深 圳 大 学 实 验 报 告

实验课程名称：最优化方法

实验项目名称：线性方程组求解

学院：计算机与软件学院 专业：软件工程（腾班）

报告人：黄亮铭 学号：**2022155028** 班级：腾班

同组人：

指导教师：李炎然

实验时间：**2024** 年 **9** 月 **7** 日

实验报告提交时间：**2024** 年 **10** 月 **5** 日

教务处制

实验报告包含内容

一、 实验目的与要求

1. 掌握高斯消元法求解线性方程组；
2. 掌握 Matlab 编写代码，熟悉其画图工具，进行实验，并验证结果；
3. 锻炼数学描述能力，提高报告的书写能力。

二、 问题

已知向量$x,b\in\mathbb{R}^{4}$，矩阵$A\in\mathbb{R}^{4\times4}$，线性方程组$Ax=b$，其中：

$$A=\begin{bmatrix}1&1&2&1\\1&2&0&1\\1&4&2&1\\1&8&2&4\end{bmatrix},b=\begin{bmatrix}2\\0\\2\\3\end{bmatrix}.$$

(1) 编写代码，通过高斯消元法求解,并验证解的准确性。

(2) 是否可以通过高斯消元法，构造线性方程组求矩阵A的逆？如果可以求其逆矩阵。

(3) 如果线性方程组$Bx=b$，高斯法消元过程中如何提高解的准确性？并求其解。矩阵B定义为：

$$B=\begin{bmatrix}0.0001&1&2&1\\1&2&0&1\\1&4&2&1\\1&8&2&0.0004\end{bmatrix}.$$

(4) 通过附件A_b.mat的数据，验证你实现算法的准确性。

三、模型建立及求解

解决问题思路，算法实现描述，误差分析，存在问题等方面进行阐述；代码不要放在报告里面，作为附件提交！实验报告内容要有条理，叙述清楚，图文结合，公式正确，书写符合中文习惯！

问题 1 解决思路 I

利用高斯消元法直接求解线性方程组，具体步骤如下：

① 构造增广矩阵：将系数矩阵和常数向量合并成一个增广矩阵。

$$\begin{bmatrix}a_{11}&a_{12}&\cdots&a_{1m}&b_1\\a_{21}&\cdots&\cdots&a_{2m}&b_2\\\vdots&\ddots&\cdots&\vdots&\vdots\\a_{n1}&\cdots&\cdots&a_{nm}&b_n\end{bmatrix}$$

② 通过行变换，将矩阵转化为上三角行列式。过程中主要操作为加减消元，即利用第 i 行的 a[i][:]为基数，乘上可以消去 a[j][i]的系数，然后加到第 j 上。 其中，i<j。

③ 回代：增广矩阵被转换成上三角形式，就可以从最后一行开始，逐个解出每个未知数。将上三角矩阵的最后一行写成$a_{nm}x_n=b_n$的形式，解出 x_n。

然后，将 x_n 的值代入到前一行中，解出x_{n-1}，依此类推，直到解出所有未知数。

④ 检查解的存在性：如果在任何步骤中，主元为 0 且该行的常数项不为 0，那么方程组无解。如果主元为 0 且该行的常数项也为 0，那么方程组有无穷多解。

⑤ 显示解：最后，将所有解出的未知数值打印在命令行，就是线性方程组的解。

问题 1 算法实现

高斯消元法的算法实现具体思路如下：

① 预处理：1）将常数向量处理为行向量；2）获取系数矩阵和常数向量的行列数；3）将系数矩阵和常数向量合并成一个增广矩阵。

② 检查解的存在性：如果系数矩阵的行数少于列数，说明无解，直接返回。

③ 求解：按照解决思路的②③编写代码。

④ 验证解的准确性：1）利用公式$Ax = dev_b$然后检查dev_b和b之间的误差是否在允许的范围内，如果在允许的范围内则进行下一步，否则直接返回。2）利用 matlab 自带函数 linsolve()获得正确的解，然后用自己的解和正确的解对比，看误差是否则允许的范围内，处理同上。3）如果 1）和 2）均能满足，则说明解是正确的。

