Chapter 8

Issues of the FE Method in one space dimension

Boundary Conditions, Implementation & the Lax-Milgram Lemma

8.1 Boundary conditions

For a second order two-point boundary value problem, typical boundary conditions (BC) include one of the following at each end, say at $x = x_l$,

1. a Dirichlet condition, *e.g.*, $u(x_l) = u_l$ is given;
2. a Neumann condition, *e.g.*, $u'(x_l)$ is given; or
3. a Robin (mixed) condition, *e.g.*, $\alpha u(x_l) + \beta u'(x_l) = \gamma$ is given, where α, β, and γ are known but $u(x_l)$ and $u'(x_l)$ are both unknown.

Boundary conditions affect the bilinear and linear form, and the solution space.

Example 8.1.

For example,
$$\begin{aligned} -u'' &= f, \qquad 0 < x < 1, \\ u(0) &= 0, \qquad u'(1) = 0, \end{aligned}$$
involves a Dirichlet BC at $x = 0$ and a Neumann BC at $x = 1$. To derive the weak form, we again follow the familiar procedure:
$$\begin{aligned} \int_0^1 -u'' v dx &= \int_0^1 f v \, dx \,, \\ -u' \, v|_0^1 + \int_0^1 u' v' dx &= \int_0^1 f v \, dx \,, \\ -u'(1)v(1) + u'(0)v(0) + \int_0^1 u' v' dx &= \int_0^1 f v \, dx \,. \end{aligned}$$
For a conforming finite element method, the solution function of u and the test functions v should be in the same space. So it is natural to require that the test functions v satisfy the same homogenous Dirichlet BC, *i.e.*, we require $v(0) = 0$; and the Dirichlet condition is therefore called an *essential* boundary condition. On the other hand, since $u'(1) = 0$, the first term in the final expression is zero, so it does not matter what $v(1)$ is, *i.e.*, there

is no constraint on $v(1)$; so the Neumann BC is called a *natural* boundary condition. It is noted that $u(1)$ is unknown. The weak form of this example is the same as before for homogeneous Dirichlet BC at both ends; but now the solution space is different:

$$(u', v') = (f, v)\ ,\ \forall v \in H^1_E(0,1)\,,$$
$$\text{where} \qquad H^1_E(0,1) = \left\{ v(x)\ ,\ v(0) = 0\ ,\ v \in H^1(0,1) \right\},$$

where we use the subscript E in $H^1_E(0,1)$ to indicated an essential boundary condition.

8.1.1 Mixed boundary conditions

Consider a Sturm-Liouville problem

$$-(pu')' + qu = f\,, \quad x_l < x < x_r\,, \quad p(x) \geq p_{min} > 0\,, \quad q(x) \geq 0\,, \tag{8.1}$$
$$u(x_l) = 0\,, \quad \alpha u(x_r) + \beta u'(x_r) = \gamma\,, \quad \beta \neq 0\,, \quad \frac{\alpha}{\beta} \geq 0\,, \tag{8.2}$$

where α, β, and γ are known constants but $u(x_r)$ and $u'(x_r)$ are unknown. Integration by parts again gives

$$-p(x_r)u'(x_r)v(x_r) + p(x_l)u'(x_l)v(x_l) + \int_{x_l}^{x_r} \left(pu'v' + quv \right) dx = \int_{x_l}^{x_r} fv\, dx\,. \tag{8.3}$$

As explained earlier, we set $v(x_l) = 0$ (an essential BC). Now we re-express the mixed BC as

$$u'(x_r) = \frac{\gamma - \alpha u(x_r)}{\beta}\,, \tag{8.4}$$

and substitute this into (8.3) to obtain

$$-p(x_r)v(x_r)\frac{\gamma - \alpha u(x_r)}{\beta} + \int_{x_l}^{x_r} \left(pu'v' + quv \right) dx = \int_{x_l}^{x_r} fv\, dx$$

or equivalently

$$\int_{x_l}^{x_r} \left(pu'v' + quv \right) dx + \frac{\alpha}{\beta} p(x_r)u(x_r)v(x_r) = \int_{x_l}^{x_r} fv\, dx + \frac{\gamma}{\beta} p(x_r)v(x_r)\,, \tag{8.5}$$

which is the corresponding weak (variational) form of the Sturm-Liouville problem. We define

$$a(u,v) = \int_{x_l}^{x_r} \left(pu'v' + quv \right) dx + \frac{\alpha}{\beta} p(x_r)u(x_r)v(x_r)\,, \text{ the bilinear form}\,, \tag{8.6}$$
$$L(v) = (f,v) + \frac{\gamma}{\beta} p(x_r)v(x_r)\,, \text{ the linear form.} \tag{8.7}$$

We can prove that:

1. $a(u,v) = a(v,u)$, *i.e.*, $a(u,v)$ is symmetric;

2. $a(u,v)$ is a bilinear form, *i.e.*,

$$a(ru+sw,v) = ra(u,v) + sa(w,v)\,,$$
$$a(u,rv+sw) = ra(u,v) + sa(u,w)\,,$$

for any real numbers r and s; and

3. $a(u,v)$ is an inner product, and the corresponding energy norm is

$$\|u\|_a = \sqrt{a(u,u)} = \left\{ \int_{x_l}^{x_r} \left(pu'^2 + qu^2 \right) dx + \frac{\alpha}{\beta} p(x_r) u(x_r)^2 \right\}^{\frac{1}{2}}.$$

It is now evident why we require $\beta \neq 0$, and $\alpha/\beta \geq 0$. Using the inner product, the solution of the weak form $u(x)$ satisfies

$$a(u,v) = L(v)\,, \quad \forall v \in H^1_E(x_l,x_r)\,, \tag{8.8}$$
$$H^1_E(x_l,x_r) = \left\{ v(x)\,, \quad v(x_l) = 0\,, \;\; v \in H^1(x_l,x_r) \right\}, \tag{8.9}$$

and we recall that there is no restriction on $v(x_r)$. The boundary condition is essential at $x = x_l$, but natural at $x = x_r$. The solution u is also the minimizer of the functional

$$F(v) = \frac{1}{2} a(v,v) - L(v)$$

in the $H^1_E(x_l,x_r)$ space.

8.1.2 Non-homogeneous Dirichlet boundary conditions

Suppose now that $u(x_l) = u_l \neq 0$ in (8.2). In this case, the solution can be decomposed as the sum of the particular solution

$$\begin{aligned} -(pu_1')' + qu_1 &= 0\,, \quad x_l < x < x_r\,, \\ u_1(x_l) = u_l\,, &\qquad \alpha u_1(x_r) + \beta u_1'(x_r) = 0\,, \quad \beta \neq 0\,, \quad \frac{\alpha}{\beta} \geq 0, \end{aligned} \tag{8.10}$$

and

$$\begin{aligned} -(pu_2')' + qu_2 &= f\,, \quad x_l < x < x_r\,, \\ u_2(x_l) = 0\,, &\qquad \alpha u_2(x_r) + \beta u_2'(x_r) = \gamma, \quad \beta \neq 0\,, \quad \frac{\alpha}{\beta} \geq 0\,. \end{aligned} \tag{8.11}$$

We can use the weak form to find the solution $u_2(x)$ corresponding to the homogeneous Dirichlet BC at $x = x_l$. If we can find a particular solution $u_1(x)$ of (8.10), then the solution to the original problem is $u(x) = u_1(x) + u_2(x)$.

Another simple way is to choose a function $u_0(x)$, $u_0(x) \in H^1(x_l,x_r)$ that satisfies

$$u_0(x_l) = u_l, \quad \alpha u_0(x_r) + \beta u_0'(x_r) = 0.$$

For example, the function $u_0(x) = u_l \phi_0(x)$ would be such a function, where $\phi_0(x)$ is the hat function centered at x_l if a mesh $\{x_i\}$ is given. Then $\hat{u}(x) = u(x) - u_0(x)$ would satisfy a homogeneous Dirichlet BC at x_l and the following S-L problem:

$$-(p\hat{u}')' + q\hat{u} = f(x) + (pu_0')' - qu_0, \quad x_l < x < x_r,$$
$$\hat{u}(x_l) = 0, \quad \alpha\hat{u}'(x_r) + \beta\hat{u}(x_r) = \gamma.$$

We can apply the finite element method for $\hat{u}(x)$ previously discussed with the modified source term $f(x)$, where the weak form $u(x)$ after substituting back is the same as before:

$$a_1(\hat{u}, v) = L_1(v), \qquad \forall\, v(x) \in H^1_E(x_l, x_r),$$

where

$$\begin{aligned} a_1(\hat{u}, v) &= \int_{x_l}^{x_r} \left(p\hat{u}'v' + q\hat{u}v \right)\, dx + \frac{\alpha}{\beta} p(x_r)\hat{u}(x_r)v(x_r) \\ L_1(v) &= \int_{x_l}^{x_r} fv dx + \frac{\gamma}{\beta} p(x_r)v(x_r) + \int_{x_l}^{x_r} \left((pu_0')'v - qu_0 v \right)\, dx \\ &= \int_{x_l}^{x_r} fv dx + \frac{\gamma}{\beta} p(x_r)v(x_r) - \int_{x_l}^{x_r} \left(pu_0'v' + qu_0 v \right)\, dx - \frac{\alpha}{\beta} p(x_r)u_0(x_r)v(x_r). \end{aligned}$$

If we define

$$\begin{aligned} a(u, v) &= \int_{x_l}^{x_r} \left(pu'v' + quv \right)\, dx + \frac{\alpha}{\beta} p(x_r)u(x_r)v(x_r), \\ L(v) &= \int_{x_l}^{x_r} fv dx + \frac{\gamma}{\beta} p(x_r)v(x_r), \end{aligned}$$

as before, then we have

$$a_1(u - u_0, v) = a(u - u_0, v) = L_1(v) = L(v) - a(u_0, v), \quad \text{or} \quad a(u, v) = L(v). \tag{8.12}$$

While we still have $a(u, v) = L(v)$, the solution is not in $H^1_E(x_l, x_r)$ space due to the non-homogeneous Dirichlet boundary condition. Nevertheless $u - u_0$ is in $H^1_E(x_l, x_r)$. The formula above is also the basis of the numerical treatment of Dirichlet boundary conditions later.

8.2 The FE method for Sturm-Liouville problems

Let us now consider the finite element method using the piecewise linear function over a mesh $x_1 = x_l, x_2, \cdots, x_M = x_r$ for the Sturm-Liouville problem

$$\begin{aligned} &-(pu')' + qu = f\,, \quad x_l < x < x_r\,, \\ &u(x_l) = u_l\,, \quad \alpha u(x_r) + \beta u'(x_r) = \gamma, \quad \beta \neq 0\,, \quad \frac{\alpha}{\beta} \geq 0\,. \end{aligned}$$

We again use the hat functions as the basis such that

$$u_h(x) = \sum_{i=0}^{M} \alpha_i \phi_i(x)\,,$$

and now focus on the treatment of the BC.

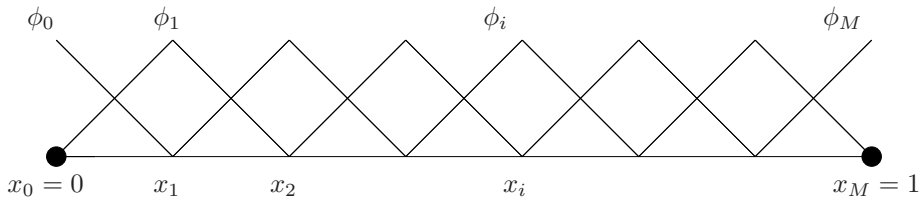


Figure 8.1. *Diagram of a 1D mesh where $x_l = 0$ and $x_r = 1$.*

The solution is unknown at $x = x_r$, so it is not surprising to have $\phi_M(x)$ for the natural BC. The first term $\phi_0(x)$ is the function used as $u_0(x)$, to deal with the nonhomogeneous Dirichlet BC. The local stiffness matrix is

$$K_i^e = \begin{bmatrix} a(\phi_i,\phi_i) & a(\phi_i,\phi_{i+1}) \\ a(\phi_{i+1},\phi_i) & a(\phi_{i+1},\phi_{i+1}) \end{bmatrix}_{(x_i,x_{i+1})} = \begin{bmatrix} \int_{x_i}^{x_{i+1}} p{\phi_i'}^2\,dx & \int_{x_i}^{x_{i+1}} p\phi_i'\phi_{i+1}'\,dx \\ \int_{x_i}^{x_{i+1}} p\phi_{i+1}'\phi_i'\,dx & \int_{x_i}^{x_{i+1}} p{\phi_{i+1}'}^2\,dx \end{bmatrix}$$

$$+ \begin{bmatrix} \int_{x_i}^{x_{i+1}} q{\phi_i}^2 dx & \int_{x_i}^{x_{i+1}} q\phi_i\phi_{i+1}\,dx \\ \int_{x_i}^{x_{i+1}} q\phi_{i+1}\phi_i\,dx & \int_{x_i}^{x_{i+1}} q\phi_{i+1}^2\,dx \end{bmatrix} + \frac{\alpha}{\beta}\,p(x_r) \begin{bmatrix} \phi_i^2(x_r) & \phi_i(x_r)\phi_{i+1}(x_r) \\ \phi_{i+1}(x_r)\phi_i(x_r) & \phi_{i+1}^2(x_r) \end{bmatrix},$$

and the local load vector is

$$F_i^e = \begin{bmatrix} L(\phi_i) \\ L(\phi_{i+1}) \end{bmatrix} = \begin{bmatrix} \int_{x_i}^{x_{i+1}} f\phi_i dx \\ \int_{x_i}^{x_{i+1}} f\phi_{i+1} dx \end{bmatrix} + \frac{\gamma}{\beta} p(x_r) \begin{bmatrix} \phi_i(x_r) \\ \phi_{i+1}(x_r) \end{bmatrix}.$$

We can see clearly the contributions from the BC; and in particular, that the only nonzero contribution of the BC to the stiffness matrix and the load vector is from the last element (x_{M-1}, x_M) due to the compactness of the hat functions.

