课题组组会-练习 2

王程

2023 年 10 月 23 日

一　练习及结果

1. 已知方程组 $Ax = b$, 其中 $A \in \mathcal{R}^{20\times 20}$, 定义为

$$\begin{bmatrix} 3 & -1/2 & -1/4 & & & \\ -1/2 & 3 & -1/2 & -1/4 & & \\ -1/4 & -1/2 & 3 & -1/2 & \ddots & \\ & \ddots & \ddots & \ddots & \ddots & -1/4 \\ & & -1/4 & -1/2 & 3 & -1/2 \\ & & & -1/4 & -1/2 & 3 \end{bmatrix}$$

试通过迭代法求解此方程组，认识迭代法收敛的含义以及迭代初值和方程组系数矩阵性质对收敛速度的影响。

实验要求：

(1) 选取不同的初始向量 $x^{(0)}$ 和不同的方程组右端项向量 b，给定迭代误差要求，用雅克比迭代法和高斯-赛德尔迭代法计算，观测得到的迭代向量序列是否收敛？若收敛，记录迭代次数，分析计算结果并给出你的结论。

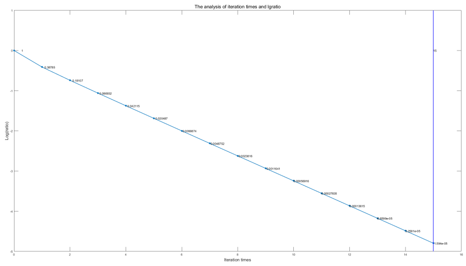
(2) 取定右端向量 b 和初始向量 $x^{(0)}$, 将 A 的主对角线元素成倍增长若干次，非主对角线元素不变，每次用雅克比迭代法计算，要求迭代误差满足 $||x^{(k+1)} - x^{(k)}||_\infty < 10^{-5}$，比较收敛速度，分析现象，并得出你的结论。

(3) 取定右端向量 b 和初始向量 $x^{(0)}$, 分别使用 a）高斯-赛德尔迭代，b）对称高斯-赛德尔迭代，c）对以上两种方法分别加上松弛因子 ω 进行求解，比较收敛速度，分析现象，得出结论。

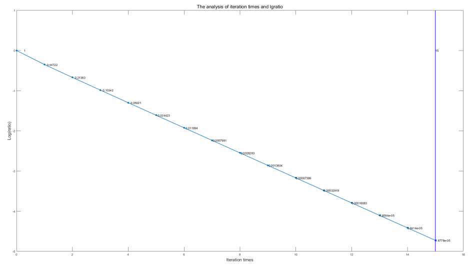
解：

(1) 采用随机数生成的方法构建初始向量 $x^{(0)}$，$x^{(0)}$ 中的元素为 $0 \sim 1$ 之间的随机数。同样的，采用随机数生成的方法构建右端项向量 b，b 中的元素为 $0 \sim 100$ 之间的随机数，要求迭代误差满足 $||x^{(k+1)} - x^{(k)}||_\infty < 10^{-5}$。以下分别给出两种迭代法的迭代次数与误差量级变化图，其中高斯-赛德尔迭代分别采用了 Forward sweep 和 Backward sweep。

