Zach Swain 5/7/17

Math 353

Computer Project 2

**2.5 c2**

Function definition

function [a,b] = sparsesetup1(n)

%where n is system size, a and b are sparse matrices

e=ones(n,1);

a=spdiags([e 2\*e e],-1:1 ,n, n);

b=zeros(n,1);

b(1)=1;

b(n)=-1;

b(2:n-1)=0;

end

Function definition

function [m,fe,be] = jacobi\_2\_que(a,b,tol)

%where a is external sparse, rhs b, and tolerance; m iterations,

%forward error, backward error

n=length(b);

d=diag(a);

r=a-diag(d);

xc=zeros(n,1);

p=zeros(n,1);

c=zeros([n,1]);

e=zeros([n,1]);

n1=0;

err=0;

relerr=1;

while (relerr>tol)

xc=(b-r\*xc)./d;

err=abs(norm(xc-p,inf));

relerr=err/(norm(xc)+eps);

p=xc;

n1=n1+1;

end

x=xc;

m=n1;

for i=1:n

for j=1:n

if (mod(j,2)==0)

xa(j)=-1;

else

xa(j)=1;

end

c(i)=c(i)+a(i,j)\*xa(j);

e(i)=e(i)+a(i,j)\*xc(j);

end

end

for i=1:n

dif(i)=abs(xa(i)-xc(i));

dif2(i)=abs(e(i)-c(i));

end

fe=max(dif,[],2);

be=max(dif2,[],2);

end

Command Window

>> [a,b]=sparsesetup1(100);

>> [m,fe,be]=jacobi\_2\_que(a,b,0.00025)

m =

447

fe =

0.9662

be =

0.0022

Number of steps = 447

Backward Error = .0022

**2.5 c6**

**a)**

Function definition – use same as in c2

function [a,b] = sparsesetup1(n)

%where n is system size, a and b are sparse matrices

e=ones(n,1);

a=spdiags([e 2\*e e],-1:1 ,n, n);

b=zeros(n,1);

b(1)=1;

b(n)=-1;

b(2:n-1)=0;

end

Function definition

function [m,be] = gauss\_seidel(a,b,tol)

%where a is external sparse, rhs b, and tolerance; m iterations, forward

%error, backward error

n=length (b);

p=zeros(n,1);

c=zeros([n,1]);

e=zeros([n,1]);

n1=0;

err=0;

relerr=1;

while (relerr>tol)

for j=1:n

if j==n

x(1)=(b(1)-a(1,2:n)\*p(2:n))/a(1,1);

elseif j==n

x(n)=(b(n)-a(n,1:n-1)\*(x(1:n,1))')/a(n,n);

else

x(j)=(b(j)-a(j,1:j-1)\*(x(1:j-1))'-a(j,j+1:n)\*p(j+1:n,1))/a(j,j);

end

end

err=abs(norm(x'-p));

relerr=err/(norm(x)+eps);

p=x';

n1=n1+1;

end

x=x';

m=n1;

for i=1:n

for j=1:n

if (mod(j,2)==0)

xa(j)=-1;

else

xa(j)=1;

end

c(i)=c(i)+a(i,j)\*xa(j);

e(i)=e(i)+a(i,j)\*x(j);

end

end

for i=1:n

dif(i)=abs(xa(i)-x(i));

dif2(i)=abs(e(i)-c(i));

end

fe=max(dif,[],2);

be=max(dif2,[],2);

end

Command Window

>> [a,b]=sparsesetup1(100);

>> [m,be]=gauss\_seidel(a,b,0.0005)

m =

959

be =

.0005

Number of steps = 959

Backward Error = .0005

**b)**

Function Definition

function [m,be] = SOR\_1(a,b,tol)

%where a is external sparse, rhs b, and tolerance; m iterations, backward

%error

n=size(a,1);

k=1;

x=zeros(n,1);

xc=zeros(n,1);

w=1.5;

c=zeros([n,1]);

e=zeros([n,1]);

n1=0;

err=0;

relerr=1;

D=diag(diag(a));

L=tril(-a,-1);

U=triu(-a,1);

tw=inv(D-w\*L)\*((1-w)\*D+w\*U);

cw=w\*inv(D-w\*L)\*b;

while (relerr>tol)

x(:,k+1)=tw\*x(:,k)+cw;

err=abs(norm(x(:,k+1)-x(:,k)));

n1=n1+1;

k=k+1;

end

xc=x(:,k);

m=n1;

for i=1:n

for j=1:n

if (mod(j,2)==0)

xa(j)=-1;

else

xa(j)=1;

end

c(i)=c(i)+a(i,j)\*xa(j);

e(i)=e(i)+a(i,j)\*xc(j);

end

end

for i=1:n

dif(i)=abs(xa(i)-xc(i));

dif2(i)=abs(e(i)-c(i));

end

fe=max(dif,[],2);

be=max(dif2,[],2);

end

Command Window

>> [a,b]=sparsesetup1(100);

>> [m,be]=SOR\_1(a,b,0.0001)

m =

959

be =

.0005

Number of steps = 959

Backward Error = .0005