8.2.1 Numerical treatments of Dirichlet BC

The finite element solution defined on the mesh can be written as

$$u_h(x) = u_l\,\phi_0(x) + \sum_{j=1}^{M} \alpha_j \phi_j(x)\,,$$

where α_j, $j = 1, 2, \cdots, M$ are the unknowns. Note that $u_l\phi_0(x)$ is an approximate particular solution in $H^1(x_l, x_r)$, and satisfies the Dirichlet boundary condition at $x = x_l$ and homogeneous Robin boundary condition at $x = x_r$. To use the finite element method to determine the coefficients, we enforce the weak form for $u_h(x) - u_a\phi_0(x)$ for the modified

differential problem,

$$\begin{aligned} -(pu')' + qu &= f + u_l\,(p\,\phi_0')' - u_l q\,\phi_0\,, \quad x_l < x < x_r\,, \\ u(x_l) = 0\,, &\qquad \alpha u(x_r) + \beta u'(x_r) = \gamma, \quad \beta \neq 0\,, \quad \frac{\alpha}{\beta} \geq 0\,. \end{aligned} \tag{8.13}$$

Thus the system of linear equations is

$$\hat{a}\,(u_h(x), \phi_i(x)) = \hat{L}(\phi_i)\,, \quad i = 1, 2, \cdots, M,$$

where $\hat{a}(:,:)$ and $\hat{L}(:)$ are the bilinear and linear for the BVP above, or equivalently

$$\begin{aligned} a(\phi_1,\phi_1)\alpha_1 + a(\phi_1,\phi_2)\alpha_2 + \cdots + a(\phi_1,\phi_M)\alpha_M &= L(\phi_1) - a(\phi_0,\phi_1)\,u_l \\ a(\phi_2,\phi_1)\alpha_1 + a(\phi_2,\phi_2)\alpha_2 + \cdots + a(\phi_2,\phi_M)\alpha_M &= L(\phi_2) - a(\phi_0,\phi_2)\,u_l \\ \cdots\cdots\cdots\cdots &= \cdots\cdots\cdots\cdots \\ a(\phi_M,\phi_1)\alpha_1 + a(\phi_M,\phi_2)\alpha_2 + \cdots + a(\phi_M,\phi_M)\alpha_M &= L(\phi_M) - a(\phi_0,\phi_M)\,u_l\,, \end{aligned}$$

where the bilinear and linear forms are still

$$\begin{aligned} a(u,v) &= \int_{x_l}^{x_r} \left(pu'v' + quv\right)\,dx + \frac{\alpha}{\beta}p(x_r)u(x_r)v(x_r); \\ L(v) &= \int_{x_l}^{x_r} fv dx + \frac{\gamma}{\beta}p(x_r)v(x_r), \end{aligned}$$

since

$$\int_{x_l}^{x_r} \left(u_l(p\,\phi_0')' - u_l q\,\phi_0\right)\phi_i(x)dx = -u_l a(\phi_0,\phi_i).$$

After moving the $a(\phi_i, u_l\phi_0(x)) = a(\phi_i,\phi_0(x))\,u_l$ to the right-hand side, we get the matrix-vector form

$$\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & a(\phi_1,\phi_1) & \cdots & a(\phi_1,\phi_M) \\ \vdots & \vdots & \vdots & \vdots \\ 0 & a(\phi_M,\phi_1) & \cdots & a(\phi_M,\phi_M) \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_M \end{bmatrix} = \begin{bmatrix} u_l \\ L(\phi_1) - a(\phi_1,\phi_0)u_l \\ \vdots \\ L(\phi_M) - a(\phi_M,\phi_0)u_l \end{bmatrix}.$$

This outlines one method to deal with a non-homogeneous Dirichlet boundary condition.

8.2.2 Contributions from Neumann or mixed BC

The contribution of mixed boundary condition at x_r using the hat basis functions is zero until the last element $[x_{M-1}, x_M]$, where $\phi_M(x_r)$ is not zero. The local stiffness matrix of the last element is

$$\begin{bmatrix} \displaystyle\int_{x_{M-1}}^{x_M} \left(p{\phi'_{M-1}}^2 + q\phi_{M-1}^2\right)dx & \displaystyle\int_{x_{M-1}}^{x_M} \left(p\phi'_{M-1}\phi'_M + q\phi_{M-1}\phi_M\right)dx \\ \displaystyle\int_{x_{M-1}}^{x_M} \left(p\phi'_M\phi'_{M-1} + q\phi_M\phi_{M-1}\right)dx & \displaystyle\int_{x_{M-1}}^{x_M} \left(p{\phi'_M}^2 + \phi_M^2\right)dx + \frac{\alpha}{\beta}p(x_r) \end{bmatrix},$$

and the local load vector is

$$F^e_{M-1} = \begin{bmatrix} \displaystyle\int_{x_{M-1}}^{x_M} f(x)\phi_{M-1}(x)\,dx \\ \displaystyle\int_{x_{M-1}}^{x_M} f(x)\phi_M(x)\,dx + \frac{\gamma}{\beta}p(x_r) \end{bmatrix}.$$

8.2.3 Pseudo-code of the FE method for 1D Sturm-Liouville problems using the hat basis functions

- Initialize:

```
for i=0, M
    F(i) = 0
    for j=0, M
        A(i,j) = 0
    end
end
```

- Assemble the coefficient matrix element by element:

for i=1, M

$$A(i-1,i-1) = A(i-1,i-1) + \int_{x_{i-1}}^{x_i} \left(p{\phi'_{i-1}}^2 + q\phi_{i-1}^2 \right) dx$$

$$A(i-1,i) = A(i-1,i) + \int_{x_{i-1}}^{x_i} \left(p\phi'_{i-1}\phi'_i + q\phi_{i-1}\phi_i \right) dx$$

$$A(i,i-1) = A(i-1,i)$$

$$A(i,i) = A(i,i) + \int_{x_{i-1}}^{x_i} \left(p{\phi'_i}^2 + q\phi_i^2 \right) dx$$

$$F(i-1) = F(i-1) + \int_{x_{i-1}}^{x_i} f(x)\phi_{i-1}(x)\,dx$$

$$F(i) = F(i) + \int_{x_{i-1}}^{x_i} f(x)\phi_i(x)\,dx$$

end

- Deal with the Dirichlet BC:

$A(0,0) = 1; \qquad F(0) = u_a$

for i=1, M

$F(i) = F(i) - A(i,0) * ua;$

$A(i,0) = 0; \qquad A(0,i) = 0;$

end

- Deal with the Mixed BC at $x = b$.

$$A(M, M) = A(M, M) + \frac{\alpha}{\beta} p(b);$$
$$F(M) = F(M) + \frac{\gamma}{\beta} p(b);$$

- Solve $AU = F$.
- Carry out the error analysis.

8.3 High order elements

To solve the Sturm-Liouville or other problems involving second order differential equations, we can use the piecewise linear finite dimensional space over a mesh. The error is usually $\mathrm{O}(h)$ in the energy and H^1 norms, and $\mathrm{O}(h^2)$ in the L^2 and L^∞ norms. If we want to improve the accuracy, we can choose to:

- refine the mesh, *i.e.*, decrease h; or
- use more accurate (high order) and larger finite dimensional spaces, *i.e.*, the piecewise quadratic or piecewise cubic basis functions.

Let us use the Sturm-Liouville problem

$$-\left(p'u\right)' + qu = f\,, \quad x_l < x < x_r\,,$$
$$u(x_l) = 0, \quad u(x_r) = 0\,,$$

again as the model problem for the discussion here. The other boundary conditions can be treated in a similar way, as discussed before. We assume a given mesh

$$x_0 = x_l,\ x_1,\ \cdots,\ x_M = x_r\,, \quad \text{and the elements,}$$
$$\Omega_1 = (x_0, x_1)\,, \quad \Omega_2 = (x_1, x_2)\,,\ \cdots,\ \Omega_M = (x_{M-1}, x_M),$$

and consider piecewise quadratic and piecewise cubic functions, but still require the finite dimensional spaces to be in $H_0^1(x_l, x_r)$ so that the finite element methods are conforming.

8.3.1 Piecewise quadratic basis functions

Define

$$V_h = \left\{\, v(x)\,,\ \text{where } v(x) \text{ is continuous piecewise quadratic in } H^1 \right\},$$

over a given mesh. The piecewise linear finite dimensional space is obviously a subspace of the space defined above, so the finite element solution is expected to be more accurate than the one obtained using the piecewise linear functions.

To use the Galerkin finite element method, we need to know *the dimension of the space* V_h of the piecewise quadratic functions in order to choose a set of basis functions. The dimension of a finite dimensional space is sometimes called the *the degree of freedom* (DOF). Given a function $\phi(x)$ in V_h, on each element a quadratic function has the form

$$\phi(x) = a_i x^2 + b_i x + c_i, \quad x_i \le x < x_{i+1}\,,$$

so there are three parameters to determine a quadratic function in the interval (x_i, x_{i+1}). In total, there are M elements, and so $3M$ parameters. However, they are not totally free, because they have to satisfy the continuity condition

$$\lim_{x\to x_i-} \phi(x) = \lim_{x\to x_i+} \phi(x)$$

for x_1, x_2, $\cdots$, x_{M-1}, or more precisely,

$$a_{i-1}x_i^2 + b_{i-1}x_i + c_{i-1} = a_i x_i^2 + b_i x_i + c_i\,, \quad i = 1, 2, \cdots M\,.$$

There are $M-1$ interior nodal points, so there are $M-1$ constraints, and $\phi(x)$ should also satisfy the BC $\phi(x_l) = \phi(x_r) = 0$. Thus the total degree of the freedom, the dimension of the finite element space, is

$$3M - (M-1) - 2 = 2M - 1\,.$$

We now know that the dimension of V_h is $2M-1$. If we can construct $2M-1$ basis functions that are linearly independent, then all of the functions in V_h can be expressed as linear combinations of them. The desired properties are similar to those of the hat basis functions; and they should

- be continuous piecewise quadratic functions;
- have minimum support, *i.e.*, be zero almost everywhere; and
- be determined by the values at nodal points (we can choose the nodal values to be unity at one point and zero at the other nodal points).

Since the degree of the freedom is $2M-1$ and there are only $M-1$ interior nodal points, we add M auxiliary points (not nodal points) between x_i and x_{i+1} and define

$$z_{2i} = x_i\,, \quad \text{nodal points}\,, \tag{8.14}$$

$$z_{2i+1} = \frac{x_i + x_{i+1}}{2}\,, \quad \text{auxiliary points}\,. \tag{8.15}$$

For instance, if the nodal points are $x_0 = 0$, $x_1 = \pi/2$, $x_2 = \pi$, then $z_0 = x_0$, $z_1 = \pi/4$, $z_2 = x_1 = \pi/2$, $z_3 = 3\pi/4$, $z_4 = x_2 = \pi$. Note that in general all the basis functions should be one piece in one element (z_{2k}, z_{2k+2}), $k = 0, 1, \cdots, M-1$. Now we can define the piecewise quadratic basis functions as

$$\phi_i(z_j) = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise}\,. \end{cases} \tag{8.16}$$

We can derive analytic expressions of the basis functions using the properties of quadratic polynomials. As an example, let us consider how to construct the basis functions in the first element (x_0, x_1) corresponding to the interval (z_0, z_2). In this element, z_0 is the boundary point, z_2 is a nodal point, and $z_1 = (z_0 + z_2)/2$ is the mid-point (the auxiliary point). For $\phi_1(x)$, we have $\phi_1(z_0) = 0$, $\phi_1(z_1) = 1$ and $\phi_1(z_2) = 0$, $\phi_1(z_j) = 0$, $j = 3, \cdots, 2M-1$; so in the interval (z_0, z_1), $\phi_1(x)$ has the form

$$\phi_1(x) = C(x - z_0)(x - z_2)\,,$$

because z_0 and z_2 are roots of $\phi_1(x)$. We choose C such that $\phi_1(z_1) = 1$, so

$$\phi_1(z_1) = C(z_1 - z_0)(z_1 - z_2) = 1, \quad \Longrightarrow C = \frac{1}{(z_1 - z_0)(z_1 - z_2)}$$

and the basis function $\phi_1(x)$ is

$$\phi_1(x) = \begin{cases} \dfrac{(x - z_0)(x - z_2)}{(z_1 - z_0)(z_1 - z_2)} & \text{if } z_0 \le x < z_2\,, \\ 0 & \text{otherwise}\,. \end{cases}$$

It is easy to verify that $\phi_1(x)$ is a continuous piecewise quadratic function in the domain (x_0, x_M). Similarly, we have

$$\phi_2(x) = \frac{(z - z_1)(z - z_0)}{(z_2 - z_1)(z_2 - z_0)}\,.$$

Generally the three global basis functions that are nonzero in the element (x_i, x_{i+1}) have the following forms:

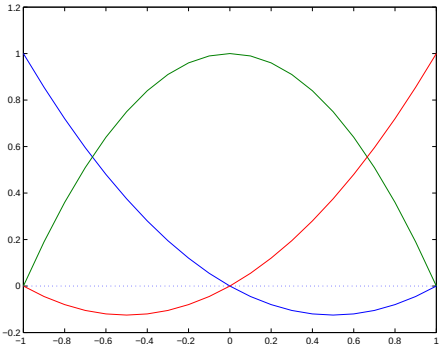
$$\phi_{2i}(z) = \begin{cases} 0 & \text{if } z < x_{i-1} \\ \dfrac{(z - z_{2i-1})(z - z_{2i-2})}{(z_{2i} - z_{2i-1})(z_{2i} - z_{2i-2})} & \text{if } x_{i-1} \le z < x_i \\ \dfrac{(z - z_{2i+1})(z - z_{2i+2})}{(z_{2i} - z_{2i+1})(z_{2i} - z_{2i+2})} & \text{if } x_i \le z < x_{i+1} \\ 0 & \text{if } x_{i+1} < z. \end{cases}$$

$$\phi_{2i+1}(z) = \begin{cases} 0 & \text{if } x < x_i \\ \dfrac{(z - z_{2i})(z - z_{2i+2})}{(z_{2i+1} - z_{2i})(z_{2i+1} - z_{2i+2})} & \text{if } x_i \le z < x_{i+1} \\ 0 & \text{if } x_{i+1} < z\,, \end{cases}$$

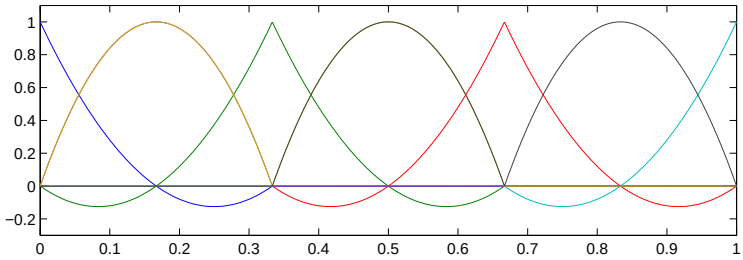
$$\phi_{2i+2}(z) = \begin{cases} 0 & \text{if } x < x_i \\ \dfrac{(z - z_{2i})(z - z_{2i+1})}{(z_{2i+2} - z_{2i})(z_{2i+2} - z_{2i+1})} & \text{if } x_i \le z < x_{i+1} \\ \dfrac{(z - z_{2i+3})(z - z_{2i+4})}{(z_{2i+2} - z_{2i+3})(z_{2i+2} - z_{2i+4})} & \text{if } x_i \le z < x_{i+1} \\ 0 & \text{if } x_{i+2} < z\,. \end{cases}$$

In Fig. 8.2, we plot some quadratic basis functions in H^1. Fig. 8.2 (a) is the plot of the shape functions, that is, the nonzero basis functions defined in the interval $(-1,\ 1)$. In Fig. 8.2 (b), we plot all the basis functions over a three-node mesh in $(0,\ 1)$. In Fig. 8.2 (c), we plot some basis functions over the entire domain, $\phi_0(x)$, $\phi_1(x)$, $\phi_2(x)$, $\phi_4(x)$, $\phi_4(x)$, where $\phi_1(x)$ is centered at the auxiliary point z_1 and nonzero at only one element while $\phi_2(x)$, $\phi_4(x)$ are nonzero at two elements.

