实验三

张天乐 计96 2018011038

上机题6

实验内容

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

实验过程

编程生成 Hilbert 矩阵 H_n , 以及 n 维向量 $b = H_n x$, 其中 x 为所有分量都是 1 的向量。

In [ ]:

```
import numpy as np

def create_hilbert(n):
    a = np.zeros((n, n))
    for i in range(n):
        for j in range(n):
            a[i][j] = 1 / (i + j + 1)
    return a
```

实现 Cholesky 分解算法

In [ ]:

```
def cholesky(A, n):
    L = np.zeros_like(A)

    for j in range(n):
        sum = 0
        for k in range(j):
            sum += L[j][k] ** 2
        L[j][j] = np.sqrt(abs(A[j][j] - sum))
        for i in range(j + 1, n):
            sum = 0
            for k in range(j):
                sum += L[i][k] * L[j][k]
            L[i][j] = (A[i][j] - sum) / L[j][j]

    return L
```

求解方程 $H_n x = b : Ly = b$, $L^T x = y$

In [ ]:

```
def solve(L, b, n):
    y = np.zeros_like(b)

    for i in range(n):
        sum = 0
```

```
        for j in range(0, i):
            sum += L[i][j] * y[j]
        y[i] = (b[i] - sum) / L[i][i]

    x = np.zeros_like(b)

    for i in range(n - 1, -1, -1):
        sum = 0
        for j in range(n - 1, i, -1):
            sum += L[j][i] * x[j]
        x[i] = (y[i] - sum) / L[i][i]

    return x
```

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

In [ ]:
```
n = 10

# 生成 H, b
H = create_hilbert(n)
ones = np.ones(n)
b = np.dot(H, ones)

# Cholesky 分解
L = cholesky(H, n)

# 解方程
x = solve(L, b, n)

# 计算残差和误差
r = max(abs(b - np.dot(H, x)))
delta = max(abs(x - ones))
print("r = {}, delta = {}".format(r, delta))
```

```
r = 2.220446049250313e-16, delta = 9.652351543287807e-05
```

残差 $||r||_\infty = 2.220 \times 10^{-16}$，误差 $||\Delta x||_\infty = 9.652 \times 10^{-05}$

(2) 在右端项上施加大小为 10^{-7} 的随机扰动，然后再解上述方程组，观察残差和误差的变化情况

In [ ]:
```
x = solve(L, b + np.random.normal(0, 1e-7, n), n)
r = max(abs(b - np.dot(H, x)))
delta = max(abs(x - ones))
print('施加大小为 1e-7 的随机扰动:')
print("r = {}, delta = {}".format(r, delta))
```

```
施加大小为 1e-7 的随机扰动:
r = 9.789978694385582e-08, delta = 354951.5402634121
```

发现残差变化很小，但误差变化极大。说明 Hilbert 矩阵是病态的，符合课本上的说明。

(3)改变 n 的值为 8 和 12、14，求解相应的方程, 观察 $||r||_\infty$、$||\Delta x||_\infty$ 的变化情况

In [ ]:
```
def test(n):
    print('n = {}'.format(n))
```

```python
    # 生成 H, b
    H = create_hilbert(n)
    ones = np.ones(n)
    b = np.dot(H, ones)

    # Cholesky 分解
    L = cholesky(H, n)

    # 解方程，计算残差和误差
    x = solve(L, b, n)
    r = max(abs(b - np.dot(H, x)))
    delta = max(abs(x - ones))
    print("r = {}, delta = {}".format(r, delta))

    # 施加大小为 1e-7 的随机扰动
    x = solve(L, b + np.random.normal(0, 1e-7, n), n)
    r = max(abs(b - np.dot(H, x)))
    delta = max(abs(x - ones))
    print('施加大小为 1e-7 的随机扰动:')
    print("r = {}, delta = {}\n".format(r, delta))

test(8)
test(12)
test(14)
```

```
n = 8
r = 2.220446049250313e-16, delta = 3.735335705190579e-07
施加大小为 1e-7 的随机扰动:
r = 2.041973496957894e-07, delta = 721.4380426637857

n = 12
r = 4.440892098500626e-16, delta = 0.007003226172676014
施加大小为 1e-7 的随机扰动:
r = 2.289919280862307e-07, delta = 241646854.96826518

n = 14
r = 2.6989521728637555e-13, delta = 849.0754440508019
施加大小为 1e-7 的随机扰动:
r = 4.649670116718685e-05, delta = 123390334297.71236
```

观察发现残差 $||r||_\infty$ 都不大。误差 $||\Delta x||_\infty$ 非常大，并且 n 越大，误差越大。说明 Hilbert 矩阵是病态的，并且 n 越大，矩阵越病态。