A Finite Element Implementation in One Dimension

3

3.1 Introduction

Classical techniques construct approximations from globally kinematically admissible functions, which we define as functions that satisfy the displacement boundary condition beforehand. Two main obstacles arise: (1) it may be very difficult to find a kinematically admissible function over the entire domain and (2) if such functions are found they lead to large, strongly coupled and complicated systems of equations. These problems have been overcome by the fact that local approximations (posed over very small partitions of the entire domain) can deliver high-quality solutions and simultaneously lead to systems of equations which have an advantageous mathematical structure amenable to large-scale computation by high-speed computers. These piecewise or "elementwise" approximations have been recognized at least 60 years ago by Courant [1] as being quite advantageous. There have been a variety of such approximation methods to solve equations of mathematical physics. The most popular method of this class is the finite element method (FEM). The central feature of the method is to partition the domain in a systematic manner into an assembly of discrete subdomains or "elements," and then to approximate the solution of each of these pieces in a manner that couples them to form a global solution valid over the whole domain. The process is designed to keep the resulting algebraic systems as computationally manageable, and memory efficient, as possible.

T. I. Zohdi, *A Finite Element Primer for Beginners*, The Basics,
https://doi.org/10.1007/978-3-319-70428-9_3

3.2 Weak Formulation

Consider the following general weak form introduced earlier

$$\boxed{\begin{array}{l}\text{Find } u \in H^1(\Omega)\ u|_{\Gamma_u} = d \text{ such that } \forall \nu \in H^1(\Omega), \nu|_{\Gamma_u} = 0 \\[2ex] \displaystyle\int_\Omega \frac{d\nu}{dx} E \frac{du}{dx}\, dx = \int_\Omega f \nu \, dx + t^* \nu|_{\Gamma_t}.\end{array}} \tag{3.1}$$

3.3 FEM Approximation

We approximate u by

$$u^h(x) = \sum_{j=1}^{N} a_j \phi_j(x). \tag{3.2}$$

If we choose ν with the same approximation functions, but a different linear combination

$$\nu^h(x) = \sum_{i=1}^{N} b_i \phi_i(x), \tag{3.3}$$

then we may write

$$\begin{aligned}&\underbrace{\int_\Omega \frac{d}{dx}\left(\sum_{i=1}^{N} b_i \phi_i(x)\right) E \frac{d}{dx}\left(\sum_{j=1}^{N} a_j \phi_j(x)\right) dx}_{\overset{\text{def}}{=}\text{stiffness contribution}} \\ &= \underbrace{\int_\Omega \left(\sum_{i=1}^{N} b_i \phi_i(x)\right) f \; dx}_{\overset{\text{def}}{=}\text{body load contribution}} + \underbrace{\left(\left(\sum_{i=1}^{N} b_i \phi_i(x)\right) t^*\right)|_{\Gamma_t}}_{\overset{\text{def}}{=}\text{traction load contribution}}.\end{aligned} \tag{3.4}$$

Since the ν's are arbitrary, the b_i are arbitrary, i.e., $\forall \nu \Rightarrow \forall b_i$, therefore

$$\boxed{\begin{array}{l}\sum_{i=1}^{N} b_i \left(\sum_{j=1}^{N} K_{ij} a_j - R_i\right) = 0 \Rightarrow [K]\{a\} = \{R\}, \\[2ex] K_{ij} \overset{\text{def}}{=} \int_\Omega \frac{d\phi_i}{dx} E \frac{d\phi_j}{dx}\, dx \text{ and} \\[2ex] R_i \overset{\text{def}}{=} \int_\Omega \phi_i f \, dx + \phi_i t^*|_{\Gamma_t},\end{array}} \tag{3.5}$$

where $[K]$ is an $N \times N$ ("stiffness") matrix with components K_{ij} and $\{R\}$ is an $N \times 1$ ("load") vector with components R_i. This is the system of equations that is to be solved. Thus, large N implies large systems of equations and more computational effort. However, with increasing N, we obtain more accurate approximate solutions. We remark that large N does not seem like much of a concern for one-dimensional problems, but is of immense concern for three-dimensional problems.