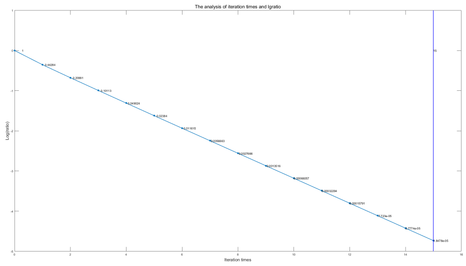
雅克比迭代法



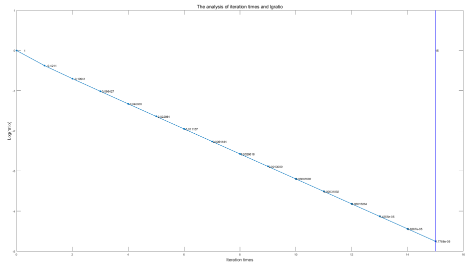
(a) random1



(b) random2



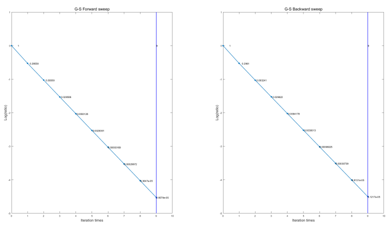
(c) random3



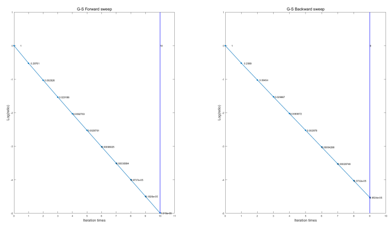
(d) random4

图 1: 雅克比迭代次数和误差量级图

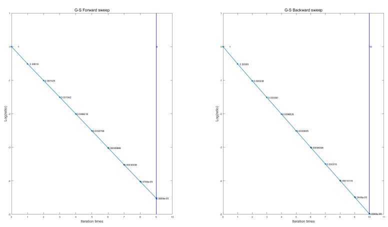
高斯-赛德尔迭代法



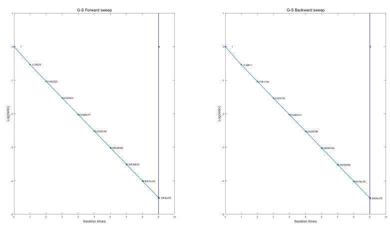
(a) random1



(b) random2



(c) random3

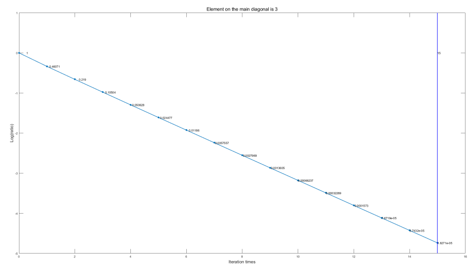


(d) random4

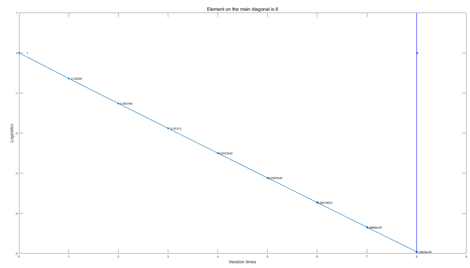
图 2: 高斯迭代次数和误差量级图

观察图 1 可以看出，需要进行 15 次左右的雅克比迭代，方程的解才能小于迭代误差限制。观察图 2 可以看出，需要进行 9 次左右的高斯-赛德尔迭代，方程的解才能小于迭代误差限制。高斯-赛德尔迭代的收敛速度快于雅克比迭代。

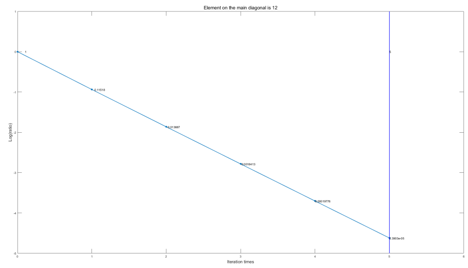
(2) 取右端向量 $b = [1, 2, \cdots, 20]^T$, 初始向量 $x^{(0)} = 0$, 将 A 的主对角线元素成倍增长，非主对角线元素不动，得到下图：



