Chapter 1

Introduction

1.1 Historical Perspective

The finite element method is a computational technique for obtaining approximate solutions to the partial differential equations that arise in scientific and engineering applications. Rather than approximating the partial differential equation directly as with, *e.g.*, finite difference methods, the finite element method utilizes a *variational problem* that involves an integral of the differential equation over the problem domain. This domain is divided into a number of subdomains called *finite elements* and the solution of the partial differential equation is approximated by a simpler polynomial function on each element. These polynomials have to be pieced together so that the approximate solution has an appropriate degree of smoothness over the entire domain. Once this has been done, the variational integral is evaluated as a sum of contributions from each finite element. The result is an algebraic system for the approximate solution having a finite size rather than the original infinite-dimensional partial differential equation. Thus, like finite difference methods, the finite element process has *discretized* the partial differential equation but, unlike finite difference methods, the approximate solution is known throughout the domain as a pieceise polynomial function and not just at a set of points.

Logan [10] attributes the discovery of the finite element method to Hrennikof [8] and McHenry [11] who decomposed a two-dimensional problem domain into an assembly of one-dimensional bars and beams. In a paper that was not recognized for several years, Courant [6] used a *variational formulation* to describe a partial differential equation with a piecewise linear polynomial approximation of the solution relative to a decomposition of the problem domain into triangular elements to solve equilibrium and vibration problems. This is essentially the modern finite element method and represents the first application where the elements were pieces of a continuum rather than structural members.

Turner *et al.* [13] wrote a seminal paper on the subject that is widely regarded

as the beginning of the finite element era. They showed how to solve one- and two-dimensional problems using actual structural elements and triangular- and rectangular-element decompositions of a continuum. Their timing was better than Courant's [6], since success of the finite element method is dependent on digital computation which was emerging in the late 1950s. The concept was extended to more complex problems such as plate and shell deformation (*cf.* the historical discussion in Logan [10], Chapter 1) and it has now become one of the most important numerical techniques for solving partial differential equations. It has a number of advantages relative to other methods, including

- the treatment of problems on complex irregular regions,
- the use of nonuniform meshes to reflect solution gradations,
- the treatment of boundary conditions involving fluxes, and
- the construction of high-order approximations.

Originally used for steady (elliptic) problems, the finite element method is now used to solve transient parabolic and hyperbolic problems. Estimates of discretization errors may be obtained for reasonable costs. These are being used to verify the accuracy of the computation, and also to control an adaptive process whereby meshes are automatically refined and coarsened and/or the degrees of polynomial approximations are varied so as to compute solutions to desired accuracies in an optimal fashion [1, 2, 3, 4, 5, 7, 14].

1.2 Weighted Residual Methods

Our goal, in this introductory chapter, is to introduce the basic principles and tools of the finite element method using a linear two-point boundary value problem of the form

$$\mathcal{L}[u] := -\frac{d}{dx}(p(x)\frac{du}{dx}) + q(x)u = f(x), \qquad 0 < x < 1, \tag{1.2.1a}$$

$$u(0) = u(1) = 0. \tag{1.2.1b}$$

The finite element method is primarily used to address partial differential equations and is hardly used for two-point boundary value problems. By focusing on this problem, we hope to introduce the fundamental concepts without the geometric complexities encountered in two and three dimensions.

Problems like (1.2.1) arise in many situations including the longitudinal deformation of an elastic rod, steady heat conduction, and the transverse deflection of a supported

cable. In the latter case, for example, $u(x)$ represents the lateral deflection at position x of a cable having (scaled) unit length that is subjected to a tensile force p, loaded by a transverse force per unit length $f(x)$, and supported by a series of springs with elastic modulus q (Figure 1.2.1). The situation resembles the cable of a suspension bridge. The tensile force p is independent of x for the assumed small deformations of this model, but the applied loading and spring moduli could vary with position.