(a)



(b)



(c)

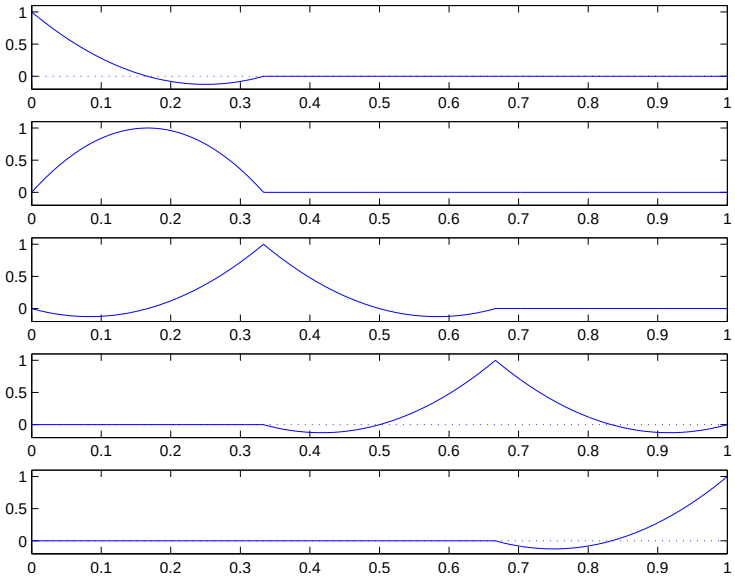


Figure 8.2. *Quadratic basis functions in H^1: (a) the shape functions (basis functions in $(-1, 1)$); (b) all the basis functions over a three-node mesh in $(0, 1)$; (c) plot of some basis functions over the entire domain, $\phi_0(x)$, $\phi_1(x)$, $\phi_2(x)$, $\phi_4(x)$, $\phi_4(x)$, where $\phi_1(x)$ is centered at the auxiliary point z_1 and nonzero at only one element, while $\phi_2(x)$, $\phi_4(x)$ are nonzero at two elements.*

8.3.2 Assembling the stiffness matrix and the load vector

The finite element solution can be written as

$$u_h(x) = \sum_{i=1}^{2M-1} \alpha_i \phi_i(x)\,.$$

The entries of the coefficient matrix are $\{a_{ij}\} = a(\phi_i, \phi_j)$ and the load vector is $F_i = L(\phi_i)$. On each element (x_i, x_{i+1}), or (z_{2i}, z_{2i+2}), there are three nonzero basis functions: ϕ_{2i}, ϕ_{2i+1}, and ϕ_{2i+2}. Thus the local stiffness matrix is

$$K_i^e = \left[\begin{array}{ccc} a(\phi_{2i}, \phi_{2i}) & a(\phi_{2i}, \phi_{2i+1}) & a(\phi_{2i}, \phi_{2i+2}) \\ a(\phi_{2i+1}, \phi_{2i}) & a(\phi_{2i+1}, \phi_{2i+1}) & a(\phi_{2i+1}, \phi_{2i+2}) \\ a(\phi_{2i+2}, \phi_{2i}) & a(\phi_{2i+2}, \phi_{2i+1}) & a(\phi_{2i+2}, \phi_{2i+2}) \end{array} \right]_{(x_i, x_{i+1})} \tag{8.17}$$

and the local load vector is

$$L_i^e = \left[\begin{array}{c} L(\phi_{2i}) \\ L(\phi_{2i+1}) \\ L(\phi_{2i+2}) \end{array} \right]_{(x_i, x_{i+1})} . \tag{8.18}$$

The stiffness matrix is still symmetric positive definite, but denser than that with the hat basis functions. It is still a banded matrix, with the band width five, a penta-diagonal matrix. The advantage in using quadratic basis functions is that the finite element solution is more accurate than that obtained on using the linear basis functions with the same mesh.

8.3.3 The cubic basis functions in $H^1(x_l, x_r)$ space

We can also construct piecewise cubic basis functions in $H^1(x_l, x_r)$. On each element (x_i, x_{i+1}), a cubic function has the form

$$\phi(x) = a_i x^3 + b_i x^2 + c_i x + d_i\,, \quad i = 0, 1, \cdots, M-1\,.$$

There are four parameters; and the total degree of freedom is $3M-1$ if a Dirichlet boundary condition is imposed at both ends. To construct cubic basis functions with properties similar to the piecewise linear and quadratic basis functions, we need to add two auxiliary points between x_i and x_{i+1}. The local stiffness matrix is then is a 4×4 matrix. We leave the construction of the basis functions, application to the Sturm-Liouville boundary value problems as a project for students.

8.4 A 1D Matlab FE package

A general 1D Matlab package has been written and is available is the book's depository, or upon request.

- The code can be used to solve a general Sturm-Liouville problem

$$-(p(x)u')' + c(x)u' + q(x)u = f(x)\,, \quad a < x < b\,,$$

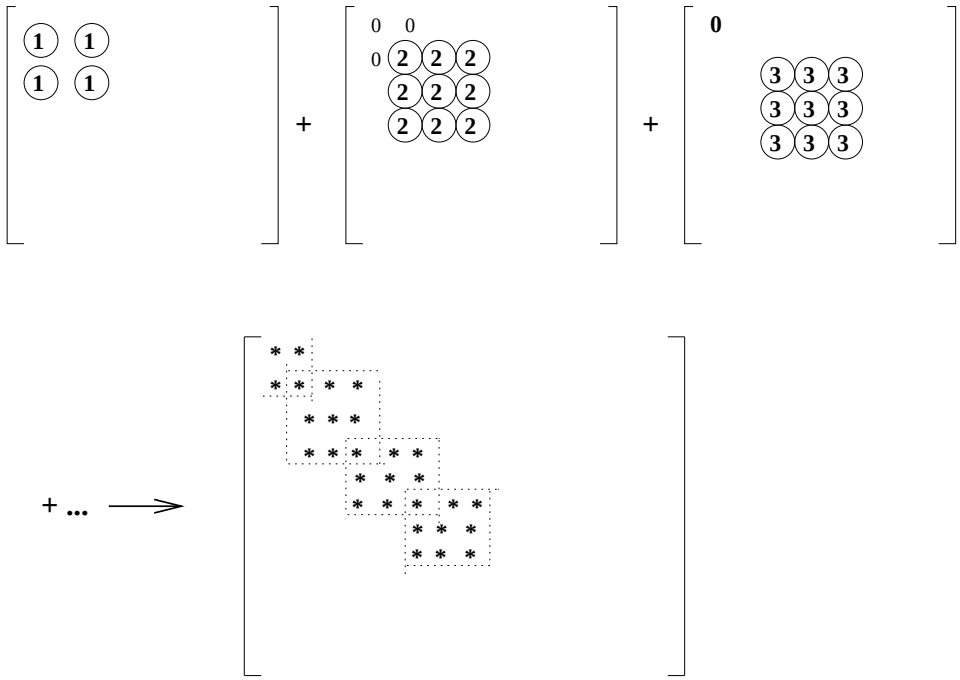


Figure 8.3. *Assembling the stiffness matrix using piecewise quadratic basis functions.*

with a Dirichlet, Neumann, or mixed boundary condition at $x = a$ and $x = b$[6].

- We use conforming finite element methods.
- The mesh is

$$x_0 = a < x_1 < x_2 \cdots < x_M = b\,,$$

as elaborated again later.

- The finite element spaces can be piecewise linear, quadratic, or cubic functions over the mesh.
- The integration formulas is the Gaussian quadrature of order one, two, three, or four.
- The matrix assembly is element by element.

8.4.1 Gaussian quadrature formulas

In a finite element method, we typically need to evaluate integrals such as $\int_a^b p(x)\phi_i'(x)\phi_j'(x)\,dx$, $\int_a^b q(x)\phi_i(x)\phi_j(x)\,dx$ and $\int_a^b f(x)\phi_i(x)\,dx$ over some intervals $(a,\ b)$ such as $(x_{i-1},\ x_i)$.

[6] In the package, $p(x)$ is expressed as $k(x)$.

Although the functions involved may be arbitrary, it is usually neither practical nor necessary to find the exact integrals. A standard approach is to transfer the interval of integration to the interval $(-1,\ 1)$ as follows

$$\int_a^b f(x)\,dx = \int_{-1}^1 \bar{f}(\xi)\,d\xi, \tag{8.19}$$

where

$$\begin{aligned} &\xi = \frac{x-a}{b-a} + \frac{x-b}{b-a} \quad \text{or} \quad x = a + \frac{b-a}{2}\Big(1+\xi\Big), \\ \Longrightarrow \qquad &d\xi = \frac{2}{b-a}dx \ \text{ or } \ dx = \frac{b-a}{2}d\xi\,. \end{aligned} \tag{8.20}$$

In this way, we have

$$\int_a^b f(x)\,dx = \frac{b-a}{2}\int_{-1}^1 f\left(a + \frac{b-a}{2}(1+\xi)\right)d\xi = \frac{b-a}{2}\int_{-1}^1 \bar{f}(\xi)\,d\xi\,, \tag{8.21}$$

where $\bar{f}(\xi) = f(a + \frac{b-a}{2}(1+\xi))$; and then to use a Gaussian quadrature formula to approximate the integral. The general quadrature formula can be written

$$\int_{-1}^1 g(\xi)d\xi \approx \sum_{i=1}^N w_i g(\xi_i)\,,$$

where the Gaussian points ξ_i and weights w_i are chosen so that the quadrature is as accurate as possible. In the Newton-Cotes quadrature formulas such as the mid-point rule, the trapezoidal rule, and the Simpson methods, the ξ_i are pre-defined independent of w_i. In Gaussian quadrature formulas, all ξ_i's and w_i's are unknowns, and are determined simultaneously such that the quadrature formula is exact for $g(x) = 1, x, \cdots, x^{2N-1}$. The number $2N-1$ is called the *algebraic precision* of the quadrature formula.

Gaussian quadrature formulas have the following features:

- accurate with the best possible algebraic precision using the fewest points;
- open with no need to use two end points where some kind of discontinuities may occur, *e.g.*, the discontinuous derivatives of the piecewise linear functions at nodal points;
- no recursive relations for the Gaussian points ξ_i's and the weights w_i's;
- accurate enough for finite element methods, because $b-a \sim h$ is generally small and only a few points ξ_i's are needed.

We discuss some Gaussian quadrature formulas below.

Gaussian quadrature of order 1 (one point):

With only one point, the Gaussian quadrature can be written as

$$\int_{-1}^1 g(\xi)d\xi = w_1 g(\xi_1)\,.$$

We choose ξ_1 and w_1 such that the quadrature formula has the highest algebraic precision ($2N-1=1$ for $N=1$) if only one point is used. Thus we choose $g(\xi)=1$ and $g(\xi)=\xi$ to have the following,

$$\text{for } g(\xi)=1\,, \quad \int_{-1}^{1} g(\xi)d\xi = 2 \implies 2 = w_1 \cdot 1\,; \quad \text{and}$$
$$\text{for } g(\xi)=\xi\,, \quad \int_{-1}^{1} g(\xi)d\xi = 0 \implies 0 = w_1\xi_1\,.$$

Thus we get $w_1=2$ and $\xi_1=0$. The quadrature formula is simply the mid-point rule.