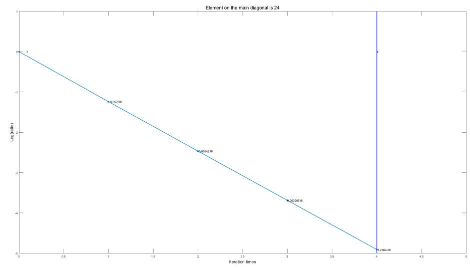
(a) element=3



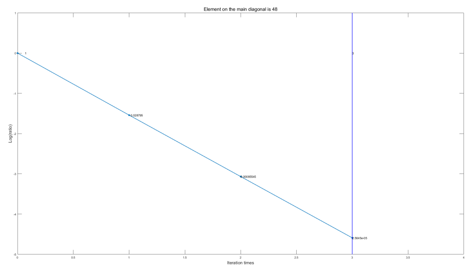
(b) element=6



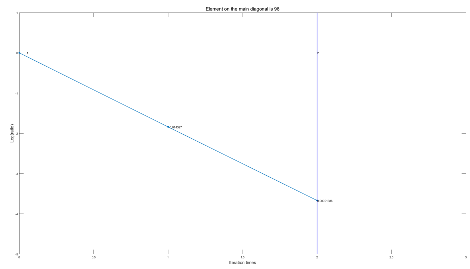
(c) element=12



(d) element=24



(e) element=48

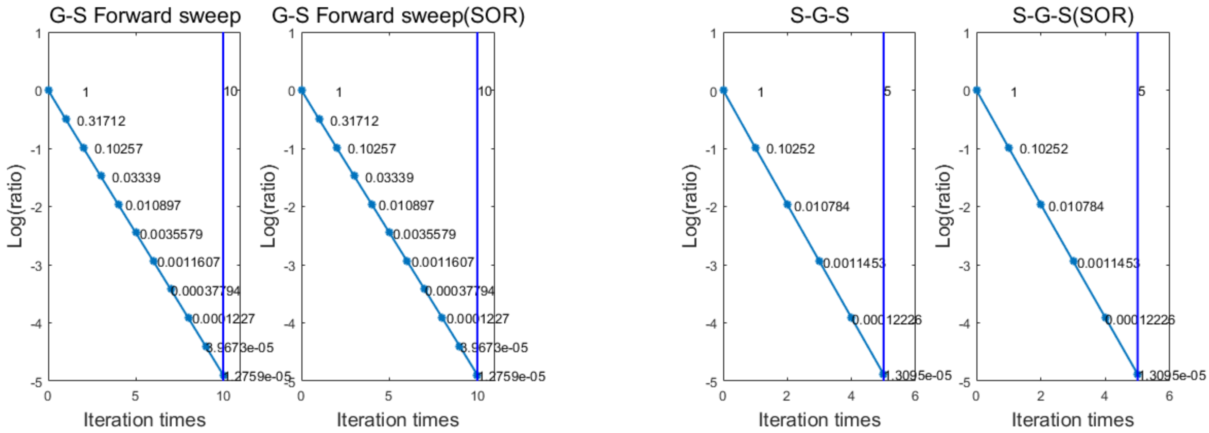


(f) element=96

图 3: 主对角线元素与迭代次数图

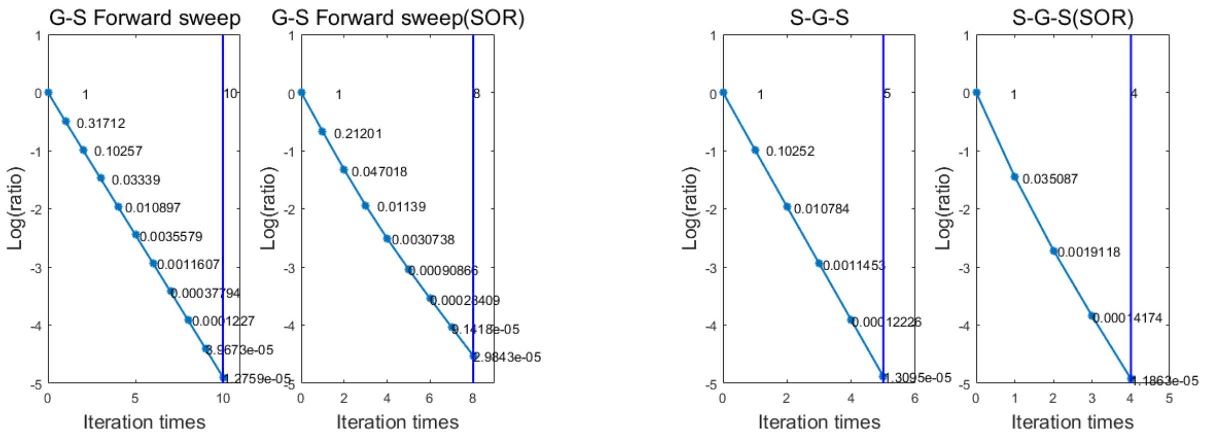
观察图 3，很明显主对角线元素越大，雅克比迭代次数约少。原因可能是随着主对角线元素的增大，矩阵 M 与 A 约接近，导致迭代次数减小。

(3) 右端项和初始向量的取值与（2）保持一致，给出不同 ω 下 G-S 迭代与 G-S(SOR) 的对比图，以及 S-G-S 迭代与 S-G-S(SOR) 的对比图。



(a) G-S and G-S(SOR)　　**(b)** S-G-S and S-G-S(SOR)

图 4: $\omega = 1$ 对比



(a) G-S and G-S(SOR)　　**(b)** S-G-S and S-G-S(SOR)

图 5: $\omega = 1.2$ 对比

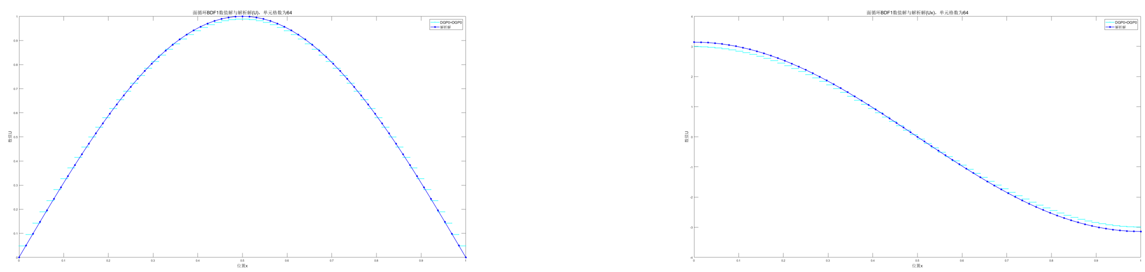
这里仅给出 $\omega = 1$ 与 $\omega = 1.2$ 的比较图，改变松弛因子的数值，最终发现当 $\omega = 1.2$ 左右时加速效果最好。

2. 对带源项的扩散方程 $u_t = u_{xx} + \pi^2 sin\left(\pi x\right), x \in [0,1], t \geq 0$，满足以下初始条件 $u\left(x,0\right) = x^2 - x$，及边界条件 $u\left(0,t\right) = u\left(1,t\right) = 0$。