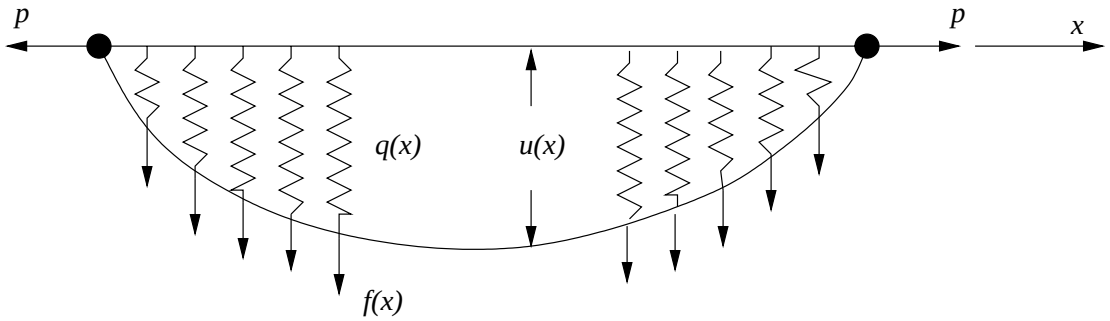

Figure 1.2.1: Deflection u of a cable under tension p, loaded by a force f per unit length, and supported by springs having elastic modulus q.

Mathematically, we will assume that $p(x)$ is positive and continuously differentiable for $x \in [0,1]$, $q(x)$ is non-negative and continuous on $[0,1]$, and $f(x)$ is continuous on $[0,1]$.

Even problems of this simplicity cannot generally be solved in terms of known functions; thus, the first topic on our agenda will be the development of a means of calculating approximate solutions of (1.2.1). With finite difference techniques, derivatives in (1.2.1a) are approximated by finite differences with respect to a mesh introduced on $[0,1]$ [12]. With the finite element method, the *method of weighted residuals (MWR)* is used to construct an integral formulation of (1.2.1) called a *variational problem*. To this end, let us multiply (1.2.1a) by a *test* or *weight function* v and integrate over $(0,1)$ to obtain

$$(v, \mathcal{L}[u] - f) = 0. \tag{1.2.2a}$$

We have introduced the L^2 *inner product*

$$(v,u) := \int_0^1 vu dx \tag{1.2.2b}$$

to represent the integral of a product of two functions.

The solution of (1.2.1) is also a solution of (1.2.2a) for all functions v for which the inner product exists. We'll express this requirement by writing $v \in L^2(0,1)$. All functions of class $L^2(0,1)$ are "square integrable" on $(0,1)$; thus, (v,v) exists. With this viewpoint and notation, we write (1.2.2a) more precisely as

$$(v, \mathcal{L}[u] - f) = 0, \qquad \forall v \in L^2(0,1). \tag{1.2.2c}$$

Equation (1.2.2c) is referred to as a *variational form* of problem (1.2.1). The reason for this terminology will become clearer as we develop the topic.

Using the method of weighted residuals, we construct approximate solutions by replacing u and v by simpler functions U and V and solving (1.2.2c) relative to these choices. Specifically, we'll consider approximations of the form

$$u(x) \approx U(x) = \sum_{j=1}^{N} c_j \phi_j(x), \tag{1.2.3a}$$

$$v(x) \approx V(x) = \sum_{j=1}^{N} d_j \psi_j(x). \tag{1.2.3b}$$

The functions $\phi_j(x)$ and $\psi_j(x)$, $j = 1, 2, \ldots, N$, are preselected and our goal is to determine the coefficients c_j, $j = 1, 2, \ldots, N$, so that U is a good approximation of u. For example, we might select

$$\phi_j(x) = \psi_j(x) = \sin j\pi x, \qquad j = 1, 2, \ldots, N,$$

to obtain approximations in the form of discrete Fourier series. In this case, every function satisfies the boundary conditions (1.2.1b), which seems like a good idea.

The approximation U is called a *trial function* and, as noted, V is called a *test function*. Since the differential operator $\mathcal{L}[u]$ is second order, we might expect $u \in C^2(0,1)$. (Actually, u can be slightly less smooth, but C^2 will suffice for the present discussion.) Thus, it's natural to expect U to also be an element of $C^2(0,1)$. Mathematically, we regard U as belonging to a finite-dimensional function space that is a subspace of $C^2(0,1)$. We express this condition by writing $U \in S^N(0,1) \subset C^2(0,1)$. (The restriction of these functions to the interval $0 < x < 1$ will, henceforth, be understood and we will no longer write the $(0,1)$.) With this interpretation, we'll call S^N the *trial space* and regard the preselected functions $\phi_j(x)$, $j = 1, 2, \ldots, N$, as forming a *basis* for S^N.

