实验二 *Gauss* 消元法

与 *Gauss-Jordan* 消元法

实验报告

课　　程	计算方法与 MATLAB
院（系）	计算机科学与技术学院
专　　业	人工智能
级　　别	2021 级
学　　号	2115102001
姓　　名	安炳旭
指导老师	陈叶旺

2023 年 12 月 1 日

实验二　*Gauss* 消元法与 *Gauss-Jordan* 消元法

1　算法思想

1.1　*Gauss* 消元法简介

对于方程$A\mathbf{x}=\mathbf{b}$而言, 其中$A\in\mathbb{R}^{m\times n}$,$\mathbf{x}\in\mathbb{R}^{n}$,$\mathbf{b}\in\mathbb{R}^{m}$,同时若矩阵$A$可逆，则方程有唯一解。一般而言，我们可以写出其增广矩阵：

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} & b_m \end{bmatrix}$$

Gauss 消元法的思想是，将线性方程组$A\mathbf{x}=\mathbf{b}$转换为同解的上三角形方程组，如(1)式所示：

$$U\mathbf{x}=L^{-1}\mathbf{b} \tag{1}$$

其中 U 为上三角矩阵，L 为下三角矩阵。因此，需要进行消元的过程，将矩阵转换为上三角矩阵，之后再回代，即可得到方程组的解。

第 1 步：第 i 行-第 1 行$\times\dfrac{a_{i1}}{a_{11}}$,$i=2,\cdots,n$

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{pmatrix} \to \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ 0 & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} & b_2^{(1)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)} & b_n^{(1)} \end{pmatrix} \tag{2}$$

第 k 步：第 i 行-第 k 行$\times\dfrac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}}$,$i=k+1,\cdots,n$

$$\begin{pmatrix} a_{l,1}^{(0)} & \cdots & a_{l,k}^{(0)} & a_{l,k+1}^{(0)} & \cdots & a_{l,n}^{(0)} & b_1^{(0)} \\ & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ & & a_{k,k}^{(k-1)} & a_{k,k+1}^{(k-1)} & \cdots & a_{k,n}^{(k-1)} & b_k^{(k-1)} \\ & & a_{k,k+1}^{(k-1)} & a_{k+1,k+1}^{(k-0)} & \cdots & a_{k+1,n}^{(k-1)} & b_{k+1}^{(k-1)} \\ & & \vdots & \vdots & \ddots & \vdots & \vdots \\ & & a_{n,k}^{(k-1)} & a_{n,k+1}^{(k-1)} & \cdots & a_{n,n}^{(k-1)} & b_n^{(k-1)} \end{pmatrix} \to \begin{pmatrix} a_{1,1}^{(0)} & \cdots & a_{1,k}^{(0)} & a_{1,k+1}^{(0)} & \cdots & a_{1,n}^{(0)} & b_1^{(0)} \\ & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ & & a_{k,k}^{(k-1)} & a_{k,k+1}^{(k-1)} & \cdots & a_{k,n}^{(k-1)} & b_k^{(k-1)} \\ & & & a_{k+1,k+1}^{(k)} & \cdots & a_{k+1,n}^{(k)} & b_{k+1}^{(k)} \\ & & & \vdots & \ddots & \vdots & \vdots \\ & & & a_{n,k+1}^{(k)} & \cdots & a_{n,n}^{(k)} & b_n^{(k)} \end{pmatrix} \tag{3}$$

如此进行$n-1$次消元，最后得到同解方程组如(4)式所示：

$$\begin{pmatrix} a_{11}^{(0)} & a_{12}^{(0)} & \cdots & a_{1n}^{(0)} \\ & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} \\ & & \ddots & \vdots \\ & & & a_{nn}^{(n-1)} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2^{(1)} \\ \vdots \\ b_n^{(n-1)} \end{pmatrix} \tag{4}$$

回代，有

$$x_k=\frac{b_k^{(k-1)}-a_{k,k+1}^{(k-1)}x_{k+1}-\cdots-a_{kn}^{(k-1)}x_n}{a_{kk}^{(k-1)}}\quad k=n,n-1,...,1 \tag{5}$$

上述过程即为 Gauss 消元法的全过程，高斯消元法的复杂度大约是$O(\frac{1}{3}n^2)$.