Gaussian quadrature of order 2 (two points):

With two points, the Gaussian quadrature can be written as

$$\int_{-1}^{1} g(\xi)d\xi = w_1 g(\xi_1) + w_2 g(\xi_2)\,.$$

We choose ξ_1, ξ_2, and w_1, w_2 such that the quadrature formula has the highest algebraic precision ($2N-1=3$ for $N=2$) if two points are used. Thus we choose $g(\xi)=1$, $g(\xi)=\xi$, $g(\xi)=\xi^2$, and $g(\xi)=\xi^3$ to have the following,

$$\text{for } g(\xi)=1\,, \quad \int_{-1}^{1} g(\xi)d\xi = 2 \implies 2 = w_1 + w_2\,;$$
$$\text{for } g(\xi)=\xi\,, \quad \int_{-1}^{1} g(\xi)d\xi = 0 \implies 0 = w_1\xi_1 + w_2\xi_2\,;$$
$$\text{for } g(\xi)=\xi^2\,, \quad \int_{-1}^{1} g(\xi)d\xi = \frac{2}{3} \implies \frac{2}{3} = w_1\xi_1^2 + w_2\xi_2^2\,; \quad \text{and}$$
$$\text{for } g(\xi)=\xi^3\,, \quad \int_{-1}^{1} g(\xi)d\xi = 0 \implies 0 = w_1\xi_1^3 + w_2\xi_2^3\,.$$

On solving the four non-linear system of equations by taking advantage of the symmetry, we get

$$w_1 = w_2 = 1\,, \quad \xi_1 = -\frac{1}{\sqrt{3}} \text{ and } \xi_2 = \frac{1}{\sqrt{3}}\,.$$

So the Gaussian quadrature formula of order 2 is

$$\int_{-1}^{1} g(\xi)d\xi \simeq g\left(-\frac{1}{\sqrt{3}}\right) + g\left(\frac{1}{\sqrt{3}}\right). \tag{8.22}$$

Higher order Gaussian quadrature formulas are likewise obtained, and for efficiency

we can pre-store the Gaussian points and weights in two separate matrices:

$$\begin{matrix} & & & \xi_i & \\ \end{matrix}$$

$$\begin{bmatrix} 0 & \frac{-1}{\sqrt{3}} & \frac{-\sqrt{3}}{\sqrt{5}} & -0.8611363116 & \cdots \\ & \frac{1}{\sqrt{3}} & 0 & -0.3399810436 & \cdots \\ & & \frac{\sqrt{3}}{\sqrt{5}} & 0.3399810436 & \cdots \\ & & & 0.8611363116 & \cdots \end{bmatrix}, \quad \begin{matrix} & & & w_i & \\ \end{matrix} \quad \begin{bmatrix} 2 & 1 & \frac{5}{9} & 0.3478548451 & \cdots \\ & 1 & \frac{8}{9} & 0.6521451549 & \cdots \\ & & \frac{5}{9} & 0.6521451549 & \cdots \\ & & & 0.3478548451 & \cdots \end{bmatrix}.$$

Below is a Matlab code setint.m to store the Gaussian points and weights up to order 4.

```
function [xi,w] = setint

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                              %
%  Function setint provides the Gaussian points x(i), and the                  %
%  weights of the Gaussian quadrature formula.                                 %
%  Output:                                                                     %
%     x(4,4):  x(:,i) is the Gaussian points of order i.                       %
%     w(4,4):  w(:,i) is the weights of quadrature of order i.                 %

  clear x;   clear w

     xi(1,1) = 0;
     w(1,1) = 2; % Gaussian quadrature of order 1

     xi(1,2) = -1/sqrt(3);
     xi(2,2) = -xi(1,2);
     w(1,2) = 1;
     w(2,2) = w(1,2); % Gaussian quadrature of order 2

     xi(1,3) = -sqrt(3/5);
     xi(2,3) = 0;
     xi(3,3) = -xi(1,3);
     w(1,3) = 5/9;
     w(2,3) = 8/9;
     w(3,3) = w(1,3);         % Gaussian quadrature of order 3

     xi(1,4) = - 0.8611363116;
     xi(2,4) = - 0.3399810436;
     xi(3,4) = -xi(2,4);
     xi(4,4) = -xi(1,4);
```

```
    w(1,4) = 0.3478548451;
    w(2,4) = 0.6521451549;
    w(3,4) = w(2,4);
    w(4,4) = w(1,4); % Gaussian quadrature of order 4

    return

%-------------------------- END OF SETINT -----------------------------
```

8.4.2 Shape functions

Similar to transforming an integral over some arbitrary interval to the integral over the standard interval between -1 and 1, it is easier to evaluate the basis functions and their derivatives in the standard interval $(-1,\ 1)$. Basis functions in the standard interval $(-1,\ 1)$ are called *shape functions* and often have analytic forms.

Using the transform between x and ξ in (8.19)-(8.20) for each element, on assuming $c(x) = 0$ we have

$$\int_{x_i}^{x_{i+1}} \left(p(x)\phi_i'\phi_j' + q(x)\phi_i\phi_j\right)\, dx = \int_{x_i}^{x_{i+1}} f(x)\phi_i(x)\, dx$$

which is transformed to

$$\frac{x_{i+1}-x_i}{2}\int_{-1}^{1} \left(\bar{p}(\xi)\psi_i'\psi_j' + \bar{q}(\xi)\psi_i\psi_j\right)\, d\xi = \frac{x_{i+1}-x_i}{2}\int_{-1}^{1} \bar{f}(\xi)\psi_i\, d\xi$$

where

$$\bar{p}(\xi) = p\left(x_i + \frac{x_{i+1}-x_i}{2}(1+\xi)\right),$$

and so on. Here ψ_i and ψ_j are the local basis functions under the new variables, *i.e.*, the shape functions and their derivatives. For piecewise linear functions, there are only two nonzero shape functions

$$\psi_1 = \frac{1-\xi}{2}, \qquad \psi_2 = \frac{1+\xi}{2}, \tag{8.23}$$

$$\text{with derivatives} \quad \psi_1' = -\frac{1}{2}, \qquad \psi_2' = \frac{1}{2}. \tag{8.24}$$

There are three nonzero quadratic shape functions

$$\psi_1 = \frac{\xi(\xi-1)}{2}, \qquad \psi_2 = 1-\xi^2, \qquad \psi_3 = \frac{\xi(\xi+1)}{2}, \tag{8.25}$$

$$\text{with derivatives} \quad \psi_1' = \xi - \frac{1}{2}, \qquad \psi_2' = -2\xi, \qquad \psi_3' = \xi + \frac{1}{2}. \tag{8.26}$$

These hat (linear) and quadratic shape functions are plotted in Fig. 8.4.

It is noted that there is an extra factor in the derivatives with respect to x, due to the transform:

$$\frac{d\phi_i}{dx} = \frac{d\psi_i}{d\xi}\frac{d\xi}{dx} = \psi_i'\frac{2}{x_{i+1}-x_i}.$$

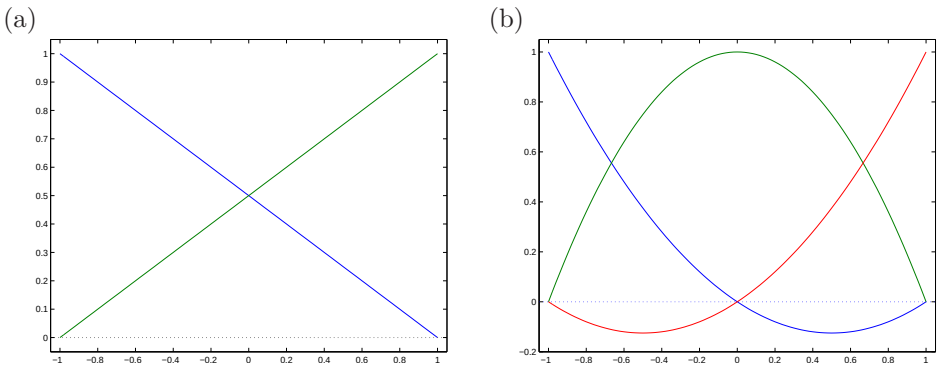


Figure 8.4. *Plot of some shape functions: (a) the hat (linear) functions; (b) the quadratic functions.*

The shape functions can be defined in a Matlab function

$$[psi,\ \ dpsi] = shape(xi, n)\,,$$

where $n = 1$ renders the linear basis function, $n = 2$ the quadratic basis function, and $n = 3$ the cubic basis function values. For example, with $n = 2$ the outputs are

$$\begin{aligned} &psi(1)\,, \quad psi(2)\,, \quad psi(3)\,, \qquad \text{three basis function values,}\\ &dpsi(1)\,, \quad dpsi(2)\,, \quad dpsi(3)\,, \quad \text{three derivative values.}\end{aligned}$$

The Matlab subroutine is as follows.

```
function [psi,dpsi]=shape(xi,n);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                               %
%   Function ‘‘shape’’ evaluates the values of the basis functions   %
%   and their derivatives at a point xi.                                        %
%                                                                               %
%   n: The basis function. n=2, linear, n=3, quadratic, n=3, cubic.  %
%   xi: The point where the base function is evaluated.                         %
%   Output:                                                                     %
%   psi:  The value of the base function at xi.                                 %
%   dpsi: The derivative of the base function at xi.                            %
%-------------------------------------------------------------------------------%

   switch n
   case 2,
     % Linear base function
     psi(1) = (1-xi)/2;
     psi(2) = (1+xi)/2;
```

```
    dpsi(1) = -0.5;
    dpsi(2) = 0.5;
    return

  case 3,
    % quadratic base function
    psi(1) = xi*(xi-1)/2;
    psi(2) = 1-xi*xi;
    psi(3) = xi*(xi+1)/2;
    dpsi(1) =  xi-0.5;
    dpsi(2) = -2*xi;
    dpsi(3) = xi + 0.5;
    return

  case 4,
    % cubic  base function
    psi(1) = 9*(1/9-xi*xi)*(xi-1)/16;
    psi(2) = 27*(1-xi*xi)*(1/3-xi)/16;
    psi(3) = 27*(1-xi*xi)*(1/3+xi)/16;
    psi(4) = -9*(1/9-xi*xi)*(1+xi)/16;

    dpsi(1) = -9*(3*xi*xi-2*xi-1/9)/16;
    dpsi(2) = 27*(3*xi*xi-2*xi/3-1)/16;
    dpsi(3) = 27*(-3*xi*xi-2*xi/3+1)/16;
    dpsi(4) = -9*(-3*xi*xi-2*xi+1/9)/16;
    return

  end
%--------------------------- END OF SHAPE ------------------------------
```

8.4.3 The main data structure

In one space dimension, a mesh is a set of ordered points as described below.

- Nodal points: $x_1 = a$, x_2, $\cdots$, $x_{nnode} = b$. The number of total nodal points plus the auxiliary points is $nnode$.

- Elements: Ω_1, Ω_2, $\cdots$, Ω_{nelem}. The number of elements is $nelem$.

- Connection between the nodal points and the elements: $nodes(nnode, nelem)$, where $nodes(j, i)$ is the j-th index of the nodes in the i-th element. For the linear basis function, $j = 1, 2$ since there are two nodes in an element; for the quadratic basis function, $j = 1, 2, 3$ since there are two nodes and an auxiliary point.

Example. Given the mesh and the indexing of the nodal points and the elements in Fig. 8.5, for linear basis functions, we have

$$\begin{aligned} nodes(1,1) &= 1\,, & nodes(1,2) &= 3\,,\\ nodes(2,1) &= 3\,, & nodes(2,2) &= 4\,,\\ nodes(1,3) &= 4\,, & nodes(1,4) &= 2\,,\\ nodes(2,3) &= 2\,, & nodes(2,4) &= 5\,. \end{aligned}$$

Example. Given the mesh and the indexing of the nodal points and the elements in Fig. 8.5, for quadratic basis functions, we have

$$\begin{aligned} nodes(1,1) &= 4\,, & nodes(2,1) &= 2\,, & nodes(3,1) &= 5\,,\\ nodes(1,2) &= 1\,, & nodes(2,2) &= 3\,, & nodes(3,2) &= 4\,. \end{aligned}$$

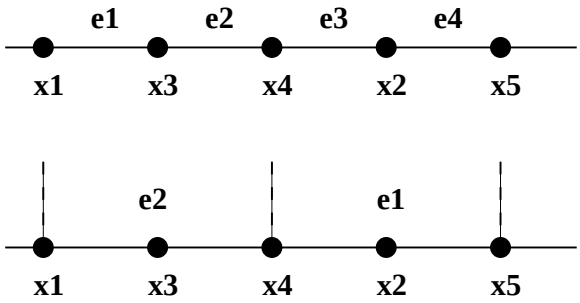


Figure 8.5. *Example of the relation between nodes and elements: (a) linear basis functions; (b) quadratic basis functions.*

8.4.4 Outline of the algorithm

```
      function  [x,u]=fem1d

        global nnode nelem
        global gk gf
        global xi  w

%%% Output: x are nodal points; u is the FE solution at
%%%         nodal points.

      [xi,w] = setint;     % Get Gaussian points and weights.

%%% Input data, pre-process
```

```
      [x,kbc,ubc,kind,nint,nodes] = prospset;

  %%%  x(nnode): Nodal points,        kbc, ubc: Boundary conditions

  %%%  kind(nelen): Choice of FE spaces. kind(i)=1,2,3 indicate
  %%%  piecewise linear, quadratic, and cubic FE space over the
  %%%  triangulation.

  %%%  nint(nelen): Choice of Gaussian quadrature. nint(i)=1,2,3,4
  %%%  indicate Gaussian order 1, 2, 3, 4.

      formkf(kind,nint,nodes,x,xi,w);

  %%%  Assembling the stiffness matrix and the load vector element by
  %%%  element.

      aplyb(kbc,ubc);

  %%%  Deal with the BC.

      u = gk\gf;             % Solve the linear system of equations

  %%%  Error analysis ...
```

8.4.5 Assembling element by element

The Matlab code is formkf.m

```
   function formkf(kind,nint,nodes,x,xi,w)

    ............

    for nel = 1:nelem,
       n = kind(nel) + 1;       % Linear FE space. n = 2, quadratic n=3, ..

       i1 = nodes(1,nel);       % The first node in nel-th element.
       i2 = nodes(n,nel);       % The last node in nel-th element.
       i3 = nint(nel);          % Order of Gaussian quadrature.
       xic = xi(:,i3);          % Get Gaussian points in the column.
       wc = w(:,i3);            % Get Gaussian weights.

%%% Evaluate the local stiffness matrix ek, and the load vector ef.
       [ek,ef] = elem(x(i1),x(i2),n,i3,xic,wc);
```

```
%%% Assembling to the global stiffness matrix gk, and the load vector gf.
       assemb(ek,ef,nel,n,nodes);

    end
```

Evaluation of local stiffness matrix and the load vector

The Matlab code is elem.m

```
   function [ek,ef] = elem(x1,x2,n,nl,xi,w)
   dx = (x2-x1)/2;

% [x1,x2] is an element [x1,x2]
% n is the choice of FE space. Linear n=2; quadratic n=3; ...

  for l=1:nl,                           % Quadrature formula that summarize.
    x = x1 + (1.0 + xi(l))*dx;          % Transform the Gaussian points.
    [xp,xc,xb,xf] = getmat(x);          % Get the coefficients at the
                                        % Gaussian points.
    [psi,dpsi] = shape(xi(l),n);        % Get the shape function and
                                        % its derivatives.
% Assembling the local stiffness matrix and the load vector.
% Notice the additional factor 1/dx in the derivatives.
    for i=1:n,
       ef(i) = ef(i) + psi(i)*xf*w(l)*dx;
       for j=1:n,
          ek(i,j)=ek(i,j)+(xk*dpsi(i)*dpsi(j)/(dx*dx) ...
               +xc*psi(i)*dpsi(j)/dx+xb*psi(i)*psi(j) )*w(l)*dx;
       end
    end
  end
```

Global assembling

The Matlab code is assemb.m

```
    function assemb(ek,ef,nel,n,nodes)
    global gk gf

    for i=1:n,                % Connection between nodes and the elements
      ig = nodes(i,nel);                  % Assemble global vector gf
      gf(ig) = gf(ig) + ef(i);

       for j=1:n,
         jg = nodes(j,nel);         % Assemble global stiffness matrix gk
```

```
      gk(ig,jg) = gk(ig,jg) + ek(i,j);
    end
  end
```

Input Data

The Matlab code is propset.m

```
function [x,kbc,vbc,kind,nint,nodes] = propset
```

- The relation between the number of nodes $nnode$ and the number of elements $nelem$:
  Linear: $nelem = nnode - 1$.
  Quadratic: $nelem = (nnode - 1)/2$.
  Cubic: $nelem = (nnode - 1)/3$.
- Nodes arranged in ascendant order. Equally spaced points are grouped together. The Matlab code is datain.m

  function [data] = datain(a,b,nnode,nelem)

  The output data has $nrec$ groups

  $data(i,1) = n1\,,$ index of the beginning of nodes.
  $data(i,2) = n2\,,$ number of points in this group.
  $data(i,3) = x(n1)\,,$ the first nodal point.
  $data(i,4) = x(n1+n2)\,,$ the last nodal point in this group.