⑤ 输出解：利用 disp()函数输出解。

问题 1 算法运算结果

结果如图所示：其中线性方程组的解为自己编写的代码获得的解，正确的解是用 matlab 自带函数获得的解。算法运算结果说明编写的代码的正确性。

图 1：求解结果展示

问题 1 存在问题及误差分析

这种解决思路存在的问题为：1）若第 i 行第 i 列的系数恰好为 0，则无法计算，需要进行额外的行交换；2）受到计算误差的影响，特别是系数之间的数量级差距过大时。

当系数矩阵的系数之间数量级相差过大（超过 double 可以表示的精度）时，使用高斯消元法直接求解线性方程组会获得错误的解。

以计算精度为小数点后 3 位为例为例，$A = \begin{bmatrix} 0.0001 & 1 \\ 1 & 1 \end{bmatrix}$，$b = [2 \quad 1]$为例，按照上述解决思路，我们将 A 和 b 合并成增广矩阵$\begin{vmatrix} 0.0001 & 1 & 2 \\ 1 & 1 & 1 \end{vmatrix}$，然后行变换得到$\begin{vmatrix} 0.0001 & 1 & 2 \\ 0 & -1e4 & -1e4 \end{vmatrix}$，回代后会得到结果$x = [0 \quad 1]$。显然，这个答案是错误的。

上述例子就是误差带来的问题，如果不想受误差影响太大，我们需要改进算法，在后面我会提出解决的办法。

计算误差的方法：利用公式$Ax = dev_b$，得到dev_b。然后通过公式$\sum_i abs(dev_b - b)$即可计算出总的误差。

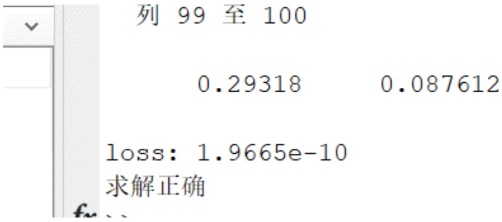


图 2：计算出的误差（A_b.mat）

问题 2 解决思路

将公式$AA^{-1} = I$展开可知

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & \cdots & \cdots & a_{2m} \\ \vdots & \ddots & \cdots & \vdots \\ a_{n1} & \cdots & \cdots & a_{nm} \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & \cdots & \cdots & a_{2n} \\ \vdots & \ddots & \cdots & \vdots \\ a_{m1} & \cdots & \cdots & a_{mn} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & \cdots & \vdots \\ 0 & \cdots & \cdots & 1 \end{bmatrix}$$

可以将上式看成是 n 个具有 m 个未知数的方程组写在一起。因此，我们分别对这 n 个线性方程组分别求解，然后将得到的答案合并成矩阵即可得到A^{-1}。

问题 2 算法实现

① 预处理前：检测矩阵 A 是否可逆，即是否满秩。构造主对角线系数为 1 其余系数为 0 的矩阵 I。

② 循环读取矩阵 I 的每一列。

③ 预处理：1）将常数向量处理为行向量；2）获取系数矩阵和常数向量的行列数；3）将系数矩阵和常数向量合并成一个增广矩阵。

④ 检查解的存在性：如果系数矩阵的行数少于列数，说明无解，直接返回。

⑤ 求解：按照问题 1 解决思路的②③编写代码。

⑥ 验证解的准确性：1）利用公式$Ax = dev_b$然后检查dev_b和b之间的误差是否在允许的范围内，如果在允许的范围内则进行下一步，否则直接返回。2）利用 matlab 自带函数 linsolve()获得正确的解，然后用自己的解和

