**Homework 2 (Partial Solution)      Posted on March 21, 1999**

**MEAM 502      Differential Equation Methods in Mechanics**

*1.      Solve the following matrix equation*

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

*by*

(1)      *Steepest Descent Method and/or Preconditioned SD Method*

Since the coefficient matrix **A** is symmetric, solving the system of linear equation is equivalent to the minimization problem:

$$\min_{\mathbf{x}} \; f(\mathbf{x}) \quad , \quad f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{x}^T\mathbf{b}$$

Suppose that **S** is a positive definite matrix. Then the steepest descent method is based on the iteration scheme:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha^{(k)}\mathbf{S}^{-1}\left(\mathbf{A}\mathbf{x}^{(k)} - \mathbf{b}\right)$$

where $\alpha^{(k)}$ is determined to be a solution of the one-dimensional minimization problem:

$$\min_{\alpha^{(k)}} \; f\left(\mathbf{x}^{(k+1)}\right).$$

Noting that

$$f\left(\mathbf{x}^{(k+1)}\right) = f\left(\mathbf{x}^{(k)} - \alpha^{(k)}\mathbf{S}^{-1}\mathbf{r}^{(k)}\right)$$
$$= \frac{1}{2}\left[\mathbf{x}^{(k)} - \alpha^{(k)}\mathbf{S}^{-1}\mathbf{r}^{(k)}\right]^T \mathbf{A}\left[\mathbf{x}^{(k)} - \alpha^{(k)}\mathbf{S}^{-1}\mathbf{r}^{(k)}\right] - \left[\mathbf{x}^{(k)} - \alpha^{(k)}\mathbf{S}^{-1}\mathbf{r}^{(k)}\right]^T \mathbf{b}$$

where the residual

$$\mathbf{r}^{(k)} = \mathbf{A}\mathbf{x}^{(k)} - \mathbf{b}$$

is defined, we have the necessary condition of the one-dimensional minimization problem:

$$\frac{\partial}{\partial \alpha^{(k)}} f\left(\mathbf{x}^{(k+1)}\right) = -\left[\mathbf{S}^{-1}\mathbf{r}^{(k)}\right]^T \mathbf{A}\left[\mathbf{x}^{(k)} - \alpha^{(k)}\mathbf{S}^{-1}\mathbf{r}^{(k)}\right] + \left[\mathbf{S}^{-1}\mathbf{r}^{(k)}\right]^T \mathbf{b}$$
$$= -\left[\mathbf{S}^{-1}\mathbf{r}^{(k)}\right]^T \mathbf{r}^{(k)} + \alpha^{(k)}\left[\mathbf{S}^{-1}\mathbf{r}^{(k)}\right]^T \mathbf{A}\left[\mathbf{S}^{-1}\mathbf{r}^{(k)}\right] = 0$$
$$\Rightarrow$$
$$\alpha^{(k)} = \frac{\left[\mathbf{S}^{-1}\mathbf{r}^{(k)}\right]^T \mathbf{r}^{(k)}}{\left[\mathbf{S}^{-1}\mathbf{r}^{(k)}\right]^T \mathbf{A}\left[\mathbf{S}^{-1}\mathbf{r}^{(k)}\right]}$$

Suppose that **A** is not symmetric. That is, we cannot use the functional f we have defined in above. We must use the least square type functional:

$$f\left(\mathbf{x}^{(k+1)}\right) = f\left(\mathbf{x}^{(k)} - \alpha^{(k)}\mathbf{S}^{-1}\mathbf{r}^{(k)}\right)$$
$$= \frac{1}{2}\left\{\mathbf{A}\left[\mathbf{x}^{(k)} - \alpha^{(k)}\mathbf{S}^{-1}\mathbf{r}^{(k)}\right] - \mathbf{b}\right\}^T \left\{\mathbf{A}\left[\mathbf{x}^{(k)} - \alpha^{(k)}\mathbf{S}^{-1}\mathbf{r}^{(k)}\right] - \mathbf{b}\right\}$$