在练习 1 的基础上，空间离散使用 DG(P0)+DG(P0) 格式，时间离散格式使用 BDF1，使用 LU-SGS 方法在均匀网格下进行求解。

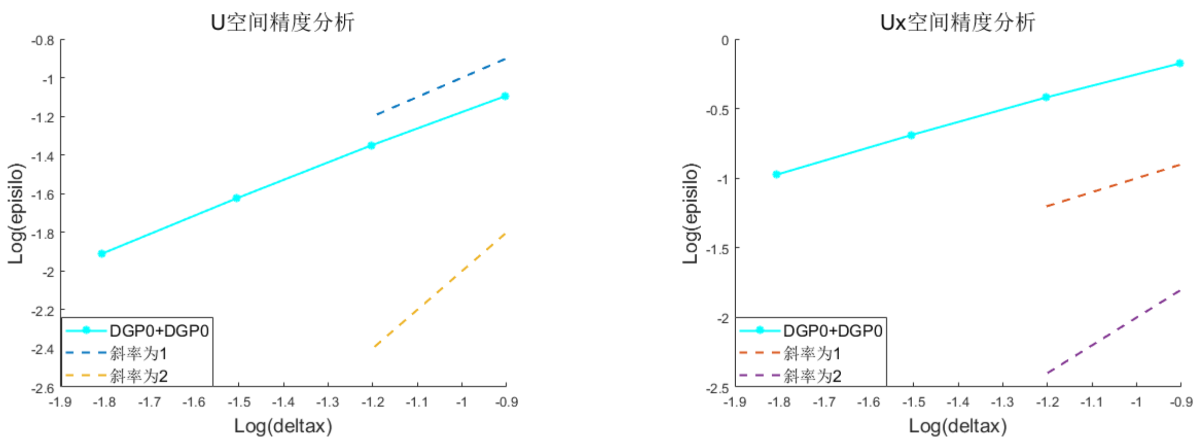
解：

与课题组练习 1 类似，仅改变迭代方法，这里给出单元格数为 64，CFL=0.01 的情况下，U 与 U_x 的数值解与解析解的对比图，以及 U 与 U_x 的空间精度分析。



(a) U 的数值解与解析解 **(b)** U_x 的数值解与解析解

图 6: U 与 U_x 的数值解与解析解



(a) U 的空间精度分析 **(b)** U_x 的空间精度分析

图 7: U 与 U_x 的空间精度分析

二　附录 (代码)

JacobiIteration

```
clc
close all
clear all
% Pre-proceeding
D=3*eye(20);
L=sparse(2:20,1:19,-1/2,20,20)+sparse(3:20,1:18,-1/4,20,20);
U=L';b=100*rand(20,1);X0=rand(20,1);Xold=zeros(20,1);Xnew=zeros(20,1);
tol=10^(-5);endtimes=100;epsilon=10^(-12);
ratioR=[0:endtimes;zeros(1,endtimes+1)];
% Proceeding
Xold=X0;resnorm0=sqrt(sum(((D+L+U)*X0-b).^2))+epsilon;
resnorm=sqrt(sum(((D+L+U)*Xold-b).^2));
ratio=resnorm/resnorm0;ratioR(2,1)=ratio;
for times=2:endtimes
Xnew=-D\(L+U)*Xold+D\b;
resnorm=sqrt(sum(((D+L+U)*Xnew-b).^2));
ratio=resnorm/resnorm0;
if ratio<tol
break
end
ratioR(2,times)=ratio;
Xold=Xnew;
end
ind=find(ratioR(2,:),1,'last');
ratioR(:,ind+1:endtimes+1)=[];
plot(ratioR(1,:),log10(ratioR(2,:)),'-*','linewidth',1.5)
xlabel('Iteration times','fontsize',14)
ylabel('Log(ratio)','fontsize',14)
title('The analysis of iteration times and lgratio','fontsize',16)
hold on
str1=num2str(ratioR(2,:)');text(ratioR(1,:),log10(ratioR(2,:)),str1,'
    linewidth',1.5);
line([ratioR(1,ind),ratioR(1,ind)], [-5,1], 'color', 'b','linewidth'
    ,1.5);
str2=num2str(ind-1);
text(ind-1,0,str2,'linewidth',1.5);
xlim([0,ind])
```

Gauss-seidelIteration

```
clc
close all
clear all
% Pre-proceeding
n=20;
D=3*eye(n);
L=sparse(2:n,1:n-1,-1/2,n,n)+sparse(3:n,1:n-2,-1/4,n,n);
U=L';
A=D+L+U;
b=100*rand(n,1);b1=b;
X0=rand(n,1);XF=zeros(n,1);XB=zeros(n,1);
tol=10^(-5);endtimes=100;epsilon=10^(-12);
ratioRF=[0:endtimes;zeros(1,endtimes+1)];
ratioRB=[0:endtimes;zeros(1,endtimes+1)];
XF=X0;XB=X0;
resnorm0=sqrt(sum((A*X0-b).^2))+epsilon;
resnorm=sqrt(sum((A*XF-b).^2));
ratio=resnorm/resnorm0;ratioRF(2,1)=ratio;ratioRB(2,1)=ratio;

% Proceeding
%Forward sweep
for times=2:endtimes
%calculate X
for i=1:n
for j=1:i-1
b(i)=b(i)-A(i,j)*XF(j);
end
for j=i+1:n
b(i)=b(i)-A(i,j)*XF(j);
end
XF(i)=b(i)/A(i,i);
end
b=b1;
resnorm=sqrt(sum((A*XF-b).^2));
ratio=resnorm/resnorm0;
if ratio<tol
break
```

```
end
ratioRF(2,times)=ratio;
end
ind=find(ratioRF(2,:),1,'last');
ratioRF(:,ind+1:endtimes+1)=[];
subplot(1,2,1)
plot(ratioRF(1,:),log10(ratioRF(2,:)),'-*','linewidth',1.5)
xlabel('Iteration times','fontsize',14)
ylabel('Log(ratio)','fontsize',14)
title('G-S Forward sweep','fontsize',16)
hold on
str1=num2str(ratioRF(2,:)');text(ratioRF(1,:),log10(ratioRF(2,:)),str1,'
    linewidth',1.5);
line([ratioRF(1,ind),ratioRF(1,ind)], [-5,1], 'color', 'b','linewidth'
    ,1.5);
str2=num2str(ind-1);
text(ind-1,0,str2,'linewidth',1.5);
xlim([0,ind])


%Backward sweep
for times=2:endtimes
%calculate X
for i=n:-1:1
for j=n:-1:i+1
b(i)=b(i)-A(i,j)*XB(j);
end
for j=i-1:-1:1
b(i)=b(i)-A(i,j)*XB(j);
end
XB(i)=b(i)/A(i,i);
end
b=b1;
resnorm=sqrt(sum((A*XB-b).^2));
ratio=resnorm/resnorm0;
if ratio<tol
break
end
ratioRB(2,times)=ratio;
end
```

```
ind=find(ratioRB(2,:),1,'last');
ratioRB(:,ind+1:endtimes+1)=[];
subplot(1,2,2)
plot(ratioRB(1,:),log10(ratioRB(2,:)),'-*','linewidth',1.5)
xlabel('Iteration times','fontsize',14)
ylabel('Log(ratio)','fontsize',14)
title('G-S Backward sweep','fontsize',16)
hold on
str1=num2str(ratioRB(2,:)');text(ratioRB(1,:),log10(ratioRB(2,:)),str1,'
    linewidth',1.5);
line([ratioRB(1,ind),ratioRB(1,ind)], [-5,1], 'color', 'b','linewidth'
    ,1.5);
str2=num2str(ind-1);
text(ind-1,0,str2,'linewidth',1.5);
xlim([0,ind])
```

