全主元高斯消去法的实现

第七小组：任浩辰　刘阳

科目：数值分析

日期：2020 年 6 月 13 日

1　实验目的

通过编程实践，熟练掌握如何使用全主元高斯消去法求解线性方程组，并验证此方法对于奇异方程的求解作用，以及选择主元对于方程解的影响。

2　实验步骤

1. 验证奇异方程的求解问题
2. 验证选主元对方程解的影响

3　实验内容

3.1　顺序高斯消去法

基本思想：用逐次消去未知数的方法，把原方程组化为上三角形方程组进行求解。

求解过程分为两步：

1. 消元过程：用初等行变换将原方程的系数矩阵化为上三角矩阵。

2. 回代过程：对上三角形方程组的最后一个方程求解，将求得的解逐步往上一个方程代入求解。

顺序高斯消去法消元过程：依次从左到右、自上而下将主对角元下方的元素化为 0，不作行交换。

顺序高斯消元法实现简单，但有如下使用条件：方程组系数矩阵 A 为严格对角占优矩阵，即 A 的每个主对角元的绝对值大于同一行其他元素绝对值之和，即

$$|a_{ii}| > \sum_{j=1 j\neq i}^{n} |a_{ij}|, i = 1, 2, ..., n \tag{1}$$

3.2　全主元高斯消去法

由高斯消去法知道，在消元的过程中可能出现 $a_{kk}^{(k)} = 0$ 的情况，这是消去法将无法进行即使主元素 $a_{kk}^{(k)} \neq 0$ 但很小时，用其作除数，会导致其他元素数量级的严重增长和舍入误差的扩

散，最后也使得计算解不可靠。

此时，我们可以用全主元高斯消去来解决这样的问题。第 k 步消元时选 A(k) 中绝对值最大的元素为主元，即

1. 先选取全主元：$|a_{i_k j_k}^{(k)}| = \max_{k \le i, j \le n} |a_{ij}^{(k)}| \ne 0$
2. 如果 $i_k \ne k$，则交换第 k 行和第 i_k 行；如果 $j_k \ne k$，则交换第 k 列和第 j_k 列
3. 消元

全主元高斯消去法具有很好的稳定性，但选全主元比较费时，故在实际计算中很少使用。

4 实验过程及主要代码

4.1 验证奇异方程组的求解问题

奇异方程组，即其系数矩阵行列式为 0。可以选用以下的方程组：

$$\begin{cases} x_1 + x_2 + x_3 = 4 \\ 2x_1 + 2x_2 + 2x_3 = 8 \\ x_1 + 2x_2 + x_3 = 6 \end{cases} \tag{2}$$

将此方程组的数据输入程序中，并观察得到的结果。

4.2 验证选主元对方程解的影响

为了验证全主元高斯消去对方程解的影响，选用如下方程，并分别使用顺序高斯消去，与全主元高斯消去，并将两种方法得到的结果进行对比。

$$\begin{cases} 10^{-7}x_1 + 2x_2 + 3x_3 = 1 \\ -x_1 + 3.217x_2 + 4.623x_3 = 2 \\ -2x_1 + 1.072x_2 + 5.643x_3 = 3 \end{cases} \tag{3}$$

4.3 主要代码

全主元高斯消去法的计算函数如下：

```
void GaussElimination(){
  int row;//保存行数
  int column;//保存列数

  for (int i = 0; i < order - 1; ++i) {
    //找到矩阵剩余部分中的绝对值最大者，并将其行数和列数记录下来
    findMaxInAll(i, &row, &column);
    //将矩阵进行行列变换，使这个位置上的元素移动至aii处
    change(i, row, column);

```

```
    cout << "第" << i + 1 << "次主元" << endl;
    print();//打印矩阵

    //将第i行的各个元素除以aii的值
    for (int j = i + 1; j <= order; ++j)
      matrix[i][j] = matrix[i][j] / matrix[i][i];
      matrix[i][i] = 1;

    //将第i行下面的各行进行消元
    for (int k = i + 1; k < order; ++k){
      float temp = matrix[k][i];
      for (int j = i; j <= order; ++j)
        matrix[k][j] = matrix[k][j] - temp * matrix[i][j];
    }
    cout << "第" << i + 1 << "次消元" << endl;
    print();
  }
  //矩阵的最后一行进行消元
  matrix[order - 1][order] /= matrix[order - 1][order - 1];
  matrix[order - 1][order - 1] = 1;
  cout << "高斯消元后的矩阵为" << endl;
  print();

  //回代
  for(int i = order - 1; i >= 0; i--){
    ans[i] = matrix[i][order];
    for(int j = i + 1; j < order; j++){
      ans[i] -= matrix[i][j] * ans[j];
    }
  }
  printf("方程组解如下：\n");
  printf("---------------------------------------------\n");

  //依次打印方程组的解
  for(int i = 1; i <= order; i++) {
    for (int j = 0; j < order; ++j) {
      if (x[j] == i)
        cout << "x" << i << "的解为" << ans[j] << endl;
    }
  }
}
```

其中，选取矩阵主元，即绝对值最大者的函数编写如下：

```
void findMaxInAll(int i, int* row, int* column) {
  *row = i;
  *column = i;

