Norwegian University of Science and Technology
Institute of mathematics

TMA4220 Finite Element Method Fall 2016

Exercise set 2

$\boxed{1}$ Consider the equation

$$-u_{xx} = f(x), \quad 0 < x < 1, \quad u(0) = 0, \quad u(1) = 0,$$

which is solved with the finite element method with equidistant grid ($x_i = ih$, $h = 1/N$), and the basis functions

$$\varphi_i(x) = \begin{cases} \frac{x - x_{i-1}}{h}, & \text{for } x_{i-1} \le x \le x_i, \\ \frac{x_{i+1} - x}{h} & \text{for } x_i \le x \le x_{i+1}, \\ 0 & \text{otherwise.} \end{cases}$$

for $i = 1, 2, \dots, N-1$.

Denote the numerical solution u_h.

a) Solve this equation with $f(x) = x^4$.

Solution:

$$u(x) = \int\int x^4 \, dx \, dx = \frac{1}{30}x^6 + C_1 x + C_2 \tag{1}$$

Letting $u(0) = u(1) = 0$, *we get* $C_1 = -\frac{1}{30}$ *and* $C_2 = 0$

b) Plot the exact solution and the computed solution. How do these compare on the gridpoints. How do they compare between gridpoints?

Solution: This reveals that the exact solution $u(x)$ *coincides with the computed solution* $u_h(x)$ *at the gridpoints. There is a discrepancy between the two in the domain between gridpoints.*

You may have noticed that the numerical and exact solution coincide on the gridpoints, that is

$$u(x_i) = u_h(x_i), \quad \text{for } i = 0, 1, 2, \dots, N.$$

We will now show that this is true for all $f(x)$. For $i = 0$ and $i = N$, there is nothing to prove (Why?), what remains is $i = 1, 2, \dots, N-1$.

c) Prove that $a(u - u_h, \varphi_i) = 0$ for $i = 1, 2, \dots, N-1$.

Solution: Let $V_h = \operatorname{span}\{\varphi_1, \varphi_2, \dots, \varphi_{N-1}\} \subset V = H_0^1([0,1])$. *Then the numerical solution satisfies*

$$a(u_h, v_h) = f(v_h) \quad \forall v \in V_h,$$

while the exact solution satisfies

$$a(u, v) = f(v) \quad \forall v \in V.$$

$\varphi_i \in V_h \subset V$, so

$$\begin{aligned} a(u-u_h,\varphi_i) &= a(u,\varphi_i) - a(u_h,\varphi_i) \quad \textit{a bilinear.} \\ &= f(\varphi_i) - f(\varphi_i) \quad \textit{u and } u_h \textit{ solves the weak problems.} \\ &= 0. \end{aligned}$$

d) Prove that for all $v \in H^1([0,1]) \cap C^0([0,1])$ and $i = 1, 2 \ldots, N-1$,

$$a(v,\varphi_i) = \frac{1}{h}(2v(x_i) - v(x_{i-1}) - v(x_{i+1})).$$

Solution:

$$a(v,\varphi_i) = \int_0^1 v_x \varphi_i' \,\mathrm{dx}.$$

$$\varphi_i' = \begin{cases} \frac{1}{h}, & \textit{for } x_{i-1} < x < x_i, \\ -\frac{1}{h}, & \textit{for } x_i < x < x_{i+1}, \\ 0, & \textit{otherwise.} \end{cases}$$

$$\begin{aligned} a(v,\varphi_i) &= \int_{x_{i-1}}^{x_i} \frac{v_x}{h}\,\mathrm{dx} - \int_{\mathrm{x_i}}^{\mathrm{x_{i+1}}} \frac{\mathrm{v_x}}{\mathrm{h}}\,\mathrm{dx} \\ &= \frac{1}{h}(v(x_i) - v(x_{i-1}) - (v(x_{i+1}) - v(x_i))) \\ &= \frac{1}{h}(2v(x_i) - v(x_{i-1}) - v(x_{i+1})). \end{aligned}$$

e) The $M \times M$ matrix

$$A = \begin{bmatrix} 2 & -1 & & \\ -1 & 2 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix}$$

is invertible for all M. Use this fact and the results of **c)** and **d)** to prove that $u(x_i) - u_h(x_i) = 0$ for $i = 1, 2 \ldots, N-1$.