  The simple case is

  $$data(i,1) = i\,, \quad data(i,2) = 0\,, \quad data(i,3) = x(i)\,, \quad data(i,4) = x(i)\,.$$

- The basis functions to be used in each element:

```
for i=1:nelem
   kind(i) = inf_ele = 1, or 2, or 3.
   nint(i) = 1, or 2, or 3, or 4.
   for j=1,kind(i)+1
     nodes(j,i) = j + kind(i)*(i-1);
   end
end
```

8.4.6 Input boundary conditions

The Matlab code aplybc.m involves an array of two elements $kbc(2)$ and a data array $vbc(2,2)$. At the left boundary

$kbc(1) = 1\,, \quad vbc(1,1) = u_a\,,$ Dirichlet BC at the left end;
$kbc(1) = 2\,, \quad vbc(1,1) = -k(a)u'(a)\,,$ Neumann BC at the left end;
$kbc(1) = 3\,, \quad vbc(1,1) = uxma\,, \quad vbc(2,1) = uaa\,,$ Mixed BC of the form:
$k(a)u'(a) = uxma(u(a) - uaa)\,.$

The BC will affect the stiffness matrix and the load vector and are handled in Matlab codes aplybc.m and drchlta.m.

- Dirichlet BC $u(a) = u_a = vbc(1,1)$.

```
for i=1:nnode,
    gf(i) = gf(i) - gk(i,1)*vbc(1,1);
    gk(i,1) = 0;    gk(1,i) = 0;
end
gk(1,1) = 1;    gf(1) = vbc(1,1);
```

  where gk is the global stiffness matrix and gf is the global load vector.

- Neumann BC $u'(a) = u_{xa}$. The boundary condition can be re-written as $-k(a)u'(a) = -k(a)u_{xa} = vbc(1,1)$. We only need to change the load vector.

```
gf(1) = gf(1) + vbc(1,1);
```

- Mixed BC $\alpha u(a) + \beta u'(a) = \gamma$, $\beta \neq 0$. The BC can be re-written as

$$\begin{aligned} k(a)u'(a) &= -\frac{\alpha}{\beta}k(a)\left(u(a) - \frac{\gamma}{\alpha}\right) \\ &= u_{xma}\left(u(a) - u_{aa}\right) = \mathbf{vbc(1,1)}\left(u(a) - \mathbf{vbc(2,1)}\right). \end{aligned}$$

  We need to change both the global stiffness matrix and the global load vector.

```
gf(1) = gf(1) + vbc(1,1)*vbc(2,1);
gk(1,1) =  gk(1,1) + vbc(1,1);
```

Examples.

1. $u(a) = 2$, we should set $kbc(1) = 1$ and $vbc(1,1) = 2$.
2. $k(x) = 2 + x^2$, $a = 2$, $u'(2) = 2$. Since $k(a) = k(2) = 6$ and $-k(a)u'(a) = -12$, we should set $kbc(1) = 2$ and $vbc(1,1) = -12$.
3. $k(x) = 2 + x^2$, $a = 2$, $2u(a) + 3u'(a) = 1$. Since

$$\begin{aligned} 3u'(a) &= -2u(a) + 1\,, \\ 6u'(a) &= -4u(a) + 2 = -4\left(u(a) - \frac{1}{2}\right), \end{aligned}$$

   we should set $kbc(1) = 3$, $vbc(1,1) = -4$ and $vbc(2,1) = 1/2$.

   Similarly, at the right BC $x = b$ we should have

$kbc(2) = 1\,,\quad vbc(1,2) = u_b\,,\quad$ Dirichlet BC at the right end;
$kbc(2) = 2\,,\quad vbc(1,2) = k(b)u'(b)\,,\quad$ Neumann BC at the right end;
$kbc(2) = 3\,,\quad vbc(1,2) = uxmb\,,\quad vbc(2,2) = ubb\,,\quad$ Mixed BC of the form
$-k(b)u'(b) = uxmb(u(b) - ubb)\,.$

The BC will affect the stiffness matrix and the load vector and are handled in Matlab codes aplybc.m and drchlta.m.

- Dirichlet BC $u(b) = u_b = vbc(1,2)$.

```
for i=1:nnode,
    gf(i) = gf(i) - gk(i,nnode)*vbc(1,2);
    gk(i,nnode) = 0;    gk(nnode,i) = 0;
end
gk(nnode,nnode) = 1;    gf(nnode) = vbc(1,1).
```

- Neumann BC $u'(b) = u_{xb}$ is given. The boundary condition can be re-written as $k(b)u'(b) = k(b)u_{xb} = vbc(1,2)$. We only need to change the load vector.

```
gf(nnode) = gf(nnode) + vbc(1,2);
```

- Mixed BC $\alpha u(b) + \beta u'(b) = \gamma$, $\beta \neq 0$. The BC can be re-written as

$$\begin{aligned} -k(b)u'(b) &= \frac{\alpha}{\beta}k(b)\left(u(b) - \frac{\gamma}{\alpha}\right) \\ &= u_{xmb}\left(u(b) - u_{bb}\right) = \mathbf{vbc}(\mathbf{1},\mathbf{2})\left(u(b) - \mathbf{vbc}(\mathbf{2},\mathbf{2})\right). \end{aligned}$$

We need to change both the global stiffness matrix and the global load vector.

```
gf(nnode) = gf(nnode) + vbc(1,2)*vbc(2,2);
gk(nnode,nnode) =  gk(nnode,nnode) + vbc(1,2);
```

8.4.7 A testing example

To check the code, we often try to compare the numerical results with some known exact solution, *e.g.*, we can choose

$$u(x) = \sin x\,, \qquad a \le x \le b\,.$$

If we set the material parameters as

$$p(x) = 1 + x\,, \quad c(x) = \cos x\,, \quad q(x) = x^2,$$

then the right hand side can be calculated as

$$f(x) = (pu')' + cu' + qu = (1+x)\sin x - \cos x + \cos^2 x + x^2 \sin x\,.$$

These functions are defined in the Matlab code getmat.m

```
function [xp,xc,xq,xf] = getmat(x);
  xp = 1+x; xc = cos(x); xq = x*x;
  xf = (1+x)*sin(x)-cos(x)+cos(x)*cos(x)+x*x*sin(x);
```

The mesh is defined in the Matlab code datain.m. All other parameters used for the finite element method are defined in the Matlab code propset.m, including the following:

- The boundary $x = a$ and $x = b$, *e.g.*, $a = 1$, $b = 4$.
- The number of nodal points, *e.g.*, $nnode = 41$.
- The choice of basis functions. If we use the same basis function, then for example we can set $inf\_ele = 2$, which is the quadratic basis function $kind(i) = inf\_ele$.
- The number of elements. If we use uniform elements, then $nelem = (nnode - 1)/inf\_ele$. We need to make it an integer.

- The choice of Gaussian quadrature formula, *e.g.*, $nint(i) = 4$. The order of the Gaussian quadrature formula should be the same or higher than the order of the basis functions, for otherwise it may not converge! For example, if we use linear elements (i.e. $inf\_ele = 1$), then we can choose $nint(i) = 1$ or $nint(i) = 2$ *etc.*

- Determine the BC $kbc(1)$ and $kbc(2)$, and $vbc(i, j)$, $i, j = 1, 2$. Note that the linear system of equations is singular if both BC are Neumann, for the solution either does not exist or is not unique.

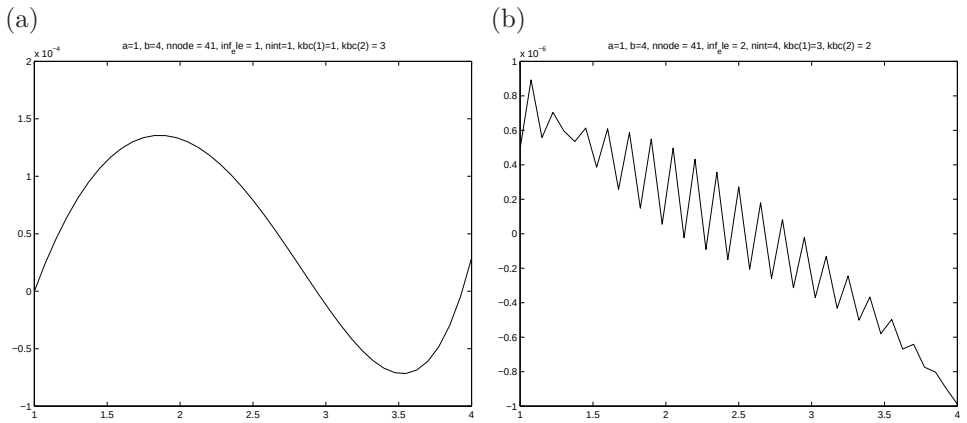


Figure 8.6. *Error plots of the FE solutions at nodal points. (a) The result is obtained with piecewise linear basis function and Gaussian quadrature of order one in the interval* $[1, 4]$. *Dirichlet BC at* $x = a$ *and mixed BC* $3u(b) + 4u'(b) = \gamma$ *from the exact solution* $u(x) = \sin x$ *are used. The magnitude of the error is* $O(10^{-4})$. *(b) Mixed BC* $3u(a) + 4u'(a) = \gamma$ *at* $x = a$ *and the Neumann BC at* $x = b$ *from the exact solution are used. The result is obtained with piecewise quadratic basis functions and Gaussian quadrature of order four. The magnitude of the error is* $O(10^{-6})$.

To run the program, simply type the following into the Matlab:

```
[x,u]= fem1d;
```

To find out the detailed usage of the finite element code, read README carefully. Fig. 8.6 gives the error plots for two different boundary conditions.

8.5 The FE method for fourth order BVPs in 1D

Let us now discuss how to solve fourth order differential equations using the finite element method. An important fourth order differential equation is the bi-harmonic equation, such

as in the model problem

$$u'''' + q(x)u = f(x)\,,\ 0 < x < 1\,,\ \text{subject to the BC}$$

$$I:\ u(0) = u'(0) = 0\,,\ u(1) = u'(1) = 0\,;\ \text{or}$$
$$II:\ u(0) = u'(0) = 0,\ \ u(1) = 0\,,\ u''(1) = 0\,;\ \text{or}$$
$$III:\ u(0) = u'(0) = 0,\ \ u''(1) = 0\,,\ u'''(1) = 0\,.$$

Note that there is no negative sign in the highest derivative term. To derive the weak form, we again multiply by a test function $v(x) \in V$ and integrate by parts to get

$$\int_0^1 (u'''' + q(x)u)v\,dx = \int_0^1 fv\,dx\,,$$
$$u'''v\big|_0^1 - \int_0^1 u'''v'\,dx + \int_0^1 quv\,dx = \int_0^1 fv\,dx\,,$$
$$u'''v\big|_0^1 - u''v'\big|_0^1 + \int_0^1 \left(u''v'' + quv\right)\,dx = \int_0^1 fv\,dx\,,$$
$$u'''(1)v(1) - u'''(0)v(0) - u''(1)v'(1) + u''(0)v'(0) + \int_0^1 \left(u''v'' + quv\right)\,dx = \int_0^1 fv\,dx\,.$$

For $u(0) = u'(0) = 0$, $u(1) = u'(1) = 0$, they are *essential boundary conditions*, thus we set

$$v(0) = v'(0) = v(1) = v'(1) = 0. \tag{8.27}$$

The weak form is

$$a(u, v) = f(v)\,, \tag{8.28}$$

where the bilinear form and the linear form are

$$a(u, v) = \int_0^1 \left(u''v'' + quv\right)\,dx\,, \tag{8.29}$$

$$L(v) = \int_0^1 fv\,dx\,. \tag{8.30}$$

Since the weak form involves second order derivatives, the solution space is

$$H_0^2(0,1) = \left\{v(x)\ ,\ v(0) = v'(0) = v(1) = v'(1) = 0\ ,\quad v\ , v' \text{ and } v'' \in L^2\right\}, \tag{8.31}$$

and from the Sobolev embedding theorem we know that $H^2 \subset C^1$.

For the boundary conditions $u(1) = u''(1) = 0$, we still have $v(1) = 0$, but there is no restriction on $v'(1)$ and the solution space is

$$H_E^2 = \left\{v(x)\,,\quad v(0) = v'(0) = v(1) = 0\,,\quad v \in H^2(0,1)\right\}. \tag{8.32}$$

For the boundary conditions $u''(1) = u'''(1) = 0$, there are no restrictions on both $v(1)$ and $v'(1)$ and the solution space is

$$H_E^2 = \left\{v(x)\,,\quad v(0) = v'(0) = 0\,,\quad v \in H^2(0,1)\right\}. \tag{8.33}$$

For non-homogeneous *natural or mixed* boundary conditions, the weak form and the linear form may be different. For homogeneous *essential BC*, the weak form and the linear form will be the same. We often need to do something to adjust the essential boundary conditions.

