

CP 第一次作业

黄羽翔 2100011536

October 18, 2023

1 数值误差的避免

解题步骤：设置变量存储每次计算结果，且统一使用 double 精度计算。停止条件为 $|x_{k+1} - x_k| < \epsilon$ ，其中 ϵ 为机器精度，即当 $x_k = x_{k+1}$ 时停止迭代。

x 取值	直接展开法	递归法	方法 2 计算 e^x 值	求倒数得 e^{-x} 值
0	1.0	1.0	1.0	1.0
10	4.53999294E-5	4.53999296E-5	2.20264657E+4	4.5399929E-5
20	6.13825973E-9	5.62188447E-9	4.85165195E+8	2.06115362E-9
30	-1.51730129E-4	-3.06681235E-5	1.06864745E+13	9.35762296E-14
40	3.74217741E+0	-3.16573189E+0	2.35385267E+0	4.24835426E-18
50	6.93150014E+4	1.10729334E+4	5.18470553E+21	1.92874985E-22
60	-1.13320491E+9	-3.35168107E+8	1.14200739E+26	8.75651076E-27
70	ERROR	-3.29796047E+13	2.51543867E+30	3.97544974E-31
80	ERROR	9.18056822E+16	5.54062238E+34	1.80485139E-35
90	ERROR	-5.05162535E+20	1.22040329E+39	8.19401262E-40
100	ERROR	-2.91375565E+24	2.68811714E+43	3.72007598E-44

Table 1: 三种方法程序计算结果

结果分析：从表中可看出，直接展开法与递归法在前三项的计算结果在一定范围内相近，体现了算法的正确性。但在 $x \geq 20$ 后，三种方法计算结果产生极大差异，而总有第三种方法最靠近真实值。

其原因在于，直接展开法与递归法在计算过程中，每次计算都会产生误差，并且在 x 较大的计算中，中间项存在远大于 1 的值，而最后计算结果过小，导致前两种算法由精度不足产生的误差此时会非常大。特别的，算法 1 产生的误差是来自于总计算结果，算法 2 产生的误差来自于每一项的计算结果，会导致算法 1 的误差大于算法 2 的误差，而二者差距会保持在一定范围（在表中可见算法 1、2 误差相差在 x 较大时保持在一个数量级左右）。而算法 3 尽管也存在同样的误差，但由于首先计算的是 e^x 的值，在求倒数后误差精度体现在有效位数的后面，因此误差相对较小。

2 矩阵的模与条件数

(a)

对于上三角矩阵（或下三角矩阵），其矩阵行列式值为对角项的乘积（代码有体现计算过程），又题目定义对角项均为 1，故其行列式值为 1。

(b)

利用高斯法，将 A 矩阵化为上三角矩阵：

$$\begin{bmatrix} 1 & -1 & \dots & -1 & 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & -1 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & 0 & 0 & \dots & 1 \end{bmatrix}$$

↓ Gauss

$$\begin{bmatrix} 1 & 0 & \dots & 0 & 1 & 1 & \dots & 2^{n-2} \\ 0 & 1 & \dots & 0 & 0 & 1 & \dots & 2^{n-3} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & 0 & 0 & \dots & 1 \end{bmatrix}$$

则 A^{-1} 为：

$$A^{-1} = \begin{bmatrix} 1 & 1 & \dots & n-1 \\ 0 & 1 & \dots & n-2 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{bmatrix}$$

(c)

若采用矩阵 p 模的定义（以及矢量 p 模的定义），则对于 ∞ 模，有：

$$\|\vec{x}\|_{\infty} = \left(\sum_{i=1}^n |x_i|^{\infty} \right)^{\frac{1}{\infty}} = \max_{1 \leq i \leq n} |x_i|$$

$$\|A\|_{\infty} = \sup_{\vec{x} \neq 0} \frac{\|A\vec{x}\|_{\infty}}{\|\vec{x}\|_{\infty}} = \sup_{\vec{x} \neq 0} \frac{\|A\vec{x}\|_{\infty}}{\max_{1 \leq i \leq n} |x_i|}$$

设

$$\vec{y} = A\vec{x}$$

则

$$\|\vec{y}\|_{\infty} = \max_{1 \leq i \leq n} |y_i| = \max_{1 \leq i \leq n} \left| \sum_{j=1}^n a_{ij} x_j \right|$$