3.4 Construction of FEM Basis Functions

As mentioned, a primary problem with Galerkin's method is that it provides no systematic way of constructing approximation functions. The difficulties that arise include (1) ill-conditioned systems due to poor choices of approximation functions and (2) domains with irregular geometries. To circumvent these problems, the FEM defines basis (approximation) functions in a piecewise manner over a subdomain, "the finite elements," of the entire domain. The basis functions are usually simple polynomials of low degree. The following three criteria are important:

- The basis functions are smooth enough to be members of $H^1(\Omega)$.
- The basis functions are simple piecewise polynomials, defined element by element.
- The basis functions form a simple nodal basis where $\phi_i(x_j) = 0$ $(i \neq j)$ and $\phi_i(x_i) = 1$, furthermore, $\sum_{i=1}^{N} \phi_i(x) = 1$ for all x and $\phi_i(x) = 0$ outside of the elements that share node i.

A set of candidate functions are defined by

$$\phi(x) = \frac{x - x_{i-1}}{h_i} \qquad \text{for } x_{i-1} \leq x \leq x_i, \tag{3.6}$$

where $h_i = x_i - x_{i-1}$ and

$$\phi(x) = 1 - \frac{x - x_i}{h_{i+1}} \qquad \text{for } x_i \leq x \leq x_{i+1}, \tag{3.7}$$

and $\phi(x) = 0$ otherwise. The derivative of the function is

$$\frac{d\phi}{dx} = \frac{1}{h_i} \qquad \text{for } x_{i-1} \leq x \leq x_i, \tag{3.8}$$

and

$$\frac{d\phi}{dx} = -\frac{1}{h_{i+1}} \qquad \text{for } x_i \leq x \leq x_{i+1}. \tag{3.9}$$

The functions are arranged so that the "apex" of the ith function coincides with the ith node (Fig. 3.1). This framework provides many advantages, for example simple numerical integration.

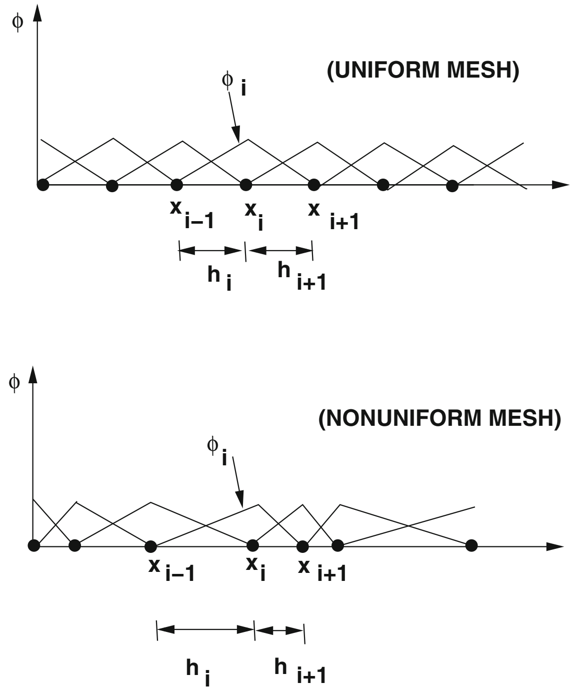


Fig. 3.1 A one-dimensional finite element basis. At the top, is a uniform mesh example and at the bottom, nonuniform

3.5 Integration and Gaussian Quadrature

Gauss made the crucial observation that one can integrate a polynomial of order 2G-1 exactly with G "sampling" points and appropriate weights. Thus, in order to automate the integration process, one redefines the function $F(x)$ over a normalized unit domain $-1 \leq \zeta \leq +1$

$$\int_0^L F(x)\,dx = \int_{-1}^1 F(x(\zeta))\,J(\zeta)d\zeta = \sum_{i=1}^G w_i F(\zeta_i) J(\zeta_i) = \sum_{i=1}^G w_i \hat{F}(\zeta_i), \tag{3.10}$$

where J is the Jacobian of the transformation. Unlike most integration schemes, Gaussian quadrature relaxes the usual restriction that the function sampling locations be evenly spaced. According to the above, we should be able to integrate a cubic (and lower order) term exactly with $G = 2$ points, since $(2G - 1) = 3$. Therefore

- For a cubic (ζ^3):

$$\int_{-1}^1 \zeta^3\,d\zeta = 0 = w_1\zeta_1^3 + w_2\zeta_2^3 \tag{3.11}$$

- For a quadratic (ζ^2):

$$\int_{-1}^{1} \zeta^2 \, d\zeta = 2/3 = w_1 \zeta_1^2 + w_2 \zeta_2^2 \tag{3.12}$$