and then

$$\frac{\partial}{\partial \alpha^{(k)}} f\left(\mathbf{x}^{(k+1)}\right) = \left\{\mathbf{A}\mathbf{S}^{-1}\mathbf{r}^{(k)}\right\}^T \left\{\mathbf{r}^{(k)} - \alpha^{(k)}\mathbf{A}\mathbf{S}^{-1}\mathbf{r}^{(k)}\right\} = 0$$
$$\Leftrightarrow$$
$$\alpha^{(k)} = \frac{\left\{\mathbf{A}\mathbf{S}^{-1}\mathbf{r}^{(k)}\right\}^T \mathbf{r}^{(k)}}{\left\{\mathbf{A}\mathbf{S}^{-1}\mathbf{r}^{(k)}\right\}^T \left\{\mathbf{A}\mathbf{S}^{-1}\mathbf{r}^{(k)}\right\}}$$

*(2) Conjugate Gradient Method and/or Pre-conditioned CG Method*

Conjugate gradient method is based on the iteration scheme

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)}\left(\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\right) + \beta^{(k)}\mathbf{S}^{-1}\left(\mathbf{A}\mathbf{x}^{(k)} - \mathbf{b}\right)$$

where the parameters $\alpha^{(k)}, \beta^{(k)}$ are the solutions of the two-dimensional minimization problem

$$\min_{\alpha^{(k)},\beta^{(k)}} f\left(\mathbf{x}^{(k+1)}\right)$$

Noting that

$$f\left(\mathbf{x}^{(k+1)}\right) = \frac{1}{2}\left(\mathbf{A}\mathbf{x}^{(k+1)} - \mathbf{b}\right)^T\left(\mathbf{A}\mathbf{x}^{(k+1)} - \mathbf{b}\right)$$
$$= \frac{1}{2}\left(\mathbf{A}\left(\mathbf{x}^{(k)} + \alpha^{(k)}\mathbf{d}^{(k)} + \beta^{(k)}\mathbf{S}^{-1}\mathbf{r}^{(k)}\right) - \mathbf{b}\right)^T\left(\mathbf{A}\left(\mathbf{x}^{(k)} + \alpha^{(k)}\mathbf{d}^{(k)} + \beta^{(k)}\mathbf{S}^{-1}\mathbf{r}^{(k)}\right) - \mathbf{b}\right)$$

where

$$\mathbf{r}^{(k)} = \mathbf{A}\mathbf{x}^{(k)} - \mathbf{b} \quad \text{and} \quad \mathbf{d}^{(k)} = \mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}$$

is the residual, we have

$$\frac{\partial f\left(\mathbf{x}^{(k+1)}\right)}{\partial \alpha^{(k)}} = \left(\mathbf{Ad}^{(k)}\right)^T \left(\mathbf{r}^{(k)} + \alpha^{(k)}\mathbf{Ad}^{(k)} + \beta^{(k)}\mathbf{AS}^{-1}\mathbf{r}^{(k)}\right) = 0$$

$$\frac{\partial f\left(\mathbf{x}^{(k+1)}\right)}{\partial \beta^{(k)}} = \left(\mathbf{AS}^{-1}\mathbf{r}^{(k)}\right)^T \left(\mathbf{r}^{(k)} + \alpha^{(k)}\mathbf{Ad}^{(k)} + \beta^{(k)}\mathbf{AS}^{-1}\mathbf{r}^{(k)}\right) = 0$$

$$\Leftrightarrow$$

$$\begin{bmatrix} \left(\mathbf{Ad}^{(k)}\right)^T \mathbf{Ad}^{(k)} & \left(\mathbf{Ad}^{(k)}\right)^T \mathbf{AS}^{-1}\mathbf{r}^{(k)} \\ \left(\mathbf{AS}^{-1}\mathbf{r}^{(k)}\right)^T \mathbf{Ad}^{(k)} & \left(\mathbf{AS}^{-1}\mathbf{r}^{(k)}\right)^T \mathbf{AS}^{-1}\mathbf{r}^{(k)} \end{bmatrix} \begin{Bmatrix} \alpha^{(k)} \\ \beta^{(k)} \end{Bmatrix} = -\begin{Bmatrix} \left(\mathbf{Ad}^{(k)}\right)^T \mathbf{r}^{(k)} \\ \left(\mathbf{AS}^{-1}\mathbf{r}^{(k)}\right)^T \mathbf{r}^{(k)} \end{Bmatrix}$$