S-G(SOR) and S-G-S(SOR)

```
clc
close all
clear all
% Pre-proceeding
n=20;
omega=1;%松弛因子
%构建矩阵
D=3*eye(n);
L=sparse(2:n,1:n-1,-1/2,n,n)+sparse(3:n,1:n-2,-1/4,n,n);
U=L';
A=D+L+U;
b=[1:n]';b1=b;
X0=zeros(n,1);XF=zeros(n,1);Xsgs=zeros(n,1);Xold=zeros(n,1);
%终止条件等
tol=10^(-5);endtimes=100;epsilon=10^(-12);
ratioRF=[0:endtimes;zeros(1,endtimes+1)];
ratioRsgs=[0:endtimes;zeros(1,endtimes+1)];
ratioRSORFGS=[0:endtimes;zeros(1,endtimes+1)];
ratioRSORSGS=[0:endtimes;zeros(1,endtimes+1)];
XF=X0;Xsgs=X0;Xsorfgs=X0;Xsorsgs=X0;
resnorm0=sqrt(sum((A*X0-b).^2))+epsilon;
resnorm=sqrt(sum((A*XF-b).^2));
ratio=resnorm/resnorm0;ratioRF(2,1)=ratio;ratioRsgs(2,1)=ratio;
```

```
    ratioRSORFGS(2,1)=ratio;ratioRSORSGS(2,1)=ratio;

% Proceeding
%G-S-Forward sweep
for times=2:endtimes
%calculate X
for i=1:n
for j=1:i-1
b(i)=b(i)-A(i,j)*XF(j);
end
for j=i+1:n
b(i)=b(i)-A(i,j)*XF(j);
end
XF(i)=b(i)/A(i,i);
end
b=b1;
resnorm=sqrt(sum((A*XF-b).^2));
ratio=resnorm/resnorm0;
if ratio<tol
break
end
ratioRF(2,times)=ratio;
end
ind=find(ratioRF(2,:),1,'last');
ratioRF(:,ind+1:endtimes+1)=[];
subplot(1,2,1)
plot(ratioRF(1,:),log10(ratioRF(2,:)),'-*','linewidth',1.5)
xlabel('Iteration times','fontsize',14)
ylabel('Log(ratio)','fontsize',14)
title('G-S Forward sweep','fontsize',16)
hold on
str1=num2str(ratioRF(2,:)');text(ratioRF(1,:),log10(ratioRF(2,:)),str1,'
    linewidth',1.5);
line([ratioRF(1,ind),ratioRF(1,ind)], [-5,1], 'color', 'b','linewidth'
    ,1.5);
str2=num2str(ind-1);
text(ind-1,0,str2,'linewidth',1.5);
xlim([0,ind])

%SOR-FGS
```

```matlab
R=b1-A*X0;
for times=2:endtimes
%calculate deltaX
Xold=Xsorfgs;
ie=1;
Xsorfgs(ie)=R(ie)/(D(ie,ie)/omega);
ie=2;
R(ie)=R(ie)-Xsorfgs(ie-1)*(L(ie,ie-1)/omega);
Xsorfgs(ie)=R(ie)/(D(ie,ie)/omega);
for ie=3:n
R(ie)=R(ie)-Xsorfgs(ie-1)*(L(ie,ie-1)/omega)-Xsorfgs(ie-2)*(L(ie,ie-2)/omega);
Xsorfgs(ie)=R(ie)/(D(ie,ie)/omega);
end
Xsorfgs=Xsorfgs+Xold;
resnorm=sqrt(sum((A*Xsorfgs-b1).^2));
ratio=resnorm/resnorm0;
if ratio<tol
break
end
R=b1-A*Xsorfgs;
ratioRSORFGS(2,times)=ratio;
end
ind=find(ratioRSORFGS(2,:),1,'last');
ratioRSORFGS(:,ind+1:endtimes+1)=[];
subplot(1,2,2)
plot(ratioRSORFGS(1,:),log10(ratioRSORFGS(2,:)),'-*','linewidth',1.5)
xlabel('Iteration times','fontsize',14)
ylabel('Log(ratio)','fontsize',14)
title('G-S Forward sweep(SOR)','fontsize',16)
hold on
str1=num2str(ratioRSORFGS(2,:)');text(ratioRSORFGS(1,:),log10(ratioRSORFGS(2,:)),str1,'linewidth',1.5);
line([ratioRSORFGS(1,ind),ratioRSORFGS(1,ind)], [-5,1], 'color', 'b','linewidth',1.5);