Likewise, since $v \in L^2$, we'll regard V as belonging to another finite-dimensional function space $\hat{S}^N$ called the *test space*. Thus, $V \in \hat{S}^N \subset L^2$ and $\psi_j(x)$, $j = 1, 2, \ldots, N$, provide a basis for $\hat{S}^N$.

Now, replacing v and u in (1.2.2c) by their approximations V and U, we have

$$(V, \mathcal{L}[U] - f) = 0, \qquad \forall V \in \hat{S}^N. \tag{1.2.4a}$$

The *residual*

$$r(x) := \mathcal{L}[U] - f(x) \tag{1.2.4b}$$

is apparent and clarifies the name "method of weighted residuals." The vanishing of the inner product (1.2.4a) implies that the residual is orthogonal in L^2 to all functions V in the test space $\hat{S}^N$.

Substituting (1.2.3) into (1.2.4a) and interchanging the sum and integral yields

$$\sum_{j=1}^{N} d_j(\psi_j, \mathcal{L}[U] - f) = 0, \qquad \forall d_j, \qquad j = 1, 2, \ldots, N. \tag{1.2.5}$$

Having selected the basis ψ_j, $j = 1, 2, \ldots, N$, the requirement that (1.2.4a) be satisfied for all $V \in \hat{S}^N$ implies that (1.2.5) be satisfied for all possible choices of d_k, $k = 1, 2, \ldots, N$. This, in turn, implies that

$$(\psi_j, \mathcal{L}[U] - f) = 0, \qquad j = 1, 2, \ldots, N. \tag{1.2.6}$$

Shortly, by example, we shall see that (1.2.6) represents a linear algebraic system for the unknown coefficients c_k, $k = 1, 2, \ldots, N$.

One obvious choice is to select the test space $\hat{S}^N$ to be the same as the trial space and use the same basis for each; thus, $\psi_k(x) = \phi_k(x)$, $k = 1, 2, \ldots, N$. This choice leads to *Galerkin's method*

$$(\phi_j, \mathcal{L}[u] - f) = 0, \qquad j = 1, 2, \ldots, N, \tag{1.2.7}$$

which, in a slightly different form, will be our "work horse." With $\phi_j \in C^2$, $j = 1, 2, \ldots, N$, the test space clearly has more continuity than necessary. Integrals like (1.2.4) or (1.2.6) exist for some pretty "wild" choices of V. Valid methods exist when V is a *Dirac delta* function (although such functions are not elements of L^2) and when V is a piecewise constant function (*cf.* Problems 1 and 2 at the end of this section).

There are many reasons to prefer a more symmetric variational form of (1.2.1) than (1.2.2), *e.g.*, problem (1.2.1) is symmetric (self-adjoint) and the variational form should reflect this. Additionally, we might want to choose the same trial and test spaces, as with Galerkin's method, but ask for less continuity on the trial space S^N. This is typically the case. As we shall see, it will be difficult to construct continuously differentiable approximations of finite element type in two and three dimensions. We can construct the symmetric variational form that we need by integrating the second derivative terms in (1.2.2a) by parts; thus, using (1.2.1a)

$$\int_0^1 v[-(pu')' + qu - f]dx = \int_0^1 (v'pu' + vqu - vf)dx - vpu'|_0^1 = 0 \tag{1.2.8}$$

where $(\ )' = d(\ )/dx$. The treatment of the last (boundary) term will need greater attention. For the moment, let v satisfy the same trivial boundary conditions (1.2.1b) as

u. In this case, the boundary term vanishes and (1.2.8) becomes

$$A(v,u) - (v,f) = 0 \tag{1.2.9a}$$

where

$$A(v,u) = \int_0^1 (v'pu' + vqu)dx. \tag{1.2.9b}$$

The integration by parts has eliminated second derivative terms from the formulation. Thus, solutions of (1.2.9) might have less continuity than those satisfying either (1.2.1) or (1.2.2). For this reason, they are called *weak solutions* in contrast to the *strong solutions* of (1.2.1) or (1.2.2). Weak solutions may lack the continuity to be strong solutions, but strong solutions are always weak solutions. In situations where weak and strong solutions differ, the weak solution is often the one of physical interest.

