线性方程组的直接解法

按照书上伪代码我实现了Cholesky算法

```python
def Cholesky(H_matrix_,n):
    H_matrix = H_matrix_.copy()
    for j in range(n):
        if j!=0:
            for k in range(j):
                H_matrix[j][j] -= H_matrix[j][k]**2
            H_matrix[j][j] = np.sqrt(H_matrix[j][j])
        for i in range(j+1,n):
            for k in range(j):
                H_matrix[i][j] -= H_matrix[i][k]*H_matrix[j][k]
            H_matrix[i][j] = H_matrix[i][j]/H_matrix[j][j]

    for i  in range(n):
        for j in range(i+1,n):
            H_matrix[i][j] = 0
    return H_matrix
```

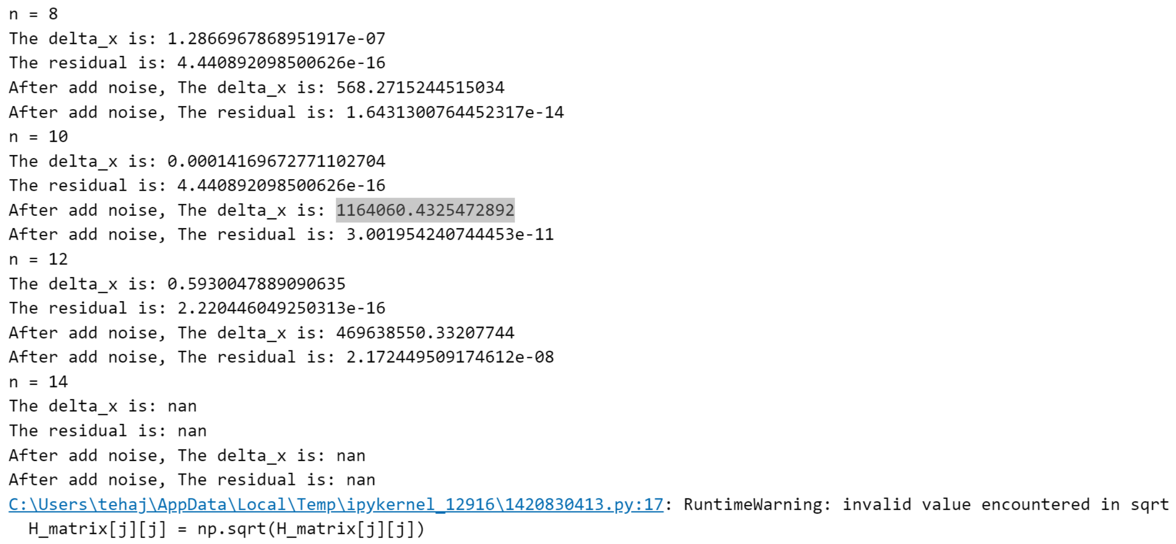
分解后，我将方程 $Hx=b$ 求解转化为，$LL^Tx=b$ 求解，也就是先求解 $Ly=b$，再求解 $L^Tx=y$，由于这些都是上三角和下三角矩阵，所以可以用课本中介绍的针对三角矩阵的求解方式求解。

```python
def solve_L(L,b,n):
    b_ = b.copy()
    for i in range(n):
        for j in range(i):
            b_[i] -= L[i][j]*b_[j]
        b_[i] = b_[i]/L[i][i]

    return b_

def solve_U(U,b,n):
    b_ = b.copy()
    for i in range(n-1,-1,-1):
        for j in range(i+1,n):
            b_[i] -= U[i][j]*b_[j]
        b_[i] = b_[i]/U[i][i]

    return b_
```

n = 10时，$||r||_\infty$ 和 $||\Delta x||_\infty$，分别为 4.440892098500626e-16 和 0.00014169672771102704

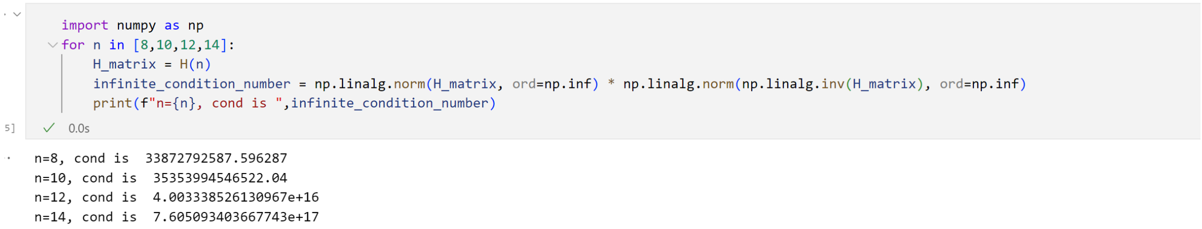
接下来，我对右端项引入了相对大小 10^{-7} 的随机扰动，重新求解方程，发现残差的无穷范数为 3.001954240744453e-11，仍然较小，而解的无穷范数变为 1164060.4325472892。这说明 $Hx=b$ 是病态的。

接下来我尝试了n=8，12，14， 结果如下



```
n = 8
The delta_x is: 1.2866967868951917e-07
The residual is: 4.440892098500626e-16
After add noise, The delta_x is: 568.2715244515034
After add noise, The residual is: 1.6431300764452317e-14
n = 10
The delta_x is: 0.00014169672771102704
The residual is: 4.440892098500626e-16
After add noise, The delta_x is: 1164060.4325472892
After add noise, The residual is: 3.001954240744453e-11
n = 12
The delta_x is: 0.5930047889090635
The residual is: 2.220446049250313e-16
After add noise, The delta_x is: 469638550.33207744
After add noise, The residual is: 2.172449509174612e-08
n = 14
The delta_x is: nan
The residual is: nan
After add noise, The delta_x is: nan
After add noise, The residual is: nan
C:\Users\tehaj\AppData\Local\Temp\ipykernel_12916\1420830413.py:17: RuntimeWarning: invalid value encountered in sqrt
  H_matrix[j][j] = np.sqrt(H_matrix[j][j])
```

可以看到，n=8，10，12时，加入扰动后 x 的偏差都会显著增大，而且n越大这个偏差的绝对数值也会越大，但残差的误差一直比较小，这进一步说明了问题的病态性，而且这种病态性会随着n的增加而增加（H矩阵的条件数增加）



```
import numpy as np
for n in [8,10,12,14]:
    H_matrix = H(n)
    infinite_condition_number = np.linalg.norm(H_matrix, ord=np.inf) * np.linalg.norm(np.linalg.inv(H_matrix), ord=np.inf)
    print(f"n={n}, cond is ",infinite_condition_number)
```

```
n=8, cond is  33872792587.596287
n=10, cond is  35353994546522.04
n=12, cond is  4.003338526130967e+16
n=14, cond is  7.605093403667743e+17
```

而n=14时，由于问题过于病态，导致了溢出现象，问题无法求解。