计算方法第四次上机作业程序文档

5.24.2021 yawning-lion

一、任务介绍

给定矩阵A右端项$\vec{b}$，容许误差，编制程序执行松弛法迭代求解方程$Ax=\vec{b}$。

二、公式说明

1、松弛法迭代解方程

取定松弛因子ω，得到松弛法计算公式（分量形式）

$$x_i^{(k+1)}=(1-\omega)x_i^{(k)}+\frac{\omega(b_i-\sum_{j=0}^{i-1}a_{ij}x_j^{(k+1)}-\sum_{j=i+1}^{n-1}a_{ij}x_j^{(k)})}{a_{ii}}$$

$$i=0,1,2...n-1,k=0,1,2...$$

2、迭代终止条件

$$\|x-x^*\|_\infty<\varepsilon$$

x^*为精确解，ε为容许误差。

四、程序说明

1、运行环境

程序编译环境为mingw-w64-v8.0.0,g++,IDE为vscode，程序文档利用markdown写作。

2、使用说明

- **输入规范**

  本程序应该有5个输入，直接在程序内部修改即可。

  `TOL`为容许误差，在一开始定义，示例如下

```
#define TOL 0.0001
```

  `n`为待求解矩阵的维度。

  `matA[n][n]`为待求解矩阵。

  `b[n]`为方程右端项。

  `m`为松弛法选取的ω的个数。

  示例如下

```
    int n=5;
    double matA[n][n];
    {
        for(int j=0;j<n;j++)
        {
            matA[i][j]=1.0/(i+j+1);
        }
    }
    double b[n]={1,0,0,0,0};
    int m=21;
```

- **输出格式**

  按选取的`omega`的值直接在程序里输出，一个值对应两行或四行输出，不同值对应的输出以空行隔开。

  当给定值下不收敛时，输出两行，示例如下

```
when omega=2
exceed the set maximum number of iteration,treated as un-
convergent
```

  当给定值下收敛时，输出四行，示例如下

```
when omega=1.9
the number of iteration is 31820
the approximate solution is
25.000004 -300.000039 1050.000100 -1400.000081 630.000015
```

依次为给定的值，迭代次数和近似解。

3、程序结构

本程序包含三个头文件

```
#include<stdio.h>
#include<math.h>
#include<stdio.h>
```

随后定义最大迭代次数和容许误差

```
#define MAXNUM 10000000//Maximum number of iterations
#define TOL 0.0001//allowable error
```

定义迭代终止判定函数`ErrorCheck`，执行松弛法迭代的函数`OrIte`，进入主函数。

完成各输入后，进入循环，一次循环对应一个给定的ω，每次开始前都对循环初始向量置0，执行迭代函数，按返回结果输出。

4、函数说明

`int ErrorCheck(int n,double* iniX,double* pX)`

输入两个指向所存储向量的指针和维度，计算分量差值的最大值，即无穷范数意义下的误差，并与容许误差比较。若达到精度要求返回1，否则返回0

`int OrIte(double *pX,int n,double* matA,double *b,double omega)`

输入存储近似解的指针，向量维度，待求解矩阵，右端项，和松弛系数。

进入`for`循环开始迭代，事实上循环终止条件亦即迭代终止条件，迭代次数超过设定的最大值时视为不收敛，迭代终止。

当近似解误差满足要求亦迭代终止，为收敛时迭代终止。

```
if(ErrorCheck(n,iniX,pX))
        {
            state=1;
            break;
        }
```

即若迭代到满足精度要求时，将`state`修改为1，退出循环。

收敛终止返回迭代次数，不收敛终止返回0.

三、算例展示

输入

```
    int n=5;
    double matA[n][n];
    {
        for(int j=0;j<n;j++)
        {
            matA[i][j]=1.0/(i+j+1);
        }
    }
    double b[n]={1,0,0,0,0};
    int m=21;
```

对于

$$\omega = 0, 0.1, 0.2, \ldots, 1.9, 2$$

输出

```
when omega=0
exceed the set maximum number of iteration,treated as un-
convergent

when omega=0.1
the number of iteration is 7423309
the approximate solution is
24.999999 -299.999985 1049.999934 -1399.999900 629.999951

when omega=0.2
the number of iteration is 3515743
```

```
the approximate solution is
24.999999 -299.999985 1049.999934 -1399.999900 629.999951

when omega=0.3
the number of iteration is 2213214
the approximate solution is
24.999999 -299.999985 1049.999934 -1399.999900 629.999951

when omega=0.4
the number of iteration is 1561865
the approximate solution is
24.999999 -299.999985 1049.999934 -1399.999900 629.999951

when omega=0.5
the number of iteration is 1171035
the approximate solution is
24.999999 -299.999985 1049.999934 -1399.999900 629.999951

when omega=0.6
the number of iteration is 910446
the approximate solution is
24.999999 -299.999985 1049.999934 -1399.999900 629.999951

when omega=0.7
the number of iteration is 724273
the approximate solution is
24.999999 -299.999984 1049.999933 -1399.999900 629.999951

when omega=0.8
the number of iteration is 584602
the approximate solution is
24.999999 -299.999984 1049.999933 -1399.999900 629.999951

when omega=0.9
the number of iteration is 475923
the approximate solution is
24.999999 -299.999984 1049.999933 -1399.999900 629.999951

when omega=1
the number of iteration is 388926
the approximate solution is
24.999999 -299.999984 1049.999933 -1399.999900 629.999951
```

```
when omega=1.1
the number of iteration is 317680
the approximate solution is
24.999999 -299.999984 1049.999933 -1399.999900 629.999951

when omega=1.2
the number of iteration is 258224
the approximate solution is
24.999999 -299.999984 1049.999933 -1399.999900 629.999952

when omega=1.3
the number of iteration is 207803
the approximate solution is
24.999999 -299.999984 1049.999933 -1399.999900 629.999952

when omega=1.4
the number of iteration is 164427
the approximate solution is
24.999999 -299.999984 1049.999932 -1399.999900 629.999952

when omega=1.5
the number of iteration is 126587
the approximate solution is
24.999999 -299.999984 1049.999932 -1399.999900 629.999952

when omega=1.6
the number of iteration is 93049
the approximate solution is
24.999999 -299.999983 1049.999931 -1399.999900 629.999952

when omega=1.7
the number of iteration is 62527
the approximate solution is
24.999999 -299.999983 1049.999930 -1399.999900 629.999953

when omega=1.8
the number of iteration is 31702
the approximate solution is
24.999999 -299.999981 1049.999927 -1399.999900 629.999955

when omega=1.9
the number of iteration is 31820
the approximate solution is
```

```
25.000004 -300.000039 1050.000100 -1400.000081 630.000015

when omega=2
exceed the set maximum number of iteration,treated as un-
convergent
```

可以看到当ω取$0, 2$时迭代次数超过设定的最大值10000000,视为不收敛，其余皆收敛，且可以看出当$\omega = 1.8$时收敛最快。

四、结果分析

可能是因为矩阵本身性质和初值选取的原因，迭代次数都很大，当ω小于1时，迭代次数都超过G-S法的迭代次数，当ω大于1时，迭代次数得到改善。