Since we've added a derivative to v by the integration by parts, v must be restricted to a space where functions have more continuity than those in L^2. Having symmetry in mind, we will select functions u and v that produce bounded values of

$$A(u,u) = \int_0^1 [p(u')^2 + qu^2]dx.$$

Actually, since p and q are smooth functions, it suffices for u and v to have bounded values of

$$\int_0^1 [(u')^2 + u^2]dx. \tag{1.2.10}$$

Functions where (1.2.10) exists are said to be elements of the Sobolev space H^1. We've also required that u and v satisfy the boundary conditions (1.2.1b). We identify those functions in H^1 that also satisfy (1.2.1b) as being elements of H_0^1. Thus, in summary, the variational problem consists of determining $u \in H_0^1$ such that

$$A(v,u) = (v,f), \qquad \forall v \in H_0^1. \tag{1.2.11}$$

The bilinear form $A(v,u)$ is called the *strain energy*. In mechanical systems it frequently corresponds to the stored or internal energy in the system.

We obtain approximate solutions of (1.2.11) in the manner described earlier for the more general method of weighted residuals. Thus, we replace u and v by their approximations U and V according to (1.2.3). Both U and V are regarded as belonging to the same finite-dimensional subspace S_0^N of H_0^1 and ϕ_j, $j = 1, 2, \ldots, N$, forms a basis for S_0^N. Thus, U is determined as the solution of

$$A(V,U) = (V,f), \qquad \forall V \in S_0^N. \tag{1.2.12a}$$

The substitution of (1.2.3b) with ψ_j replaced by ϕ_j in (1.2.12a) again reveals the more explicit form

$$A(\phi_j, U) = (\phi_j, f), \qquad j = 1, 2, \ldots, N. \tag{1.2.12b}$$

Finally, to make (1.2.12b) totally explicit, we eliminate U using (1.2.3a) and interchange a sum and integral to obtain

$$\sum_{k=1}^{N} c_k A(\phi_j, \phi_k) = (\phi_j, f), \qquad j = 1, 2, \ldots, N. \tag{1.2.12c}$$

Thus, the coefficients c_k, $k = 1, 2, \ldots, N$, of the approximate solution (1.2.3a) are determined as the solution of the linear algebraic equation (1.2.12c). Different choices of the basis ϕ_j, $j = 1, 2, \ldots, N$, will make the integrals involved in the strain energy (1.2.9b) and L^2 inner product (1.2.2b) easy or difficult to evaluate. They also affect the accuracy of the approximate solution. An example using a finite element basis is presented in the next section.

Problems

1. Consider the variational form (1.2.6) and select

   $$\psi_j(x) = \delta(x - x_j), \qquad j = 1, 2, \ldots, N,$$

   where $\delta(x)$ is the Dirac delta function satisfying

   $$\delta(x) = 0, \qquad x \neq 0, \qquad \int_{-\infty}^{\infty} \delta(x) dx = 1,$$

   and

   $$0 < x_1 < x_2 < \ldots < x_N < 1.$$

   Show that this choice of test function leads to the *collocation method*

   $$\mathcal{L}[U] - f(x)|_{x=x_j} = 0, \qquad j = 1, 2, \ldots, N.$$

   Thus, the differential equation (1.2.1) is satisfied exactly at N distinct points on $(0, 1)$.

2. The *subdomain method* uses piecewise continuous test functions having the basis

   $$\psi_j(x) := \begin{cases} 1, & \text{if } x \in (x_{j-1/2}, x_{j+1/2}) \\ 0, & \text{otherwise} \end{cases}.$$

   where $x_{j-1/2} = (x_j + x_{j-1})/2$. Using (1.2.6), show that the approximate solution $U(x)$ satisfies the differential equation (1.2.1a) on the average on each subinterval $(x_{j-1/2}, x_{j+1/2})$, $j = 1, 2, \ldots, N$.