正确的解对比，看误差是否则允许的范围内，处理同上。3）如果 1）和 2）均能满足，则说明解是正确的。

⑦ 回到步骤②，直到矩阵 I 的所有列都被读取。

⑧ 将每次获得的解合并成矩阵。

⑨ 验证结果的准确性：利用公式$AA^{-1}=dev_I$然后检查dev_I和I之间的误差是否在允许的范围内，如果在允许的范围内则进行下一步，否则直接返回。

⑩ 输出结果在命令行上。

问题 2 算法运算结果

结果如图所示：其矩阵 A 的逆为自己编写的代码获得的解，正确的解是用 matlab 自带函数获得的解。算法运算结果说明编写的代码的正确性。

```
命令行窗口
高斯消元法
求解1组中...
求解结束1组结束...
求解2组中...
求解结束2组结束...
求解3组中...
求解结束3组结束...
求解4组中...
求解结束4组结束...
矩阵A的逆为:
     0.22222           1     0.11111    -0.33333
    -0.33333           0     0.33333           0
     0.33333        -0.5     0.16667           0
     0.44444           0    -0.77778     0.33333

正确的逆为:
     0.22222           1     0.11111    -0.33333
    -0.33333           0     0.33333 -2.7756e-17
     0.33333        -0.5     0.16667 -1.3878e-17
     0.44444           0    -0.77778     0.33333

求解正确
fx >>
```

图 3：求解结果展示

问题 2 存在问题及误差分析

这种解决思路存在的问题和问题 1 中的解决思路存在的问题相同，都是 1）若第 i 行第 i 列的系数恰好为 0，则无法计算，需要进行额外的行交换；2）受到计算误差的影响，特别是系数之间的数量级差距过大时。

计算误差的方法：利用公式$AA^{-1}=dev_I$，得到dev_I。然后通过公式$\sum_{ij}abs(dev_I-I)$即可计算出总的误差。

```
loss: 2.1712e-08
求解正确
```

图 4：计算出的误差（A_b.mat）

问题 3 解决思路

我们可以使用高斯列主消元法提高解的准确性。这是高斯消元法的改进版本。具体步骤如下：

① 构造增广矩阵：将系数矩阵和常数向量合并成一个增广矩阵。

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} & b_1 \\ a_{21} & \cdots & \cdots & a_{2m} & b_2 \\ \vdots & \ddots & \cdots & \vdots & \vdots \\ a_{n1} & \cdots & \cdots & a_{nm} & b_n \end{bmatrix}$$

② 将矩阵转化为上三角行列式。过程中主要操作为 1）行交换，即交换当前列系数最大的行和当前行，2）加减消元，即利用第 i 行的 a[i][:]为基数，乘上可以消去 a[j][i]的系数，然后加到第 j 上。 其中，i≠j。3）将当前行列的系数变为 1。通过操作 1)，可以解决高斯消元法存在的一个问题：若第 i 行第 i 列的系数恰好为 0，则无法计算。通过操作 2)，我们可以将除对角线外的元素变为 0。通过操作 3)，我们可以省略回代这一步操作。

③ 检查解的存在性：如果在任何步骤中，主元为 0 且该行的常数项不为 0，那么方程组无解。如果主元为 0 且该行的常数项也为 0，那么方程组有无穷多解。

④ 显示解：最后，将所有解出的未知数值打印在命令行，就是线性方程组的解。

问题 3 算法实现

① 预处理：1）将常数向量处理为行向量；2）获取系数矩阵和常数向量的行列数；3）将系数矩阵和常数向量合并成一个增广矩阵。

② 检查解的存在性：如果系数矩阵的行数少于列数，说明无解，直接返回。

③ 求解：按照解决思路的②编写代码。

④ 验证解的准确性：1）利用公式$Ax = dev_b$然后检查dev_b和b之间的误差是否在允许的范围内，如果在允许的范围内则进行下一步，否则直接返回。2）利用 matlab 自带函数 linsolve()获得正确的解，然后用自己的解和正确的解对比，看误差是否则允许的范围内，处理同上。3）如果 1）和 2）均能满足，则说明解是正确的。

