数值分析实验报告

张钰晖

2015011372, yuhui-zh15@mails.tsinghua.edu.cn, 185-3888-2881

2017 年 6 月 15 日

1 题目描述

3-6 编程实现 Hilbert 矩阵 $\boldsymbol{H_n}$，以及 n 维向量 $\boldsymbol{b} = \boldsymbol{H_n x}$，其中，$\boldsymbol{x}$ 为所有分量都是 1 的向量. 用 Cholesky 分解算法求解方程 $\boldsymbol{H_n x} = \boldsymbol{b}$，得到近似解 $\hat{\boldsymbol{x}}$，计算残差 $\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{H_n}\hat{\boldsymbol{x}}$ 和误差 $\Delta\boldsymbol{x} = \hat{\boldsymbol{x}} - \boldsymbol{x}$ 的 ∞-范数.

(1) 设 n=10，计算 $||\boldsymbol{r}||_\infty$、$||\Delta\boldsymbol{x}||_\infty$.

(2) 在右端项上施加 10^{-7} 的扰动然后解方程组，观察残差和误差的变化情况.

(3) 改变 n 的值为 8 和 12，求解相应的方程，观察 $||\boldsymbol{r}||_\infty$、$||\Delta\boldsymbol{x}||_\infty$ 的变化情况. 通过实验说明了什么问题？

注：Hilbert 矩阵 $\boldsymbol{H_n}$ 的定义如下：

$$\boldsymbol{H_n} = \begin{bmatrix} 1 & \frac{1}{2} & \cdots & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{n} & \frac{1}{n+1} & \cdots & \frac{1}{2n-1} \end{bmatrix} \tag{1}$$

2 解题思路

H_n 是一个正定实对称矩阵，故可以进行 Cholesky 分解。求解以该矩阵为系数的方程时，可以使用 Cholesky 分解算法将矩阵 H_n 分解为 LL^T 的形式，其中 L 是一个下三角矩阵。

通过 Cholesky 分解，原方程变为 $LL^Tx = b$。先求解 $Ly = b$，再求解 $L^Tx = y$，得到 $\hat{x}$。进而计算残差 $r = b - H_n\hat{x}$ 和误差 $\Delta x = \hat{x} - x$。

在右端项 b 增加 10^{-7} 的扰动重新求解方程，计算残差和误差。

3 实验结果

算法运行的结果如下：

(1) ORIGIN: $n = 10, ||r|| = 2.2204 * 10^{-16}, ||\Delta x|| = 4.4459 * 10^{-4}$

(2) INTERFERENCE: $n = 10, ||r|| = 4.4409 * 10^{-16}, ||\Delta x|| = 7.0071 * 10^{-1}$

(3) ORIGIN: $n = 8, ||r|| = 2.2204 * 10^{-16}, ||\Delta x|| = 4.1154 * 10^{-7}$
INTERFERENCE: $n = 8, ||r|| = 2.2204 * 10^{-16}, ||\Delta x|| = 2.1622 * 10^{-2}$
ORIGIN: $n = 12, ||r|| = 4.4409 * 10^{-16}, ||\Delta x|| = 3.3581 * 10^{-1}$
INTERFERENCE: $n = 12, ||r|| = 4.4409 * 10^{-16}, ||\Delta x|| = 2.3620 * 10^{1}$

通过实验结果我们发现，扰动项并不会影响残差 ||r||，这是因为扰动项出现之后，求解的方程也相应的变化。但是我们发现，添加扰动项后误差 || x|| 突增，这是因为 Hilbert 是病态矩阵，条件数随 n 的增大而剧烈增大，微小的因变量扰动会造成自变量的巨大扰动。随着 n 的增加，残差和误差的范数都会跟着增加，其条件数越来越大，矩阵越来越病态，求解相应方程问题的稳定性越来越差。所以在实际求解过程中，我们应该考虑系数矩阵的病态性。如果系数矩阵过于病态，可以考虑在方程两侧同时乘上一个可逆矩阵来下调系数矩阵的病态性。

4 实验心得

通过本次实验对比 Hilbert 矩阵在 n=8, 10, 12 的不同计算结果，我加深了对问题病态性的理解，可以明显看出随着 n 的增大，问题的敏感性大大增强，输入微小的扰动将对结果产生巨大的影响，n=8 和 10 时，误差的范数基本很小，分别为 0.02 和 0.7，但 n=12 时，误差的范数达到了 23，问题已经很病态了。

同时，通过实验我还发现，病态矩阵线性方程组求解问题的病态是体现在问题的误差上而不是残差，这样的误差极大地影响我们对于准确解的求解，甚至造成工程上的失误。所以在面对数值问题求解的线性方程组的时候，我们必须认真的分析系数矩阵的条件数。

