Answer ALL FOUR Questions

Question 1

Let $I = [0, 2]$ and $f(x) = (x - 1)^3$ for $x \in I$.

(a) Let $V_h$ be the continuous space of linear piecewise function $P_1(I)$ on $I$. Write a MATLAB code to compute the $L^2$-projection $P_h f \in V_h$ of $f$.

(b) Show that $\int_{\Omega} (f - P_h f) v dx = 0$ for all $v \in V_h$, if and only if $\int_{\Omega} (f - P_h f) \phi_i dx = 0$, for $i = 0, 1, \ldots, n$, where $\{\phi_i\} \subset V_h$ is set of hat functions.

Solution 1

(a) function [Pf]=L2Projector1D()
    n = 5; % number of subintervals
    h = 2/n; % mesh size
    x = 0:h:2; % mesh
    f=@(x) (x-1)^3;
    M = MassAssembler1D(x); % assemble mass
    b = LoadAssembler1D(x,f); % assemble load
    Pf = M\b; % solve linear system
end

function M = MassAssembler1D(x)
    n = length(x)-1; % number of subintervals
    M = zeros(n+1,n+1); % allocate mass matrix
    for i = 1:n % loop over subintervals
        h = x(i+1) - x(i); % interval length
        M(i,i) = M(i,i) + h/3; % add h/3 to M(i,i)
        M(i,i+1) = M(i,i+1) + h/6;
        M(i+1,i) = M(i+1,i) + h/6;
        M(i+1,i+1) = M(i+1,i+1) + h/3;
    end
end

function b = LoadAssembler1D(x,f)
    n = length(x)-1; b = zeros(n+1,1);
    for i = 1:n h = x(i+1) - x(i);
        b(i) = b(i) + f(x(i))*h/2;
        b(i+1) = b(i+1) + f(x(i+1))*h/2;
    end
end
(b) Assume that \( \int_\Omega (f - P_h f) v dx = 0 \) for all \( v \in V_h \) and \( \{ \phi_i \} \subset V_h \). Thus \( \phi_i \in V_h \) for \( i = 0, \ldots, n \) and therefore \( \int_\Omega (f - P_h f) \phi_i dx = 0 \). If \( \int_\Omega (f - P_h f) v dx = 0 \) is first assumed, we know that \( \{ \phi_i \} \) is a set of basis functions as such \( v = \sum_{i=1}^n \alpha_i \phi_i \) and so \( \int_\Omega (f - P_h f) v dx = \int_\Omega (f - P_h f) \sum \alpha_i \phi_i dx = 0 \).

Question 2

Consider the following one-dimensional boundary value problem

\[-u'' = 7, \ x \in I = [0, 1] \]

\[u(0) = 0, \ u(1) = 3\]

(a) Define a suitable finite element space \( V_h \)? Formulate a finite element method for this problem based on variational method.  

[3 Marks]

(b) Derive the discrete system of equations (the global stiffness matrix and the load vector) using a uniform mesh with 4 nodes.  

[7 Marks]

Solution 2

(a) Let's first consider the generalized problem of the BVP in the question

\[-(au')' = f, \ x \in I = [0, L] \]

\[au'(0) = \kappa_0 (u(0) - g_0) \]

\[-au'(L) = \kappa_L (u(L) - g_L) \]

It follows treatment under the variational method

\[\int_0^L f v dx = \int_0^L -(au')' v dx \]

\[= \int_0^L av'u dx - au'(L)v(L) + au'(0)v(0) \]

\[= \int_0^L av'u dx + \kappa_L (u(L) - g_L) + \kappa_0 (u(0) - g_0) \]

\[\int_0^L av'u dx + \kappa_L u(L) + \kappa_0 u(0) = \int_0^L f v dx + \kappa_L g_L + \kappa_0 g_0 \]

The suitable finite element space follows

\[V_h = \left\{ v : \|v\|_{L^2(I)} < \infty, \|v'\|_{L^2(I)} < \infty \right\} \]

The finite element problem is to find \( u_h \in V_h \) such that

\[\int_0^L av'u_h' dx + \kappa_L u_h(L) + \kappa_0 u_h(0) = \int_0^L f v dx + \kappa_L g_L + \kappa_0 g_0, \ \forall v \in V_h \]
and for specific BVP in the question, set \( a = 1, L = 1, \kappa_0 = \kappa_L = 10^6, g_0 = 0 \) and \( g_L = 3 \).