1.2　*Gauss-Jordan* 消元法简介

与一般的 *Gauss* 消元法相比，*Gauss-Jordan* 消元法是一种无回代的算法。线性方程组经过 *k*-1 次消元后得到：

$$\begin{pmatrix} a_{1,1}^{(0)} & \cdots & a_{l,k}^{(0)} & a_{l,k+1}^{(0)} & \cdots & a_{l,n}^{(0)} & b_1^{(0)} \\ & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ & & a_{k,k}^{(k-1)} & a_{k,k+1}^{(k-1)} & \cdots & a_{k,n}^{(k-1)} & b_k^{(k-1)} \\ & & a_{k,k+1}^{(k-1)} & a_{k+1,k+1}^{(k-0)} & \cdots & a_{k+1,n}^{(k-1)} & b_{k+1}^{(k-1)} \\ & & \vdots & \vdots & \ddots & \vdots & \vdots \\ & & a_{n,k}^{(k-1)} & a_{n,k+1}^{(k-1)} & \cdots & a_{n,n}^{(k-1)} & b_n^{(k-1)} \end{pmatrix}$$

第 *k* 步消元：首先从 $a_{ik}^{(k)}$ $(i=k,k+1,...,n)$ 中选取主元，即 $|a_{i_k k}|=max_{k\le i\le n}|a_{ik}|$,并记$l=i_k$,若$l\neq k$,则交换 *k* 与 *l* 行，这时新的$a_{kk}^{(k)}\neq 0$时，使用新的$a_{kk}^{(k)}$除以第 *k* 行，即

$$\begin{cases} a_{kj}^{(k+1)}=a_{kj}^{(k)}/a_{kk}^{(k)} \\ b_k^{(k+1)}=b_k^{(k)}/a_{kk}^{(k)} \end{cases}\ (j=k,k+1,...n)$$

再用－$a_{kk}^{(k)}$乘第 *k* 行加到第 *i* 行$(i=1,2,...,n,i\neq k)$,即

$$\begin{cases} a_{ij}^{(k+1)}=a_{ik}^{(k)}-a_{ik}^{(k)}a_{kj}^{(k+1)}(i=1,2,...,n,i\neq k,j=k \\ b_i^{(k+1)}=b_i^{(k)}-a_{ik}^{(k)}b_k^{(k+1)}\quad (i=1,2,...,n,i\neq k) \end{cases}$$

如此进行了 *n* 步消元，最后得到的解为

$$\boldsymbol{x}=(b_1^{(n+1)},b_2^{(n+1)},...,b_n^{(n+1)})^T$$

2　算法步骤

本次实验需要使用 MATLAB 来具体实现这两个算法，根据第一部分的介绍，很容易能够得到这两个算法的伪代码，首先来分析 *Gauss* 顺序消元法的伪代码，详情如表 1 所示：

表 1 *Gauss-Jordan* 消元法伪代码

输入：

方程矩阵 A 系数矩阵 b

输出：解向量 x

For $k = 1, 2, \ldots, n$, Do
  For $i = k+1, \ldots, n$, Do
    $a_{ik} = a_{ik}/a_{kk}$;
  Enddo
  For $j = k+1, \ldots, n$, Do
    For $i = k+1, \ldots, n$, Do
      $a_{ij} = a_{ij} - a_{ik}a_{kj}$;
    Enddo
  Enddo
Enddo

