数值分析上机报告
第三章

院 (系) 名 称：　微电子学院

学 生 姓 名：　周玉乾

学　　号：　220205764

二〇二〇年十一月

第三章

一、 问题

列主元 Gauss 消去

对于某电路的分析，归结于求解线性方程组 $RI = V$，其中

$$R = \begin{bmatrix} 31 & -13 & 0 & 0 & 0 & -10 & 0 & 0 & 0 \\ -13 & 35 & -9 & 0 & -11 & 0 & 0 & 0 & 0 \\ 0 & -9 & 31 & -10 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -10 & 79 & -30 & 0 & 0 & 0 & -9 \\ 0 & 0 & 0 & -30 & 57 & -7 & 0 & -5 & 0 \\ 0 & 0 & 0 & 0 & -7 & 47 & -30 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -30 & 41 & 0 & 0 \\ 0 & 0 & 0 & 0 & -5 & 0 & 0 & 27 & -2 \\ 0 & 0 & 0 & -9 & 0 & 0 & 0 & -2 & 29 \end{bmatrix}$$

$$V^T = (-15, 27, -23, 0, -20, 12, -7, 7, 10)^T$$

1. 编制解 n 阶线性方程组 $Ax = b$ 的列主元 Gauss 消去的通用程序；
2. 用所编程序解线性方程组 $RI = V$，并打印出解向量，保留 5 位有效数字；
3. 本题编程之中，你提高了哪些能力?