str2=num2str(ind-1);
text(ind-1,0,str2,'linewidth',1.5);
xlim([0,ind])

%SGS
```

```
for times=2:endtimes
%calculate X Forward sweep
for i=1:n
for j=1:i-1
b(i)=b(i)-A(i,j)*Xsgs(j);
end
for j=i+1:n
b(i)=b(i)-A(i,j)*Xsgs(j);
end
Xsgs(i)=b(i)/A(i,i);
end
b=b1;
%Backward sweep
for i=n:-1:1
for j=n:-1:i+1
b(i)=b(i)-A(i,j)*Xsgs(j);
end
for j=i-1:-1:1
b(i)=b(i)-A(i,j)*Xsgs(j);
end
Xsgs(i)=b(i)/A(i,i);
end
b=b1;
resnorm=sqrt(sum((A*Xsgs-b).^2));
ratio=resnorm/resnorm0;
if ratio<tol
break
end
ratioRsgs(2,times)=ratio;
end
ind=find(ratioRsgs(2,:),1,'last');
ratioRsgs(:,ind+1:endtimes+1)=[];
figure
subplot(1,2,1)
plot(ratioRsgs(1,:),log10(ratioRsgs(2,:)),'-*','linewidth',1.5)
xlabel('Iteration times','fontsize',14)
ylabel('Log(ratio)','fontsize',14)
title('S-G-S','fontsize',16)
hold on
str1=num2str(ratioRsgs(2,:)');text(ratioRsgs(1,:),log10(ratioRsgs(2,:)),
```

```
    str1,'linewidth',1.5);
line([ratioRsgs(1,ind),ratioRsgs(1,ind)], [-5,1], 'color', 'b','
    linewidth',1.5);
str2=num2str(ind-1);
text(ind-1,0,str2,'linewidth',1.5);
xlim([0,ind])
hold off


%SOR-SGS
R=omega*(2-omega)*(b1-A*X0);
for times=2:endtimes
%calculate deltaX
%Forward sweep
Xold=Xsorsgs;
ie=1;
Xsorsgs(ie)=R(ie)/D(ie,ie);
ie=2;
R(ie)=R(ie)-Xsorsgs(ie-1)*(L(ie,ie-1)*omega);
Xsorsgs(ie)=R(ie)/D(ie,ie);
for ie=3:n
R(ie)=R(ie)-Xsorsgs(ie-1)*(L(ie,ie-1)*omega)-Xsorsgs(ie-2)*(L(ie,ie-2)*
    omega);
Xsorsgs(ie)=R(ie)/D(ie,ie);
end
%Backward sweep
R=D*Xsorsgs;
ie=n;
Xsorsgs(ie)=R(ie)/D(ie,ie);
ie=n-1;
R(ie)=R(ie)-Xsorsgs(ie+1)*(U(ie,ie+1)*omega);
Xsorsgs(ie)=R(ie)/D(ie,ie);
for ie=n-2:-1:1
R(ie)=R(ie)-Xsorsgs(ie+1)*(U(ie,ie+1)*omega)-Xsorsgs(ie+2)*(U(ie,ie+2)*
    omega);
Xsorsgs(ie)=R(ie)/D(ie,ie);
end
%此时 Xsorsgs 里面存储的是 deltaX
Xsorsgs=Xsorsgs+Xold;
resnorm=sqrt(sum((A*Xsorsgs-b1).^2));
```

```
ratio=resnorm/resnorm0;
if ratio<tol
break
end
R=omega*(2-omega)*(b1-A*Xsorsgs);
ratioRSORSGS(2,times)=ratio;
end
ind=find(ratioRSORSGS(2,:),1,'last');
ratioRSORSGS(:,ind+1:endtimes+1)=[];
subplot(1,2,2)
plot(ratioRSORSGS(1,:),log10(ratioRSORSGS(2,:)),'-*','linewidth',1.5)
xlabel('Iteration times','fontsize',14)
ylabel('Log(ratio)','fontsize',14)
title('S-G-S(SOR)','fontsize',16)
hold on
str1=num2str(ratioRSORSGS(2,:)');text(ratioRSORSGS(1,:),log10(
    ratioRSORSGS(2,:)),str1,'linewidth',1.5);
line([ratioRSORSGS(1,ind),ratioRSORSGS(1,ind)], [-5,1], 'color', 'b','
    linewidth',1.5);
str2=num2str(ind-1);
text(ind-1,0,str2,'linewidth',1.5);
xlim([0,ind])
```

