线性方程组求解——Gauss-Seidel 迭代法

湘潭大学, 数学与计算科学学院, 21 级王艺博

一、Gauss-Seidel 迭代法

$n=3$ 时

$$A=\begin{pmatrix}a_{11}&a_{12}&a_{13}\\a_{21}&a_{22}&a_{23}\\a_{31}&a_{32}&a_{33}\end{pmatrix},\quad b=\begin{pmatrix}b_1\\b_2\\b_3\end{pmatrix},$$

Jacobi 公式为

$$\begin{aligned}x_1^{(k+1)}&=\frac{b_1-a_{12}x_2^{(k)}-a_{13}x_3^{(k)}}{a_{11}}\\x_2^{(k+1)}&=\frac{b_2-a_{21}x_1^{(k)}-a_{23}x_3^{(k)}}{a_{22}}\\x_3^{(k+1)}&=\frac{b_3-a_{31}x_1^{(k)}-a_{32}x_2^{(k)}}{a_{33}}\end{aligned}$$

可以看出 Jacobi 方法每次迭代使用的是上一步的结果，每行全部独立计算完后才进入下一轮迭代。而实际上计算 x_2 时, 本次迭代产生的 x_1 已经更新, 使用 x_3 时，x_1,x_2 已经更新。由此想法（尽可能使用最新产生的结果），得到 Gauss-Seidel 方法

Gauss-Seidel 公式为

$$\begin{aligned}x_1^{(k+1)}&=\frac{b_1-a_{12}x_2^{(k)}-a_{13}x_3^{(k)}}{a_{11}}\\x_2^{(k+1)}&=\frac{b_2-a_{21}x_1^{(k+1)}-a_{23}x_3^{(k)}}{a_{22}}\\x_3^{(k+1)}&=\frac{b_3-a_{31}x_1^{(k+1)}-a_{32}x_2^{(k+1)}}{a_{33}}\end{aligned}$$

或等价的，将 A 分解为 $A=D-L-U$，其中

$$D=diag(a_{11},a_{22},a_{33}),$$

$$L=-\begin{bmatrix}0&0&0\\a_{21}&0&0\\a_{31}&a_{32}&0\end{bmatrix},\quad U=-\begin{bmatrix}0&a_{12}&a_{13}\\0&0&a_{23}\\0&0&0\end{bmatrix}.$$

$$
\begin{aligned}
(D-L-U)x &= b \\
Dx &= b+(L+U)x \\
Dx^{(k+1)} &= b+Lx^{(k+1)}+Ux^{(k)} \\
(D-L)x^{(k+1)} &= b+Ux^{(k)} \\
x^{(k+1)} &= (D-L)^{-1}(b+Ux^{(k)})
\end{aligned}
$$

其中 $x^{(k+1)}=D^{-1}(b+Lx^{(k+1)}+Ux^{(k)})$ 写成矩阵的形式

$$
\begin{bmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \\ x_3^{(k+1)} \end{bmatrix} = \begin{bmatrix} \frac{1}{a_{11}} & & \\ & \frac{1}{a_{22}} & \\ & & \frac{1}{a_{33}} \end{bmatrix} \left(\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} - \begin{bmatrix} & & \\ a_{21} & & \\ a_{31} & a_{32} & \end{bmatrix} \begin{bmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \\ x_3^{(k+1)} \end{bmatrix} - \begin{bmatrix} & a_{12} & a_{13} \\ & & a_{23} \\ & & \end{bmatrix} \begin{bmatrix} x_1^{(k)} \\ x_2^{(k)} \\ x_3^{(k)} \end{bmatrix} \right)
$$

下面是其一般形式下的算法

二、算法

♡ Gauss-Seidel 迭代法

主要思路 输入: $A, b, x^{(0)}$, 输出 x

$$
\begin{aligned}
x^{(0)} &= \text{ initial vector} \\
x^{(k+1)} &= (D-L)^{-1}(b+Ux^{(k)})
\end{aligned}
$$

添加一些限制

- 容许误差 e_tol,
- 最大迭代步 N.