*Solution: Combining the results of **c)** and **d)**, we see that $v_i = u(x_i) - u_h(x_i)$ satisfies*

$$-v_{i-1} + 2v_i - v_{i+1} = 0$$

for $i = 1, 2, \ldots, N-1$. Removing v_0 and v_h, (which are equal to zero) this is equivalent to

$$A\mathbf{v} = \mathbf{0}$$

where A is as above and $\mathbf{v} = [v_1 v_2 \ldots v_{N-1}]^\top$. Since A is invertible, $A\mathbf{v} = \mathbf{0} \Rightarrow \mathbf{v} = 0$.

$\boxed{2}$ Given the Helmholtz problem

$$\begin{aligned} -u_{xx} + \sigma u &= f \text{ on } (0,1), \\ u(0) &= u(1) = 0. \end{aligned}$$

where $\sigma > 0$ is a constant. Set up the weak form for this problem. Show that, when this problem is solved by a Galerkin method, using $V_h = \text{span}\ \{\varphi_i\}\ _{i=1}^{N}$, the discrete problem can be written as

$$(A + \sigma M)\mathbf{u} = \mathbf{f}.$$

Set up the matrix M for $V_h = X_h^1$ on a uniform grid.

Solution: Multiply by the equation by a test function v, integrate over the domain $(0,1)$, use partial integration and get rid of the boundary terms by require $v(0) = v(1) = 0$. The the weak formualtion becomes:

$$\textit{Find } u \in H_0^1(\Omega) \textit{ such that } \int_0^1 v_x u_x dx + \sigma \int_0^1 uv dx = \int_0^1 fv dx, \quad \forall v \in H_0^1(0,1).$$

Choose $V_h = span\{\varphi_1, \varphi_2, \dots, \varphi_h\}$, let our unknown approximation be written as

$$u_h(x) = \sum_{j=1}^{N} u_j\, \varphi_j(x),$$

where the coefficients u_j is found from

$$\sum_{j=1}^{N} \int_0^1 \left(\frac{d\varphi_j}{dx} \frac{d\varphi_i}{dx} \right) dx + \sigma \sum_{j=1}^{N} \int_0^1 (\varphi_j\, \varphi_i) dx = \int_0^1 (f\, \varphi_i) dx, \qquad i = 1, 2, \cdots, N.$$

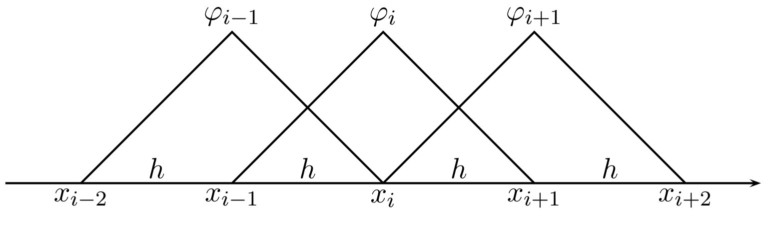
This is a linear system of equations

$$A\mathbf{u} + \sigma M \mathbf{u} = \mathbf{f}$$

where

$$(A_{i,j} = \int_0^1 \left(\frac{d\varphi_j}{dx} \frac{d\varphi_i}{dx} \right) dx, \qquad M_{i,j} = \int_0^1 (\varphi_j\, \varphi_i) dx \quad \textit{and} \quad f_i = \int_0^1 (f \varphi_i) dx.$$

Let $V_h = X_h^1$ on a uniform grid, so that $h = 1/N$ and $x_i = ih$, $i = 0, 1, \dots, N$.