(b) A basis for \( V_h \) is given by a set of \( n + 1 \) hat functions \( \{ \phi_i \}_{i=0}^n \). Inserting the trial function

\[
u_h = \sum_{i=0}^n \xi_i \phi_i
\]

into the finite element problem and choosing \( v = \phi_i \) ends up with a linear system

\[
(A + R) \xi = b + r
\]

where the entries of the \( (n + 1) \times (n + 1) \) matrices \( A \) and \( R \), and the \( n + 1 \) vectors \( b \) and \( r \) are given by

\[
A_{ij} = \int_I a \phi_j' \phi_i' dx
\]

\[
R_{ij} = \kappa_L \phi_j(L) \phi_i(L) + \kappa_0 \phi_j(0) \phi_i(0)
\]

\[
b_i = \int_I f \phi_i dx
\]

\[
r_i = \kappa_L g_L \phi_i(L) + \kappa_0 g_0 \phi_i(0)
\]

and

\[
\phi_i = \begin{cases} 
(x - x_{i-1})/h_i, & \text{if } x \in I_i \\
(x_{i+1} - x)/h_{i+1}, & \text{if } x \in I_{i+1} \\
0, & \text{otherwise.}
\end{cases}
\]

Hence the derivative

\[
\phi_i' = \begin{cases} 
1/h_i, & \text{if } x \in I_i \\
-1/h_{i+1}, & \text{if } x \in I_{i+1} \\
0, & \text{otherwise.}
\end{cases}
\]

It can be shown that for four nodes the linear system for specific BVP where \( a = 1, \kappa_0 = \kappa_L = 10^6, g_0 = 0, g_1 = 3 \) and \( h_i = h = 1/3 \) is

\[
A + R = \begin{bmatrix}
3 + 10^6 & -3 & 0 & 0 \\
-3 & 6 & -3 & 0 \\
0 & -3 & 6 & -3 \\
0 & 0 & -3 & 3 + 10^6
\end{bmatrix}
\]

\[
b + r = \begin{bmatrix}
7/6 \\
7/3 \\
7/3 \\
7/6 + 3 \times 10^6
\end{bmatrix}
\]
Question 3

Consider the triangulation by Figure 1

(a) Write down the point matrix \( P \) and the connectivity matrix \( T \). Express the area of an arbitrary triangle in terms of its corner coordinates \((x_1, y_1), (x_2, y_2), (x_3, y_3)\).

(b) Determine the basis functions for piecewise linear functions on an arbitrary triangle with corner coordinates \((x_1, y_1), (x_2, y_2), (x_3, y_3)\).

Solution 3

(a) The point matrix is

\[
P = \begin{bmatrix}
0 & 0.5 & 1 & 0 & 0.5 & 1 & 0 & 0.5 & 1 \\
0 & 0 & 0 & 0.5 & 0.5 & 0.5 & 1 & 1 & 1 \\
\end{bmatrix}
\]

and the connectivity matrix is

\[
T = \begin{bmatrix}
1 & 2 & 1 & 2 & 4 & 5 & 4 & 5 \\
2 & 3 & 5 & 6 & 5 & 6 & 8 & 9 \\
5 & 6 & 4 & 5 & 8 & 9 & 7 & 8 \\
\end{bmatrix}
\]

Area for a triangle with corner coordinates \((x_1, y_1), (x_2, y_2), (x_3, y_3)\) is

\[
\triangle = \frac{1}{2} \begin{vmatrix}
1 & x_1 & y_1 \\
1 & x_2 & y_2 \\
1 & x_3 & y_3 \\
\end{vmatrix}
= \frac{1}{2} \left| x_1 y_2 + x_2 y_3 + x_3 y_1 - (x_1 y_3 + x_2 y_1 + x_3 y_2) \right|
\]
(b) Let $K$ be a triangle and let $P_1(K)$ be the space of linear functions on $K$, defined by

$$P_1(K) = \{ v : v = c_0 + c_1 x + c_2 y, (x, y) \in K, c_0, c_1, c_2 \in \mathbb{R} \}.$$ 

Suppose the evaluation at each corner $i = 1, 2, 3$ gives nodal values $\alpha_i = v(x_i, y_i)$ thus

$$\begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix}$$

Therefore in terms of nodal values

$$v = \begin{bmatrix} 1 & x & y \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \frac{1}{2\triangle} \begin{bmatrix} 1 & x & y \end{bmatrix} \begin{bmatrix} x_2y_3 - x_3y_2 & x_3y_1 - x_1y_3 & x_1y_2 - x_2y_1 \\ y_2 - y_3 & y_3 - y_1 & y_1 - y_2 \\ x_3 - x_2 & x_1 - x_3 & x_2 - x_1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix}$$

where the basis functions are

$$\lambda_1(x, y) = \frac{1}{2\triangle} [(x_2y_3 - x_3y_2) + (y_2 - y_3)x + (x_3 - x_2)y]$$

$$\lambda_2(x, y) = \frac{1}{2\triangle} [(x_3y_1 - x_1y_3) + (y_3 - y_1)x + (x_1 - x_3)y]$$

$$\lambda_3(x, y) = \frac{1}{2\triangle} [(x_1y_2 - x_2y_1) + (y_1 - y_2)x + (x_2 - x_1)y]$$