8.5.1 The finite element discretization

Given a mesh

$$0 = x_0 < x_1 < x_2 < \cdots < x_M = 1\,,$$

we want to construct a finite dimensional space V_h. For conforming finite element methods we have $V_h \in H^2(0,1)$, therefore we cannot use the piecewise linear functions since they are in the Sobolev space $H^1(0,1)$ but not in $H^2(0,1)$.

For piecewise quadratic functions, theoretically we can find a finite dimensional space that is a subset of $H^2(0,1)$; but this is not practical as the basis functions would have large support and involve at least six nodes. The most practical conforming finite dimensional space in 1D is the piecewise cubic functions over the mesh

$$V_h = \left\{ v(x),\ \ v(x) \text{ is a continuous piecewise cubic function},\ \ v \in H_0^2(0,1) \right\}. \quad (8.34)$$

The degree of freedom. On each element, we need four parameters to determine a cubic function. For essential boundary conditions at both $x = a$ and $x = b$, there are $4M$ parameters for M elements; and at each interior nodal point, the cubic and its derivative are continuous and there are four boundary conditions, so the dimension of the finite element space is

$$4M - 2(M-1) - 4 = 2(M-1)\,.$$

Construct the basis functions in H^2 in 1D

Since the derivative has to be continuous, we can use piecewise Hermite interpolation and construct the basis function in two categories. The first category is

$$\begin{aligned} \phi_i(x_j) &= \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise}\,, \end{cases} \\ \text{and } \phi_i'(x_j) &= 0\,, \quad \text{for any } x_j, \end{aligned} \quad (8.35)$$

i.e., the basis functions in this group have unity at one node and are zero at other nodes, and the derivatives are zero at all nodes. To construct the local basis function in the element (x_i, x_{i+1}), we can set

$$\phi_i(x) = \frac{(x - x_{i+1})^2\,(a(x - x_i) + 1)}{(x_i - x_{i+1})^2}\,.$$

It is obvious that $\phi_i(x_i) = 1$ and $\phi_i(x_{i+1}) = \phi_i'(x_{i+1}) = 0$ *i.e.*, x_{i+1} is a double root of the polynomial. We use $\phi'(x_i) = 0$ to find the coefficient a, to finally obtain

$$\phi_i(x) = \frac{(x - x_{i+1})^2 \left(\dfrac{2(x - x_i)}{(x_{i+1} - x_i)} + 1 \right)}{(x_i - x_{i+1})^2}\,. \quad (8.36)$$

The global basis function can thus be written as

$$\phi_i(x) = \begin{cases} 0 & \text{if } x \le x_{i-1}\,, \\ \dfrac{(x-x_{i-1})^2\left(\dfrac{2(x-x_i)}{(x_{i-1}-x_i)}+1\right)}{(x_i-x_{i-1})^2} & \text{if } x_{i-1} \le x \le x_i\,, \\ \dfrac{(x-x_{i+1})^2\left(\dfrac{2(x-x_i)}{(x_{i+1}-x_i)}+1\right)}{(x_i-x_{i+1})^2}, & \text{if } x_i \le x \le x_{i+1}\,, \\ 0 & \text{if } x_{i+1} \le x\,. \end{cases} \tag{8.37}$$

There are $M-1$ such basis functions. The second group of basis functions satisfy

$$\bar{\phi}_i'(x_j) = \begin{cases} 1 & \text{if } i=j, \\ 0 & \text{otherwise}\,, \end{cases} \tag{8.38}$$
$$\text{and } \bar{\phi}_i(x_j) = 0\,, \quad \text{for any } x_j,$$

i.e., the basis functions in this group are zero at all nodes, and the derivatives are unity at one node and zero at other nodes. To construct the local basis functions in an element (x_i, x_{i+1}), we can set

$$\bar{\phi}_i(x) = C(x-x_i)(x-x_{i+1})^2,$$

since x_i and x_{i+1} are zeros of the cubic and x_{i+1} is a double root of the cubic. The constant C is chosen such that $\psi_i'(x_i) = 1$, so we finally obtain

$$\bar{\phi}_i(x) = \frac{(x-x_i)(x-x_{i+1})^2}{(x_i-x_{i+1})^2}\,. \tag{8.39}$$

The global basis function for this category is thus

$$\bar{\phi}_i(x) = \begin{cases} 0 & \text{if } x \le x_{i-1}\,, \\ \dfrac{(x-x_i)(x-x_{i-1})^2}{(x_i-x_{i-1})^2} & \text{if } x_{i-1} \le x \le x_i\,, \\ \dfrac{(x-x_i)(x-x_{i+1})^2}{(x_{i+1}-x_i)^2} & \text{if } x_i \le x \le x_{i+1}\,, \\ 0 & \text{if } x_{i+1} \le x\,. \end{cases} \tag{8.40}$$

8.5.2 The shape functions

There are four shape functions in the interval $(-1, 1)$, namely,

$$\psi_1(\xi) = \frac{(\xi-1)^2(\xi+2)}{4}\,,$$
$$\psi_2(\xi) = \frac{(\xi+1)^2(-\xi+2)}{4}\,,$$
$$\psi_3(\xi) = \frac{(\xi-1)^2(\xi+1)}{4}\,,$$
$$\psi_4(\xi) = \frac{(\xi+1)^2(\xi-1)}{4}\,.$$

It is noted that there are two basis functions *centered* at each node, so called a *double node*. The finite element solution can be written as

$$u_h(x) = \sum_{j=1}^{M-1} \alpha_j \phi_j(x) + \sum_{j=1}^{M-1} \beta_j \bar{\phi}_j(x)\,, \tag{8.41}$$

and after the coefficients α_j and β_j are found we have

$$u_h(x_j) = \alpha_j\,, \qquad u_h'(x_j) = \beta_j\,.$$

There are four nonzero basis functions on each element (x_i, x_{i+1}); and on adopting the order $\phi_1, \phi_2, \psi_1, \psi_2, \phi_3, \phi_3, \psi_3, \cdots$, the local stiffness matrix has the form

$$\left[\begin{array}{cccc} a(\phi_i,\phi_i) & a(\phi_i,\phi_{i+1}) & a(\phi_i,\bar{\phi}_i) & a(\phi_i,\bar{\phi}_{i+1}) \\ a(\phi_{i+1},\phi_i) & a(\phi_{i+1},\phi_{i+1}) & a(\phi_{i+1},\bar{\phi}_i) & a(\phi_{i+1},\bar{\phi}_{i+1}) \\ a(\bar{\phi}_i,\bar{\phi}_i) & a(\bar{\phi}_i,\phi_{i+1}) & a(\bar{\phi}_i\,,\bar{\phi}_i) & a(\bar{\phi}_i,\bar{\phi}_{i+1}) \\ a(\bar{\phi}_{i+1},\bar{\phi}_i) & a(\bar{\phi}_{i+1},\bar{\phi}_{i+1}) & a(\bar{\phi}_{i+1},\bar{\phi}_i) & a(\bar{\phi}_{i+1},\bar{\phi}_{i+1}) \end{array}\right]_{(x_i,x_{i+1})}.$$

This global stiffness matrix is still banded, and has band width six.

8.6 The Lax-Milgram Lemma and the existence of FE solutions

One of the most important issues is whether the weak form has a solution, and if so under what assumptions. Further, if the solution does exist, is it unique, and how close is it to the solution of the original differential equations? Answers to these questions are based on the Lax-Milgram Lemma.

8.6.1 General settings: assumptions, and conditions

Let V be a Hilbert space with inner product $(u\,,\,v)_V$ and norm $\|u\|_V = \sqrt{(u,u)_V}$, *e.g.*, C^m, the Sobolev spaces H^1 and H^2, *etc.* Assume there is a *bilinear* form

$$a(u,v), \quad V \times V \longmapsto R\,,$$

and a linear form

$$L(v), \quad V \longmapsto R\,,$$

that satisfy the following conditions:

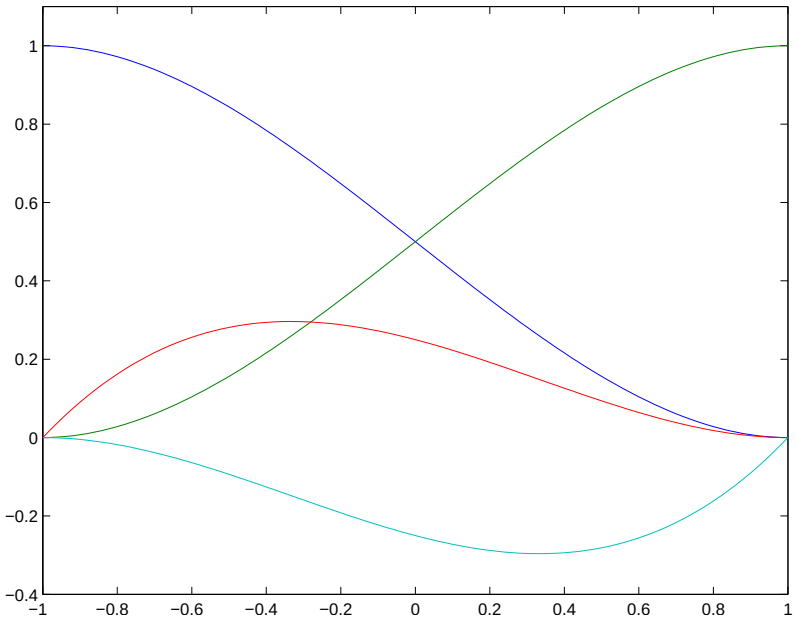
1. $a(u,v)$ is symmetric, *i.e.*, $a(u,v) = a(v,u)$;
2. $a(u,v)$ is continuous in both u and v, *i.e.*, there is a constant γ such that

$$|a(u,v)| \le \gamma \|u\|_V \|v\|_V\,,$$

for any u and $v \in V$; the norm of the operator $a(u,v)$[7];

[7] If this condition is true, then $a(u,v)$ is called a bounded operator and the least lower bound of such a $\gamma > 0$ is called the norm of $a(u,v)$.

(a)



(b)

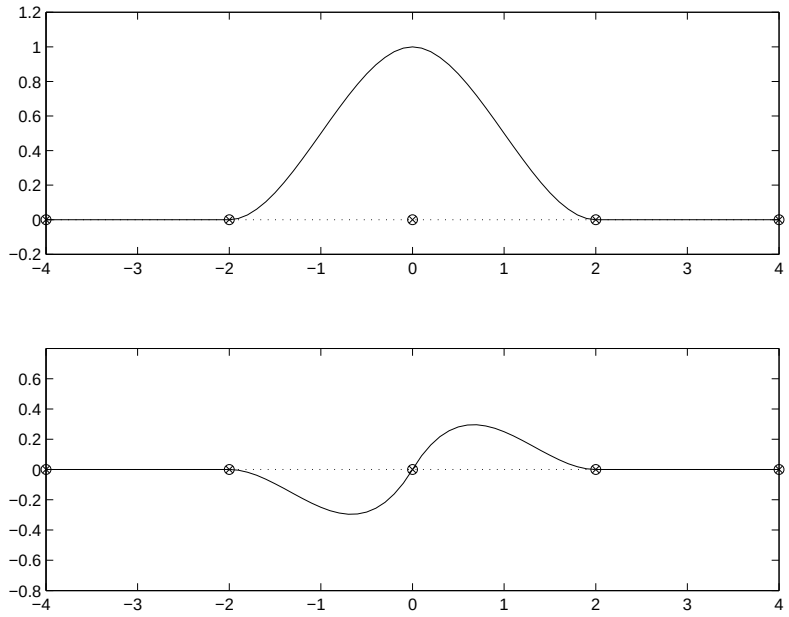


Figure 8.7. *(a) Hermite cubic shape functions on a master element; and (b) corresponding global functions at a node i in a mesh.*

3. $a(u,v)$ is V-elliptic, *i.e.*, there is a constant α such that

$$a(v,v) \geq \alpha \|v\|_V^2$$

for any $v \in V$ (alternative terms are *coercive*, or inf-sup condition); and

4. L is continuous, *i.e.*, there is a constant Λ such that

$$|L(v)| \leq \Lambda \|v\|_V \,,$$

for any $v \in V$.

8.6.2 The Lax-Milgram lemma

Theorem 8.1. *Under the above conditions 2 to 4, there* exists *a* unique *element $u \in V$ such that*

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

Furthermore, if the condition 1 is also true, i.e., $a(u,v)$ is symmetric, then

1. $\|u\|_V \leq \dfrac{\Lambda}{\alpha}$*; and*
2. *u is the unique global minimizer of*

$$F(v) = \frac{1}{2} a(v,v) - L(v)\,.$$

Sketch of the proof. The proof exploits the Riesz representation theorem from functional analysis. Since $L(v)$ is a bounded linear operator in the Hilbert space V with the inner product $a(u,v)$, there is unique element u^* in V such that

$$L(v) = a(u^*, v), \quad \forall\, v \in V\,.$$

The a-norm is equivalent to V norm. From the continuity condition of $a(u,v)$, we get

$$\|u\|_a = \sqrt{a(u,u)} \leq \sqrt{\gamma \|u\|_V^2} = \sqrt{\gamma}\, \|u\|_V\,.$$

From the V-elliptic condition, we have

$$\|u\|_a = \sqrt{a(u,u)} \geq \sqrt{\alpha \|u\|_V^2} = \sqrt{\alpha}\, \|u\|_V\,,$$

therefore

$$\sqrt{\alpha}\, \|u\|_V \leq \|u\|_a \leq \sqrt{\gamma}\, \|u\|_V,$$

or

$$\frac{1}{\sqrt{\gamma}}\, \|u\|_a \leq \|u\|_V \leq \frac{1}{\sqrt{\alpha}} \|u\|_a\,.$$

Often $\|u\|_a$ is called the energy norm.