subDG(P0)+DG(P0) L-U Iteration

```
function [Unumsolution,n]=subDGP0plusDGP0(Unit,CFL,endtau)
%Some basic paramater
endx=1;deltax=endx/Unit;numberx=endx/deltax+1;
tol=10^(-10);
nu=1;Lr=1/(2*pi);Tr=Lr^2/nu;
abslambda=sqrt(nu/Tr);deltatau=CFL*deltax/abslambda;%伪时间变量
B1=1;
C=[B1,0;0,B1/deltax];Mtau=[deltax,0;0,1/deltax];%此为推导出的 U=CV 中的 C
A=[abslambda,0;0,abslambda];epsilon=10^(-12);
R=zeros(2*Unit,1);
Rd=zeros(2,numberx-1);
Rb=zeros(2,numberx-1);
Fn=zeros(2,numberx);
X=zeros(2*Unit,1);


```

```
%构建 LHS
%Mtau/deltatau
LHS1=sparse(1:2:2*Unit-1,1:2:2*Unit-1,deltax/deltatau,2*Unit,2*Unit);
LHS1=LHS1+sparse(2:2:2*Unit,2:2:2*Unit,1/(deltax*deltatau),2*Unit,2*Unit);
%Rdomain
LHS2=-sparse(2:2:2*Unit,2:2:2*Unit,-1/(Tr*deltax),2*Unit,2*Unit);
%Rboundary
LHS3=zeros(2*Unit,2*Unit);
for iface=2:numberx-1
ieL=iface-1;
ieR=iface;
%diag
LHS3(2*ieL-1:2*ieL,2*ieL-1:2*ieL)=LHS3(2*ieL-1:2*ieL,2*ieL-1:2*ieL)+C'*[abslambda/2,-nu/(2*deltax);-1/(2*Tr),abslambda/(2*deltax)];
LHS3(2*ieR-1:2*ieR,2*ieR-1:2*ieR)=LHS3(2*ieR-1:2*ieR,2*ieR-1:2*ieR)-C'*[-abslambda/2,-nu/(2*deltax);-1/(2*Tr),-abslambda/(2*deltax)];
%upper
LHS3(2*ieL-1:2*ieL,2*ieR-1:2*ieR)=LHS3(2*ieL-1:2*ieL,2*ieR-1:2*ieR)+C'*[-abslambda/2,-nu/(2*deltax);-1/(2*Tr),-abslambda/(2*deltax)];
%lower
LHS3(2*ieR-1:2*ieR,2*ieL-1:2*ieL)=LHS3(2*ieR-1:2*ieR,2*ieL-1:2*ieL)-C'*[abslambda/2,-nu/(2*deltax);-1/(2*Tr),abslambda/(2*deltax)];
end
LHS3(2*1-1:2*1,2*1-1:2*1)=LHS3(2*1-1:2*1,2*1-1:2*1)-C'*([abslambda/2,-nu/2;-1/(2*Tr),abslambda/2]*[0,0;0,1]+[-abslambda/2,-nu/2;-1/(2*Tr),-abslambda/2])*C;
LHS3(2*(numberx-1)-1:2*(numberx-1),2*(numberx-1)-1:2*(numberx-1))=LHS3(2*(numberx-1)-1:2*(numberx-1),2*(numberx-1)-1:2*(numberx-1))+C'*([abslambda/2,-nu/2;-1/(2*Tr),abslambda/2]+[-abslambda/2,-nu/2;-1/(2*Tr),-abslambda/2]*[0,0;0,1])*C;

LHS=LHS1+LHS2+LHS3;

%取出我们所需要的 D
D=zeros(2*Unit,2*Unit);
for iface=2:numberx
ieL=iface-1;
D(2*ieL-1:2*ieL,2*ieL-1:2*ieL)=LHS(2*ieL-1:2*ieL,2*ieL-1:2*ieL);
end
```

```
%取出我们所需要的 L
L=zeros(2*Unit,2*Unit);
for iface=2:numberx-1
ieR=iface;
ieL=iface-1;
L(2*ieR-1:2*ieR,2*ieL-1:2*ieL)=LHS(2*ieR-1:2*ieR,2*ieL-1:2*ieL);
end

%取出我们所需要的 U
U=zeros(2*Unit,2*Unit);
for iface=2:numberx-1
ieR=iface;
ieL=iface-1;
U(2*ieL-1:2*ieL,2*ieR-1:2*ieR)= LHS(2*ieL-1:2*ieL,2*ieR-1:2*ieR);
end


%为循环所预设的一些量

Ucurrent=zeros(2,numberx-1);
Unext=zeros(2*Unit,1);
%initial condition set up
x=0;

for k=1:numberx-1
Ucurrent(1,k)=(x+deltax/2)^2-(x+deltax/2);
Ucurrent(2,k)=(2*(x+deltax/2)-1)*deltax;
x=x+deltax;
end

%Rdomain
x=0;
for k=1:numberx-1
Rd(1,k)=pi*(cos(pi*x)-cos(pi*(x+deltax)));
Rd(2,k)=-Ucurrent(2,k)/(Tr*deltax);
x=x+deltax;
end
%Rboundary
for iface=2:numberx-1
ieL=iface-1;
```

```
ieR=iface;
Fn(:,iface)=0.5*([-nu*Ucurrent(2,ieL)/deltax;-Ucurrent(1,ieL)/Tr]+[-nu*