这种方法的主要特点是，每一行的消元操作都会影响下一行，因此在每一步迭代中，都需要使用当前行的信息来影响下一行。

下面再来分析 *Gauss-Jordan* 消元法的伪代码，详情如表 2 所示：

表 2 Gauss-Jordan 消元法伪代码

输入：增广矩阵 [A | I]，其中 I 是单位矩阵

输出：矩阵 A 的右侧部分，即解向量

1. For $k = 1, 2, \ldots, n$, Do
2.   交换 $\mathbb{A}$ 的第 k 行和第 p_k 行，其中 p_k 为列主元；
3. $a_{kk} = 1/a_{kk}$;
4.   For $i = 1, \ldots, n$ 且 $i \neq k$, Do $a_{ik} := -a_{ik}a_{kk}$; Enddo
5.   For $i = 1, \ldots, n$ 且 $i \neq k$, Do
6.   For $j = 1, \ldots, n$ 且 $j \neq k$, Do
7. $a_{ij} := a_{ij} + a_{ik}a_{kj}$;
8.   Enndo
9.   Enddo
10.   For $j = 1, \ldots, n$ 且 $j \neq k$, Do $a_{kj} := a_{kk}a_{kj}$; Enndo
11. Enddo
12. For $k = n, n-1, \ldots, 1$, Do
13.   交换 $\mathbb{A}$ 的第 k 列和第 p_k 列；
14. Enddo

这段伪代码实现了高斯-约当消元法的主要步骤，代码包含三步基本操作：计算同列的消元乘子（3－4 行），执行其余各列的消元（5－9 行），进行同行的单位化操作（10 行）。这些步骤将矩阵 A 转化为行梯阵形式，从而可以求解线性

方程组或求解矩阵的逆矩阵。

3 代码实现

本次实验代码及其功能如下表所示，具体功能请参考注释与第二部分的伪代码：

main.m

```
clc;clear;
% Example
A=input("请输入系数方阵 A:");
b=input("请输入常数向量 b:");

solution1 = gaussian_elimination(A, b);
solution2=GaussJordan(A,b);
disp('使用 Gauss 消元法的方程组的解为:');
disp(solution1);
disp('使用 Gauss-Jordan 消元法的方程组的解为:');
disp(solution2);
```

gaussian_elimination.m

```
function x = gaussian_elimination(A, b)
    n = length(b);
    % 消元过程
    for k = 1:n-1
        for i = k+1:n
            if A(k,k) == 0
                error('主对角元素错误! ');
            else
                L_ik = A(i,k) / A(k,k);
                for j = k+1:n
                    A(i,j) = A(i,j) - L_ik * A(k,j);
                end
                b(i) = b(i) - b(k) * L_ik;
            end
        end
    end
    % 回代过程
    x = zeros(n, 1);
    for i = n:-1:1
        x(i) = b(i);
```

```
        for j = i+1:n
            x(i) = x(i) - A(i,j) * x(j);
        end
        x(i) = x(i) / A(i,i);
    end
end
```

GaussJordan.m

```
function x = GaussJordan(A, b)
    matrix = [A, b];
    n = length(matrix(:, end));
    for i = 1 : n
        % 找第 i 列最大值
        [max_col, position_to_i] = max(matrix(i:end, i));
        if max_col == 0
            error('主对角元是 0，无唯一解，无法用高斯主列消元法解');
        end
        % 交换最大元列的行
        max_col_rowposition = position_to_i + i - 1;
        if max_col_rowposition ~= i
            temp = matrix(i, :);
            matrix(i, :) = matrix(max_col_rowposition, :);
            matrix(max_col_rowposition, :) = temp;
        end
        % 主对角元化 1
        matrix(i, :) = matrix(i, :) / matrix(i, i);
        % i+1 行后全部行消元
        for j = (i + 1) : n
            scale = matrix(j, i) / matrix(i, i);
            matrix(j, :) = matrix(j, :) - scale * matrix(i, :);
        end
    end
    % 后替换法
    for i = n : -1 : 1
        for j = 1 : i - 1
            matrix(j, :) = matrix(j, :) - matrix(j, i) * matrix(i, :);
        end
    end
    x = matrix(:, end);
end
```