- For a linear (ζ):

$$\int_{-1}^{1} \zeta \, d\zeta = 0 = w_1 \zeta_1 + w_2 \zeta_2 \tag{3.13}$$

- For a constant (1):

$$\int_{-1}^{1} 1 \, d\zeta = 2 = w_1 1 + w_2 1 = w_1 + w_2 \tag{3.14}$$

There are four variables, ζ_1, ζ_2, w_1, w_2, to solve for. The solution that satisfies all of the requirements is $\zeta_1 = \sqrt{1/3} = -\zeta_2$ and $w_1 = w_2 = 1$. For the general case of G points, we have

$$\int_{-1}^{1} \hat{F}(\zeta) d\zeta = \sum_{i=1}^{G} w_i \hat{F}(\zeta_i) \tag{3.15}$$

and subsequently 2G nonlinear equations for the ζ_i's and w_i's. Fortunately, the ζ_i's are the roots to the Gth degree Legendre polynomial, defined via the recursion (Fig. 3.2)

$$(G+1)L_{G+1}(\zeta) - (2G+1)\zeta L_G(\zeta) + G L_{G-1}(\zeta) = 0, \tag{3.16}$$

with $L_o(\zeta) = 1$, $L_1(\zeta) = \zeta$. The roots of the Legendre polynomial are well known and tabulated. Once the roots are determined the remaining equations for the w_i's are linear and easy to solve. Fortunately, the roots are precomputed over a normalized unit domain, and one does not need to compute them. The only task is to convert the domain of each element to a standard unit domain (in the next section). A table of Gauss weights can be found in Table 3.1.

3.5.1 An Example

Consider the following integral

$$I \overset{\text{def}}{=} \int_{0.2}^{1.5} 10 e^{-x^2} \, dx. \tag{3.17}$$

This integral is of the form

$$I \overset{\text{def}}{=} \int_{a}^{b} f(x) \, dx = \int_{-1}^{1} f\left(\frac{(b-a)\zeta + b + a}{2}\right) \underbrace{\frac{(b-a)}{2}}_{J} \, d\zeta, \tag{3.18}$$

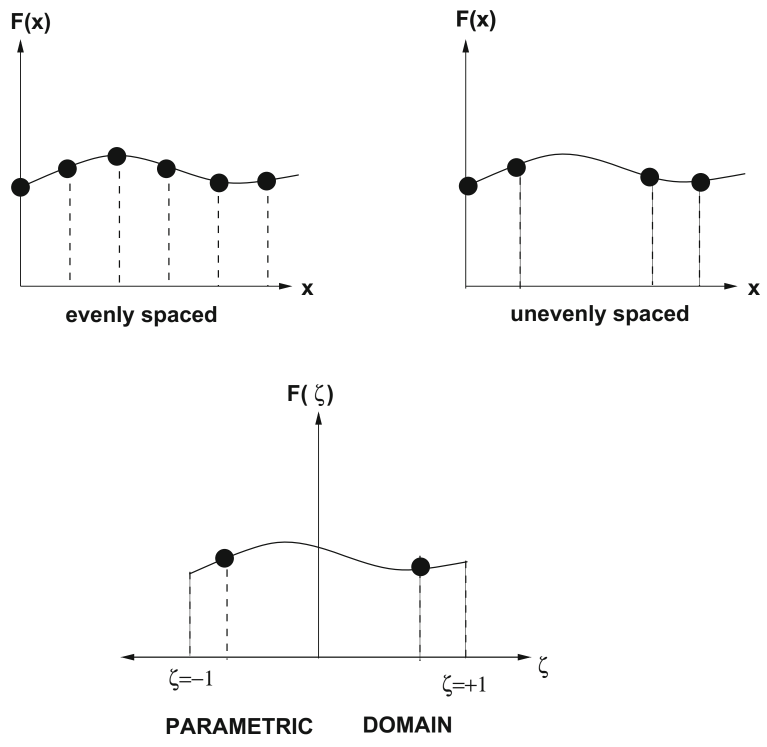


Fig. 3.2 Integration using Gaussian quadrature

Table 3.1 Gauss integration rules

Gauss rule	ζ_i	w_i
2	0.577350269189626	1.000000000000000
	−0.577350269189626	1.000000000000000
3	0.000000000000000	0.888888888888889
	0.774596669224148	0.555555555555556
	−0.774596669224148	0.555555555555556
4	0.339981043584856	0.652145154862546
	0.861136311594053	0.347854845137454
	−0.339981043584856	0.652145154862546
	−0.861136311594053	0.347854845137454
5	0.000000000000000	0.568888888888889
	0.538469310105683	0.478628670499366
	0.906179845938664	0.236926885056189
	−0.538469310105683	0.478628670499366
	−0.906179845938664	0.236926885056189

where we have the following mapping

$$x = \frac{(b-a)\zeta + b + a}{2} \Rightarrow dx = \frac{b-a}{2} d\zeta. \tag{3.19}$$