当残差 <e_tol 或迭代步数 $\geq N$ 时，都会停止迭代，输出结果

实现步骤

- 步骤 1: $k=0, x=x^{(0)}$;
- 步骤 2: 计算残差 $r=\|b-Ax\|$,
  - 如果残差 r >e_tol 且 $k<N$，转步骤 3;
  - 否则, 转步骤 5;
- 步骤 3: 更新解向量

$$
x=(D-L)^{-1}(b+Ux^{(0)})
$$

- 步骤 4：$x0=x, k=k+1$，转步骤 2;

- 步骤 5: 输出 x.

算法 1 线性方程组的 Gauss-Seidel 迭代法

输入：系数矩阵 A, 右端向量 b, 初值 $x0$, 容许误差 e_tol, 最大迭代步 N;
输出：数值解 x
$k = 0, x = x0;$
计算残差 $r = \|b - Ax\|;$
while $r >$e_tol & $k < N$ **do**
    $x = (D - L)^{-1}(b + Ux(0))$
    $x0 = x;$
    $r = \|b - Ax\|;$
    $k = k + 1;$
end while
if $k > N$ **then**
    输出: 算法超出最大迭代次数;
else
    输出: 迭代次数 k;
end if
返回 x;

三、北太天元源程序

```
function [x,k,r] = myGS(A,b,x0,e_tol,N)
% Gauss-Seidel迭代法解线性方程组
% Input: A, b(列向量), x0(初始值)
%       e_tol: error tolerant
%       N: 限制迭代次数小于 N 次
% Output: x , k(迭代次数),r:残差
%  Version:          1.0
%  last modified:   01/29/2024
   n = length(b); k = 0;
   x=zeros(n,N); % 记录每一次迭代的结果，方便后续作误差分析
   x(:,1)=x0;
   L = -tril(A,-1); U = -triu(A,1); D = diag(diag(A));
   r = norm(b - A*x,2);
   while r > e_tol && k < N
       x(:,k+2) = inv(D-L)*(b+U*x(:,k+1)); % 不同之处
       r = norm(b - A*x(:,k+2),2); % 残差
       k = k+1;
   end
   x = x(:,2:k+1); % x取迭代时的结果

   if k>N
       fprintf('迭代超出最大迭代次数');
   else
       fprintf('迭代次数=%i\n',k);
   end
end
```

将上述代码保存为 myGS.m 文件。

四、数值算例

例 1 利用 *Gauss-Seidel* 迭代法求解线性方程组

$$\begin{bmatrix} 10 & -1 & 2 & 0 \\ -1 & 11 & -1 & 3 \\ 2 & -1 & 10 & -1 \\ 0 & 3 & -1 & 8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 6 \\ 25 \\ -11 \\ 15 \end{bmatrix}$$

取初值为

$$x^{(0)}=\begin{bmatrix}0\\0\\0\\0\end{bmatrix}$$

容许误差及最大迭代步分别为 `e_tol=1.0e-08`, `N=100`。

调用函数 `[x,k,r] = myGS(A,b,x0,e_tol,N)`，进行实现

容易验证，线性方程组的真解为

$$x=\begin{bmatrix}1\\2\\-1\\1\end{bmatrix}$$

在计算过程中，我们同时将每一迭代步求得的解向量 $x^{(k)}$，并计算残差 $r^{(k)}=\|b-Ax^{(k)}\|$ 和误差向量 $e^{(k)}=|x-x^{(k)}|$，如图1-3 所示。从图1可以看出，随着迭代步的增加，数值解向量的每一个分量均收敛到真解。图2显示方程的残差 $r^{(k)}$ 快速收敛到 0，且开始几个迭代步残差下降的非常快。类似的，图3显示误差 $e^{(k)}$ 快速收敛到 0。