The non-zeros elements of the stiffness matrix A and the mass matrix M is

$$A_{i,i} = \int_{x_{i-1}}^{x_{i+1}} \left(\frac{d\varphi_i}{dx} \right) dx = \frac{2}{h}, \qquad A_{i,i+1} = \int_{x_i}^{x_{i+1}} \frac{d\varphi_{i+1}}{dx} \frac{d\varphi_i}{dx} dx = -\frac{1}{h} = A_{i+1,i}$$

$$M_{i,i} = \int_{x_{i-1}}^{x_{i+1}} \varphi_i^2 dx = \frac{2}{3} h, \qquad M_{i,i+1} = \int_{x_i}^{x_{i+1}} \varphi_{i+1} \varphi_i dx = \frac{1}{6} h = M_{i+1,i}.$$

All the other elements are 0. So

$$M = \frac{h}{6}\begin{pmatrix} 4 & 1 & & & 0 \\ 1 & 4 & 1 & & \\ & 1 & 4 & \ddots & \\ & & \ddots & \ddots & 1 \\ 0 & & & 1 & 4 \end{pmatrix} \in \mathbb{R}^{(N-1)\times(N-1)}.$$

$\boxed{3}$ Write a MATLAB program for solving the Helmholtz problem

$$-u_{xx} + \sigma u = f(x), \qquad 0 < x < 1, \qquad u(0) = u(1) = 0.$$

or, using the weak formulation

$$\text{find } u \in H_0^1(0,1) \text{ s.t. } \int_0^1 u_x v_x dx + \sigma \int_0^1 uv dx = \int_0^1 fv dx, \text{ for all } v \in H_0^1(0,1) \quad (2)$$

by the finite element method on X_h^2, using the algorithm outlined in the supplementary note.

To test your code, let $\sigma = 1$, $f = \sin(\pi x)$ in which case $u(x) = \sin(\pi x)/(1+\pi^2)$.

Use for example $[0, 0.1, 0.25, 0.3, 0.4, 0.45, 0.5, 0.55, 0.6, 0.7, 0.8, 0.9, 1]$ for the partition of the elements.

As already pointed out in exercise 2, the discrete problem can be written as

$$(A + \sigma M)\mathbf{u} = \mathbf{b}. \quad (3)$$

So the task is to set up the matrices A and M and the load vector $\mathbf{b}$, and solve the system. What you have to do is described in the following:

a) Preliminaries:
Set up the element matrices A_h^K and M_h^K, corresponding to contribution from element K to the first and second integrals of (2) resp.
Solution: Starting from the very beginning: The quadratic shape functions defined on the reference element $\hat{K} = (0,1)$, corresponding to the nodes $x_1 = 0$, $x_2 = 1/2$ and $x_3 = 1$ is given by

$$\psi_1(\xi) = 2(\xi - \frac{1}{2})(\xi - 1), \quad \psi_2(x) = -4\xi(\xi - 1), \quad \psi_3(x) = 2\xi(\xi - \frac{1}{2}).$$

The mapping from the $\hat{K}$ to $K = (x_k, x_{k+1})$ is given by $x(\xi) = x_k + h_k\xi$ and the inverse mapping is $\xi(x) = (x - x_k)/h_k$, where $h_k = x_{k+1} - x_k$.

$$A_h^K = \frac{1}{3h_k}\begin{pmatrix} 7 & -8 & 1 \\ -8 & 16 & -8 \\ 1 & -8 & 7 \end{pmatrix}, \qquad M_h^K = \frac{h_k}{30}\begin{pmatrix} 4 & 2 & -1 \\ 2 & 16 & 2 \\ -1 & 2 & 4 \end{pmatrix}$$

So the contribution from element K to the element matrices M_h^K and A_h^K becomes:

$$(M_h^K)_{\alpha,\beta} = \int_{x_k}^{x_{k+1}} \varphi_\alpha^K(x)\varphi_\beta^K(x)dx = h_k \int_0^1 \psi_\alpha(\xi)\psi_\beta(\xi)d\xi$$

$$(A_h^K)_{\alpha,\beta} = \int_{x_k}^{x_{k+1}} \frac{d\varphi_\alpha^K}{dx}\frac{d\varphi_\beta^K}{dx}dx = \frac{1}{h_k}\int_0^1 \frac{\psi_\alpha}{d\xi}\frac{d\psi_\beta}{d\xi}d\xi$$

se the slides on the webpage for details. Altogether, we ends up with:

$$A_h^K = \frac{1}{3h_k}\begin{pmatrix} 7 & -8 & 1 \\ -8 & 16 & -8 \\ 1 & -8 & 7 \end{pmatrix}, \qquad M_h^K = \frac{h_k}{30}\begin{pmatrix} 4 & 2 & -1 \\ 2 & 16 & 2 \\ -1 & 2 & 4 \end{pmatrix}$$

b) Write a function computing integrals by the following quadrature formula:

$$\int_0^1 g(x)dx \approx \frac{1}{2}(g(c_1) + g(c_2)), \qquad c_{1,2} = \frac{1}{2} \pm \frac{\sqrt{3}}{6}.$$

