^{i-1} \right)^2 \mathrm{d}x &= \int_0^1 \sum_{i=0}^{n} c_i x^{i-1} \cdot \sum_{j=0}^{n} c_j x^{j-1} \mathrm{d}x \\
&= \int_0^1 \sum_{i=0}^{n} \sum_{j=0}^{n} c_i c_j x^{i+j-2} \mathrm{d}x \\
&= \sum_{i=0}^{n} \sum_{j=0}^{n} \frac{c_i c_j}{i+j-1} \\
&= c^{\mathrm{T}} H_n c
\end{aligned} \tag{16}$$

而积分半正定，因此 $c^{\mathrm{T}} H_n c \geq 0$，当且仅当 $\sum_{i=0}^{n} c_i x^{i-1} = 0$，即 $c = 0$ 时成立。

由于 H_n 为对称正定实矩阵，因此其本征值均大于 0。故

$$\det(H_n) = \prod_{i=1}^{s} \lambda^{r_i} > 0 \tag{17}$$

即 H_n 是非奇异的。

(c)

取对数

$$\ln[\det{(H_n)}] = 3\sum_{i=1}^{n-1}\ln i! - \sum_{i=n}^{2n-1}\ln i! \tag{18}$$

行列式的值如下表 (计算程序见 det.py)

n	1	2	3	4	5
$\vert H_n\vert$	1.0×10^{0}	8.3×10^{-2}	4.6×10^{-4}	1.7×10^{-7}	3.7×10^{-12}
n	6	7	8	9	10
$\vert H_n\vert$	5.4×10^{-18}	4.8×10^{-25}	2.7×10^{-33}	9.7×10^{-43}	2.2×10^{-53}

(d)

GEM 和 Cholesky 方法得到的 n 取 1-10 时 x_1 的值与理论值的偏差如下 (计算程序见 equation.py)

n	1	2	3	4	5
GEM	0	-9×10^{-16}	2×10^{-14}	7×10^{-13}	3×10^{-13}
Cholesky	0	-4×10^{-16}	2×10^{-14}	4×10^{-14}	1×10^{-11}
n	6	7	8	9	10
GEM	-1×10^{-9}	4×10^{-8}	2×10^{-7}	-6×10^{-5}	2×10^{-3}
Cholesky	-9×10^{-10}	3×10^{-8}	3×10^{-7}	-5×10^{-5}	1×10^{-3}

可以发现 Cholesky 的精度基本好于 GEM。这主要是因为 H_n 在 n 较大时条件数 $K(H_n)$ 相当大，而误差基本与条件数正相关，因此对于 Cholesky 分解 $H_n = A^\dagger A$ 其误差比 GEM 小。

4. 级数求和与截断误差

(a)

$$f(q^2 = 0.5) = 1.106217 \tag{19}$$

具体过程见 (c) 问。

(b)

引入截断 Λ。首先计算主值积分

$$\mathrm{P.V.}\int \mathrm{d}^3 n \frac{1}{|\vec{n}|^2 - q^2} = 4\pi\Lambda + 2\pi q \ln\frac{\Lambda - q}{\Lambda + q} \tag{20}$$

为了估算收敛速度，取中心为 Λ(Λ 很大)，厚度为 1 的球壳估算 $f(q^2)$ 的改变量。积分的改变量为

$$\Delta_{int} = 4\pi\left(1+\frac{q}{\Lambda}\right) + O\left(\frac{1}{\Lambda^2}\right) \tag{21}$$

求和的改变量可预估为

$$\Delta_{sum} = \frac{4}{3}\pi[(\Lambda+1/2)^3-(\Lambda-1/2)^3]\frac{1}{\Lambda^2-q^2} = 4\pi + O\left(\frac{1}{\Lambda^2}\right) \tag{22}$$

故函数改变量

$$\Delta f = \Delta_{sum} - \Delta_{int} \propto \frac{1}{\Lambda} \tag{23}$$

即收敛速度为 $1/\Lambda$ 量级。因此，若要求精度为 10^{-5}，至少有 $\Lambda \sim 10^6$。实际上程序运行到 10^4 量级时就几乎无法跑出结果，且精度也只有 10^{-2} 左右。(计算程序见 series_try.py)