若满足条件的项分别为 y_{i1}, x_{i2} ，若 $i1 \neq i2$ ，则 $x_{i1} < x_{i2}$ ，有

$$\frac{\|\vec{y}\|}{\|\vec{x}\|} = \left| \frac{y_{i1}}{x_{i2}} \right| \leq \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}|$$

当且仅当 $i1 = i2$ 时，不等式取等，此时满足 sup 条件，即

$$\|A\|_{\infty} = \left| \frac{y_{i1}}{x_{i1}} \right| = \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}|$$

(d)

对于 p=2 的欧式模，么正矩阵 U 有：

$$\|U\|_2 = \sup_{\vec{x} \neq 0} \frac{\|U\vec{x}\|_2}{\|\vec{x}\|_2} = \sup_{\vec{x} \neq 0} \frac{\|U\vec{x}\|_2}{|\vec{x}|^2}$$

由定义

$$\|U\vec{x}\|_2 = \sqrt{\sum_{i=1}^n \left(\sum_{j=1}^n u_{ij} x_j \right)^2} = \sqrt{x^\dagger U^\dagger U x} = \sqrt{x^\dagger x} = \|\vec{x}\|_2 = |\vec{x}|^2$$

则有

$$\|U\|_2 = \sup_{\vec{x} \neq 0} \frac{|\vec{x}|^2}{|\vec{x}|^2} = 1$$

将全体 U 替换为 U^\dagger , 所有 U^\dagger 替换为 U , 则完成另一个证明类似的, 对于 $\|UA\|_2$ 有

$$\|UA\|_2 = \sup_{x \neq 0} \frac{\|UA\vec{x}\|_2}{\|\vec{x}\|_2} = \sup_{x \neq 0} \frac{\|UA\vec{x}\|_2}{|\vec{x}|^2}$$

而

$$\|UA\vec{x}\|_2 = \sqrt{\sum_{i=1}^n \left(\sum_{j=1}^n u_{ij} a_{jk} x_k \right)^2} = \sqrt{x^\dagger A^\dagger U^\dagger U A x} = \sqrt{x^\dagger A^\dagger A x} = \|A\vec{x}\|_2$$

则

$$\|UA\|_2 = \sup_{x \neq 0} \frac{\|A\vec{x}\|_2}{|\vec{x}|^2} = \|A\|_2$$

如果利用欧氏模来定义条件数, 则

$$K_2(A) = \|A\|_2 \|A^{-1}\|_2 = \|UA\|_2 \|U^\dagger A^{-1}\|_2 = K_2(UA)$$

(e)

对于上述 A 定义, 结合前文所求, 有

$$\begin{aligned} \|A\|_\infty &= \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}| = \sum_{i=1, j} |a_{ij}| = n \\ \|A^{-1}\|_\infty &= \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}| = \sum_{i=1, j} |a_{ij}| = 2^{n-1} \\ K_\infty(A) &= \|A\|_\infty \|A^{-1}\|_\infty = n \times 2^{n-1} \end{aligned}$$

3 Hilbert 矩阵

(a)

若满足

$$D = D_{\min} = \int_0^1 \left(\sum_{i=1}^n c_i x^{i-1} - f(x) \right)^2 dx$$

则有

$$\delta \left(\int_0^1 \left(\sum_{i=1}^n c_i x^{i-1} - f(x) \right)^2 dx \right) = \delta D = 0$$

又

$$\begin{aligned}
\delta D &= \delta \left(\int_0^1 \left(\sum_{i=1}^n c_i x^{i-1} - f(x) \right)^2 dx \right) \\
&= \int_0^1 \delta \left(\left(\sum_{i=1}^n c_i x^{i-1} - f(x) \right)^2 \right) dx \\
&= \int_0^1 dx \left(\frac{\partial}{\partial c_1} \left(\sum_{i=1}^n c_i x^{i-1} - f(x) \right)^2 \delta c_1 + \dots \frac{\partial}{\partial c_n} \left(\sum_{i=1}^n c_i x^{i-1} - f(x) \right)^2 \delta c_n \right) \\
&= \int_0^1 \left(2 \left(\sum_{i=1}^n c_i x^{i-1} - f(x) \right) \left(\sum_{j=1}^n \delta c_j x^{j-1} \right) \right) dx \\
&= 2 \sum_{j=1}^n \delta c_j \left(\sum_{i=1}^n \frac{1}{i+j-1} c_i - \int_0^1 f(x) x^{j-1} dx \right) = 0
\end{aligned}$$