$F(u^*)$ is the global minimizer. For any $v \in V$, if $a(u,v) = a(v,u)$, then

$$\begin{aligned} F(v) = F(u^* + v - u^*) = F(u^* + w) &= \frac{1}{2}a(u^* + w, u^* + w) - L(u^* + w) \\ &= \frac{1}{2}\left(a(u^* + w, u^*) + a(u^* + w, w)\right) - L(u^*) - L(w) \\ &= \frac{1}{2}\left(a(u^*, u^*) + a(w, u^*) + a(u^*, w) + a(w, w)\right) - L(u^*) - L(w) \\ &= \frac{1}{2}a(u^*, u^*) - L(u^*) + \frac{1}{2}a(w, w) + a(u^*, w) - L(w) \\ &= F(u^*) + \frac{1}{2}a(w, w) - 0 \\ &\geq F(u^*)\,. \end{aligned}$$

Proof of the stability. We have

$$\alpha\|u^*\|_V^2 \leq a(u^*, u^*) = L(u^*) \leq \Lambda\|u^*\|_V\,,$$

therefore

$$\alpha\|u^*\|_V^2 \leq \Lambda\|u^*\|_V \implies \|u^*\|_V \leq \frac{\Lambda}{\alpha}\,.$$

Remark: The Lax-Milgram Lemma is often used to prove the existence and uniqueness of the solutions of ODEs/PDEs.

8.6.3 An example using the Lax-Milgram lemma

Let us consider the 1D Sturm-Liouville problem once again:

$$\begin{aligned} &-(pu')' + qu = f\,, \quad a < x < b\,, \\ &u(a) = 0, \quad \tilde{\alpha}u(b) + \tilde{\beta}u'(b) = \tilde{\gamma}\,, \quad \tilde{\beta} \neq 0, \quad \frac{\tilde{\alpha}}{\tilde{\beta}} \geq 0\,. \end{aligned}$$

The bilinear form is

$$a(u,v) = \int_a^b \left(pu'v' + quv\right)\,dx + \frac{\tilde{\alpha}}{\tilde{\beta}}p(b)u(b)v(b)\,,$$

and the linear form is

$$L(v) = (f,v) + \frac{\tilde{\gamma}}{\tilde{\beta}}p(b)v(b)\,.$$

The space is $V = H_E^1(a,b)$. To consider the conditions of the Lax-Milgram theorem, we need the *Poincaré inequality*:

Theorem 8.2. *If $v(x) \in H^1$ and $v(a) = 0$, then*

$$\int_a^b v^2\,dx \leq (b-a)^2\int_a^b |v'(x)|^2\,dx \quad \text{or} \quad \int_a^b |v'(x)|^2\,dx \geq \frac{1}{(b-a)^2}\int_a^b v^2\,dx\,. \tag{8.42}$$

Proof: We have

$$v(x) = \int_a^x v'(t)\,dt$$

$$\implies |v(x)| \le \int_a^x |v'(t)|\,dt \le \left\{\int_a^x |v'(t)|^2 dt\right\}^{1/2} \left\{\int_a^x dt\right\}^{1/2} \le \sqrt{b-a}\left\{\int_a^b |v'(t)|^2\right\}^{1/2},$$

so that

$$v^2(x) \le (b-a)\int_a^b |v'(t)|^2\,dt$$

$$\implies \int_a^b v^2(x)\,dx \le (b-a)\int_a^b |v'(t)|^2 dt \int_a^b dx \le (b-a)^2 \int_a^b |v'(x)|\,dx\,.$$

This completes the proof.

We now verify the Lax-Milgram Lemma conditions for the Sturm-Liouville problem.

- Obviously $a(u,v) = a(v,u)$.
- The bilinear form is continuous:

$$\begin{aligned}
|a(u,v)| &= \left|\int_a^b \left(pu'v' + quv\right)\,dx + \frac{\tilde{\alpha}}{\tilde{\beta}}p(b)u(b)v(b)\right| \\
&\le \max\{\,p_{max},\,q_{max}\,\}\left(\int_a^b \left(|u'v'| + |uv|\right)\,dx + \frac{\tilde{\alpha}}{\tilde{\beta}}|u(b)v(b)|\right) \\
&\le \max\{\,p_{max},\,q_{max}\,\}\left(\int_a^b |u'v'|dx + \int_a^b |uv|\,dx + \frac{\tilde{\alpha}}{\tilde{\beta}}|u(b)v(b)|\right) \\
&\le \max\{\,p_{max},\,q_{max}\,\}\left(2\|u\|_1\|v\|_1 + \frac{\tilde{\alpha}}{\tilde{\beta}}|u(b)v(b)|\right).
\end{aligned}$$

From the inequality

$$\begin{aligned}
|u(b)v(b)| &= \left|\int_a^b u'(x)\,dx \int_a^b v'(x)\,dx\right| \\
&\le (b-a)\sqrt{\int_a^b |u'(x)|^2\,dx}\sqrt{\int_a^b |v'(x)|^2\,dx} \\
&\le (b-a)\sqrt{\int_a^b \left(|u'(x)|^2 + |u(x)|^2\right)\,dx}\sqrt{\int_a^b \left(|v'(x)|^2 + |v(x)|^2\right)\,dx} \\
&\le (b-a)\|u\|_1\|v\|_1\,,
\end{aligned}$$

we get

$$|a(u,v)| \le \max\{\,p_{max},\,q_{max}\}\left(2 + \frac{\tilde{\alpha}}{\tilde{\beta}}(b-a)\right)\|u\|_1\|v\|_1$$

i.e., the constant γ can be determined as

$$\gamma = \max\{\,p_{max},\,q_{max}\}\left(2 + \frac{\tilde{\alpha}}{\tilde{\beta}}(b-a)\right).$$

- $a(v,v)$ is V-elliptic. We have

$$\begin{aligned} a(v,v) &= \int_a^b \left(p(v')^2 + qv^2\right)\, dx + \frac{\tilde{\alpha}}{\tilde{\beta}} p(b)v(b)^2 \\ &\geq \int_a^b p(v')^2\, dx \\ &\geq p_{min} \int_a^b (v')^2\, dx \\ &= p_{min} \left(\frac{1}{2}\int_a^b (v')^2\, dx + \frac{1}{2}\int_a^b (v')^2\, dx \right) \\ &\geq p_{min} \left(\frac{1}{2}\frac{1}{(b-a)^2}\int_a^b v^2\, dx + \frac{1}{2}\int_a^b (v')^2\, dx \right) \\ &= p_{min}\ \min\left\{ \frac{1}{2(b-a)^2}, \frac{1}{2} \right\} \|v\|_1^2 , \end{aligned}$$

  i.e., the constant α can be determined as

$$\alpha = p_{min}\ \min\left\{ \frac{1}{2(b-a)^2}, \frac{1}{2} \right\}.$$

- $L(v)$ is continuous because

$$\begin{aligned} L(v) &= \int_a^b f(x)v(x)\, dx + \frac{\tilde{\gamma}_1}{\tilde{\beta}} p(b)v(b) \\ |L(v)| &\leq (|f|,|v|)_0 + \left|\frac{\tilde{\gamma}_1}{\tilde{\beta}}\right| p(b)\sqrt{b-a}\, \|v\|_1 \\ &\leq \|f\|_0 \|v\|_1 + \left|\frac{\tilde{\gamma}_1}{\tilde{\beta}}\right| p(b)\sqrt{b-a}\, \|v\|_1 \\ &\leq \left(\|f\|_0 + \left|\frac{\tilde{\gamma}_1}{\tilde{\beta}}\right| p(b)\sqrt{b-a} \right) \|v\|_1 , \end{aligned}$$

  i.e., the constant Λ can be determined as

$$\Lambda = \|f\|_0 + \left|\frac{\tilde{\gamma}_1}{\tilde{\beta}}\right|\ p(b)\,\sqrt{b-a}\,.$$

Thus we have verified the conditions of the Lax-Milgram lemma under certain assumptions such as $p(x) \geq p_{min} > 0$, $q(x) \geq 0$, *etc.*, and hence conclude that there is the unique solution in $H_e^1(a,b)$ to the original differential equation. The solution also satisfies

$$\|u\|_1 \leq \frac{\|f\|_0 + \left|\tilde{\gamma}/\tilde{\beta}\right|\ p(b)}{p_{min}\ \min\left\{ \frac{1}{2(b-a)}, 1 \right\}}\,.$$

8.6.4 Abstract FE methods

In the same setting, let us assume that V_h is a finite dimensional subspace of V and that $\{\phi_1, \phi_2, \cdots \phi_M\}$ is a basis for V_h. We can formulate the following abstract finite element method using the finite dimensional subspace V_h. We seek $u_h \in V_h$ such that

$$a(u_h, v) = L(v)\,, \quad \forall v \in V_h\,, \tag{8.43}$$

or equivalently

$$F(u_h) \leq F(v), \quad \forall v \in V_h\,. \tag{8.44}$$

We apply the weak form in the finite dimensional V_h:

$$a(u_h, \phi_i) = L(\phi_i)\,, \qquad i = 1, \cdots, M\,. \tag{8.45}$$

Let the finite element solution u_h be

$$u_h = \sum_{j=1}^{M} \alpha_j \phi_j\,.$$

Then from the weak form in V_h we get

$$a\left(\sum_{j=1}^{M} \alpha_j \phi_j, \phi_i\right) = \sum_{j=1}^{M} \alpha_j a(\phi_j, \phi_i) = L(\phi_i)\,, \qquad i = 1, \cdots, M\,,$$

which in the matrix-vector form is

$$AU = F,$$

where $U \in R^M$, $F \in R^M$ with $F(i) = L(\phi_i)$ and A is an $M \times M$ matrix with entries $A(i,j) = a(\phi_j, \phi_i)$. Since any element in V_h can be written as

$$v = \sum_{i=1}^{M} \eta_i \phi_i\,,$$

we have

$$a(v,v) = a\left(\sum_{i=1}^{M} \eta_i \phi_i, \sum_{j=1}^{M} \eta_j \phi_j\right) = \sum_{i,j=1}^{M} \eta_i a(\phi_i, \phi_j) \eta_j = \eta^T A \eta > 0$$

provided $\eta^T = \{\eta_1, \cdots \eta_M\} \neq 0$. Consequently, A is symmetric positive definite. The minimization form using V_h is

$$\frac{1}{2} U^T A U - F^T \mathbf{U} = \min_{\eta \in R^M} \left(\frac{1}{2} \eta^T A \eta - F^T \eta\right). \tag{8.46}$$

The existence and uniqueness of the abstract FE method.
Since the matrix A is symmetric positive definite and it is invertible, so there is a unique

solution to the discrete weak form. Also from the conditions of Lax-Milgram lemma, we have

$$\alpha\|u_h\|_V^2 \le a(u_h,u_h) = L(u_h) \le \Lambda\|u_h\|_V\,,$$

whence

$$\|u_h\|_V \le \frac{\Lambda}{\alpha}\,.$$

Error estimates. If $e_h = u - u_h$ is the error, then:

- $a(e_h, v_h) = (e_h, v_h)_a = 0\,,\ \forall v_h \in V_h$;
- $\|u-u_h\|_a = \sqrt{a(e_h,e_h)} \le \|u - v_h\|_a\,,\ \forall v_h \in V_h$, *i.e.*, u_h is the best approximation to u in the energy norm; and
- $\|u-u_h\|_V \le \frac{\gamma}{\alpha}\|u-v_h\|_V\,,\ \forall v_h \in V_h$, which gives the error estimates in the V norm.

Sketch of the proof: From the weak form, we have

$$a(u,v_h) = L(v_h)\,,\quad a(u_h,v_h) = L(v_h) \quad\Longrightarrow\quad a(u-u_h,v_h) = 0\,.$$

This means the finite element solution is the projection of u onto the space V_h. It is the best solution in V_h in the energy norm, because

$$\begin{aligned}\|u-v_h\|_a^2 &= a(u-v_h,u-v_h) = a(u-u_h+w_h,u-u_h+w_h)\\ &= a(u-u_h,u-u_h) + a(u-u_h,w_h) + a(w_h,u-u_h) + a(w_h,w_h)\\ &= a(u-u_h,u-u_h) + a(w_h,w_h)\\ &\ge \|u-u_h\|_a^2\,,\end{aligned}$$

where $w_h = u_h - v_h \in V_h$. Finally, from the condition (3), we have

$$\begin{aligned}\alpha\|u-u_h\|_V^2 &\le a(u-u_h,u-u_h) = a(u-u_h,u-u_h) + a(u-u_h,w_h)\\ &= a(u-u_h,u-u_h+w_h) = a(u-u_h,u-v_h)\\ &\le \gamma\|u-u_h\|_V\|u-v_h\|_V\,.\end{aligned}$$

The last inequality is obtained from condition (2).

8.7 *1D IFEM for discontinuous coefficients

Now we revisit the 1D interface problems discussed in Section 2.10

$$-(pu')' \;=\; f(x), \qquad 0<x<1, \qquad u(0)=0,\quad u(1)=0, \tag{8.47}$$

and consider the case in which the coefficient has a finite jump,

$$p(x) = \begin{cases} \beta^-(x) & \text{if } 0<x<\alpha,\\ \beta^+(x) & \text{if } \alpha<x<1.\end{cases} \tag{8.48}$$

The theoretical analysis about the solution still holds if the natural jump conditions

$$[u]_\alpha = 0, \qquad \left[\beta\, u'\right]_\alpha = 0, \tag{8.49}$$

are satisfied, where $[u]_\alpha$ means the jump defined at α.

