

数值实验报告 4

实验名称	高斯消去法与三角分解法的实现				实验时间	2025 年 3 月 21 日	
姓名	秦浩政 郭凯平 刘桂凡 刘佳鑫	班级	数据科学 2301	学号	2306030214 2306020510 2309050116 2309050117	成绩	

一、实验目的，内容

目的：掌握高斯消去法与三角分解法的原理以及实现

内容：1.实现高斯消去法求解线性方程组，通过消元与回代过程求出方程的解。

2.实现三角分解法，通过将矩阵分解为下三角矩阵 L 与上三角矩阵 U，再进行前代和回代求解方程组。

3.通过示例验证算法的正确性，分析误差来源以及优化方法。

二、算法描述

1.高斯消去法

（1）消元：对方程组的增广矩阵进行初等行变换，逐列将主元下方的元素消为 0，最后形成上三角矩阵。

（2）回代：从最后一行开始，依次对方程进行求解。
$$x_i = (b_i(i) - \sum_{j=i+1}^n a_{ij}(i)x_j) / a_{ii}$$

2.三角分解法

（1）分解：将系数矩阵 A 分解为下三角矩阵 L 和上三角矩阵 U，满足 A = LU。

（2）求解：通过前代法求解 Ly = b 得 y，再通过回代法求解 Ux = y 得 x。

三、程序代码

1. 高斯消去法

```
import numpy as np

1 usage
def gauss(A, b):
    n = len(b)
    # 构造增广矩阵
    a = np.hstack((A.astype(float), b.reshape(-1, 1).astype(float)))

    for k in range(n-1):
        for i in range(k+1, n):
            p = a[i, k] / a[k, k]
            a[i, k:] = a[i, k:] - p*a[k, k:]

    x = np.zeros(n)
    for i in range(n-1, -1, -1):
        x[i] = (a[i, -1] - np.dot(a[i, i+1:n], x[i+1:n])) / a[i, i]
    # 判断是否为整数或者浮点数
    y = [int(num) if num.is_integer() else round(num, 2) for num in x]
    return y

if __name__ == "__main__":
    A = np.array([[1, -2, 2],
                  [2, -3, -3],
                  [4, 1, 6]])
    b = np.array([-2, 4, 3])
    x = gauss(A, b)
    print("高斯消去法解得：", x)
```

2.三角分解法

```
def LU(A):
    n = len(A)
    L = [[0.0] * n for i in range(n)]
    U = [[0.0] * n for i in range(n)]

    for k in range(n):
        # 计算 U 的第 k 行
        for j in range(k, n):
            U[k][j] = A[k][j]
            for m in range(k):
                U[k][j] = U[k][j] - L[k][m] * U[m][j]

        # 计算 L 的第 k 列
        L[k][k] = 1.0
        for i in range(k+1, n):
            L[i][k] = A[i][k]
            for m in range(k):
                L[i][k] = L[i][k] - L[i][m] * U[m][k]
            L[i][k] = L[i][k] / U[k][k]

    return L, U

1 usage
def sanjiaofenjie(L, U, b):
    n = len(b)
    y = [0.0] * n
    x = [0.0] * n

    # 前代解 Ly = b
```

```
23 def sanjiaofenjie(L, U, b):
30     y[i] = b[i]
31     for j in range(i):
32         y[i] = y[i] - L[i][j] * y[j]
33     # 回代解 Ux = y
34     for i in range(n-1, -1, -1):
35         x[i] = y[i]
36         for j in range(i+1, n):
37             x[i] = x[i] - U[i][j] * x[j]
38         x[i] = x[i] / U[i][i]
39
40     x = [round(num) if num.is_integer() else round(num, 2) for num in x]
41     return x
42
43 if __name__ == "__main__":
44     A = [
45         [1, -2, 2],
46         [2, -3, -3],
47         [4, 1, 6]
48     ]
49     b = [-2, 4, 3]
50     # 进行 LU 分解
51     L, U = LU(A)
52     print("L 矩阵: ")
53     for i in L:
54         print([f"{x}" for x in i])
55     print("\nU 矩阵: ")
56     for i in U:
57         print([f"{x}" for x in i])
58     # 求解方程
59     x = sanjiaofenjie(L, U, b)
60     print("三角分解法解得: ", x)
```

四. 数值结果

1. 高斯消去法

```
C:\Users\21567\python\my projects\python project\venv\scripts\python.exe C:\Users\21567\python\my projects\python project\venv\scripts\python.exe
高斯消去法解得:  [2, 1, -1]

进程已结束，退出代码为 0
```

2.三角分解法

```
C:\Users\21507\PycharmProjects\pythonProject\.venv\Scripts\python.exe C:\Users\21507\Pychar
L 矩阵:
['1.0', '0.0', '0.0']
['2.0', '1.0', '0.0']
['4.0', '9.0', '1.0']

U 矩阵:
['1', '-2', '2']
['0.0', '1.0', '-7.0']
['0.0', '0.0', '61.0']
三角分解解得:  [2, 1, -1]

进程已结束，退出代码为 0
```

五．计算结果分析

正确性分析：两种算法通过示例均得到正确解[2, 1, -1]，与理论计算一致。
高斯消去法对矩阵中的主元要求较高，主元过小容易导致舍入误差。
三角分解法则需要保证主元非零，可以通过列主元分解进而提高稳定性。

六．计算中出现的问题，解决方法及体会

- 问题：1.初始代码未处理主元为零的情况，导致矩阵奇异错误。
2.浮点数精度问题导致求解过程中的小数部分出现微小误差。
- 解决方法：1.添加主元选择判断，避免主元为零的情况。
2.使用四舍五入或者符号处理优化最终结果。
- 体会：1.高斯分解法适用于中小规模方程组，但需要注意数值稳定性。
2.三角分解法适用于多次求解。
3.在实际应用中需要结合矩阵的特性选择合适的算法。

教师评语	指导教师：年 月 日
------	------------

