```
clc
clear all;
M=39;
dx= 1/(M+1);
A= zeros(M,M);
B= 6.*dx.^3.*(1:39)';
coeff=[1 -2 1];
cc=1;
A(1,1:2)= [-2 1];
A(end,end-1:end) = [1 -2];


B(1)=B(1)-0;
B(end)=B(end)-1;

for r = 2:M-1
    A(r,cc : cc+2)= coeff;
    cc= cc+1;
end

u= linsolve(A,B);
x= dx.*(1:39);
tol=1e-8;
```

POINT-JACOBI METHOD

```
Res1=1;
v=zeros(M,1);
% v1=v;
itr1=1;
v1orders=[];
v2orders=[];
v3orders=[];
while(Res1(itr1)>tol)
    v1=v;
    for i= 1:size(A)
        sum=0;
        for j=1:size(A)
            if j~=i
        sum=sum + A(i,j).*v1(j);
            end
        end
        v(i)=(1./A(i,i)).*(B(i)-sum);
    end
    itr1=itr1+1;
    err=norm(B-A*v);
  Res1=[Res1,err./norm(B)];

%   v1=v;
if Res1(itr1)<1e-3 && isempty(v3orders)
    v3orders=v;
elseif Res1(itr1)<1e-2 && isempty(v2orders)
    v2orders=v;
elseif Res1(itr1)<1e-1 && isempty(v1orders)
```

```
    v1orders=v;
end
end
```

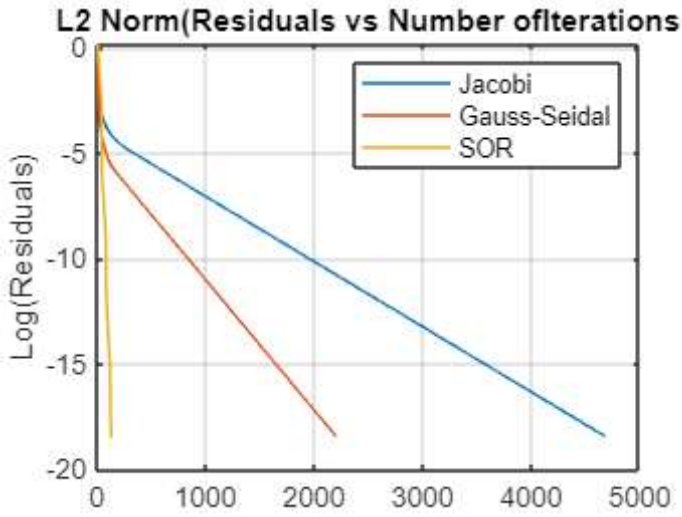
GAUSS SEIDAL METHOD

```
Res2=1;
w=zeros(M,1);
w1=w;
w1orders=[];
w2orders=[];
w3orders=[];


itr2=1;
while(Res2(itr2)>tol)

    for i= 1:size(A)
        sum=0;
        for j=1:(i-1)
        sum=sum + A(i,j).*w(j);
        end
        for j=i+1:size(A)
        sum=sum+A(i,j).*w1(j);
        end
        w(i)=(1./A(i,i)).*(B(i)-sum);

    end
    itr2=itr2+1;
    err=norm(B-A*w);
    Res2=[Res2,err./norm(B)];
    w1=w;
    if Res2(itr2)<1e-3 && isempty(w3orders)
    w3orders=w;
elseif Res2(itr2)<1e-2 && isempty(w2orders)
    w2orders=w;
elseif Res2(itr2)<1e-1 && isempty(w1orders)
    w1orders=w;
    end
end
```