    Ucurrent(2,ieR)/deltax;-Ucurrent(1,ieR)/Tr])-0.5*A*([Ucurrent(1,ieR);
    Ucurrent(2,ieR)/deltax]-[Ucurrent(1,ieL);Ucurrent(2,ieL)/deltax]);
Rb(:,ieL)=Rb(:,ieL)-C'*Fn(:,iface);
Rb(:,ieR)=Rb(:,ieR)+C'*Fn(:,iface);
end
Fn(:,1)=0.5*([-nu*Ucurrent(2,1)/deltax;0]+[-nu*Ucurrent(2,1)/deltax;-
    Ucurrent(1,1)/Tr])-0.5*A*([Ucurrent(1,1);Ucurrent(2,1)/deltax]-[0;
    Ucurrent(2,1)/deltax]);
Fn(:,numberx)=0.5*([-nu*Ucurrent(2,numberx-1)/deltax;-Ucurrent(1,numberx
    -1)/Tr]+[-nu*Ucurrent(2,numberx-1)/deltax;0])-0.5*A*([0;Ucurrent(2,
    numberx-1)/deltax]-[Ucurrent(1,numberx-1);Ucurrent(2,numberx-1)/
    deltax]);
Rb(:,1)=Rb(:,1)+C'*Fn(:,1);
Rb(:,numberx-1)=Rb(:,numberx-1)-C'*Fn(:,numberx);

%R 组装
for k=1:numberx-1
R(2*k-1:2*k,1)=Rd(:,k)+Rb(:,k);
end
%进行必要的向量等价转变
for k=1:numberx-1
Unext(2*k-1:2*k,1)=Ucurrent(:,k);
end

%循环迭代
for n=deltatau:deltatau:endtau
X=0;
b=R;
%Forward sweep
ie=1;
X(ie:ie+1,1)=D(ie:ie+1,ie:ie+1)\R(ie:ie+1,1);
for ie=2:numberx-1
R(2*ie-1:2*ie,1)=R(2*ie-1:2*ie,1)-L(2*ie-1:2*ie,2*(ie-1)-1:2*(ie-1))*X
    (2*(ie-1)-1:2*(ie-1),1);
X(2*ie-1:2*ie,1)=D(2*ie-1:2*ie,2*ie-1:2*ie)\R(2*ie-1:2*ie,1);
end
%Backward sweep
for ie=1:numberx-1
```

```
R(2*ie-1:2*ie,1)= D(2*ie-1:2*ie,2*ie-1:2*ie)*X(2*ie-1:2*ie,1);
end

ie=numberx-1;
X(2*ie-1:2*ie)=D(2*ie-1:2*ie,2*ie-1:2*ie)\R(2*ie-1:2*ie,1);
for ie=numberx-2:-1:1
R(2*ie-1:2*ie,1)=R(2*ie-1:2*ie,1)-U(2*ie-1:2*ie,2*(ie+1)-1:2*(ie+1))*X(2*(ie+1)-1:2*(ie+1),1);
X(2*ie-1:2*ie,1)=D(2*ie-1:2*ie,2*ie-1:2*ie)\R(2*ie-1:2*ie,1);
end
if max(X)<tol
break
end
Unext=Unext+X;
Rd=zeros(2,numberx-1);
Rb=zeros(2,numberx-1);
for k=1:numberx-1
Ucurrent(:,k)=Unext(2*k-1:2*k,1);
end
%Rdomain
x=0;
for k=1:numberx-1
Rd(1,k)=pi*(cos(pi*x)-cos(pi*(x+deltax)));
Rd(2,k)=-Ucurrent(2,k)/(Tr*deltax);
x=x+deltax;
end
%Rboundary
for iface=2:numberx-1
ieL=iface-1;
ieR=iface;
Fn(:,iface)=0.5*([-nu*Ucurrent(2,ieL)/deltax;-Ucurrent(1,ieL)/Tr]+[-nu*Ucurrent(2,ieR)/deltax;-Ucurrent(1,ieR)/Tr])-0.5*A*([Ucurrent(1,ieR);Ucurrent(2,ieR)/deltax]-[Ucurrent(1,ieL);Ucurrent(2,ieL)/deltax]);
Rb(:,ieL)=Rb(:,ieL)-C'*Fn(:,iface);
Rb(:,ieR)=Rb(:,ieR)+C'*Fn(:,iface);
end

Fn(:,1)=0.5*([-nu*Ucurrent(2,1)/deltax;0]+[-nu*Ucurrent(2,1)/deltax;-Ucurrent(1,1)/Tr])-0.5*A*([Ucurrent(1,1);Ucurrent(2,1)/deltax]-[0;Ucurrent(2,1)/deltax]);
```

```
Fn(:,numberx)=0.5*([-nu*Ucurrent(2,numberx-1)/deltax;-Ucurrent(1,numberx-1)/Tr]+[-nu*Ucurrent(2,numberx-1)/deltax;0])-0.5*A*([0;Ucurrent(2,numberx-1)/deltax]-[Ucurrent(1,numberx-1);Ucurrent(2,numberx-1)/deltax]);
Rb(:,1)=Rb(:,1)+C'*Fn(:,1);
Rb(:,numberx-1)=Rb(:,numberx-1)-C'*Fn(:,numberx);

%R 组装
for k=1:numberx-1
R(2*k-1:2*k,1)=Rd(:,k)+Rb(:,k);
end

for k=1:numberx-1
Unext(2*k-1:2*k,1)=Ucurrent(:,k);
end
end
Unumsolution(1,:)=Ucurrent(1,:);Unumsolution(2,:)=Ucurrent(2,:)/deltax;
end
```