⑤ 输出解：利用 disp()函数输出解。

问题 3 算法运算结果

结果如图所示：其中线性方程组的解为自己编写的代码获得的解，正确的解是用 matlab 自带函数获得的解。算法运算结果说明编写的代码的正确性。

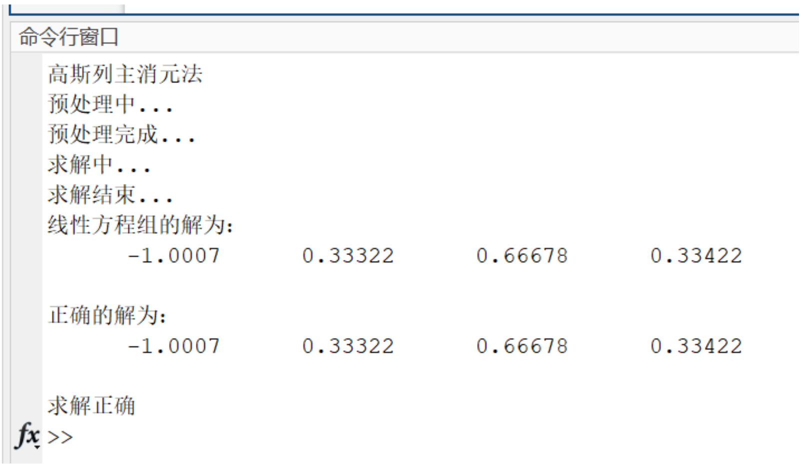


图 5：求解结果展示

问题 3 存在问题及误差分析

列主消元法的主要问题是相比于高斯消元法，列主元消元法的计算量较大，因为在每一步中都需要选择最大的元素作为主元素进行行交换，较大的增加了计算的复杂性。在矩阵较大或者线性方程组的组数比较多时，时间的消耗较大。但是使用列主元消元法可以避免在高斯消元法中可能出现的数值不稳定性问题，并且可以得到更精确的解。

与问题 1 的计算误差方法相同，问题 3 的计算误差的方法：利用公式$Ax = dev_b$，得到dev_b。然后通过公式$\sum_i abs(dev_b - b)$即可计算出总的误差。