$$\Rightarrow$$

$$\begin{Bmatrix} \alpha^{(k)} \\ \beta^{(k)} \end{Bmatrix} = -\begin{bmatrix} \left(\mathbf{Ad}^{(k)}\right)^T \mathbf{Ad}^{(k)} & \left(\mathbf{Ad}^{(k)}\right)^T \mathbf{AS}^{-1}\mathbf{r}^{(k)} \\ \left(\mathbf{AS}^{-1}\mathbf{r}^{(k)}\right)^T \mathbf{Ad}^{(k)} & \left(\mathbf{AS}^{-1}\mathbf{r}^{(k)}\right)^T \mathbf{AS}^{-1}\mathbf{r}^{(k)} \end{bmatrix}^{-1} \begin{Bmatrix} \left(\mathbf{Ad}^{(k)}\right)^T \mathbf{r}^{(k)} \\ \left(\mathbf{AS}^{-1}\mathbf{r}^{(k)}\right)^T \mathbf{r}^{(k)} \end{Bmatrix}$$

*(3) Gauss Elimination Method such as LU decomposition or using MATLAB*

*Where*

$$\mathbf{A} = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 3 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -2 & 4 & -2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 4 & -2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 2 \end{bmatrix}, \quad \mathbf{b} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -0.1 \\ -0.2 \\ -0.2 \\ -0.2 \\ -0.2 \\ -0.1 \end{Bmatrix}$$

Since the above problem possesses a symmetric coefficient matrix, we shall consider a unsymmetric matrix:

$$\mathbf{A} = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 3 & -\mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -2 & 4 & -2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 4 & -2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 2 \end{bmatrix}, \quad \mathbf{b} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -0.1 \\ -0.2 \\ -0.2 \\ -0.2 \\ -0.2 \\ -0.1 \end{Bmatrix}$$

Using the following MATLAB program

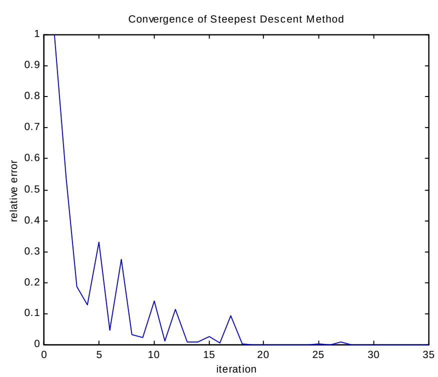
```
% Homework #2(1)_99W
%MEAM 502      Winter 1999
%
____________________________________________________________________________
____________
%
% set up the coefficient matrix A and the right hand side b
A=zeros(10);
b=zeros(10,1);
for i=1:10
   A(i,i)=2;
   if i>1, A(i-1,i)=-1;, end
   if i<10, A(i+1,i)=-1;, end
end
A(7,7)=3;A(7,8)=-1;A(8,7)=-2;A(8,8)=4;A(8,9)=-2;A(9,8)=-2;A(9,9)=4;A(9,10)=-2;A(10,9)=-
2;A(10,10)=2;
A
b=[0,0,0,0,-0.1,-0.2,-0.2,-0.2,-0.2,-0.1]'
%
% set up a preconditioned matrix S
```