对每一个 δc_j 均满足约束，则满足约束矩阵方程

$$H\vec{c} = \vec{b}$$

其中 $H_{ij} = \frac{1}{i+j-1}$ 为 Hilbert 矩阵, $\vec{c} = \{c_1, \dots, c_n\}^T$ 为待求系数向量, $\vec{b} = \{\int_0^1 f(x)x^0 dx, \dots, \int_0^1 f(x)x^{n-1} dx\}$ 为约束向量

(b)

利用数学归纳法，对于 $n = 1$ ，显然成立。假设对于 $n = k$ 成立，即

$$|H_n| = \begin{vmatrix} 1 & \frac{1}{2} & \dots & \frac{1}{k} \\ \frac{1}{2} & \frac{1}{3} & \dots & \frac{1}{k+1} \\ \dots & \dots & \dots & \dots \\ \frac{1}{k} & \frac{1}{k+1} & \dots & \frac{1}{2k-1} \end{vmatrix} > 0$$

则对于 $n = k + 1$, 有

$$\begin{aligned}
 |H_{k+1,0}| &= \begin{vmatrix} 1 & \frac{1}{2} & \cdots & \frac{1}{k} & \frac{1}{k+1} \\ \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{k+1} & \frac{1}{k+2} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{1}{k} & \frac{1}{k+1} & \cdots & \frac{1}{2k-1} & \frac{1}{2k} \\ \frac{1}{k+1} & \frac{1}{k+2} & \cdots & \frac{1}{2k} & \frac{1}{2k+1} \end{vmatrix} \\
 &= \begin{vmatrix} 1 & \frac{1}{2} & \cdots & \frac{1}{k} & \frac{1}{k+1} \\ \frac{1}{2} - \frac{1}{2} & \frac{1}{3} - \frac{1}{2} \times \frac{1}{2} & \cdots & \frac{1}{k+1} - \frac{1}{2} \times \frac{1}{k} & \frac{1}{k+2} - \frac{1}{2} \times \frac{1}{k+1} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{1}{k} - \frac{1}{k} & \frac{1}{k+1} - \frac{1}{k} \times \frac{1}{2} & \cdots & \frac{1}{2k-1} - \frac{1}{k} \times \frac{1}{k} & \frac{1}{2k} - \frac{1}{k} \times \frac{1}{k+1} \\ \frac{1}{k+1} - \frac{1}{k+1} & \frac{1}{k+2} - \frac{1}{k+1} \times \frac{1}{2} & \cdots & \frac{1}{2k} - \frac{1}{k+1} \times \frac{1}{k} & \frac{1}{2k+1} - \frac{1}{k+1} \times \frac{1}{k+1} \end{vmatrix} \\
 &= \begin{vmatrix} 1 & \frac{1}{2} & \cdots & \frac{1}{k} & \frac{1}{k+1} \\ 0 & \frac{1}{12} & \cdots & \frac{k-1}{2k(k+1)} & \frac{k+1}{2(k+1)(k+2)} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \frac{k-1}{2k(k+1)} & \cdots & \frac{k^2-2k+1}{k^2(2k-1)} & \frac{k^2-k}{2k^2(k+1)} \\ 0 & \frac{k}{2(k+1)(k+2)} & \cdots & \frac{k^2-k}{2k^2(2k+1)} & \frac{k^2}{(k+1)^2(2k+3)} \end{vmatrix} \\
 &= \begin{vmatrix} \frac{1}{12} & \cdots & \frac{k-1}{2k(k+1)} & \frac{k}{2(k+1)(k+2)} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{k-1}{2k(k+1)} & \cdots & \frac{k^2-2k+1}{k^2(2k-1)} & \frac{k^2-k}{2k^2(k+1)} \\ \frac{k}{2(k+1)(k+2)} & \cdots & \frac{k^2-k}{2k^2(2k+1)} & \frac{k^2}{(k+1)^2(2k+3)} \end{vmatrix} \\
 &= (k!)^2 \frac{1}{2^2 \times 3^2 \times \cdots \times (k+1)^2} H_{k+1,2} \\
 &\neq 0
 \end{aligned}$$

即证明 H_n 正定, 而其行列式不为 0 也显然说明 H_n 是非奇异的

(c)