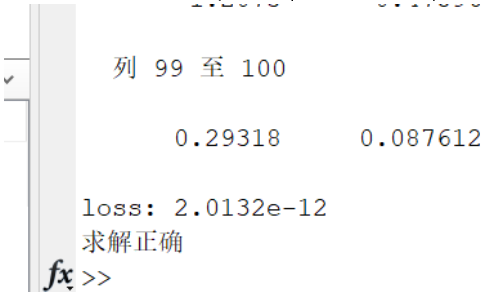


图 6：计算出的误差（A_b.mat）

可以看到，采用列主消元法进行计算的误差比使用高斯消元法计算的误差更小，证明解的准确确实提高了。

问题 4 验证算法准确性

① 高斯消元法求解线性方程组结果。

使用自己编写的函数和 matlab 自带的求解函数得到的结果的误差在允许的范围之内，说明算法的正确性。

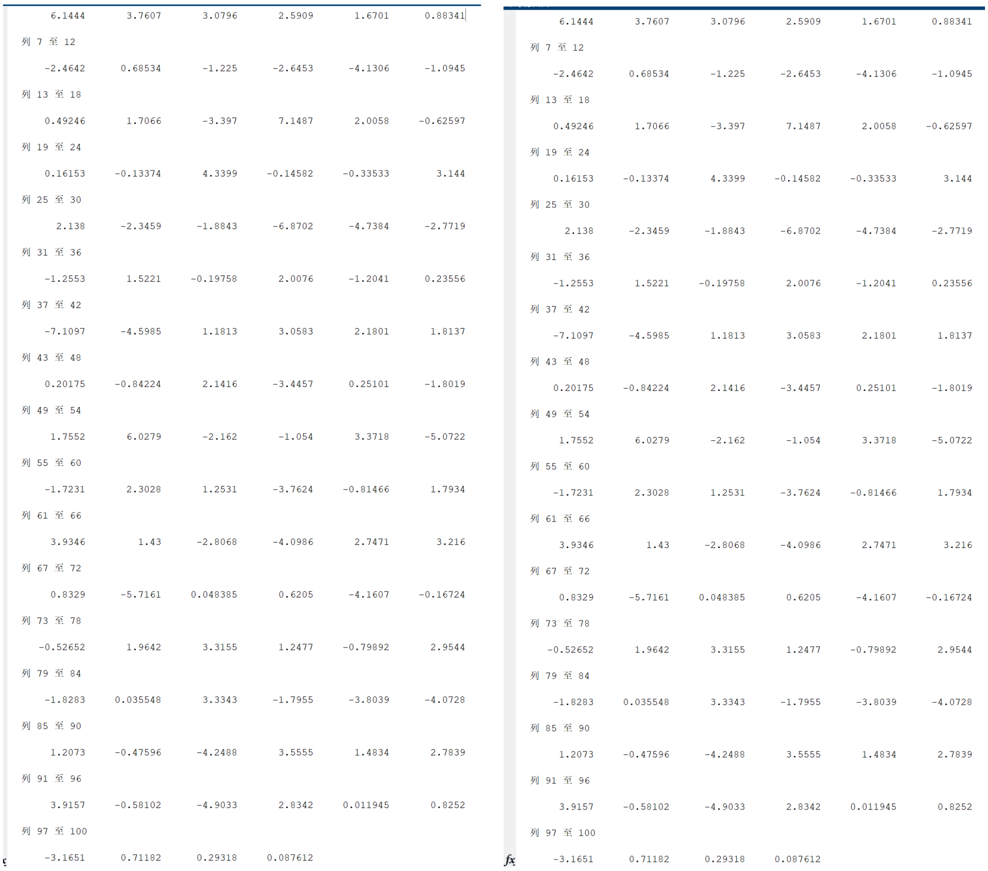


图 7：求解结果展示

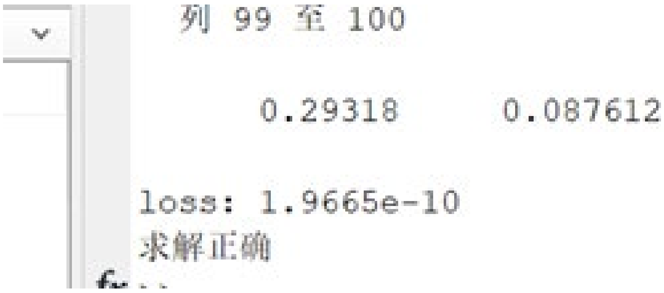
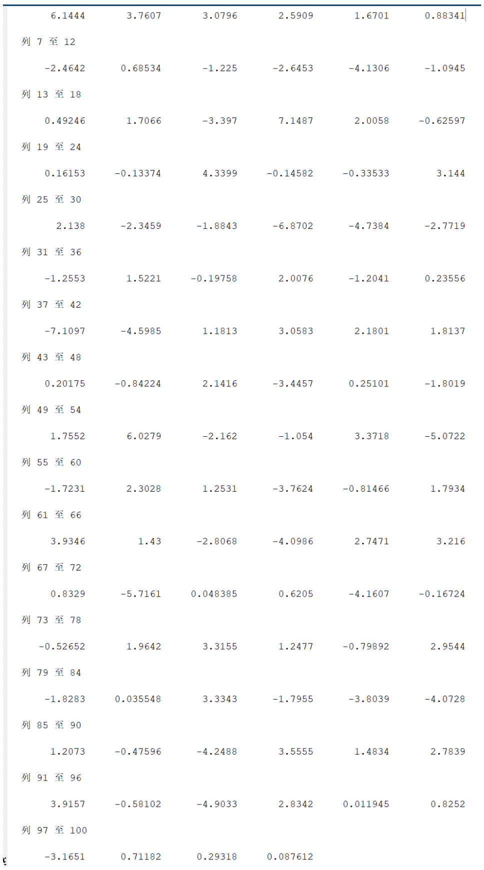
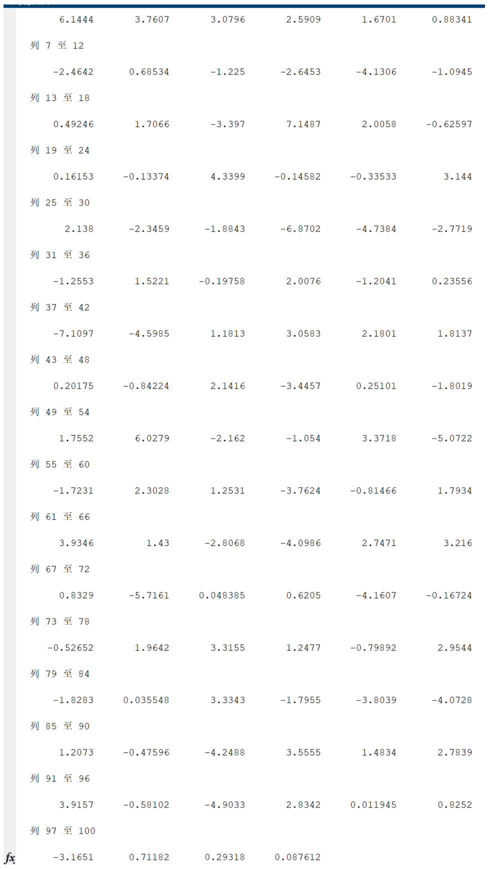


