TMA4220: Supplementary note 1
Implemention of FEM in 1D.

Anne Kværnø

September 21, 2012

This is a supplement to Q: section 4.3. I have also included some slides with an example at the course website. It might be helpful to use these when you read this note.

Given the prototype problem (no boundary conditions)

$$-u_{xx} = f(x), \qquad a \le x \le b. \tag{1}$$

The weak formulation of this problem is

$$\text{Find } u \in H^1(a,b) \text{ such that } a(u,v) = F(v), \quad \forall v \in H^1(a,b) \tag{2}$$

where

$$a(u,v) = \int_a^b u_x v_x dx, \qquad F(x) = \int fv dx.$$

Next, choose a subspace $V_h = \text{span}\{\varphi_1, \varphi_2, \ldots, \varphi_{N_h}\} \subset H^0(a,b)$. The Galerkin method then becomes

$$\text{Find } u_h \in V_h \text{ such that } a(u_h, v_h) = F(v_h), \quad \forall v \in V_h. \tag{3}$$

or simply

$$\tilde{A_h}\tilde{u} = \tilde{\mathbf{b}_h}, \quad \text{where} \quad A_{h,i,j} = a(\varphi_j, \varphi_i) = \int_a^b \frac{d\varphi_j}{dx}\frac{d\varphi_i}{dx}dx, \qquad b_{h,i} = F(\varphi_i) = \int_a^b f\phi_i dx. \tag{4}$$

NB! We do not expect this system to have a unique solution (why not?)

We will now use $V_h = X_h^r$, which is defined as follows: Choose a partition $\mathcal{T}_h$ of the interval (a,b), that is, choose $a = x_0 < x_1 < \cdots < x_N < x_N < b$. Let $K_k = [x_{k-1}, x_k]$, $h_k = x_k - x_{k-1}$, and

$$X_h^r = \{v_h \in C^0[a,b] \ : \ v_h|_{K_k} \in \mathbb{P}_r, \ \forall K_k \in \mathcal{T}_h\}$$

where $v_h|_K$ is the restriction of v_h to the element K. The idea is:

$$a(\varphi_i, \varphi_j) = \int_a^b \frac{d\varphi_j}{dx}\frac{d\varphi_i}{dx}dx = \sum_{k=1}^N \int_{K_k} \frac{d\varphi_j}{dx}\frac{d\varphi_i}{dx}dx = \sum_{k=1}^N a^K(\varphi_i, \varphi_j) \tag{5}$$

so if we can find a reasonable representation of the basis functions on each element, we can calculate the contribution to a_{ij} from each element, and then sum over all the elements. Similar can of course be done for the load vector b_i.

In this note we cover:

(i) How to find a representation of the basis functions restricted to elements.
(ii) How to compute the contributions $A^K_{h,ij}$ and $b^K_{h,i}$ from a given element K.
(iii) How to set up the matrix $\tilde{A}_h$ and the vector $\tilde{\mathbf{b}}_h$.
(iv) How to impose boundary conditions.

(i) The basis function restricted to elements (shape functions).

On each element K we need a basis for $\mathbb{P}_r$. There are several options for this of course, but a quite convenient one is to choose a lagrangian basis, that is, choose $r+1$ nodes, and use the cardinal functions as basis functions. If the boundary points of the elements are included in the set of nodes, then you get continuity of the basis functions over the elements for free.

In practice, the polynomials are defined on a reference element: $\hat{K} = [0,1]$. On this element, choose $r+1$ distinct nodes ξ_i, $i = 0, 1, \cdots, r$, where $\xi_0 = 0$ and $\xi_r = 1$. The shape functions on the reference element are given by

$$\Psi_\alpha(\xi) = \prod_{\beta=1,\beta\neq\alpha}^{r} \frac{\xi - \xi_\beta}{\xi_\alpha - \xi_\beta}, \qquad \alpha = 0, 1, \cdots, r, \qquad \xi \in [0,1].$$

We next have to map these shape functions from the reference element $\hat{K}$ to the physical element K_k. And now we run into the real nightmare of indices. Let us define a *local to global mapping* $i = \theta(k, \alpha)$. So α refer to a node ξ_α on the reference element, k refer to element K_k on the interval $[a, b]$ and i refer to node x_i on $[a, b]$ (puh!). In the following I am going to use $x^k_\alpha = x_i$. Notice that $x^k_r = x^{k+1}_0$, the last node on element K_k is the first one on element K_{k+1}. The mapping between $\hat{K}$ and K_k is

$$x(\xi) = \Phi_k(\xi) = x^k_0 + h_k\xi, \qquad \xi(x) = \Phi^{-1}_k(x) = \frac{x - x^k_0}{h_k} \tag{6}$$

where $h_k = x^k_r - x^k_0$ is the size of the element. So finally

$$\phi_i|_{K_k}(x) = \Psi_\alpha(\Phi^{-1}_k(x)), \quad i = \theta(k,\alpha), \quad \alpha = 0, 1, \ldots, r$$

defines all the (nonzero) basis functions on the element K_k, and thus the whole space X^r_h.