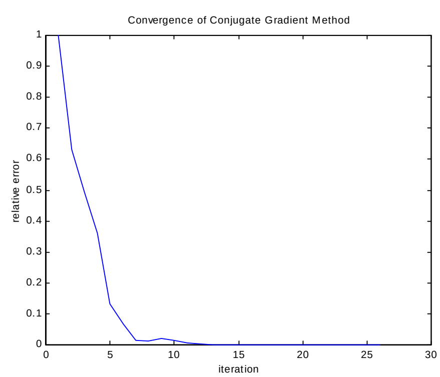
```
for i=1:10
    S(i,i)=A(i,i);
end
%S=(A'+A)/2;
Sinv=inv(S);
%
% steepest descent method
%
omega=1;
xk=zeros(10,1);
error=1;
iteration=0;
tolerance=10^-5;
maxiter=200;
errorh=[];
alfa=[];
while error>tolerance
    rk=A*xk-b;
    ASr=A*Sinv*rk;
    alfak=(ASr'*rk)/(ASr'*ASr);
    xk1=xk-omega*alfak*Sinv*rk;
    iteration=iteration+1
    error=sqrt((xk1-xk)'*(xk1-xk))/sqrt(xk1'*xk1)
    errorh(iteration)=error;
    alfa(iteration)=alfak;
    xk=xk1;
    if iteration>maxiter, break, end
end
xk
plot(errorh)
xlabel('iteration')
ylabel('relative error')
title('Convergence of Steepest Descent Method')
```

```
pause
%
% Conjugate Gradient Method
%
omega=1;
xk=zeros(10,1);
xkm1=xk;
error=1;
iteration=0;
tolerance=10^-5;
maxiter=200;
errorh=[];
alfa=[];
beta=[];
while error>tolerance
    rk=A*xk-b;
    dk=xk-xkm1;
    xkm1=xk;
    ASr=A*Sinv*rk;
    Ad=A*dk;
    CM=[Ad'*Ad,Ad'*ASr;ASr'*Ad,ASr'*ASr];
    alfabeta=-pinv(CM)*[Ad'*rk;ASr'*rk];
    alfak=alfabeta(1);
    betak=alfabeta(2);;
    xk1=xk+alfak*dk+betak*Sinv*rk;
    iteration=iteration+1
    error=sqrt((xk1-xk)'*(xk1-xk))/sqrt(xk1'*xk1)
    errorh(iteration)=error;
    alfa(iteration)=alfak;
    beta(iteration)=betak;
    xk=xk1;
    if iteration>maxiter, break, end
end
```

```
xk
plot(errorh)
xlabel('iteration')
ylabel('relative error')
title('Convergence of Conjugate Gradient Method')
%
% Direct Method
%
x=inv(A)*b
```

we shall solve the problem as follows:




At 35th iteration, the steepest descent method provides the solution

xk =

-0.1437
-0.2874
-0.4311
-0.5749
-0.7186
-0.7624
-0.6062
-0.8561

-1.0061
-1.0561

At the 26th iteration, the conjugate gradient method provides a solution

xk =

-0.1437
-0.2875
-0.4312
-0.5749
-0.7187
-0.7624
-0.6062
-0.8561
-1.0061
-1.0561

while the direct Gaussian elimination schemes the solution

x =

-0.1438
-0.2875
-0.4312
-0.5750
-0.7188
-0.7625
-0.6063
-0.8562
-1.0062
-1.0563

Thus, you should work out for the case of symmetric coefficients by defining the functional

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T \mathbf{A}\mathbf{x} - \mathbf{x}^T \mathbf{b}\,.$$

*2. Solve the following nonlinear differential equation*

$$-\frac{d}{dx}\left\{\left(\frac{1}{\sqrt{1+(du/dx)^2}}\frac{du}{dx}\right)\right\} + u^4 = \sin(\pi x) \quad in \quad (0,1)$$

*with the boundary condition* $u(0) = u(1) = 0$*, by using the Newton method or modified Newton method, after it would be approximated by FDM, FEM, or weighted residual methods.*

We shall derive the Newton scheme of the above nonlinear differential equation by going back to the principle. To this end, we shall approximate the k+1 approximation of the solution:

$$u \approx u^{(k+1)} = u^{(k)} + \Delta u^{(k)}$$

Substituting this into the nonlinear differential equation

$$-\frac{d}{dx}\left\{\left(\frac{1}{\sqrt{1+\left(du^{(k+1)}/dx + d\Delta u^{(k+1)}/dx\right)^2}}\left(\frac{du^{(k)}}{dx} + \frac{d\Delta u^{(k)}}{dx}\right)\right)\right\} + \left(u^{(k)} + \Delta u^{(k)}\right)^4 = \sin(\pi x)$$