计算结果见代码

可以发现 H_n 以 2^{-n^2} 的量级多级指数衰减

(d)

具体计算结果见代码, 结果如下图所示: (实在懒得打表 hh)

```

C:\1CPhomework\homework1\3.exe
(c)
输入1-10之间的整数n以计算H_n, 输入其它数以跳过此问
0
跳过计算!
(d)
输入保留小数位数a
5
n=1时
使用GEM算法得到解为
x=( 1.00000 )
使用Choleskey算法得到解
x=( 1.00000 )
n=2时
使用GEM算法得到解为
x=( -2.00000 6.00000 )
使用Choleskey算法得到解
x=( -2.00000 6.00000 )
n=3时
使用GEM算法得到解为
x=( 3.00000 -24.00000 30.00000 )
使用Choleskey算法得到解
x=( 3.00000 -24.00000 30.00000 )
n=4时
使用GEM算法得到解为
x=( -4.00000 60.00000 -180.00000 140.00000 )
使用Choleskey算法得到解
x=( -4.00000 60.00000 -180.00000 140.00000 )

```

Figure 1

```

C:\1CPHomework\homework1\3.exe
n=5时
使用GEM算法得到解为
x=( 5.00000 -120.00000 630.00000 -1120.00000 630.00000 )
使用Choleskey算法得到解
x=( 5.00000 -120.00000 630.00000 -1120.00000 630.00000 )

n=6时
使用GEM算法得到解为
x=( -6.00000 210.00000 -1680.00000 5040.00000 -6300.00000 2772.00000 )
使用Choleskey算法得到解
x=( -6.00000 210.00000 -1680.00000 5040.00000 -6300.00000 2772.00000 )

n=7时
使用GEM算法得到解为
x=( 7.00000 -336.00000 3780.00002 -16800.00009 34650.00016 -33264.00014 12012.00004 )
使用Choleskey算法得到解
x=( 7.00000 -336.00000 3780.00000 -16800.00001 34650.00002 -33264.00001 12012.00000 )

n=8时
使用GEM算法得到解为
x=( -8.00000 504.00000 -7559.99997 46199.99972 -138599.99901 216215.99832 -168167.99864 51479.99958 )
使用Choleskey算法得到解
x=( -8.00000 503.99994 -7559.99920 46199.99562 -138599.98804 216215.98289 -168167.98771 51479.99650 )

n=9时
使用GEM算法得到解为
x=( 8.99994 -719.99563 13859.92494 -110879.45775 450447.99209 -1009003.86822 1261255.22402 -823677.09938 218789.27995 )
使用Choleskey算法得到解
x=( 8.99993 -719.99532 13859.92050 -110879.43033 450447.90420 -1009003.71020 1261255.06321 -823677.01289 218789.26085 )

```

Figure 2

可看出两种解法结果在 n 较大时有微小的差别。由于 Choleskey 分解具有高稳定性，其计算精度应当更高。

4 矩阵与二次型

(a)
满足

$$|\lambda I - B| = 0$$

$$(\lambda I - B)\vec{x} = \vec{0}$$

化为

$$(\lambda - 1)(\lambda - 2)(\lambda - 1) + (1 - \lambda) \times 1 \times 1 + (1 - \lambda) \times 1 \times 1 = 0$$

$$\lambda^3 - 4\lambda^2 + 3\lambda = 0$$

$$\lambda(\lambda - 1)(\lambda - 3) = 0$$

特征值为

$$\lambda_1 = 0, \lambda_2 = 1, \lambda_3 = 3$$

对应特征向量为

$$\vec{x}_1 = \left\{ \frac{\sqrt{3}}{3}, \frac{\sqrt{3}}{3}, \frac{\sqrt{3}}{3} \right\}$$

$$\vec{x}_2 = \left\{ \frac{\sqrt{2}}{2}, 0, -\frac{\sqrt{2}}{2} \right\}$$

$$\vec{x}_3 = \left\{ \frac{\sqrt{6}}{6}, -\frac{2\sqrt{6}}{6}, \frac{\sqrt{6}}{6} \right\}$$

(b)