Given a uniform mesh x_i, $i = 0, 1, \cdots n$, $x_{i+1} - x_i = h$. Unless the interface α in (8.47) itself is a node, the solution obtained from the standard finite element method using the linear basis functions is only first order accurate in the maximum norm. In [20], modified basis functions that are defined below

$$\phi_i(x_k) = \begin{cases} 1, & \text{if } k = i, \\ 0, & \text{otherwise}, \end{cases} \tag{8.50}$$

$$[\phi_i]_\alpha = 0, \qquad [\beta\, \phi_i']_\alpha = 0, \tag{8.51}$$

are proposed. Obviously, if $x_j \le \alpha < x_{j+1}$, then only ϕ_j and ϕ_{j+1} need to be changed to satisfy the second jump condition. Using the method of undetermined coefficients, that is, we look for the basis function $\phi_j(x)$ in the interval (x_j, x_{j+1}) as

$$\phi_j(x) = \begin{cases} a_0 + a_1 x & \text{if } x_j \le x < \alpha, \\ b_0 + b_1 x & \text{if } \alpha \le x \le x_{j+1}, \end{cases} \tag{8.52}$$

which should satisfy $\phi_j(x_j) = 1$, $\phi_j(x_{j+1}) = 0$, $\phi_j(\alpha-) = \phi_j(\alpha+)$, and $\beta^- \phi_j'(\alpha-) = \beta^+ \phi_j'(\alpha+)$. There are four unknowns and four conditions. It has been proved in [20] that the coefficients are unique determined and have the following closed form if β is a piecewise constant and $\beta^- \beta^+ > 0$,

$$\phi_j(x) = \begin{cases} 0, & 0 \le x < x_{j-1}, \\ \dfrac{x - x_{j-1}}{h}, & x_{j-1} \le x < x_j, \\ \dfrac{x_j - x}{D} + 1, & x_j \le x < \alpha, \\ \dfrac{\rho\,(x_{j+1} - x)}{D}, & \alpha \le x < x_{j+1}, \\ 0, & x_{j+1} \le x \le 1, \end{cases} \qquad \phi_{j+1}(x) = \begin{cases} 0, & 0 \le x < x_j, \\ \dfrac{x - x_j}{D}, & x_j \le x < \alpha, \\ \dfrac{\rho\,(x - x_{j+1})}{D} + 1, & \alpha \le x < x_{j+1}, \\ \dfrac{x_{j+2} - x}{h}, & x_{j+1} \le x \le x_{j+2}, \\ 0, & x_{j+2} \le x \le 1. \end{cases}$$

where

$$\rho = \frac{\beta^-}{\beta^+}, \quad D = h - \frac{\beta^+ - \beta^-}{\beta^+}\,(x_{j+1} - \alpha).$$

Fig. 8.8 shows several plots of the modified basis functions $\phi_j(x)$, $\phi_{j+1}(x)$, and some neighboring basis functions, which are the standard hat functions. At the interface α, we can see clearly kinks in the basis functions which reflect the natural jump condition.

Using the modified basis functions, it has been shown in [20] that the finite element solution obtained from the Galerkin finite method with the new basis functions is second order accurate in the maximum norm.

For 1D interface problems, the finite difference and finite element methods are not much different. The finite element method likely performs better for self-adjoint problems, while the finite difference method is more flexible for general elliptic interface problems.

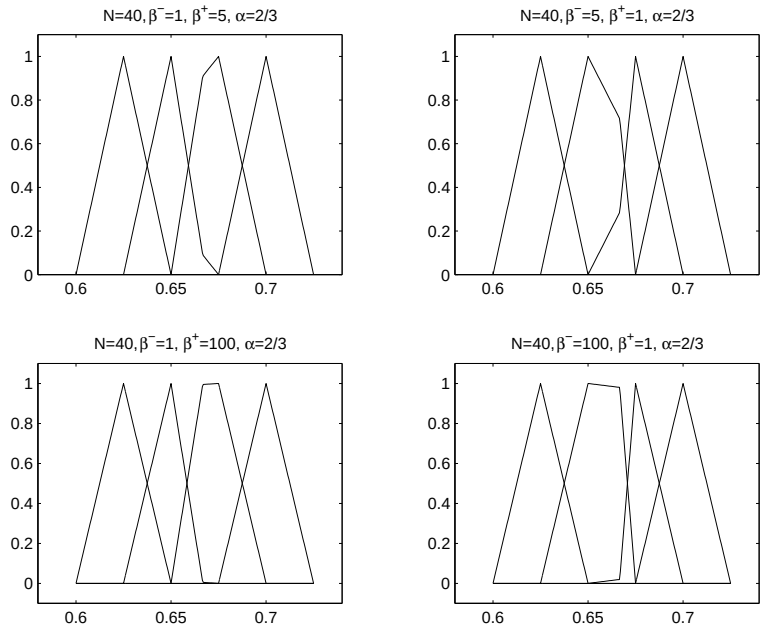


Figure 8.8. *Plot of some basis function near the interface with different* β^- *and* β^+*. The interface is at* $\alpha = \frac{2}{3}$*.*

8.8 Exercises

1. (Purpose: Review abstract FE methods.) Consider the Sturm-Liouville problem

$$-u'' + u = f\,, \quad 0 < x < \pi\,,$$
$$u(0) = 0\,, \quad u(\pi) + u'(\pi) = 1\,.$$

   Let V_f be the finite dimensional space

$$V_f = span\,\{\, x,\ \sin(x), \sin(2x)\,\}\,.$$

   Find the best approximation to the solution of the weak form from V_f in the energy norm ($\|:\|_a = \sqrt{a(:,:)}$). You can use either analytic derivation or computer software packages (e.g. Maple, Matlab, SAS, etc.). Take $f = 1$ for the computation. Compare this approach with the finite element method using three hat basis functions. Find the true solution, and plot the solution and the error of the finite element solution.

2. Consider the Sturm-Liouville problem

$$-((1+x^2)u')' + xu = f, \quad 0 < x < 1,$$
$$u(1) = 2.$$

Transform the problem to a problem with homogeneous Dirihlet boundary condition at $x = 1$. Write down the weak form for each of the following case:

(a) $u(0) = 3$. **Hint:** Construct a function $u_0(x) \in H^1$ such that $u_0(0) = 3$ and $u_0(1) = 2$.

(b) $u'(0) = 3$. **Hint:** Construct a function $u_0(x) \in H^1$ such that $u_0(1) = 2$ and $u_0'(0) = 0$.

(c) $u(0) + u'(0) = 3$. **Hint:** Construct a function $u_0(x) \in H^1$ such that $u_0(1) = 2$ and $u_0(0) + u_0'(0) = 0$.

3. Consider the Sturm-Liouville problem

$$-(pu')' + qu = f\,, \quad a < x < b\,,$$
$$u(a) = 0\,, \qquad u(b) = 0\,.$$

Consider a mesh $a = x_0 < x_1 \cdots < x_M = b$ and the finite element space

$V_h = \left\{ v(x) \in H_0^1(a,b) \right\}$, the set of piecewise cubic functions over the mesh.

(a) Find the dimension of V_h.

(b) Find all nonzero shape functions $\psi_i(\xi)$ where $-1 \le \xi \le 1$, and plot them.

(c) What is the size of the local stiffness matrix and load vector? Sketch the assembling process.

(d) List some advantages and disadvantages of this finite element space, compared with the piecewise continuous linear finite dimensional space (the hat functions).

4. Down-load the files of the 1D finite element Matlab package. Consider the following analytic solution and parameters,

$$u(x) = e^x \sin x\,, \quad p(x) = 1 + x^2\,, \quad q(x) = e^{-x}\,, \quad c(x) = 1\,,$$

and $f(x)$ determined from the differential equation

$$-(pu')' + c(x)u' + qu = f\,, \quad a < x < b\,.$$

Use this example to become familiar with the 1D finite element Matlab package, by trying the following boundary conditions:

(a) Dirichlet BC at $x = a$ and $x = b$, where $a = -1$, $b = 2$;

(b) Neumann BC at $x = a$ and Dirichlet BC at $x = b$, where $a = -1$ and $b = 2$.

(c) Mixed BC $\gamma = 3u(a) - 5u'(a)$ at $x = a = -1$, and Neumann BC at $x = b = 2$.

Using linear, quadratic, and cubic basis functions, tabulate the errors in the infinity norm

$$e_M = \max_{0\le i\le M} |u(x_i) - U_i|$$

at the nodes and auxiliary points as follows:

M	Basis	Gaussian	error	e_M/e_{2M}

for different $M = 4, 8, 16, 32, 64$ (nnode= $M+1$), or the **closest integers** if necessary. What are the respective convergence orders?
(Note: the method is second, third or fourth order convergent if the ratio e_M/e_{2M} approaches 4, 8 or 16, respectively.)

For the **last case**:
(1) print out the stiffness matrix for the *linear basis function* with $M = 5$. Is it symmetric?
(2) Plot the computed solution against the exact one, and the error plot for the case of the linear basis function. Take enough points to plot the exact solution to see the whole picture.
(3) Plot the error versus $h = 1/M$ in log-log scale for the three different bases.
The slope of such a plot is the convergence order of the method employed. For this problem, you will only produce *five* plots for the last case.

Find the energy norm, H^1 norm and L^2 norm of the error and do the grid refinement analysis.

5. Use the Lax-Milgram Lemma to show whether the following two-point value problem has a unique solution:

$$\begin{aligned} -u'' + q(x)u &= f\,, \quad 0 < x < 1\,, \\ u'(0) = u'(1) &= 0\,, \end{aligned} \tag{8.53}$$

where $q(x) \in C(0,1)$, $q(x) \ge q_{min} > 0$. What happens if we relax the condition to $q(x) \ge 0$? Give counter-examples if necessary.

6. Consider the general fourth order two-point BVP

$$a_4 u'''' + a_3 u''' + a_2 u'' + a_1 u' + a_0 u = f\,, \quad a < x < b\,,$$

with the mixed BC

$$2u'''(a) - u''(a) + \gamma_1 u'(a) + \rho_1 u(a) = \delta_1\,, \tag{8.54}$$

$$u'''(a) + u''(a) + \gamma_2 u'(a) + \rho_2 u(a) = \delta_2\,, \tag{8.55}$$

$$u(b) = 0\,,$$

$$u'(b) = 0\,.$$

Derive the weak form for this problem.
Hint: Solve for $u'''(a)$ and $u''(a)$ from (8.54) and (8.55). The weak form should only involve up to second order derivatives.

7. (An eigenvalue problem.) Consider

$$-(pu')' + qu - \lambda u = 0\,, \quad 0 < x < \pi\,, \tag{8.56}$$

$$u(0) = 0, \quad u(\pi) = 0\,. \tag{8.57}$$

   (a) Find the weak form of the problem.

   (b) Check whether the conditions of the Lax-Milgram Lemma are satisfied. Which condition is violated? Is the solution unique for arbitrary λ?
   Note: It is obvious that $u = 0$ is a solution. For some λ, we can find nontrivial solutions $u(x) \neq 0$. Such a λ is an eigenvalue of the system, and the nonzero solution is an eigenfunction corresponding to that eigenvalue. The problem to find the eigenvalues and the eigenfunctions is called an eigenvalue problem.

   (c) Find all the eigenvalues and eigenfunctions when $p(x) = 1$ and $q(x) = 0$.
   Hint: $\lambda_1 = 1$ and $u(x) = \sin(x)$ is one pair of the solutions.

8. Use the 1D finite element package with linear basis functions and a uniform grid to solve the eigenvalue problem

$$-(pu')' + qu - \lambda u = 0\,, \quad 0 < x < \pi\,,$$

$$u(0) = 0\,,\ u'(\pi) + \alpha u(\pi) = 0\,,$$

$$\text{where} \quad p(x) \geq p_{min} > 0\,,\ q(x) \geq 0\,,\ \alpha \geq 0\,.$$

   in each of the following two cases:

   (a) $p(x) = 1$, $q(x) = 1$, $\alpha = 1$.

   (b) $p(x) = 1 + x^2$, $q(x) = x$, $\alpha = 3$.

   Try to solve the eigenvalue problem with $M = 5$ and $M = 20$. Print out the eigenvalues but not the eigenfunctions. Plot all the eigenfunctions in a single plot for $M = 5$, and plot two typical eigenfunctions for $M = 20$ (6 plots in total).
   Hint: The approximate eigenvalues λ_1, λ_2, $\cdots$, λ_M and the eigenfunction $u_{\lambda_i}(x)$ are the generalized eigenvalues of

$$Ax = \lambda Bx\,,$$

   where A is the stiffness matrix and $B = \{b_{ij}\}$ with $b_{ij} = \int_0^\pi \phi_i(x)\phi_j(x)dx$. You can generate the matrix B either numerically or analytically; and in Matlab you can use $[V, D] = EIG(A, B)$ to find the generalized eigenvalues and the corresponding eigenvectors. For a computed eigenvalue λ_i, the corresponding eigenfunction is

$$u_{\lambda_i}(x) = \sum_{j=1}^{M} \alpha_{i,j}\phi_j(x)\,,$$

   where $[\alpha_{i,1}, \alpha_{i,2}, \cdots, \alpha_{i,M}]^T$ is the eigenvector corresponding to the generalized eigenvalue.
   Note: if we can find the eigenvalues and corresponding eigenfunctions, the solution to the differential equation can be expanded in terms of the eigenfunctions, similar to Fourier series.

9. (An application.) Consider a nuclear fuel element of spherical form, consisting of a sphere of "fissionable" material surrounded by a spherical shell of aluminium "cladding" as shown in the figure. We wish to determine the temperature distribution in the nuclear fuel element and the aluminium cladding. The governing equations for the two regions are the same, except that there is no heat source term for the aluminium cladding. Thus

$$-\frac{1}{r^2}\frac{d}{dr}r^2k_1\frac{dT_1}{dr} = q\,, \quad 0 \le r \le R_F\,,$$
$$-\frac{1}{r^2}\frac{d}{dr}r^2k_2\frac{dT_2}{dr} = 0\,, \quad R_F \le r \le R_C\,,$$

where the subscripts 1 and 2 refer to the nuclear fuel element and the cladding, respectively. The heat generation in the nuclear fuel element is assumed to be of the form

$$q_1 = q_0\left[1 + c\left(\frac{r}{R_F}\right)^2\right],$$

where q_0 and c are constants depending on the nuclear material. The BC are

$$kr^2\frac{dT_1}{dr} = 0 \text{ at } r = 0 \quad \text{(natural BC)},$$
$$T_2 = T_0 \text{ at } r = R_C\,,$$

where T_0 is a constant. Note the temperature at $r = R_F$ is continuous.

- Derive a weak form for this problem. (**Hint:** first multiply both sides by r^2.)
- Use two linear elements $[0, R_F]$ and $[R_F, R_C]$ to determine the finite element solution.
- Compare the nodal temperatures $T(0)$ and $T(R_f)$ with the values from the exact solution

$$T_1 = T_0 + \frac{q_0R_F^2}{6k_1}\left\{\left[1 - \left(\frac{r}{R_F}\right)^2\right] + \frac{3}{10}c\left[1 - \left(\frac{r}{R_F}\right)^4\right]\right\} + \frac{q_0R_F^2}{3k_2}\left(1 + \frac{3}{5}c\right)\left(1 - \frac{R_F}{R_C}\right),$$

$$T_2 = T_0 + \frac{q_0R_F^2}{3k_2}\left(1 + \frac{3}{5}c\right)\left(\frac{R_F}{r} - \frac{R_F}{R_C}\right).$$

Take $T_0 = 80$, $q_0 = 5$, $k_1 = 1$, $k_2 = 50$, $R_F = 0.5$, $R_C = 1$, $c = 1$ for plotting and comparison.