Neglecting the higher order terms than the linear in $\Delta u^{(k)}$, we have

$$-\frac{d}{dx}\left\{\frac{\frac{du^{(k)}}{dx}}{\sqrt{1+\left(du^{(k+1)}/dx\right)^2}}\right\}+\left(u^{(k)}\right)^4$$

$$-\frac{d}{dx}\left\{\left(-\frac{\frac{du^{(k)}}{dx}}{\left\{\sqrt{1+\left(du^{(k+1)}/dx\right)^2}\right\}^3}+\frac{1}{\sqrt{1+\left(du^{(k+1)}/dx\right)^2}}\right)\frac{d\Delta u^{(k)}}{dx}\right\}+4\left(u^{(k)}\right)^3\Delta u^{(k)}$$

$$=\sin(\pi x)$$

that is, the increment $\Delta u^{(k)}$ for the k+1 approximation is a solution of the differential equation

$$-\frac{d}{dx}\left\{\left(-\frac{\frac{du^{(k)}}{dx}}{\left\{\sqrt{1+\left(du^{(k+1)}/dx\right)^2}\right\}^3}+\frac{1}{\sqrt{1+\left(du^{(k+1)}/dx\right)^2}}\right)\frac{d\Delta u^{(k)}}{dx}\right\}+4\left(u^{(k)}\right)^3\Delta u^{(k)}$$

$$=\frac{d}{dx}\left\{\frac{\frac{du^{(k)}}{dx}}{\sqrt{1+\left(du^{(k+1)}/dx\right)^2}}\right\}+\left(u^{(k)}\right)^4+\sin(\pi x)$$

Applying the finite difference approximation, we have

$$
-\frac{1}{\Delta x}\left\{\left(-\frac{\dfrac{u^{(k)}_{i+1}-u^{(k)}_{i}}{\Delta x}}{\left\{\sqrt{1+\left(\dfrac{u^{(k)}_{i+1}-u^{(k)}_{i}}{\Delta x}\right)^2}\right\}^3}+\frac{1}{\sqrt{1+\left(\dfrac{u^{(k)}_{i+1}-u^{(k)}_{i}}{\Delta x}\right)^2}}\right)\frac{\Delta u^{(k)}_{i+1}-\Delta u^{(k)}_{i}}{\Delta x}\right\}
$$

$$
+\frac{1}{\Delta x}\left\{\left(-\frac{\dfrac{u^{(k)}_{i}-u^{(k)}_{i-1}}{\Delta x}}{\left\{\sqrt{1+\left(\dfrac{u^{(k)}_{i}-u^{(k)}_{i-1}}{\Delta x}\right)^2}\right\}^3}+\frac{1}{\sqrt{1+\left(\dfrac{u^{(k)}_{i}-u^{(k)}_{i-1}}{\Delta x}\right)^2}}\right)\frac{\Delta u^{(k)}_{i-1}-\Delta u^{(k)}_{i-1}}{\Delta x}\right\}
$$

$$
+4\left(u^{(k)}_{i}\right)^3\Delta u^{(k)}_{i}
$$

$$
=\frac{1}{\Delta x}\left\{\frac{\dfrac{u^{(k)}_{i+1}-u^{(k)}_{i}}{\Delta x}}{\sqrt{1+\left(\dfrac{u^{(k)}_{i+1}-u^{(k)}_{i}}{\Delta x}\right)^2}}\right\}-\frac{1}{\Delta x}\left\{\frac{\dfrac{u^{(k)}_{i}-u^{(k)}_{i-1}}{\Delta x}}{\sqrt{1+\left(\dfrac{u^{(k)}_{i}-u^{(k)}_{i-1}}{\Delta x}\right)^2}}\right\}+\left(u^{(k)}_{i}\right)^4+\sin(\pi x_i)
$$