3. Consider the two-point boundary value problem

$$-u'' + u = x, \qquad 0 < x < 1, \qquad u(0) = u(1) = 0,$$

which has the exact solution

$$u(x) = x - \frac{\sinh x}{\sinh 1}.$$

Solve this problem using Galerkin's method (1.2.12c) using the trial function

$$U(x) = c_1 \sin \pi x.$$

Thus, $N = 1$, $\phi_1(x) = \psi_1(x) = \sin \pi x$ in (1.2.3). Calculate the error in strain energy as $A(u,u) - A(U,U)$, where $A(u,v)$ is given by (1.2.9b).

1.3 A Simple Finite Element Problem

Finite element methods are weighted residuals methods that use bases of piecewise polynomials having small support. Thus, the functions $\phi(x)$ and $\psi(x)$ of (1.2.3, 1.2.4) are nonzero only on a small portion of problem domain. Since continuity may be difficult to impose, bases will typically use the minimum continuity necessary to ensure the existence of integrals and solution accuracy. The use of piecewise polynomial functions simplify the evaluation of integrals involved in the L^2 inner product and strain energy (1.2.2b, 1.2.9b) and help automate the solution process. Choosing bases with small support leads to a sparse, well-conditioned linear algebraic system (1.2.12c)) for the solution.

Let us illustrate the finite element method by solving the two-point boundary value problem (1.2.1) with constant coefficients, *i.e.*,

$$-pu'' + qu = f(x), \qquad 0 < x < 1, \qquad u(0) = u(1) = 0, \tag{1.3.1}$$

where $p > 0$ and $q \geq 0$. As described in Section 1.2, we construct a variational form of (1.2.1) using Galerkin's method (1.2.11). For this constant-coefficient problem, we seek to determine $u \in H_0^1$ satisfying

$$A(v,u) = (v,f), \qquad \forall v \in H_0^1, \tag{1.3.2a}$$

where

$$(v,u) = \int_0^1 vu dx, \tag{1.3.2b}$$

$$A(v,u) = \int_0^1 (v'pu' + vqu) dx. \tag{1.3.2c}$$

With u and v belonging to H_0^1, we are sure that the integrals (1.3.2b,c) exist and that the trivial boundary conditions are satisfied.

We will subsequently show that functions (of one variable) belonging to H^1 must necessarily be continuous. Accepting this for the moment, let us establish the goal of finding the simplest continuous piecewise polynomial approximations of u and v. This would be a piecewise linear polynomial with respect to a mesh

$$0 = x_0 < x_1 < \ldots < x_N = 1 \tag{1.3.3}$$

introduced on $[0, 1]$. Each subinterval (x_{j-1}, x_j), $j = 1, 2, \ldots, N$, is called a *finite element*. The basis is created from the "hat function"

$$\phi_j(x) = \begin{cases} \frac{x - x_{j-1}}{x_j - x_{j-1}}, & \text{if } x_{j-1} \le x < x_j \\ \frac{x_{j+1} - x}{x_{j+1} - x_j}, & \text{if} x_j \le x < x_{j+1} \\ 0, & \text{otherwise} \end{cases} . \tag{1.3.4a}$$