Notice that this is a *nodal* or lagrangian basis for X^r_h: To each node x_i on $[a, b]$ there is a corresponding basis function $\varphi_i(x)$ s.t. $\varphi_i(x_j) = \delta_{ij}$. Moreover, the support of φ_i (where $\varphi_i(x)$ is nonzero) is the elements in which x_i is a node.

(ii) Contributions from each element K_k.

The contribution to element $\tilde{A}_{h,ij}$ from element K_k is given by

$$\tilde{A}^k_{h,\alpha,\beta} = \int_{K_k} \frac{d\phi_j}{dx}\frac{d\phi_i}{dx}dx = \frac{1}{h_k}\int_0^1 \frac{d\Psi_\beta}{d\xi}\frac{d\Psi_\alpha}{d\xi}d\xi, \qquad i = \theta(k,\alpha),\ j = \theta(k,\beta).$$

since

$$\frac{d\phi_i}{dx} = \frac{d\Psi_\alpha(\Phi^{-1}_k(x))}{dx} = \frac{\partial\Psi_\alpha}{\partial\xi}\cdot\frac{d\Phi^{-1}_k(x)}{dx} = \frac{1}{h_k}\frac{d\Psi_\alpha}{d\xi}, \qquad dx = h_k\, d\xi.$$

Similar

$$\tilde{b}^k_{h,\alpha} = \int_{K_k} f(x)\phi_i(x)dx = h_k \int_0^1 f(\Phi_k^{-1}(\xi))\, \Psi_\alpha(\xi)d\xi.$$

In practice, the last integrals have to be approximated by some numerical quadrature, e.g.

$$\int_0^1 g(x)dx \approx \sum_{j=1}^{s} w_j f(c_j)$$

Using Gauss quadrature as examples, we have

$$\begin{aligned} s = 1: \quad & c_1 = 1/2, && w_1 = 1 \\ s = 2: \quad & c_1 = (3-\sqrt{3})/6, \quad c_2 = (3+\sqrt{3})/6, && w_1 = w_2 = 1/2 \end{aligned}$$

Using this, we get

$$\tilde{b}^k_{h,\alpha} \approx h_k \sum_{j=1}^{s} w_j f(x_0^k + c_j h_k)\Psi_\alpha(c_j).$$

Strictly speaking, this is a *variational crime*, a deviation from the variational framework.

Notice that, by using this approach, all the integrals can be done directly on the reference element, and the result is then mapped to the real element.

(iii) The assembly process

In this case, we collect contributions to the stiffness matrix and the load vector, from each element k_k, see (5). This can most easily be described by the algorithm:

$\tilde{A}_h = 0$, $\tilde{\mathbf{b}}_h = 0$.
for $K_k \in \mathcal{T}_h$
    for $\alpha = 0, \ldots, r$
        $i = \theta(k, \alpha)$
        for $\beta = 0, \ldots, r$
            $j = \theta(k, \beta)$
            $\tilde{A}_{h,ij} = \tilde{A}_{h,ij} + \tilde{A}^k_{\alpha,\beta}$
        $\tilde{b}_{h,i} = \tilde{b}_{h,i} + \tilde{b}^k_\alpha$
    end
end

We now have the system of linear equations

$$\tilde{A}_h \tilde{u}_h = \tilde{\mathbf{b}}_h \tag{7}$$

where $\tilde{A}_h$ is singular, so we can expect no unique solution.

Boundary conditions

By the construction of the basis of X_h^r, our (so far unknown) numerical solution $u_h \in X_h^r$ can be written as

$$u_h(x) = \sum_{i=0}^{N_h+1} u_i \varphi_i(x), \qquad u(x_i) = u_i, i = 0, 1, 2, \cdots, N_h + 1,$$

where $N_h + 2$ is the total number of nodes. Let us now include boundary conditions on (1).

a) *Dirichlet (essential) boundary conditions*: Let $u(a) = g_0$ and $u(b) = g_1$, where g_0, g_1 are given. Then our numerical solution will be of the form:

$$u_h(x) = g_0\varphi_0(x) + \sum_{i=1}^{N_h} u_i \varphi_i(x) + g_1\varphi_{N_h+1}(x)$$

leaving us with N_h equations to be solved. First, we move what is known, that is the first column of $\tilde{A}_h$ times g_0 and the last column times g_1 to the right hand side. Second, we remove the corresponding equations, that is the first and the last rows of (7). We are now left with a solvable system

$$A_h \mathbf{u}_h = \mathbf{b_h} - A_{h,:,0}\, g_0 - g_1 A_{h,:,N_h+1}$$

where $A_{h,:,j}$ refer to column j of the matrix $\tilde{A}_h$ with the first and last row removed.

b) *Neumann (natural) boundary conditions*; Let $u(a) = 0$ and $u_x(b) = r$. This is left as an exercise!