Rearranging this form, we have the finite difference equation for the increment $\Delta u^{(k)}$:

$$-\frac{a_{i+1}}{\Delta x^2}\Delta u^{(k)}{}_{i+1}+\left(\frac{a_{i+1}}{\Delta x^2}+\frac{a_{i-1}}{\Delta x^2}+k_i\right)\Delta u^{(k)}{}_i-\frac{a_{i-1}}{\Delta x^2}\Delta u^{(k)}{}_{i-1}=f_i$$

where

$$a_{i+1}=-\frac{\dfrac{u^{(k)}{}_{i+1}-u^{(k)}{}_i}{\Delta x}}{\left\{\sqrt{1+\left(\dfrac{u^{(k)}{}_{i+1}-u^{(k)}{}_i}{\Delta x}\right)^2}\right\}^3}+\frac{1}{\sqrt{1+\left(\dfrac{u^{(k)}{}_{i+1}-u^{(k)}{}_i}{\Delta x}\right)^2}}$$

$$a_{i-1}=-\frac{\dfrac{u^{(k)}{}_i-u^{(k)}{}_{i-1}}{\Delta x}}{\left\{\sqrt{1+\left(\dfrac{u^{(k)}{}_i-u^{(k)}{}_{i-1}}{\Delta x}\right)^2}\right\}^3}+\frac{1}{\sqrt{1+\left(\dfrac{u^{(k)}{}_i-u^{(k)}{}_{i-1}}{\Delta x}\right)^2}}$$

$$k_i=4\left(u^{(k)}{}_i\right)^3$$

$$f_i=\frac{1}{\Delta x}\left\{\frac{\dfrac{u^{(k)}{}_{i+1}-u^{(k)}{}_i}{\Delta x}}{\sqrt{1+\left(\dfrac{u^{(k)}{}_{i+1}-u^{(k)}{}_i}{\Delta x}\right)^2}}\right\}-\frac{1}{\Delta x}\left\{\frac{\dfrac{u^{(k)}{}_i-u^{(k)}{}_{i-1}}{\Delta x}}{\sqrt{1+\left(\dfrac{u^{(k)}{}_i-u^{(k)}{}_{i-1}}{\Delta x}\right)^2}}\right\}+\left(u^{(k)}{}_i\right)^4+\sin(\pi x_i)$$

Using the MATLAB program

```
% Homework #2 : Problem 2 / hw2p2
% MEAM 502      Winter 1999
%
% Newton's Method for a Nonlinear Differential Equation
% ______________________________________________________________________________
%
nx=21;
dx=1/(nx-1);
x=0:dx:1;
```

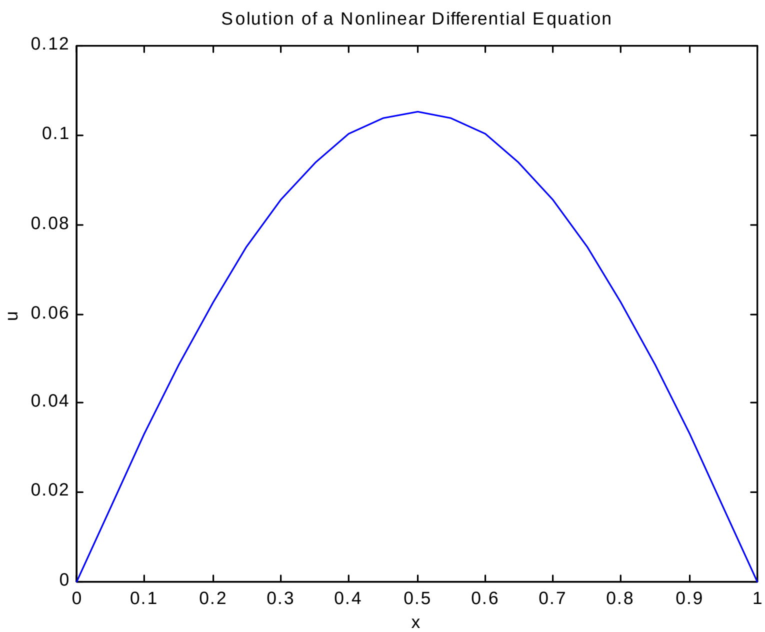
```
u=zeros(nx,1);
error=1;
errorh=[];
tolerance=10^-5;
iteration=0;
maxiter=20;
KM=zeros(nx);
bv=zeros(nx,1);
KM(1,1)=1;
KM(nx,nx)=1;
%
while error>tolerance
    %
    for i=2:nx-1
        duip1=(u(i+1)-u(i))/dx;
        duim1=(u(i)-u(i-1))/dx;
        aip1=-duip1/(sqrt(1+duip1^2))^3+1/sqrt(1+duip1^2);
        aim1=-duim1/(sqrt(1+duim1^2))^3+1/sqrt(1+duim1^2);
        ki=4*u(i)^3;
        bv(i)=(duip1/sqrt(1+duip1^2))/dx-(duim1/sqrt(1+duim1^2))/dx-u(i)^4+sin(pi*x(i));
        KM(i-1,i)=-aim1/dx^2;
        KM(i,i)=aim1/dx^2+aip1/dx^2+ki;
        KM(i+1,i)=-aip1/dx^2;
    end
    KM(1,2)=0;KM(nx,nx-1)=0;
    du=KM¥bv;
    u=u+du;
    iteration=iteration+1;
    error=sqrt(du'*du)/(u'*u);
    errorh(iteration)=error;
    if iteration>maxiter, break, end
    %
end
```

```
iteration
error
plot(x,u)
xlabel('x')
ylabel('u')
title('Solution of a Nonlinear Differential Equation')
```