通过本次实验，我也加深了对 Cholesky 分解的理解，也自己编程实现了该分解算法和分解后求解方程的算法，由于使用 Cholesky 分解算法，右端项发生扰动时，不必重新计算 L，也显示了这种方式的优越性。

5 源代码

```
% 对矩阵A进行Cholesky分解，输入矩阵A，输出分解后的矩阵A
function [A] = cholesky(A)
n = size(A, 1);
% 将A提取为下三角矩阵
A = tril(A);
% 执行算法过程
for j = 1: n
   for k = 1: j - 1
      A(j, j) = A(j, j) - A(j, k) ^ 2;
   end
   A(j, j) = sqrt(A(j, j));
   for i = j + 1: n
      for k = 1: j - 1
         A(i, j) = A(i, j) - A(i, k) * A(j, k);
      end
      A(i, j) = A(i, j) / A(j, j);
   end
end
```

```
% 对Cholesky分解后的矩阵L解线性方程组，输入矩阵L，b，输出求解结果x
function [x] = solve_cholesky(L, b)
n = size(L, 1);
y = zeros(n, 1);
% 回代法解线性方程组
for i = 1: n
   y(i) = b(i);
   for j = 1: i - 1
      y(i) = y(i) - L(i, j) * y(j);
   end
   y(i) = y(i) / L(i, i);
end
x = zeros(n, 1);
for i = n: -1: 1
   x(i) = y(i);
   for j = n: -1: i + 1
      x(i) = x(i) - L(j, i) * x(j);
   end
   x(i) = x(i) / L(i, i);
end
```

```
% 生成n阶Hilbert矩阵，输入n，输出H
function [H] = hilbert(n)
% 便捷的生成方法
a = repmat(1: n, n, 1);
H = 1 ./ (a + a' - 1);
end
```

```
% 生成10阶Hilbert矩阵
n = 10;
H = hilbert(n);
x = ones(n, 1);
b = H * x;

% 使用Cholesky分解法解线性方程组，计算残差和误差
L = cholesky(H);
x_ = solve_cholesky(L, b);
r = b - H * x_;
delta_x = x_ - x;
fprintf('ORIGIN: n = %d, ||r|| = %.4e, ||delta_x|| = %.4e\n', n, max(abs(r)), max(abs(delta_x))
    );

% 增加扰动项，使用Cholesky分解法解线性方程组，计算残差和误差
b_ = b;
b_ = b_ + ones(n, 1) * 1e-7;
x_ = solve_cholesky(L, b_);
r = b_ - H * x_;
delta_x = x_ - x;
fprintf('INTERFERENCE: n = %d, ||r|| = %.4e, ||delta_x|| = %.4e\n', n, max(abs(r)), max(abs(
    delta_x)));

% 生成8阶Hilbert矩阵
n = 8;
H = hilbert(n);
x = ones(n, 1);
b = H * x;

% 使用Cholesky分解法解线性方程组，计算残差和误差
L = cholesky(H);
x_ = solve_cholesky(L, b);
r = b - H * x_;
delta_x = x_ - x;
fprintf('ORIGIN: n = %d, ||r|| = %.4e, ||delta_x|| = %.4e\n', n, max(abs(r)), max(abs(delta_x))
    );

% 增加扰动项，使用Cholesky分解法解线性方程组，计算残差和误差
b_ = b;
b_ = b_ + ones(n, 1) * 1e-7;
x_ = solve_cholesky(L, b_);
r = b_ - H * x_;
delta_x = x_ - x;
fprintf('INTERFERENCE: n = %d, ||r|| = %.4e, ||delta_x|| = %.4e\n', n, max(abs(r)), max(abs(
    delta_x)));

% 生成12阶Hilbert矩阵
```

```
n = 12;
H = hilbert(n);
x = ones(n, 1);
b = H * x;

% 使用Cholesky分解法解线性方程组，计算残差和误差
L = cholesky(H);
x_ = solve_cholesky(L, b);
r = b - H * x_;
delta_x = x_ - x;
fprintf('ORIGIN: n = %d, ||r|| = %.4e, ||delta_x|| = %.4e\n', n, max(abs(r)), max(abs(delta_x))
    );

% 增加扰动项，使用Cholesky分解法解线性方程组，计算残差和误差
b_ = b;
b_ = b_ + ones(n, 1) * 1e-7;
x_ = solve_cholesky(L, b_);
r = b_ - H * x_;
delta_x = x_ - x;
fprintf('INTERFERENCE: n = %d, ||r|| = %.4e, ||delta_x|| = %.4e\n', n, max(abs(r)), max(abs(
    delta_x)));
```

参考文献

[1] 数值分析与算法（第二版）, 喻文健, 2015.