图 8：求解误差（高斯消元）

② 高斯列主消元法求解线性方程组结果。

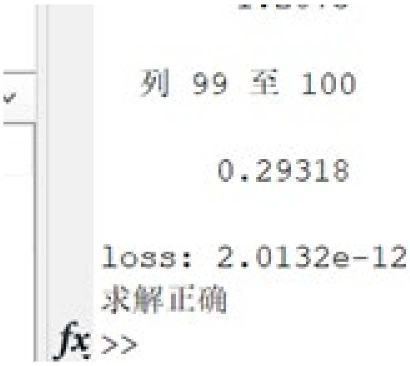


```
    6.1444    3.7607    3.0796    2.5909    1.6701    0.88341
列 7 至 12
   -2.4642   0.68534    -1.225   -2.6453   -4.1306    -1.0945
列 13 至 18
   0.49246    1.7066    -3.397    7.1487    2.0058   -0.62597
列 19 至 24
   0.16153  -0.13374    4.3399  -0.14582  -0.33533      3.144
列 25 至 30
     2.138   -2.3459   -1.8843   -6.8702   -4.7384    -2.7719
列 31 至 36
   -1.2553    1.5221  -0.19758    2.0076   -1.2041    0.23556
列 37 至 42
   -7.1097   -4.5985    1.1813    3.0583    2.1801     1.8137
列 43 至 48
   0.20175  -0.84224    2.1416   -3.4457   0.25101    -1.8019
列 49 至 54
    1.7552    6.0279    -2.162    -1.054    3.3718    -5.0722
列 55 至 60
   -1.7231    2.3028    1.2531   -3.7624  -0.81466     1.7934
列 61 至 66
    3.9346      1.43   -2.8068   -4.0986    2.7471      3.216
列 67 至 72
    0.8329   -5.7161  0.048385    0.6205   -4.1607   -0.16724
列 73 至 78
  -0.52652    1.9642    3.3155    1.2477  -0.79892     2.9544
列 79 至 84
   -1.8283  0.035548    3.3343   -1.7955   -3.8039    -4.0728
列 85 至 90
    1.2073  -0.47596   -4.2488    3.5555    1.4834     2.7839
列 91 至 96
    3.9157  -0.58102   -4.9033    2.8342  0.011945     0.8252
列 97 至 100
   -3.1651   0.71182   0.29318  0.087612
```




```
    6.1444    3.7607    3.0796    2.5909    1.6701    0.88341
列 7 至 12
   -2.4642   0.68534    -1.225   -2.6453   -4.1306    -1.0945
列 13 至 18
   0.49246    1.7066    -3.397    7.1487    2.0058   -0.62597
列 19 至 24
   0.16153  -0.13374    4.3399  -0.14582  -0.33533      3.144
列 25 至 30
     2.138   -2.3459   -1.8843   -6.8702   -4.7384    -2.7719
列 31 至 36
   -1.2553    1.5221  -0.19758    2.0076   -1.2041    0.23556
列 37 至 42
   -7.1097   -4.5985    1.1813    3.0583    2.1801     1.8137
列 43 至 48
   0.20175  -0.84224    2.1416   -3.4457   0.25101    -1.8019
列 49 至 54
    1.7552    6.0279    -2.162    -1.054    3.3718    -5.0722
列 55 至 60
   -1.7231    2.3028    1.2531   -3.7624  -0.81466     1.7934
列 61 至 66
    3.9346      1.43   -2.8068   -4.0986    2.7471      3.216
列 67 至 72
    0.8329   -5.7161  0.048385    0.6205   -4.1607   -0.16724
列 73 至 78
  -0.52652    1.9642    3.3155    1.2477  -0.79892     2.9544
列 79 至 84
   -1.8283  0.035548    3.3343   -1.7955   -3.8039    -4.0728
列 85 至 90
    1.2073  -0.47596   -4.2488    3.5555    1.4834     2.7839
列 91 至 96
    3.9157  -0.58102   -4.9033    2.8342  0.011945     0.8252
列 97 至 100
   -3.1651   0.71182   0.29318  0.087612
```