we have convergence of the Newton's method:

```
iteration = 9
error = 2.6169e-006
```

and the solution




3. *Find the best approximation* $h_0 \in H$ *of a given function* $f(x) = (1 - x^2)\exp(x)$ *in the Sobolev*

*space* $V = H^1(-1,1) = \{v \mid v, \partial v \in L^2(-1,1)\}$ *with the inner product*

$$(u,v) = \int_{-1}^{1} \left\{ u(x)v(x) + \left(1 + \frac{1}{2}\sin(\pi x)\right)\partial u(x)\partial v(x) \right\} dx$$

*where* $H = \{v \in V \mid v(x) = c_0 + c_1 x + c_2 x^2 + c_3 x^3 + c_4 x^4 \ , c_i \in R \ , i = 0,1,2,3,4 \ \}$.

Let the best approximation $f_h \in H$ of f be represented by

$$f_h = \{1 \quad x \quad x^2 \quad x^3 \quad x^4\} \begin{Bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{Bmatrix} = \mathbf{N}^T \mathbf{f}$$

while an arbitrary element $v_h \in H$ be similarly represented by $v_h = \mathbf{N}^T \mathbf{v}$. Then the best approximation solution is determined by

$$(v_h, f - f_h) = 0 \quad , \quad \forall v_h \in H$$

Thus, we have

$$(v_h, f - f_h) = \int_{-1}^{1} \left\{ \mathbf{v}^T \mathbf{N}(f - \mathbf{N}^T \mathbf{f}) + \mathbf{v}^T \partial \mathbf{N}\left(1 + \frac{1}{2}\sin(\pi x)\right)(\partial f - \partial \mathbf{N}^T \mathbf{f}) \right\} dx = \mathbf{v}^T (\mathbf{b} - \mathbf{K}\mathbf{f}),$$

$$\mathbf{b} = \int_{-1}^{1} \left\{ \mathbf{N} f + \partial \mathbf{N}\left(1 + \frac{1}{2}\sin(\pi x)\right)\partial f \right\} dx$$

$$\mathbf{K} = \int_{-1}^{1} \left\{ \mathbf{N}\mathbf{N}^T + \partial \mathbf{N}\left(1 + \frac{1}{2}\sin(\pi x)\right)\partial \mathbf{N}^T \right\} dx$$

This means that the coefficient **f** can be obtained by solving the matrix equation

$$\mathbf{Kf} = \mathbf{b}$$

where **K** is a symmetric 5x5 matrix, and **b** is a five component vector.