Applying this transformation, we have

$$I \overset{\text{def}}{=} \int_{0.2}^{1.5} 10e^{-x^2}\, dx = \frac{1.5-0.2}{2} \int_{-1}^{1} 10e^{-(0.65\zeta+0.85)^2}\, d\zeta, \tag{3.20}$$

where $x = 0.65\zeta + 0.85$. Applying a three-point rule yields (*the exact answer is* 6.588)

$$\begin{aligned} I &= \frac{1.5-0.2}{2} \int_{-1}^{1} 10e^{-(0.65\zeta+0.85)^2}\, d\zeta \\ &= 6.5 \left(0.5555e^{-(0.65(-0.77459)+0.85)^2} + 0.8888e^{-(0.65(0)+0.85)^2} + 0.5555e^{-(0.65(0.77459)+0.85)^2}\right) \\ &= 6.586. \end{aligned} \tag{3.21}$$

3.6 Global/Local Transformations

One strength of the finite element method is that most of the computations can be done in an element-by-element manner. Accordingly, we define the entries of the stiffness matrix $[K]$ as

$$K_{ij} = \int_{\Omega} \frac{d\phi_i}{dx} E \frac{d\phi_j}{dx}\, dx, \tag{3.22}$$

and the load vector as

$$R_i = \int_{\Omega} \phi_i f\, dx + \phi_i t^*|_{\Gamma_t}. \tag{3.23}$$

We partition the domain Ω into elements, $\Omega_1, \Omega_2, ..., \Omega_e, ...\Omega_N$, and can consequently break the calculations (integrals over Ω) into elements (integrals over Ω_e), $K_{ij} = \sum_e K_{ij}^e$, where

$$K_{ij}^e = \int_{\Omega_e} \frac{d\phi_i}{dx} E \frac{d\phi_j}{dx}\, dx \tag{3.24}$$

and

$$R_i^e = \int_{\Omega_e} \phi_i f\, dx + \phi_i t^*|_{\Gamma_{t,e}}, \tag{3.25}$$

where $R_i = \sum_e R_i^e$ and $\Gamma_{t,e} = \Gamma_t \cap \Omega_e$.

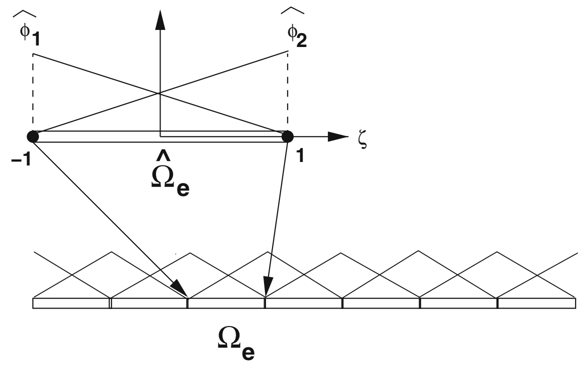


Fig. 3.3 A one-dimensional linear finite element mapping

In order to make the calculations systematic we wish to use the generic or master element defined in a local coordinate system (ζ). Accordingly, we need the following mapping functions, from the master coordinates to the real spatial coordinates, $M_x(\zeta) \mapsto x$ (Fig. 3.3)

$$x = \sum_{i=1}^{2} \mathcal{X}_i \hat{\phi}_i \overset{\text{def}}{=} M_x(\zeta), \tag{3.26}$$

where the $\mathcal{X}_i$ are the true spatial coordinates of the ith node, and where $\hat{\phi}(\zeta) \overset{\text{def}}{=} \phi(x(\zeta))$. These types of mappings are usually termed "parametric" maps. If the polynomial order of the shape functions is as high as the Galerkin approximation over the element, it is called an "isoparametric" map, lower, then "subparametric" map, higher, then "superparametric".

3.7 Differential Properties of Shape Functions

The master element shape functions form a nodal bases of linear approximation given by

$$\hat{\phi}_1 = \frac{1}{2}(1 - \zeta) \qquad \text{and} \qquad \hat{\phi}_2 = \frac{1}{2}(1 + \zeta). \tag{3.27}$$

They have the following properties:

- For linear elements we have a nodal basis consisting of two nodes, and thus two degrees of freedom.
- The nodal shape functions can be derived quite easily, by realizing that it is a nodal basis; i.e., they are unity at the corresponding node and zero at all other nodes.