Q 为 \vec{x} 的组合，满足

$$Q = [\vec{x}_1^T, \vec{x}_2^T, \vec{x}_3^T] = \begin{bmatrix} \frac{\sqrt{3}}{3} & \frac{\sqrt{2}}{2} & \frac{\sqrt{6}}{6} \\ \frac{\sqrt{3}}{3} & 0 & -\frac{2\sqrt{6}}{6} \\ \frac{\sqrt{3}}{3} & -\frac{\sqrt{2}}{2} & \frac{\sqrt{6}}{6} \end{bmatrix}$$

$$Q^{-1} = Q^T = \begin{bmatrix} \frac{\sqrt{3}}{3} & \frac{\sqrt{3}}{3} & \frac{\sqrt{3}}{3} \\ \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{6}}{6} & -\frac{2\sqrt{6}}{6} & \frac{\sqrt{6}}{6} \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

设

$$\vec{u} = \{x, y, z\}^T$$

则有

$$\vec{u}^T B \vec{u} = x^2 + 2y^2 + z^2 - 2xy - yz = (x - y)^2 + 1/2(y - z)^2 + 1/2y^2 + 1/2z^2 = 2$$

利用 mathematica 绘图

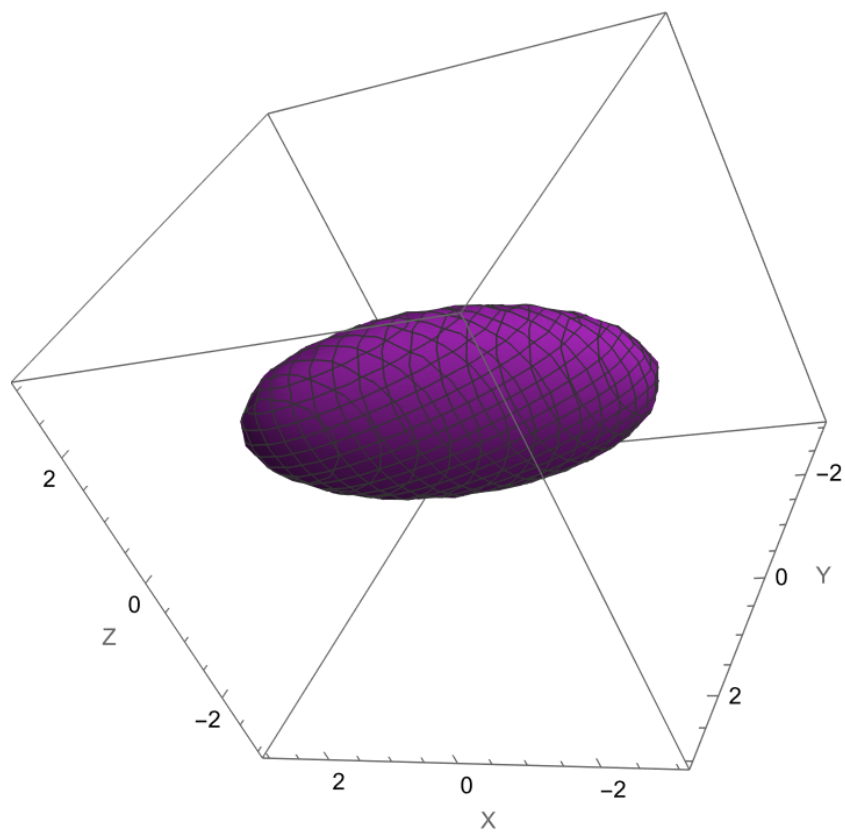


Figure 3

类似的，三个特征向量图像为

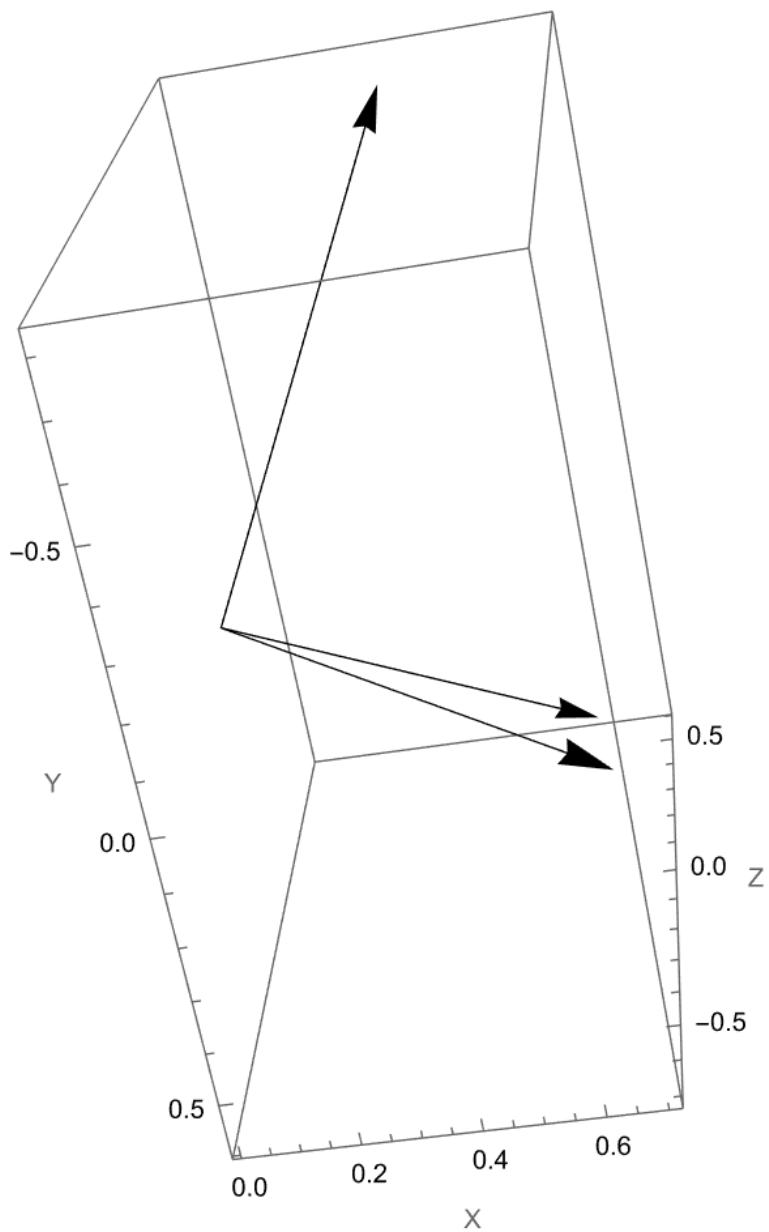


Figure 4

绘图 mathematica 代码附在文件中

5 正定矩阵

(a)

对于 $\vec{u} = \{x_1, x_2, x_3, x_4\}^T$, 有

$$\vec{u}^T A^T = \{x_1, x_2 - x_1, x_3 - x_2, x_4 - x_3, -x_4\}^T$$

