



Norwegian University of Science  
and Technology  
Department of Mathematical  
Sciences

TMA4220 Numerical  
Solution of Partial  
Differential Equations  
Using Element Methods  
Høst 2012

**Exercise set 3**

1 Quarteroni: Section 3.7: Exercise 4 and 5.

2 Write a MATLAB program for solving the Helmholtz problem

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

or, using the weak formulation

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

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

To test you 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. (That is, the first element is  $[0, 0.1]$ , and there is an extra node in the middle).

As already pointed out in Exercise 2.4, the discrete problem can be written as

$$(A + \sigma M)\mathbf{u} = \mathbf{b}. \quad (2)$$

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 (1) 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*

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

*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  $h_k = x_{k+1} - x_k$ .*

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

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

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

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

see the slides on the webpage for details. Altogether, we end up with:

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

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

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

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  $\int_{x_k}^{x_{k+1}} \phi_i(x) f(x) dx$ , 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

```

1 function b = bk(f,x)
2 % Calculate a numerical approximation to the elemental load vector
3 % of the function f on an element [x(1),x(2)].
4 % using quadratic elements.
5
6 h = x(2)-x(1);
7 c1 = 1/2-sqrt(3)/6;
8 c2 = 1/2+sqrt(3)/6;
9 b = h*[(c1-0.5)*(c1-1)*f(x(1)+h*c1) + (c2-0.5)*(c2-1)*f(x(1)+h*c2); ...
10        2*c1*(1-c1)*f(x(1)+h*c1) + 2*c2*(1-c2)*f(x(1)+h*c2);
11        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 (2), and plot the solution

```

1 f = @(x) sin(pi*x);
  sigma = 1;
3
  % The partition of [0,1].
5 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];

7 Ak = [7/3, -8/3, 1/3;      % Element stiffness matrix
        -8/3, 16/3, -8/3;
9       1/3, -8/3, 7/3];
  Mk = [2/15, 1/15, -1/30; % Element mass matrix
11      1/15, 8/15, 1/15;
        -1/30, 1/15, 2/15];

13
  Nk = length(x)-1;          % Number of elements.
15 N = 2*Nk+1;              % Number of nodes (including the boundaries)
  theta = @(k,alpha) 2*(k-1)+alpha; % local-to-global mapping.

17
  Ah = sparse(N,N);         % Stiffness matrix
19 Mh = sparse(N,N);         % Mass matrix resp.
  bh = zeros(N,1);          % Load vector

21
  % Assemble process:
23 for k = 1:Nk
      h = x(k+1)-x(k);      % Size of the element
25     gi = theta(k,1):theta(k,1)+2;
      Ah(gi,gi) = Ah(gi,gi) + Ak/h;
27     Mh(gi,gi) = Mh(gi,gi) + Mk*h;
      bh(gi) = bh(gi) + bk(f,[x(k),x(k+1)]);
29 end

31 % For the case u(0)=u(1)=0
  % Remove contributions concerning the Dirichlet boundaries.
33 A = Ah(2:end-1,2:end-1);
  M = Mh(2:end-1,2:end-1);
35 b = bh(2:end-1);

37 % Solve the system
  u = (A+sigma*M)\b;
39
  % Include the boundaries.
41 u = [0;u;0];

43 % For the plot: Create a x-vector with all the nodes
  % including the midpoints of each element.
45 xi = x(1);
  for k=1:Nk
47     xi = [xi;x(k)+(x(k+1)-x(k))/2; x(k+1)];
  end
49
  % Plot the numerical and the exact solution in the nodes.
51 u_exact = sin(pi*xi)/(1+pi^2);
  plot(xi,u_exact,'r',xi,u,'b')
53
  legend('u_exact','u_h');

```

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

- f) Change the boundary conditions to  $u(0) = 1$  and  $u_x(1) = 2$ . Which changes has to be done in the code?

*Solution:* The weak formulation of the Helmholtz problem is now

$$\text{find } u \in V \text{ s.t. } \int_0^1 u_x v_x dx + \sigma \int_0^1 u v dx = \int_0^1 f v dx - \int_0^1 R_{g,x} v_x dx + 2v(1), \text{ for all } v \in V$$

where  $v = \{v \in H^1(0,1) : v(0) = 0\}$  and  $R_g \in H^1(0,1)$  is some function satisfying  $R_g(0) = 1$ . For the numerical solution, we choose  $R_g = \varphi_0(x)$ . The corresponding discrete formulation is

$$\sum_{j=0}^N u_j \int_0^1 \varphi_{j,x} \varphi_{i,x} dx + \sigma \sum_{j=1}^N u_j \int_0^1 \varphi_j \varphi_i dx = \int_0^1 f \varphi_i dx - \int_0^1 \varphi_{i,x} \varphi_{0,x} dx + 2\phi_i(1)$$

for  $i = 1, 2, \dots, N$ . In practice, this means that the column corresponding to the left hand side Dirichlet condition is moved to the right hand side. In addition, we add one term in the last equation, to include the contribution from the Neumann boundary conditions. Line 31-41 in Figure 1 has to be changed to

```

32 % For the case u(0)=1, u_x(1)=2.
32 bh = bh - Ah(:,1)*1;
32 bh(end) = bh(end)+2;
34
34 A = Ah(2:end,2:end);
36 M = Mh(2:end,2:end);
36 b = bh(2:end);
38
38 % Solve the system
40 u = (A+sigma*M)\b;
42
42 % Include the Dirichlet boundary.
42 u = [1;u];

```