This will be used for to approximate the contribution from an element to the load vector.
Solution: On an element, the approximation becomes

$$\int_{x_k}^{x_{k+1}} g(x)dx \approx \frac{g(x_k + c_1 h_k) + g(x_k + c_2 h_k)}{2}.$$

But the contributions to the load vector is $\int_{x_k}^{x_{k+1}} \varphi_i(x)f(x)dx$*, so that*

$$b_i^K = \int_{x_k}^{x_{k+1}} \varphi_i(x)f(x)dx \approx \sum_{j=1}^{2} \psi_i(c_j)f(x_k + c_j).$$

The corresponding function can be

```
function b = bk(f,x)
% Calculate a numerical approximation to the elemental load vector
% of the function f on an element [x(1),x(2)].
% using quadratic elements.

h = x(2)-x(1);
c1 = 1/2-sqrt(3)/6;
c2 = 1/2+sqrt(3)/6;
b = h*[(c1-0.5)*(c1-1)*f(x(1)+h*c1) + (c2-0.5)*(c2-1)*f(x(1)+h*c2); ...
     2*c1*(1-c1)*f(x(1)+h*c1) + 2*c2*(1-c2)*f(x(1)+h*c2);
     c1*(c1-0.5)*f(x(1)+h*c1)+c2*(c2-0.5)*f(x(1)+h*c2)];
```

c) Assemble the prototype matrices $\tilde{A}_h$ and $\tilde{M}_h$ as well as the load vector $\tilde{\mathbf{b}}$.

Solution: For points ***b)-e)****, see Figure 1.*

d) Remove the rows and columns corresponding to the boundary conditions.

e) Solve (3), and plot the solution.

```
f = @(x) sin(pi*x);
sigma = 1;

% The partition of [0,1].
x = [0,0.1,0.25,0.3,0.4,0.45,0.5,0.55,0.6,0.7,0.8,0.9,1];

Ak = [7/3, -8/3, 1/3;     % Element stiffness matrix
      -8/3, 16/3, -8/3;
      1/3, -8/3, 7/3];
Mk = [2/15, 1/15, -1/30; % Element mass matrix
      1/15, 8/15, 1/15;
      -1/30, 1/15, 2/15];

Nk = length(x)-1;         % Number of elements.
N = 2*Nk+1;               % Number of nodes (including the boundaries)
theta = @(k,alpha) 2*(k-1)+alpha;  % local-to-global mapping.

Ah = sparse(N,N);             % Stiffness matrix
Mh = sparse(N,N);             % Mass matrix resp.
bh = zeros(N,1);              % Load vector

% Assemble process:
for k = 1:Nk
    h = x(k+1)-x(k);      % Size of the element
    gi = theta(k,1):theta(k,1)+2;
    Ah(gi,gi) = Ah(gi,gi) + Ak/h;
    Mh(gi,gi) = Mh(gi,gi) + Mk*h;
    bh(gi) = bh(gi) + bk(f,[x(k),x(k+1)]);
end

% For the case u(0)=u(1)=0
% Remove contributions concerning the Dirichlet boundaries.
A = Ah(2:end-1,2:end-1);
M = Mh(2:end-1,2:end-1);
b = bh(2:end-1);

% Solve the system
u = (A+sigma*M)\b;

% Include the boundaries.
u = [0;u;0];

% For the plot: Create a x-vector with all the nodes
% including the midpoints of each element.
xi = x(1);
for k=1:Nk
    xi = [xi;x(k)+(x(k+1)-x(k))/2; x(k+1)];
end

% Plot the numerical and the exact solution in the nodes.
u_exact = sin(pi*xi)/(1+pi^2);
plot(xi,u_exact,'r',xi,u,'b')

legend('u_exact','u_h');
```

Figure 1: Code for solving the Helmholtz equation in 1D by a quadratic FEM