We note that the ϕ_i's are never really computed; we actually start with the $\hat{\phi}_i$'s and then map them into the actual problem domain. Therefore in the stiffness matrix and right-hand side element calculations, all terms must be defined in terms of the local coordinates. With this in mind, we introduce some fundamental quantities, such as the finite element mapping deformation gradient

$$F \overset{\text{def}}{=} \frac{dx}{d\zeta}. \tag{3.28}$$

The corresponding one-dimensional determinant is $|F| = \frac{dx}{d\zeta} \overset{\text{def}}{=} J$, which is known as the Jacobian. We will use $|F|$ and J interchangeably throughout this monograph. The differential relations $\zeta \rightarrow x$ are

$$\frac{d()}{d\zeta} = \frac{dx}{d\zeta}\frac{d()}{dx} = J\frac{d()}{dx}. \tag{3.29}$$

The inverse differential relations $x \rightarrow \zeta$ are

$$\frac{d()}{dx} = \frac{d\zeta}{dx}\frac{d()}{d\zeta} = \frac{1}{J}\frac{d()}{d\zeta}. \tag{3.30}$$

We can now express $\frac{d}{dx}$ in terms ζ, via

$$\frac{d\phi}{dx} = \frac{d}{dx}\phi(M(\zeta)) = \frac{d\zeta}{dx}\frac{d}{d\zeta}\phi(M(\zeta)) = \frac{d\zeta}{dx}\frac{d}{d\zeta}\hat{\phi}(\zeta). \tag{3.31}$$

Finally with quadrature for each element

$$K_{ij}^e = \sum_{q=1}^{g} \underbrace{w_q \left(\frac{d}{d\zeta}(\phi_i(M(\zeta))\right)\frac{d\zeta}{dx}E\left(\frac{d}{d\zeta}(\phi_j(M(\zeta))\right)\frac{d\zeta}{dx}|F|}_{\text{evaluated at } \zeta=\zeta_q} \tag{3.32}$$

and

$$R_i^e = \sum_{q=1}^{g} \underbrace{w_q\phi_i(M(\zeta))f|F|}_{\text{evaluated at } \zeta=\zeta_q} + \underbrace{\phi_i(M(\zeta))t^*}_{\text{evaluated on traction endpoints}}, \tag{3.33}$$

where the w_q are Gauss weights.

Remarks: It is permitted to have material discontinuities within the finite elements. On the implementation level, the system of equations to be solved is $[K]\{a\} = \{R\}$, where the stiffness matrix is represented by $K(I, J)$, where (I, J) are the global entries. However, one can easily take advantage of the element-by-element structure and store the entries via $k^e(e, i, j)$, where (e, i, j) are the local (element) entries. For the local storage approach, a global/local index relation must be made to connect the local entry to the global entry when the linear algebraic solution process begins. This is a relatively simple and efficient storage system to encode. The element-by-element strategy has other advantages with regard to element-by-element system solvers. This is trivial in one dimension; however, it can be complicated in three dimensions. This is discussed later.

3.8 Post-Processing

Post-processing for the stress, strain, and energy from the existing displacement solution, i.e., the values of the nodal displacements, the shape functions, are straightforward. Essentially the process is the same as the formation of the weak form in the system. Therefore, for each element

$$\frac{du}{dx} = \frac{d}{dx}\sum_{i=1}^{2} a_i\phi_i = \left(\frac{d}{d\zeta}\sum_{i=1}^{2} a_i\hat{\phi}_i\right)\frac{d\zeta}{dx}. \tag{3.34}$$

3.9 A Detailed Example

3.9.1 Weak Form

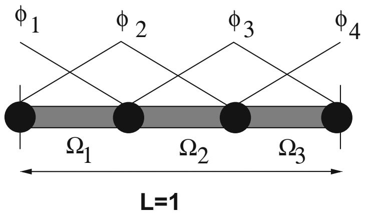
Consider the following problem (Fig. 3.4)

$$\frac{d}{dx}(E(x)\frac{du}{dx}) + f(x) = 0, \tag{3.35}$$

$u(0) = 0$ and $\frac{du}{dx}(1) = t$, posed over a domain of unit length. The weak form is

$$\int_o^{L=1} \frac{dv}{dx}E(x)\frac{du}{dx}\,dx = \int_o^{L=1} f(x)v\,dx + \underbrace{(E(x)\frac{du}{dx}v)|_0^1}_{=t^*v}. \tag{3.36}$$

Fig. 3.4 Three elements and four nodes



Using three elements (four nodes), each of equal size, the following holds:

- Over element 1 (Ω_1): $\mathcal{X}_i = \mathcal{X}_1 = 0$ and $\mathcal{X}_{i+1} = \mathcal{X}_2 = 1/3$, $\phi_1(x) = 1 - 3x$ and $\phi_2(x) = 3x$,
- Over element 2 (Ω_2): $\mathcal{X}_i = \mathcal{X}_2 = 1/3$ and $\mathcal{X}_{i+1} = \mathcal{X}_3 = 2/3$, $\phi_2(x) = 2 - 3x$ and $\phi_3(x) = -1 + 3x$,
- Over element 3 (Ω_3): $\mathcal{X}_i = \mathcal{X}_3 = 2/3$ and $\mathcal{X}_{i+1} = \mathcal{X}_4 = 1$, $\phi_3(x) = 3 - 3x$ and $\phi_4(x) = -2 + 3x$,

We break the calculations up element by element. *All calculations between* $0 \leq x \leq 1/3$ *belong to element number 1, while all calculations between* $1/3 \leq x \leq 2/3$ *belong to element number 2 and all calculations between* $2/3 \leq x \leq 1$ *belong to element number 3.*

3.9.2 Formation of the Discrete System

For element number 1, to compute $K_{ij}^{e=1}$, we study the following term for $i = 1, 2, 3$:

$$\sum_{j=1}^{N} \left(\int_0^{1/3} \frac{d\phi_i}{dx} E(x) \frac{d\phi_j}{dx}\, dx \right) a_j. \tag{3.37}$$

Explicitly, for $i = 1$, we have

$$\underbrace{\left(\int_0^{1/3} \frac{d\phi_1}{dx} E(x) \frac{d\phi_1}{dx}\, dx \right) a_1}_{K_{11}^{e=1}} + \underbrace{\left(\int_0^{1/3} \frac{d\phi_1}{dx} E(x) \frac{d\phi_2}{dx}\, dx \right) a_2}_{K_{12}^{e=1}} + \underbrace{\left(\int_0^{1/3} \frac{d\phi_1}{dx} E(x) \frac{d\phi_3}{dx}\, dx \right) a_3}_{=0} +0, \; etc., \tag{3.38}$$

where the zero-valued terms vanish because the basis functions are zero over the first finite element domain. The entries such as $K_{ij}^{e=1}$ multiply the term a_j, which dictate their location within the global stiffness matrix. If we repeat the procedure for $i = 2$, $j = 1, 2, 3$, we obtain the entries for the global stiffness matrix (4×4)

$$\begin{bmatrix} K_{11}^{e=1} & K_{12}^{e=1} & 0 & 0 \\ K_{21}^{e=1} & K_{22}^{e=1} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.39}$$

stemming from the placement of the local element stiffness matrix

$$\begin{bmatrix} K_{11}^{e=1} & K_{12}^{e=1} \\ K_{21}^{e=1} & K_{22}^{e=1} \end{bmatrix} \tag{3.40}$$

into the global stiffness matrix. Following a similar procedure for the right-hand side (load)

$$\underbrace{\left(\int_0^{1/3} \phi_i f(x)\, dx\right)}_{R_i^{e=1}} \tag{3.41}$$

yields ($i = 1, 2$)

$$\begin{bmatrix} R_1^{e=1} \\ R_2^{e=1} \end{bmatrix}. \tag{3.42}$$

Repeating the procedure for all three of the elements yields

$$\begin{bmatrix} K_{11}^{e=1} & K_{12}^{e=1} & 0 & 0 \\ K_{21}^{e=1} & K_{22}^{e=1} + K_{11}^{e=2} & K_{12}^{e=2} & 0 \\ 0 & K_{21}^{e=2} & K_{22}^{e=2} + K_{11}^{e=3} & K_{12}^{e=3} \\ 0 & 0 & K_{21}^{e=3} & K_{22}^{e=3} \end{bmatrix} \tag{3.43}$$

and

$$\begin{bmatrix} R_1^{e=1} \\ R_2^{e=1} + R_1^{e=2} \\ R_2^{e=2} + R_1^{e=3} \\ R_2^{e=3} \end{bmatrix}. \tag{3.44}$$

Note that the load vector

$$R_2^{e=3} = \int_{2/3}^{1} \phi_4 f(x)\, dx + E(x)\frac{du}{dx}\phi_4(1) = \int_{2/3}^{1} \phi_4 f(x)\, dx + t^* \tag{3.45}$$

has a traction contribution from the right endpoint. In summary, the basic process is to (1) compute element by element and (2) to sweep over all basis function contributions over each element.