图 9：求解结果展示

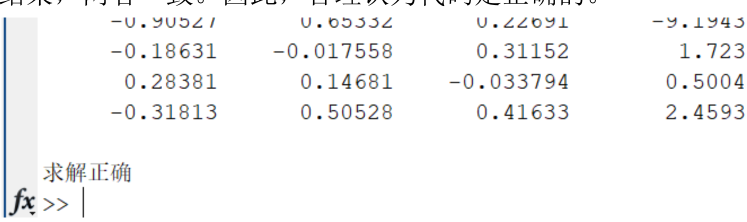


```
列 99 至 100

    0.29318    0.087612

loss: 2.0132e-12
求解正确
fx >>
```

图 10：求解误差（列主消元）

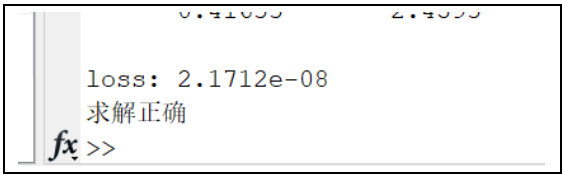
③ 高斯消元法求逆和高斯列主消元法求逆。

根据代码，只有在自己编写的代码得到的结果和标准答案即 matlab 自带函数的得到的结果之间的误差在允许的范围之内，命令行才会打印求解正确。此外，我也观察过输出的结果，两者一致。因此，合理认为代码是正确的。



```
   -0.18631   -0.017558     0.31152      1.723
    0.28381     0.14681   -0.033794     0.5004
   -0.31813     0.50528     0.41633     2.4593

求解正确
fx >>
```

图 11：求解结果展示



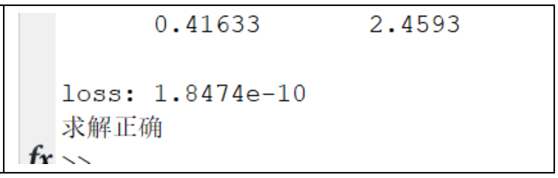


图 12：求解误差（左高斯消元，右列主消元）

四、小结（可含个人心得体会）

① 通过本次实验，我掌握了高斯消元法和高斯列主消元法求解线性方程组的方式以及他们之间的区别。

② 通过编写代码，我对如何使用 Matlab 有了更深的理解。

③ 高斯消元法大大降低了求解线性方程组的时间复杂度，但是也带来了不稳定和精度问题。而高斯列主消元法则很好地解决了上述两个问题，但是时间复杂度会稍高，因为每次都需要找到当前列最大元素所在行。

④ 验证算法的准确性也是编写代码的重要的一环，通过计算 loss 即可知道算法的准确性。

⑤ 验证算法的准确性还可以通过与 Matlab 自带函数的结果对比。由此，我能也够确保自己编写的代码的准确性

⑥ 总的来说，这次实验不仅提升了我的编程技能，也加深了我对线性代数和数值计算的理解。

指导教师批阅意见：   成绩评定：   指导教师签字： 2024 年 10 月 7 日
备注：

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。