```

```
    for (int j = i; j < order; ++j) {
      for (int k = i; k < order; ++k) {
        if (abs(matrix[j][k]) > abs(matrix[*row][*column])){
          //记录行列位置
          *row = j;
          *column = k;
        }
      }
    }
  }
```

矩阵进行行列变换的函数如下：

```
void change(int n, int row, int column){
  for (int i = 0; i <= order; ++i)
    swap(matrix[n][i],matrix[row][i]);//行变化
  for (int j = 0; j <= order; ++j) {
    swap(matrix[j][n],matrix[j][column]);//列变化
  }
  swap(x[n],x[column]);//未知数x的位置变化
}
```

主函数如下：

```
int main() {
  cout << "请输入方程个数: " << endl;
  cin >> order;//保存方程阶数
  cout << "输入每一列的数据" << endl;
  for (int i = 0; i < order; ++i) {
    for (int j = 0; j <= order; ++j) {
      cin >> matrix[i][j];//将数据保存到增广矩阵中
    }
  }
  for (int k = 0; k < order; ++k) {
    x[k] = k + 1;//记录未知数顺序
  }
  GaussElimination();//进行高斯消去求解
  return 0;
}
```

5 实验结果及分析

5.1 奇异方程组的解结果

将奇异方程组的数据写入程序，可以得到以下结果，如图所示：

```
方程组解如下：
--------------------------------------------
x1的解为nan
x2的解为nan
x3的解为nan
```

可以看出，如果方程组为奇异方程，即系数矩阵行列式为 0，此时方程的解有无数个，无法求出确定的值。

5.2 选主元对方程解的影响

为了验证顺序高斯消去，与全主元高斯消去的不同结果，我们选用如下方程组：

$$\begin{cases} 10^{-7}x_1 + 2x_2 + 3x_3 = 1 \\ -x_1 + 3.217x_2 + 4.623x_3 = 2 \\ -2x_1 + 1.072x_2 + 5.643x_3 = 3 \end{cases} \tag{4}$$

将方程组的数据写入程序，首先使用全主元高斯消去，会得到如下结果：

```
方程组解如下：
--------------------------------------------
x1的解为-0.468167
x2的解为-0.0679055
x3的解为0.378604
```

其中，对矩阵的选主元、消去步骤如图所示：

```
第1次主元
         5.64    1.07   -2.00    |     3.00
         4.62    3.22   -1.00    |     2.00
         3.00    2.00    0.00    |     1.00

第1次消元
         1.00    0.19   -0.35    |     0.53
                 2.34    0.64    |    -0.46
                 1.43    1.06    |    -0.59

第2次主元
         1.00    0.19   -0.35    |     0.53
                 2.34    0.64    |    -0.46
                 1.43    1.06    |    -0.59

第2次消元
         1.00    0.19   -0.35    |     0.53
                 1.00    0.27    |    -0.20
                         0.67    |    -0.32

高斯消元后的矩阵为
         1.00    0.19   -0.35    |     0.53
                 1.00    0.27    |    -0.20
                         1.00    |    -0.47
```

而使用顺序高斯消去法，则会得到如下结果：

```
方程组解如下:
--------------------------------------------
x1的解为-1
x2的解为-0.25
x3的解为0.5
```

其中，对矩阵的选主元、消去步骤如图所示:

```
第1次主元
        0.00      2.00      3.00      |      1.00
       -1.00      3.22      4.62      |      2.00
       -2.00      1.07      5.64      |      3.00

第1次消元
        1.00    20000000.00    30000000.00    |    10000000.00
                20000004.00    30000004.00    |    10000002.00
                40000000.00    60000004.00    |    20000004.00

第2次主元
        1.00    20000000.00    30000000.00    |    10000000.00
                20000004.00    30000004.00    |    10000002.00
                40000000.00    60000004.00    |    20000004.00

第2次消元
        1.00    20000000.00    30000000.00    |    10000000.00
                   1.00               1.50    |    0.50
                                      8.00    |    4.00

高斯消元后的矩阵为
        1.00    20000000.00    30000000.00    |    10000000.00
                   1.00               1.50    |    0.50
                                      1.00    |    0.50
```

可以看出，在顺序高斯消去法的过程中，数值差距非常大，同时也会造成很大的误差。

将两种方法得到的结果代入原方程，并将其计算出的结果与原结果进行比较，得到如下表格:

	顺序消去	全主元消去	原结果
方程 1	0.9999999	1	1
方程 2	2.505	$2+1.2\times10^{-7}$	2
方程 3	4.5525	$3+1.6\times10^{-6}$	3

从表格中可以看出，使用全主元高斯消去法，可以避免用绝对值较小的数做除数，从而避免其他元素数量级严重的增长和舍入误差的扩大，最终使得得到的结果比顺序高斯消去更加接近真实值。

6 实验体会

在这次的实验过程中，利用 c++ 编写了全主元高斯消去法的有关代码，增加了编程能力，也进一步理解了这种求解线性方程组的计算方法。

运用全主元高斯消去法，可以避免使用绝对值较小的元素作为除数，避免舍入误差的扩大。在计算过程中，每次消元前选取矩阵剩余部分中绝对值最大的元素作为主元素，用这个元素作除数，可以减小舍入误差。

但全主元高斯消去法的计算量过大，因此在平时计算中，往往会采用相对比较简单的列主元高斯消去，这样既能够减少计算量，也能保持一定的精确度。