Remark: We note that all integrals are computed using Gaussian quadrature.

3.9.3 Applying Boundary Conditions

Applying the primal (displacement) boundary conditions requires us to recall that the b_i's in the representation of the test functions are not arbitrary at the endpoints, thus the equations associated with those test functions have to be eliminated, and the value of the approximate solution enforced at the displacement boundary condition via[1]

$$u^h(x=0) = \sum_{j=1}^{4} a_j \phi_j(x=0) = a_1, \tag{3.46}$$

which is the displacement-specified boundary condition. Thus, we have the following system of equations

$$\begin{bmatrix} K_{22}^{e=1} + K_{11}^{e=2} & K_{12}^{e=2} & 0 \\ K_{21}^{e=2} & K_{22}^{e=2} + K_{11}^{e=3} & K_{12}^{e=3} \\ 0 & K_{21}^{e=3} & K_{22}^{e=3} \end{bmatrix} \begin{bmatrix} a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} R_2^{e=1} + R_1^{e=2} - K_{12}^{e=1} a_1 \\ R_2^{e=2} + R_1^{e=3} \\ R_2^{e=3} \end{bmatrix} \tag{3.47}$$

3.9.4 Massive Data Storage Reduction

The direct storage of $K(I, J)$ requires $N \times N$ entries. The element-by-element storage, $k^e(e, i, j)$, requires $4e$. The memory requirements for an element-by-element paradigm are much smaller than those for a direct scheme, which store needless zeros. For example, for $N = 10^4$ nodes, the direct storage is $(10^4)^2 = 10^8$, while the element-by-element storage is 9999×4, *which is essentially 2500 times less than direct storage.* Additionally, there is a massive reduction of mathematical operations during the algebraic solution phase, because of the element-by-element structure of FEM system.

3.10 Quadratic Elements

In many cases, if the character of the exact solution is known to be smooth, it is advantageous to use higher-order approximation elements. Generally, if the exact solution to a problem is smooth, for sufficiently fine meshes, if one compares, for the same number of nodes, the solution produced with linear basis functions to the solution produced with quadratic basis functions, the quadratically produced solution is more accurate. Similarly, if the exact solution is rough (nonsmooth), for sufficiently fine meshes, if one compares, for the same number of nodes, the solution produced with linear basis functions to the solution produced with quadratic basis functions, the linearly produced solution is more accurate (Fig. 3.5).

To illustrate how to construct a quadratic finite element approximation, we follow a similar template for linear elements, however, with three nodes instead of two. Consistent with the basic nodal basis construction, the basis function must equal

[1]The traction boundary conditions are automatically accounted for in the weak formulation.

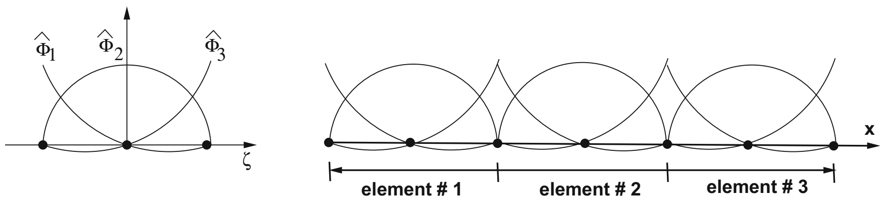


Fig. 3.5 Three quadratic elements with seven nodes

unity on the node it belongs and be zero at the others. Thus, for a generic quadratic element:

- For node # 1: $\hat{\phi}_1(\zeta) = -\frac{1}{2}(1-\zeta)\zeta$, which yields $\hat{\phi}_1(-1) = 1$, $\hat{\phi}(0) = 0$, $\hat{\phi}_1(1) = 0$,
- For node # 2: $\hat{\phi}_2(\zeta) = (1+\zeta)(1-\zeta)$, which yields $\hat{\phi}_2(-1) = 0$, $\hat{\phi}_2(0) = 1$, $\hat{\phi}_2(1) = 0$ and
- For node # 3: $\hat{\phi}_3(\zeta) = \frac{1}{2}(\zeta+1)\zeta$ which yields $\hat{\phi}_3(-1) = 0, \hat{\phi}_3(0) = 0, \hat{\phi}_3(1) = 1$.