(c)

提高计算机效率的思路是将函数改写成收敛的求和/积分形式。将式子改写为

$$f(q^2) = \lim_{\vec{r}\to 0} f\left(q^2,\vec{r}\right) = \lim_{\vec{r}\to 0}\left[\sum_{\vec{n}\in\mathbb{Z}^3}\frac{e^{i\vec{n}\cdot\vec{r}}}{|\vec{n}|^2-q^2} - \text{P.V.}\int \mathrm{d}^3\vec{n}\frac{e^{i\vec{n}\cdot\vec{r}}}{|\vec{n}|^2-q^2}\right] \tag{24}$$

改写求和项

$$\begin{aligned}\sum_{\vec{n}\in\mathbb{Z}^3}\frac{e^{i\vec{n}\cdot\vec{r}}}{|\vec{n}|^2-q^2} &= \sum_{\vec{n}\in\mathbb{Z}^3}\frac{e^{i\vec{n}\cdot\vec{r}}e^{-\left(|\vec{n}|^2-q^2\right)}}{|\vec{n}|^2-q^2} + \sum_{\vec{n}\in\mathbb{Z}^3}\frac{e^{i\vec{n}\cdot\vec{r}}\left(1-e^{-\left(|\vec{n}|^2-q^2\right)}\right)}{|\vec{n}|^2-q^2} \\ &= \sum_{\vec{n}\in\mathbb{Z}^3}\frac{e^{i\vec{n}\cdot\vec{r}}e^{-\left(|\vec{n}|^2-q^2\right)}}{|\vec{n}|^2-q^2} + \int_0^1 \mathrm{d}t e^{tq^2}\sum_{\vec{n}\in\mathbb{Z}^3}K(\vec{n},\vec{r})\end{aligned} \tag{25}$$

其中 $K(\vec{n},\vec{r}) = \exp(i\vec{n}\cdot\vec{r} - t|\vec{n}|^2)$。作傅立叶变换

$$\begin{aligned}K(\vec{k},\vec{r}) &= \int \mathrm{d}^3\vec{n}e^{i\vec{n}\cdot\vec{r}-t|\vec{n}|^2}e^{-2\pi i\vec{k}\cdot\vec{n}} \\ &= \int \mathrm{d}^3\vec{n}e^{i\vec{n}\cdot(\vec{r}-2\pi\vec{k})-t|\vec{n}|^2} \\ &= 2\pi\int_0^\infty \mathrm{d}n n^2\int_0^\pi \mathrm{d}\theta\sin\theta e^{i|\vec{r}-2\pi\vec{k}|n\cos\theta-tn^2} \\ &= 2\pi\int_0^\infty \mathrm{d}n\frac{2n\sin(n|\vec{r}-2\pi\vec{k}|)}{|\vec{r}-2\pi\vec{k}|}e^{-tn^2} \\ &= \left(\frac{\pi}{t}\right)^{3/2}e^{-\frac{|\vec{r}-2\pi\vec{k}|^2}{4t}}\end{aligned} \tag{26}$$

由泊松求和公式

$$\sum_{\vec{n}\in\mathbb{Z}^3}K(\vec{n},\vec{r}) = \sum_{\vec{k}\in\mathbb{Z}^3}K(\vec{k},\vec{r}) \tag{27}$$