G-S 收敛所需步数相比 Jacobi 明显更少，这个例子中，G-S 只需 10 步，而 Jacobi 需要 26 步。

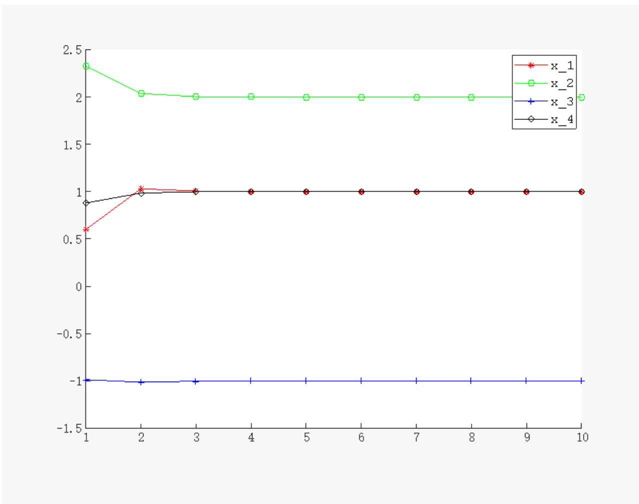

图 1 数值解

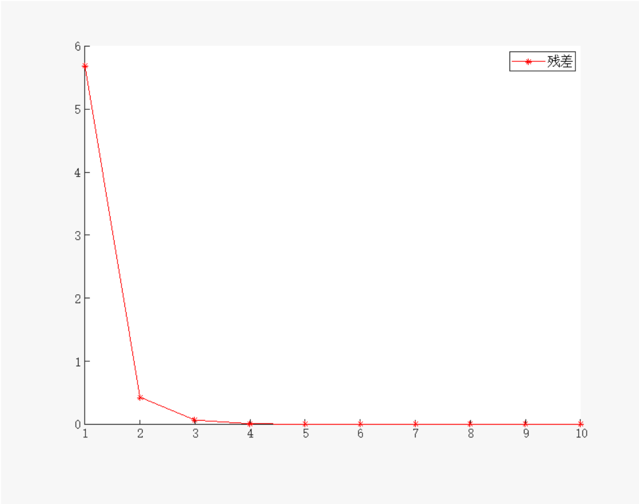

图 2　残差变化

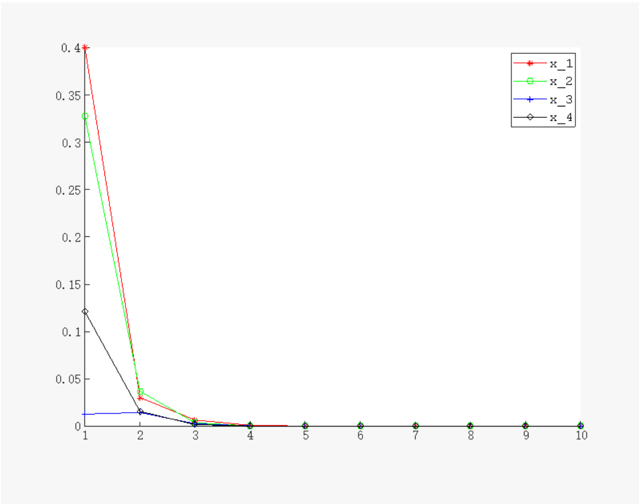

图 3　误差

```
%% Gauss-Seidel test
% time :    1/29/2023
clc;clear all,format long;
addpath("../base","../Jacobi迭代法","../Gauss-Seidel迭代法") % 加载函数文件
N = 100; e_tol = 1e-8;
%% example 1
A=[10 -1 2 0; -1 11 -1 3; 2 -1 10 -1; 0 3 -1 8];
b=[6; 25; -11; 15];
x0=[0; 0; 0; 0];
    [x11,k1] = myGS(A,b,x0,e_tol,N)
% 作图查看误差变化
    x_exact=[1;2;-1;1]; %真解
    n = length(b);
    error=zeros(n,k1);% 每个分量的误差
        error = abs(x_exact - x11);
```

```
res =zeros(1,k1); % 残差
for i=1:1:k1
    res(i) = norm(b-A*x11(:,i),2);
end
% 数值解
    figure(1);
    plot(1:k1,x11(1,:),'-*r',1:k1,x11(2,:),'-og', 1:k1,x11(3,:),'-+b',1:k1,x11(4,:),'-dk');
    legend('x_1','x_2','x_3','x_4');
% 残差变化
    figure(2);
    plot(1:k1,res,'-*r');
    legend('残差');
% 误差
    figure(3);
    plot(1:k1,error(1,:),'-*r',1:k1,error(2,:),'-og',
        1:k1,error(3,:),'-+b',1:k1,error(4,:),'-dk');
    legend('x_1','x_2','x_3','x_4');
```

将上述代码保存为 GS_test.m 文件。