**Question 4**

Based on finite element method, write a MATLAB code to solve

$$-\Delta u = 0 \text{ in } \Omega = [-2, 2] \times [-2\pi, 2\pi]$$

with Dirichlet boundary conditions

$$u = \exp(x) \arctan(y) \text{ on } \partial \Omega.$$ 

Construct the appropriate geometry matrix and call `initmesh` function from the PDE-Toolbox to generate mesh. (Hint: Consider Robin boundary condition approximation) [10 Marks]
Solution 4

```matlab
function LaplaceSolver2D()
    a=@(x,y) 1;
g = Rectg(-2,-2*pi,2,2*pi);
    [p,e,t] = initmesh(g,'hmax',0.5);
    A = StiffnessAssembler2D(p,t,a);
kappa=@(x,y) 10^6;
gD=@(x,y) exp(x).*atan(y);
gN=@(x,y) 0;
    [R,r] = RobinAssembler2D(p,e,kappa,gD,gN);
    phi = (A+R)
    end
end

function r = Rectg(xmin,ymin,xmax,ymax)
    r=[2 xmin xmax ymin ymin 1 0;
      xmin xmax ymin ymin 1 0;
      xmin xmax ymax ymax 1 0;
      xmin xmin ymin ymax 1 0]';
end

function A = StiffnessAssembler2D(p,t,a)
    np = size(p,2); nt = size(t,2);
    A = sparse(np,np); % allocate stiffness matrix
    for K = 1:nt
        loc2glb = t(1:3,K); % local-to-global map
        x = p(1,loc2glb); % node x-coordinates
        y = p(2,loc2glb); % node y-
        [area,b,c] = HatGradients(x,y);
        xc = mean(x); yc = mean(y); % element centroid
        abar = a(xc,yc); % value of a(x,y) at centroid
        AK = abar*(b*b'...
                +c*c')*area; % element stiffness matrix
        A(loc2glb,loc2glb) = A(loc2glb,loc2glb) ...
                               + AK; % add element stiffnesses to A
    end
end

function [area,b,c] = HatGradients(x,y)
    area=polyarea(x,y);
    b=[y(2)-y(3); y(3)-y(1); y(1)-y(2)]/2/area;
    c=[x(3)-x(2); x(1)-x(3); x(2)-x(1)]/2/area;
end
```
function [R,r] = RobinAssembler2D(p,e,kappa,gD,gN)
R = RobinMassMatrix2D(p,e,kappa);
r = RobinLoadVector2D(p,e,kappa,gD,gN);
end

definition of Robin Mass Matrix

function R = RobinMassMatrix2D(p,e,kappa)
np = size(p,2); % number of nodes
ne = size(e,2); % number of boundary edges
R = sparse(np,np); % allocate boundary matrix
for E = 1:ne
    loc2glb = e(1:2,E); % boundary nodes
    x = p(1,loc2glb); % node x-coordinates
    y = p(2,loc2glb); % node y-coordinates
    len = sqrt((x(1)-x(2))^2+(y(1)-y(2))^2); % edge length
    xc = mean(x); yc = mean(y); % edge mid-point
    k = kappa(xc,yc); % value of kappa at mid-point
    RE = k/6*[2 1; 1 2]*len; % edge boundary matrix
    R(loc2glb,loc2glb) = R(loc2glb,loc2glb) + RE;
end
end

definition of Robin Load Vector

function r = RobinLoadVector2D(p,e,kappa,gD,gN)
np = size(p,2);
ne = size(e,2);
r = zeros(np,1);
for E = 1:ne
    loc2glb = e(1:2,E);
    x = p(1,loc2glb);
    y = p(2,loc2glb);
    len = sqrt((x(1)-x(2))^2+(y(1)-y(2))^2);
    xc = mean(x); yc = mean(y);
    tmp = kappa(xc,yc)*gD(xc,yc)+gN(xc,yc);
    rE = tmp*[1; 1]*len/2; r(loc2glb) = r(loc2glb) + rE;
end
end

END