数值分析实验报告

第四章

姓名	王琛
学号	2016011360
班级	计 65
实验日期	2019 年 4 月 11 日
报告日期	2019 年 4 月 11 日

第 4 章第 2 题

问题描述

考虑常微分方程的两点边值问题：

$$\begin{cases} \epsilon \dfrac{d^2y}{dx^2} + \dfrac{dy}{dx} = a, (0 < a < 1) \\ y(0) = 0, y(1) = 1 \end{cases}$$

容易知道它的精确解为

$$y = \frac{1-a}{1-e^{1/\epsilon}}(1 - e^{-\frac{x}{\epsilon}}) + ax$$

为了把微分方程离散，把 [0,1] 区间 n 等分，令 $h = \frac{1}{n}$，

$$x_i = ih, (i = 1, 2, ..., n-1),$$

得到有限差分方程

$$\epsilon \frac{y_{i-1} - 2y_i + y_{i+1}}{h^2} + \frac{y_{i+1} - y_i}{h} = a,$$

简化为

$$(\epsilon + h)y_{i+1} - (2\epsilon + h)y_i + \epsilon y_{i-1} = ah^2$$

从而离散后得到的线性方程组的系数矩阵为

$$\boldsymbol{A} = \begin{bmatrix} -(2\epsilon+h) & \epsilon+h & & & \\ \epsilon & -(2\epsilon+h) & \epsilon+h & & \\ & \epsilon & -(2\epsilon+h) & \ddots & \\ & & \ddots & \ddots & \epsilon+h \\ & & & \epsilon & -(2\epsilon+h) \end{bmatrix}$$

(1) 对 $\epsilon = 1, a = \frac{1}{2}, n = 100$，分别用雅可比、G-S 和 SOR 方法求线性方程组的解，要求有 4 位有效数字，然后比较与精确解的误差.

(2) 对 $\epsilon = 0.1, \epsilon = 0.01, \epsilon = 0.0001$ 考虑同样的问题。

解题思路

首先根据题意生成矩阵 A 和向量 b，然后实现 jacobi、G-S 和 SOR 算法对方程进行求解并且计算误差。改变 ϵ 的值，重复实验。

实验结果

改变 ϵ 的值，得到的误差的无穷范数如下:

Jacobi 和 G-S:

ϵ	Jacobi	G-S
1	3.0952e-04	3.0992e-04
0.1	0.0088	0.0088
0.01	0.0661	0.0661
0.0001	0.0050	0.0050

对于 SOR 迭代法，不同的 w 对应的结果不同。对于某些 w，算法可能并不收敛。根据 D.M Young 的研究结果，SOR 的 optimal w 为 $\omega_{opt} = \frac{2}{1+\sqrt{1-\rho^2}}$，其中 ρ 为谱半径。在实验中，我选择了 w=0.5, 0.8, 1.5, 1.8 以及 w_{opt} 进行了测试，当 $\rho > 1$ 时，w_{opt} 使用 1 代替。

ϵ \ ω	0.5	0.8	1.5	1.8	ω_{opt}
1	3.2961e-4	3.1485e-04	3.0336e-04	3.01154e-04	3.0992e-04
0.1	0.0088	0.0088	0.0088	0.0088	0.0088
0.01	0.0661	0.0661	0.0661	0.0661	0.0061
0.0001	0.0050	0.0050	不收敛	不收敛	0.0050

实验结论

从结果可以看出，三种方法在 ϵ 相同时，误差基本相同（除了 $\epsilon = 1$ 时有很小的差别，基本可以看成是计算误差）。且 SOR 的不同 ω 对结果也没有影响。

随着 ϵ 的减小，误差出现先增大后减小的趋势。当 $\epsilon = 0.01$ 时，误差最大，此时 h=0.01，猜想 ϵ 和 h 数量级相当时，误差最大。

实验心得

这次实验我主要实现了 Jacobi、G-S 和 SOR 三种迭代求解线性方程组的解法。并且了解了如何通过查分近似求解线性方程组。可以得出，三种方法对于误差的影响基本相同，但是 SOR 依赖于 w_{opt} 的选择。

主要代码

jacobi

```
function[x]=jacob(x,A,b,eps)
n=size(A,1);
y=inf(n,1);
while norm(x-y,inf)>=eps
    y=x;
    for i=1:n
        x(i)=b(i);
        for j=1:n
            if j~=i
                x(i)=x(i)-A(i,j)*y(j);
            end
        end
        x(i)=x(i)/A(i,i);
    end
end
```

G-S

```
function[x]=gauss_seidel(x,A,b,eps)
n=size(A,1);
y=inf(n,1);
```

```
while norm(x-y,inf)>=eps
    y=x;
    for i=1:n
        x(i)=b(i);
        for j=1:n
            if j~=i
                x(i)=x(i)-A(i,j)*x(j);
            end
        end
        x(i)=x(i)/A(i,i);
    end
end
```

SOR

```
function[x]=SOR(x,A,b,eps,w)
n=size(A,1);
y=inf(n,1);
max_cnt=100000;
cnt=0;
while norm(x-y,inf)>=eps
    if cnt > max_cnt
        fprintf("SOR doesn't converge!!!\n");
        return
    end
    y=x;
    for i=1:n
       sum=b(i);
       for j=1:n
           if j~=i
               sum=sum-A(i,j)*x(j);
           end
       end
       x(i)=(1-w)*x(i)+w*sum/A(i,i);
    end
```

```
    cnt=cnt+1;
end
```