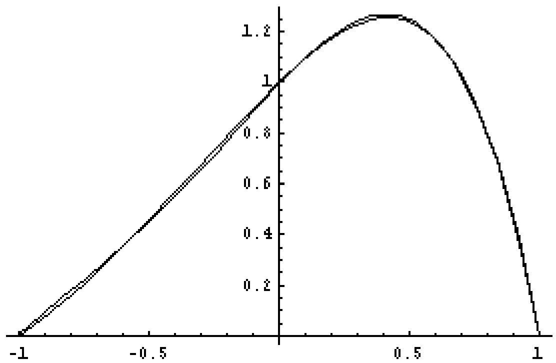
Using the following MATHEMATICA program

```
Nf={1,x,x^2,x^3,x^4};
dNf=D[Nf,x];
f=(1-x^2)*Exp[x];
df=D[f,x];
ax=1+0.5*Sin[Pi*x];
Km=Table[0,{i,1,5},{j,1,5}];
bv=Table[0,{i,1,5}];
Do[bv[[i]]=NIntegrate[Nf[[i]]*f+dNf[[i]]*ax*df,{x,-1,1}];
        Do[Km[[i,j]]=NIntegrate[Nf[[i]]*Nf[[j]]+dNf[[i]]*ax*dNf[[j]],{x,-1,1}],{j,1,
      5}],{i,1,5}]
fv=LinearSolve[Km,bv];
fh=Nf.fv
Plot[{fh,f},{x,-1,1}]
```

We have

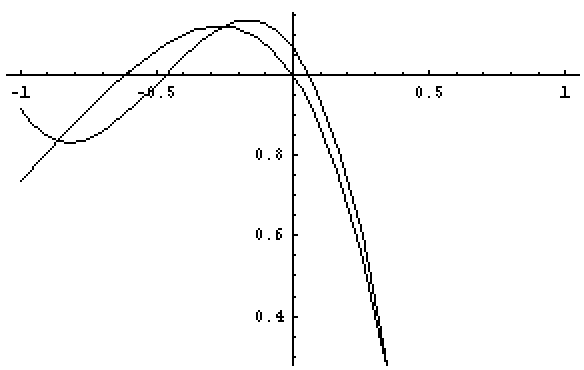
$$fh = 0.991919 + 1.06654\,x - 0.442395\,x^2 - 1.07011\,x^3 - 0.543474\,x^4$$

and




$dfh/dx = 1.06654 - 0.88479\,x - 3.21034\,x^2 - 2.1739\,x^3$

Approximation Error (seminorm) = 0.0972432

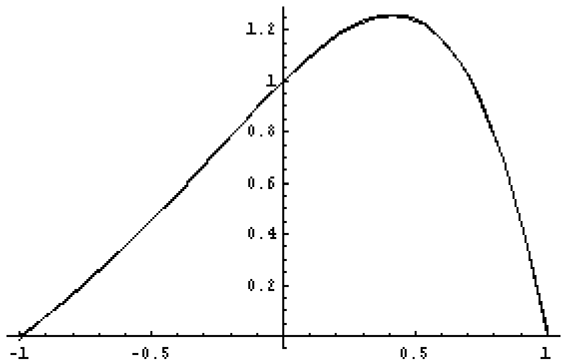



If we consider the best approximation by using the L2 inner product:

$$(u,v) = \int_{-1}^{1} \{u(x)v(x)\}dx$$

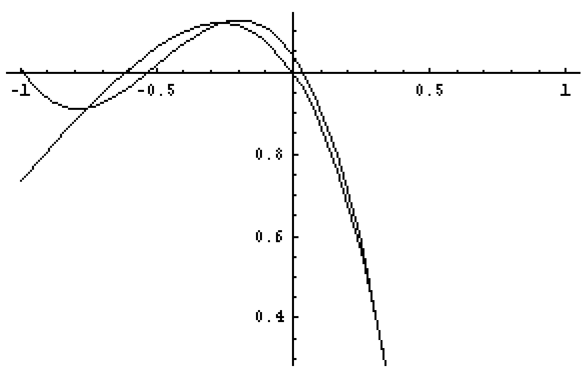
then the best approximation becomes

$fh = 0.999079 + 1.04023\,x - 0.480783\,x^2 - 1.01808\,x^3 - 0.515297\,x^4$




dfh/dx = 1.04023 - 0.961567 x - 3.05424 x$^2$ - 2.06119 x$^3$

Approximation Error (seminorm) = 0.12285




It is clear that the L2 inner product provides the approximation with larger approximation error in the seminorm of the Sobolev space $V$. Here the seminorm for the error is defined by

$$|u|_1 = \sqrt{\int_{-1}^{1} \partial u(x)^2 \, dx}$$