则

$$\vec{u}^T A^T A \vec{u} = x_1^2 + (x_2 - x_1)^2 + (x_3 - x_2)^2 + (x_4 - x_3)^2 + x_4^2 \geq 0$$

很明显，对于能分解为 $A^T A$ 形式的矩阵，均有

$$\vec{u}^T A^T A \vec{u} = \sum_i \left(\sum_j a_{ij} u_j \right)^2 \geq 0$$

又对于本式，取等条件为

$$\begin{cases} x_1 = 0 \\ x_2 = x_1 \\ x_3 = x_2 \\ x_4 = x_3 \\ x_4 = 0 \end{cases}$$

当且仅当

$$\vec{u} = \{0, 0, 0, 0\}^T$$

时成立，故 $\vec{u} \neq 0$ 时， $\vec{u}^T A^T A \vec{u} > 0$ ，得证

(b)

设 $\vec{u} = \{x, y\}$ ，有：

$$\begin{aligned} \vec{u}^T S \vec{u} &= 2x^2 + bxy + bxy + 4y^2 \\ \vec{u}^T S \vec{u} &= 2x^2 + bxy + bxy + 4y^2 \\ &= 2(x + \sqrt{2}y)^2 + (2b - 4\sqrt{2})xy \end{aligned}$$

当 $-2\sqrt{2} < b < 2\sqrt{2}$ 时，原式可配为完全平方和

$$\vec{u}^T S \vec{u} = \frac{b}{\sqrt{2}}(x + \sqrt{2}y)^2 + (2 - b/\sqrt{2})x^2 + 2(2 - b/\sqrt{2})y^2 > 0$$

此时原矩阵为正定

当 $b = \pm 2\sqrt{2}$ 时，原式恰可配为完全平方和

$$\vec{u}^T S \vec{u} = 2(x \pm \sqrt{2}y)^2 \geq 0$$

此时原矩阵为半正定

当 $b > 2\sqrt{2}$ 时，原式不可配为完全平方和，故不是正定矩阵