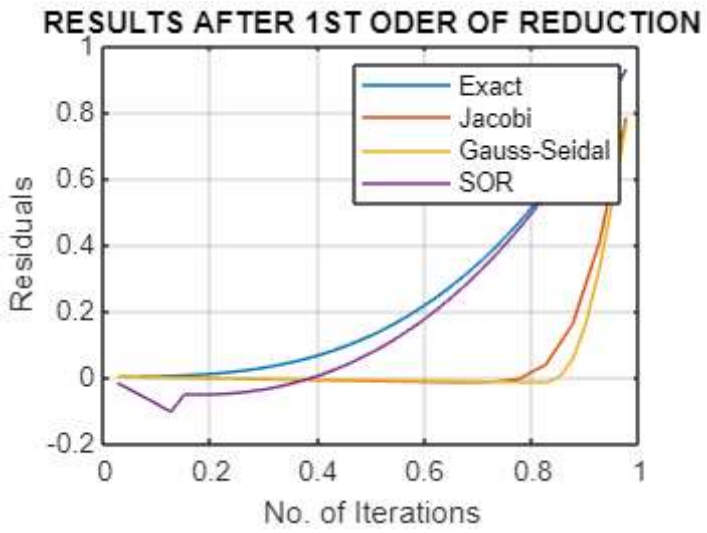
SOR METHOD

```
omega=2./(1 + sin(pi./(M+1)));
Res3=1;
itr3=1;
y=zeros(M,1);
y1=y;
y1orders=[];
y2orders=[];
y3orders=[];
while(Res3(itr3)>tol)

    for i= 1:size(A)
        sum=0;
        for j=1:(i-1)
```

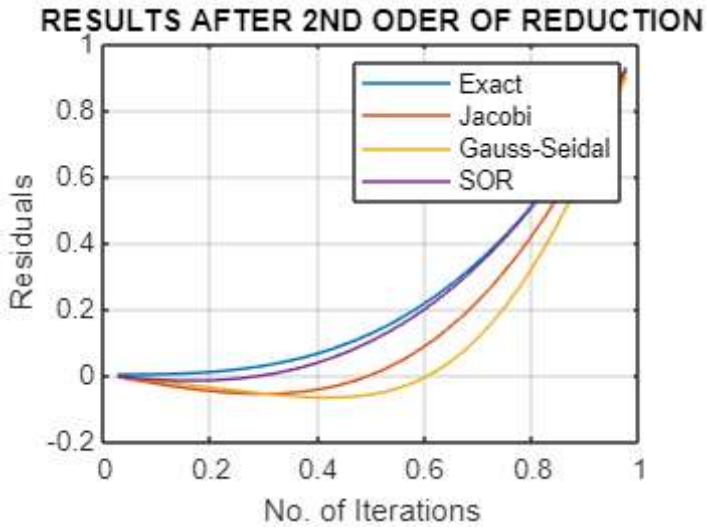
```
        sum=sum + A(i,j).*y(j);
        end
        for j=i+1:size(A)
        sum=sum+A(i,j).*y1(j);
        end
        y(i)=((1-omega).*y(i))+(omega./A(i,i)).*(B(i)-sum);

    end
    itr3=itr3+1;
    err=norm(B-A*y);
    Res3=[Res3,err./norm(B)];
    y1=y;
    if Res3(itr3)<1e-3 && isempty(y3orders)
    y3orders=y;
elseif Res3(itr3)<1e-2 && isempty(y2orders)
    y2orders=y;
elseif Res3(itr3)<1e-1 && isempty(y1orders)
    y1orders=y;
    end
end


%%ERROR VALUES
plot(1:itr1,log(Res1),1:itr2,log(Res2),1:itr3,log(Res3))
grid on
legend({'Jacobi','Gauss-Seidal','SOR'})
title('L2 Norm(Residuals vs Number ofIterations')
ylabel('Log(Residuals)')
```




```
%% 1st ODER OF REDUCTION
plot(x,x.^3,x,v1orders,x,w1orders,x,y1orders)
legend({'Exact','Jacobi','Gauss-Seidal','SOR'})
grid on
title('RESULTS AFTER 1ST ODER OF REDUCTION')
xlabel('No. of Iterations')
ylabel('Residuals')
```




```
%% 2nd ODER OF REDUCTION
plot(x,x.^3,x,v2orders,x,w2orders,x,y2orders)
legend({'Exact','Jacobi','Gauss-Seidal','SOR'})
grid on
title('RESULTS AFTER 2ND ODER OF REDUCTION')
xlabel('No. of Iterations')
ylabel('Residuals')
```




```
%% 3rd ODER OF REDUCTION
plot(x,x.^3,x,v3orders,x,w3orders,x,y3orders)
legend({'Exact','Jacobi','Gauss-Seidal','SOR'})
grid on
title('RESULTS AFTER 3RD ODER OF REDUCTION')
xlabel('No. of Iterations')
ylabel('Residuals')
```


RESULTS AFTER 3RD ODER OF REDUCTION
Residuals
No. of Iterations
Exact
Jacobi
Gauss-Seidal
SOR
1
0.8
0.6
0.4
0.2
0
-0.2
0
0.2
0.4
0.6
0.8
1