Following the approach for linear elements, the connection between x and ζ is

$$x(\zeta) = \mathcal{X}_i\hat{\phi}_1(\zeta) + \mathcal{X}_{i+1}\hat{\phi}_2(\zeta) + \mathcal{X}_{i+2}\hat{\phi}_3(\zeta). \tag{3.48}$$

Clearly, the weak form does not change for linear or quadratic approximations. Furthermore, the quadratically generated system has a similar form to the linearly generated system

$$\sum_{j=1}^{N} K_{ij}a_j = R_i \qquad i = 1, 2, ...N, \tag{3.49}$$

where N is the number of nodes in 1-D. Let us consider an example with three elements, resulting in 7 nodes. Breaking up the integral into the elements

$$\int_0^1 = \int_0^{1/3} + \int_{1/3}^{2/3} + \int_{2/3}^{1}. \tag{3.50}$$

For element #1, for $i = 1, 2...N$, we need to compute

$$\sum_{j=1}^{N} \underbrace{\int_0^{1/3} \frac{d\phi_i}{dx} E(x) \frac{d\phi_j}{dx}\, dx}_{K_{ij}^{e=1}}, \tag{3.51}$$

yielding

$$\underbrace{\sum_{j=1}^{N}\int_0^{1/3}\frac{d\phi_1}{dx}E(x)\frac{d\phi_j}{dx}\,dx}_{K_{1j}^{e=1}} = \underbrace{\int_0^{1/3}\frac{d\phi_1}{dx}E(x)\frac{d\phi_1}{dx}\,dx}_{K_{11}^{e=1}} + \underbrace{\int_0^{1/3}\frac{d\phi_1}{dx}E(x)\frac{d\phi_2}{dx}\,dx}_{K_{12}^{e=1}} + \underbrace{\int_0^{1/3}\frac{d\phi_1}{dx}E(x)\frac{d\phi_3}{dx}\,dx}_{K_{13}^{e=1}} + \underbrace{\int_0^{1/3}\frac{d\phi_1}{dx}E(x)\frac{d\phi_4}{dx}\,dx}_{K_{14}^{e=1}=0}. \quad (3.52)$$

For the right-hand side, for $i = 1, 2...N$, we need to compute

$$\int_0^{1/3}\phi_i f(x)\,dx = R_i^{e=1}, \quad (3.53)$$

thus

$$R_1^{e=1} = \int_0^{1/3}\phi_1 f(x)\,dx. \quad (3.54)$$

Repeating this for $i = 2, 3...N$, we have

$$\begin{bmatrix} K_{11}^{e=1} & K_{12}^{e=1} & K_{13}^{e=1} & 0 & 0 & 0 & 0 \\ K_{21}^{e=1} & K_{22}^{e=1} & K_{23}^{e=1} & 0 & 0 & 0 & 0 \\ K_{31}^{e=1} & K_{32}^{e=1} & K_{33}^{e=1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{bmatrix} = \begin{bmatrix} R_1^{e=1} \\ R_2^{e=1} \\ R_3^{e=1} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.55)$$

This is then repeated for elements 2 and 3, to yield

$$\begin{bmatrix} K_{11}^{e=1} & K_{12}^{e=1} & K_{13}^{e=1} & 0 & 0 & 0 & 0 \\ K_{21}^{e=1} & K_{22}^{e=1} & K_{23}^{e=1} & 0 & 0 & 0 & 0 \\ K_{31}^{e=1} & K_{32}^{e=1} & K_{33}^{e=1}+K_{11}^{e=2} & K_{12}^{e=2} & K_{13}^{e=2} & 0 & 0 \\ 0 & 0 & K_{21}^{e=2} & K_{22}^{e=2} & K_{23}^{e=2} & 0 & 0 \\ 0 & 0 & K_{31}^{e=2} & K_{32}^{e=2} & K_{33}^{e=2}+K_{11}^{e=3} & K_{12}^{e=3} & K_{13}^{e=3} \\ 0 & 0 & 0 & 0 & K_{21}^{e=3} & K_{22}^{e=3} & K_{23}^{e=3} \\ 0 & 0 & 0 & 0 & K_{31}^{e=3} & K_{32}^{e=3} & K_{33}^{e=3} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{bmatrix} = \begin{bmatrix} R_1^{e=1} \\ R_2^{e=1} \\ R_3^{e=1}+R_1^{e=2} \\ R_2^{e=2} \\ R_3^{e=2}+R_1^{e=3} \\ R_2^{e=3} \\ R_3^{e=3} \end{bmatrix} \quad (3.56)$$

One then applies boundary conditions in the same manner as for linear elements.

Remark: A logical question to ask is what is the accuracy of the finite element method? This is addressed in the next chapter.

Reference

1. Courant, R. (1943). Variational methods for the solution of problems of equilibrium and vibrations. *Bulletin of the American Mathematical Society*, *49*, 1–23.