4　实验结果

实验二　*Gauss* 消元法与 *Gauss-Jordan* 消元法

为了验证算法的正确性，此次实验选取了课后习题 3.1，3.2 与 3.4 进行检验，三题题目如下所示：

3.1 $$\begin{bmatrix} 2 & 4 & 3 \\ 3 & 5 & 1 \\ 5 & 3 & 2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}=\begin{bmatrix} 9 \\ 4 \\ 13 \end{bmatrix}$$

3.2 $$\begin{bmatrix} 2 & 1 & -3 \\ 3 & 4 & 5 \\ 6 & -2 & 3 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}=\begin{bmatrix} 5 \\ 12 \\ 17 \end{bmatrix}$$

3.5 $$\begin{bmatrix} 0.7290 & 0.8100 & 0.9000 \\ 1.0000 & 1.0000 & 1.0000 \\ 1.3310 & 1.21000 & 1.10000 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}=\begin{bmatrix} 0.6867 \\ 0.8338 \\ 1.0000 \end{bmatrix}$$

使用上述 MATLAB 代码所得到的结果如表 3-表 5 所示：

表 3 使用 *Gauss* 消元法和 *Gauss-Jordan* 消元法求解 3.1 的结果

```
>请输入系数方阵 A:[10,-2,-1;-2,10,-1;-1,-2,5]

>请输入常数向量 b:[3;15;10]

使用 Gauss 消元法的方程组的解为:

    1

    2

    3

使用 Gauss-Jordan 消元法的方程组的解为:

    1

    2

    3
```

表 4 使用 *Gauss* 消元法和 *Gauss–Jordan* 消元法求解 3.2 的结果

```
>请输入系数方阵 A:[2,4,3;3,5,1;5,3,2]

>请输入常数向量 b:[9;4;13]

使用 Gauss 消元法的方程组的解为:

    2

    -1

    3
```

```
使用 Gauss-Jordan 消元法的方程组的解为:

    2.0000

    -1.0000

    3.0000
```

表 5 使用 *Gauss* 消元法和 *Gauss−Jordan* 消元法求解 3.5 的结果

```
>请输入系数方阵 A:[0.7290,0.8100,0.9000;1.0000,1.0000,1.0000;1.3310,1.2100,1.1000]

>请输入常数向量 b:[0.6867;0.8338;1.0000]

使用 Gauss 消元法的方程组的解为:

    0.2245

    0.2814

    0.3279

使用 Gauss-Jordan 消元法的方程组的解为:

    0.2245

    0.2814

    0.3279
```

可以发现，两种消元方法均能够解出正确答案，因此基本可以验证程序的正确性。

5 实验结论

最后，给出 *Gauss* 消元法和 *Gauss-Jordan* 消元法的比较，作为本实验的结论部分：

- *Gauss* 消元法

  优点：

  1. *Gauss* 消元法是一种直观且容易理解的线性代数求解方法，它通过矩阵变换的方式将线性方程组转化为上三角矩阵，然后通过回代求解。
  2. 在实际应用中，对于一些小规模的线性方程组，Gauss 消元法是一个稳定可靠的求解方法，可以精确得到解。

  缺点：

  1. 主元为 0 则无法计算

2. 计算复杂度： 随着线性方程组规模的增大，Gauss 消元法的计算复杂度呈现出$O(\frac{1}{3}n^3)$的数量级，对于大规模矩阵，计算开销较大。

➢ *Gauss-Jordan* 消元法：

优点：

✧ 没有回代过程，消元完毕即可获得解答。

缺点：

✧ 计算复杂度： 与 Gauss 消元法一样，Gauss-Jordan 消元法在计算上的复杂度也是$O(\frac{1}{2}n^3)$，对于大规模矩阵同样存在计算开销较大的问题。