可进一步改写求和项

$$\begin{aligned}\sum_{\vec{n}\in\mathbb{Z}^3}\frac{e^{i\vec{n}\cdot\vec{r}}}{|\vec{n}|^2-q^2} &= \sum_{\vec{n}\in\mathbb{Z}^3}\frac{e^{i\vec{n}\cdot\vec{r}}e^{-\left(|\vec{n}|^2-q^2\right)}}{|\vec{n}|^2-q^2}+\sum_{\vec{k}\in\mathbb{Z}^3}\int_0^1\,\mathrm{d}te^{tq^2}\left(\frac{\pi}{t}\right)^{3/2}e^{-\frac{|\vec{r}-2\pi\vec{k}|^2}{4t}}\\ &= \sum_{\vec{n}\in\mathbb{Z}^3}\frac{e^{i\vec{n}\cdot\vec{r}}e^{-\left(|\vec{n}|^2-q^2\right)}}{|\vec{n}|^2-q^2}+\sum_{\vec{k}\in\mathbb{Z}^3,\vec{k}\neq0}\int_0^1\,\mathrm{d}te^{tq^2}\left(\frac{\pi}{t}\right)^{3/2}e^{-\frac{|\vec{r}-2\pi\vec{k}|^2}{4t}}\\ &\quad+\int_0^1\,\mathrm{d}t(e^{tq^2}-1)\left(\frac{\pi}{t}\right)^{3/2}e^{-\frac{r^2}{4t}}+\int_0^1\,\mathrm{d}t\left(\frac{\pi}{t}\right)^{3/2}e^{-\frac{r^2}{4t}}\end{aligned}\tag{28}$$

此时求和部分前三项均已收敛，而最后一项用以和积分发散部分抵消。改写积分项

$$\begin{aligned}\mathrm{P.V.}\int\mathrm{d}^3\vec{n}\frac{e^{i\vec{n}\cdot\vec{r}}}{|\vec{n}|^2-q^2} &= 2\pi\mathrm{P.V.}\int_0^\infty\mathrm{d}nn^2\int_0^\pi\mathrm{d}\theta\sin\theta\frac{e^{inr\cos\theta}}{n^2-q^2}\\ &= \frac{4\pi}{r}\mathrm{P.V.}\int_0^\infty\mathrm{d}n\frac{n\sin(nr)}{n^2-q^2}\\ &= \frac{2\pi^2}{r}\cos(qr)\\ &= \cos(qr)\int_0^\infty\,\mathrm{d}t\left(\frac{\pi}{t}\right)^{3/2}e^{-\frac{r^2}{4t}}\end{aligned}\tag{29}$$

令 $\vec{r}\to0$，可以得到

$$\begin{aligned}f(q^2) &= I_1+I_2+I_3+I_4\\ I_1 &= \sum_{\vec{n}\in\mathbb{Z}^3}\frac{e^{-\left(|\vec{n}|^2-q^2\right)}}{|\vec{n}|^2-q^2}\\ I_2 &= \sum_{\vec{k}\in\mathbb{Z}^3,\vec{k}\neq0}\int_0^1\,\mathrm{d}te^{tq^2}\left(\frac{\pi}{t}\right)^{3/2}e^{-\frac{\pi^2|\vec{k}|^2}{t}}\\ I_3 &= \int_0^1\,\mathrm{d}t(e^{tq^2}-1)\left(\frac{\pi}{t}\right)^{3/2}\\ I_4 &= -\int_1^\infty\,\mathrm{d}t\left(\frac{\pi}{t}\right)^{3/2}\end{aligned}\tag{30}$$

其中 I_4 可准确求解

$$I_4=-2\pi^{3/2}=-11.13665599\tag{31}$$

I_3 可直接积分

$$I_3=6.08324678\tag{32}$$

因为 $\pi^{3/2}e^{-2\pi^2}\sim10^{-8}$，故在 10^{-5} 精度要求下只需要截断至 $|\vec{k}|=1$，积分结果为

$$I_2=6\int_0^1\,\mathrm{d}te^{tq^2}\left(\frac{\pi}{t}\right)^{3/2}e^{-\frac{\pi^2}{t}}=0.00026490\tag{33}$$

I_1 的收敛速度非常快，取截断 $\Lambda\geq5$ 即可。结果为

$$I_1=6.15936128\tag{34}$$

最终得到

$$f(q^2) = 1.106217 \tag{35}$$

满足 10^{-5} 精度要求。实际上精度可以轻松达到 10^{-9} 甚至更高。(计算程序见 series_final.py)