二、 分析

列主元 Gauss 消去的算法流程图如图 3.1 所示。

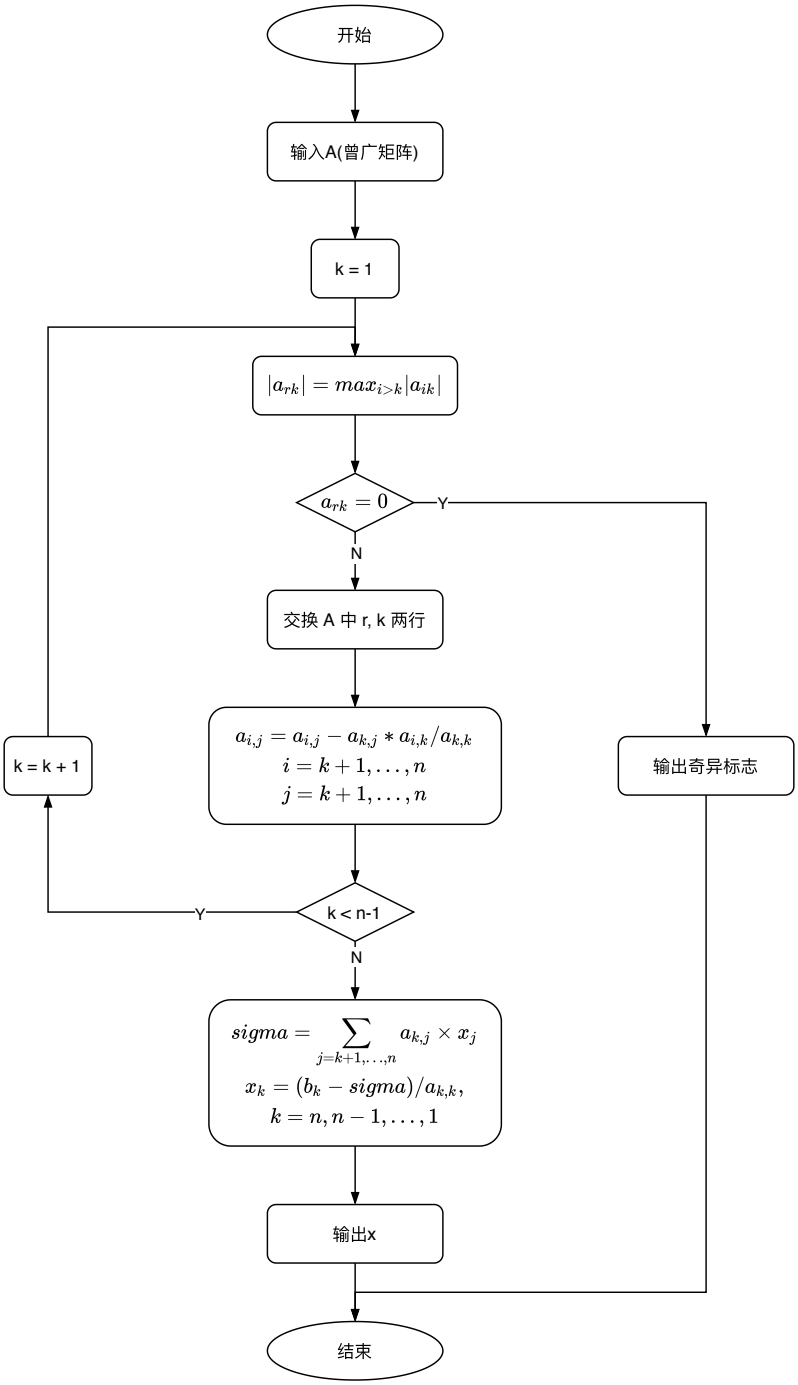

图 3.1　列主元 Gauss 消去算法流程

三、　程序

```
1 import sys
2 import numpy as np
3
4 A = [[31, -13, 0, 0, 0, -10, 0, 0, 0],
5     [-13, 35, -9, 0, -11, 0, 0, 0, 0],
```

```
    [0, -9, 31, -10, 0, 0, 0, 0, 0],
    [0, 0, -10, 79, -30, 0, 0, 0, -9],
    [0, 0, 0, -30, 57, -7, 0, -5, 0],
    [0, 0, 0, 0, -7, 47, -30, 0, 0],
    [0, 0, 0, 0, 0, -30, 41, 0, 0],
    [0, 0, 0, 0, -5, 0, 0, 27, -2],
    [0, 0, 0, -9, 0, 0, 0, -2, 29]]

b = [-15, 27, -23, 0, -20, 12, -7, 7, 10]

def find_shape(A, b):
    """
    获取 N
    """

    # N行 M列
    n1, m1 = A.shape
    n2, = b.shape

    # PRINT(N1, M1)
    # PRINT(N2)

    # 判断矩阵形状
    if n1 != m1 or n1 != n2:
        print('Error martix shape!')
        sys.exit()
    else:
        N = n1

    return N


def MGauss(A, b):
    """
    列主元 GAUSS 消去 消元
    """

    # 判断矩阵形状
    N = find_shape(A, b)

    # 列主元高斯消元
    for k in range(0, N):
        p = k
        maxabs = abs(A[k, k])
        # 找列最大值
        for i in range(k+1, N):
            if abs(A[i, k]) > maxabs:
                p = i
```

```
54                maxabs = abs(A[i, k])
55        print('maxabs', maxabs)
56        # 最大值为 0
57        if maxabs == 0:
58            print('Singular')
59            sys.exit()
60        # 最大值不在对角线，则交换两行
61        if p != k:
62            A[[p,k],:] = A[[k,p],:]
63            b[[p,k]] = b[[k,p]]
64        print('exchange r{0} and r{1}:\r\n'.format(k, p), A)
65        # 消元，将对角线以下变为 0
66        for i in range(k+1, N):
67            m_ik = A[i, k] / A[k, k]
68            for j in range(0, N):
69                A[i, j] -= A[k, j] * m_ik
70            b[i] -= b[k] * m_ik
71        print('After Elimination:\r\n', np.concatenate((A,np.asarray([b]).T), axis =
              1))
72
73    if A[N-1, N-1] == 0:
74        print('Singular')
75        sys.exit()
76
77    return A, b
78
79 def bring_back(A, b):
80    """
81    列主元 Gauss 消去 回带
82    """
83
84    # 判断矩阵形状
85    N = find_shape(A, b)
86
87    # 回带
88    X = np.zeros(N)
89    X[N-1] = b[N-1] / A[N-1, N-1]
90    for i in range(0, N-1):
91        k = N-2-i
92        sigma = sum(A[k, j]*X[j] for j in range(k+1, N))
93        # print('k:{0}, sigma:{1}'.format(k, sigma))
94        X[k] = (b[k] - sigma) / A[k, k]
95
96    return X
97
98 def main():
99    """
100   main
```

```
    """

    A_np = np.asarray(A, dtype = float)
    b_np = np.asarray(b, dtype = float)

    # B_NP = B_NP.T

    print('A:\r\n', A_np)
    print('b:\r\n', b_np)

    A_G, b_G = MGauss(A_np, b_np)
    x_G = bring_back(A_G, b_G)

    print('A_G:b_G\r\n', np.concatenate((A_G,np.asarray([b_G]).T), axis = 1))
    print('x_G', x_G)


if __name__ == "__main__":
    main()
```

四、 算例

$$A = \begin{bmatrix} 31 & -13 & 0 & 0 & 0 & -10 & 0 & 0 & 0 \\ -13 & 35 & -9 & 0 & -11 & 0 & 0 & 0 & 0 \\ 0 & -9 & 31 & -10 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -10 & 79 & -30 & 0 & 0 & 0 & -9 \\ 0 & 0 & 0 & -30 & 57 & -7 & 0 & -5 & 0 \\ 0 & 0 & 0 & 0 & -7 & 47 & -30 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -30 & 41 & 0 & 0 \\ 0 & 0 & 0 & 0 & -5 & 0 & 0 & 27 & -2 \\ 0 & 0 & 0 & -9 & 0 & 0 & 0 & -2 & 29 \end{bmatrix}$$

$$b^T = (-15, 27, -23, 0, -20, 12, -7, 7, 10)^T$$

运算结果：

```
x_G [-0.28923382 0.34543572 -0.71281173 -0.22060851 -0.43040043 0.15430874
    -0.05782287 0.20105389 0.29022866]
```

使用 MATLAB 自带的求解线性方程组的方法求解，验证编写算法的正确性：

```
>> A \ b'

```

```
ans =

   -0.2892
    0.3454
   -0.7128
   -0.2206
   -0.4304
    0.1543
   -0.0578
    0.2011
    0.2902
```

可以看到结果是一致的。

五、 结论

1. 列主元 Gauss 消去法避免了小数作除数，因此一般能保证舍入误差不增大，这个方法基本上是稳定的；
2. MATLAB 中可用 `A \ b` 求线性方程组解。