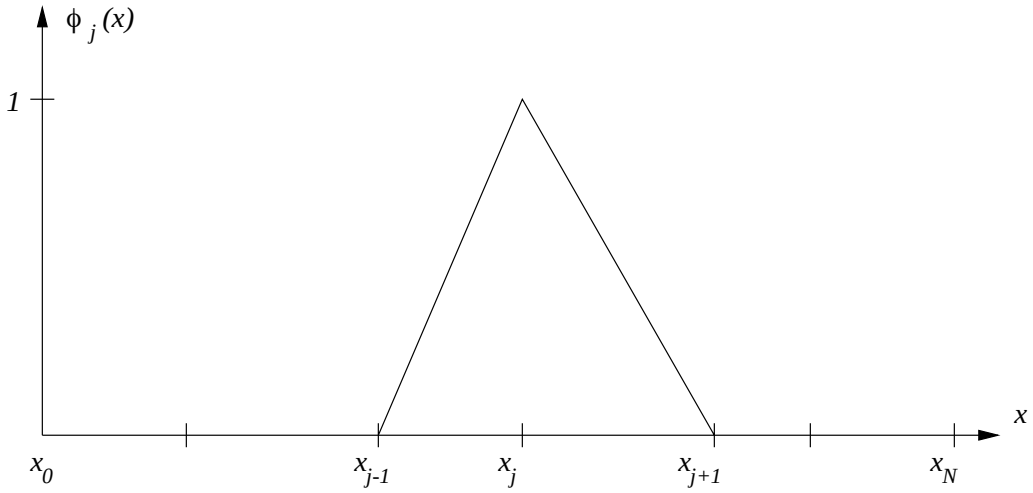

Figure 1.3.1: One-dimensional finite element mesh and piecewise linear hat function $\phi_j(x)$.

As shown in Figure 1.3.1, $\phi_j(x)$ is nonzero only on the two elements containing the *node* x_j. It rises and descends linearly on these two elements and has a maximal unit value at $x = x_j$. Indeed, it vanishes at all nodes but x_j, *i.e.*,

$$\phi_j(x_k) = \delta_{jk} := \begin{cases} 1, & \text{if } x_k = x_j \\ 0, & \text{otherwise} \end{cases} . \tag{1.3.4b}$$

Using this basis with (1.2.3), we consider approximations of the form

$$U(x) = \sum_{j=1}^{N-1} c_j \phi_j(x). \tag{1.3.5}$$

Let's examine this result more closely.

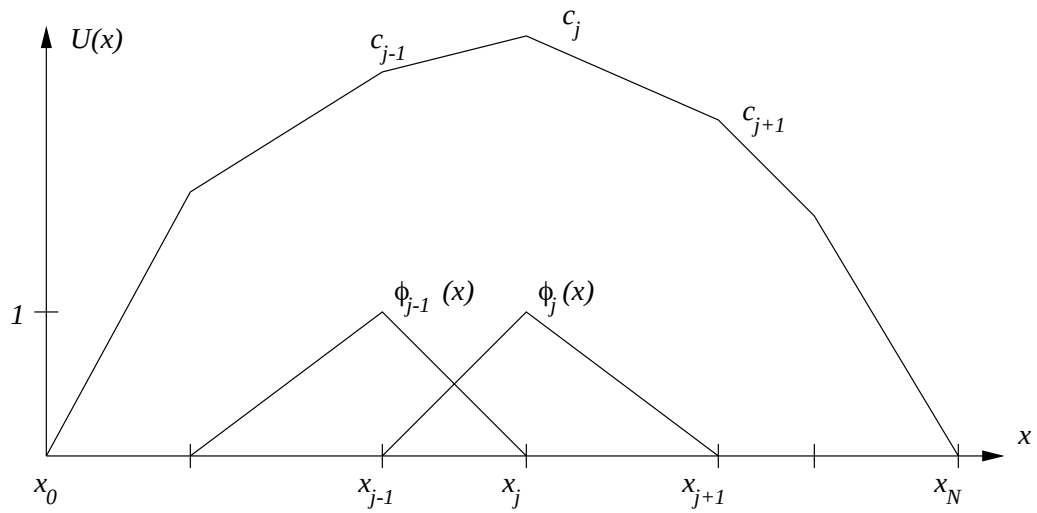

Figure 1.3.2: Piecewise linear finite element solution $U(x)$.

1. Since each $\phi_j(x)$ is a continuous piecewise linear function of x, their summation U is also continuous and piecewise linear. Evaluating U at a node x_k of the mesh using (1.3.4b) yields

$$U(x_k) = \sum_{j=1}^{N-1} c_j \phi_j(x_k) = c_k.$$

   Thus, the coefficients c_k, $k = 1, 2, \ldots, N-1$, are the values of U at the interior nodes of the mesh (Figure 1.3.2).

2. By selecting the lower and upper summation indices as 1 and $N-1$ we have ensured that (1.3.5) satisfies the prescribed boundary conditions

$$U(0) = U(1) = 0.$$

   As an alternative, we could have added basis elements $\phi_0(x)$ and $\phi_N(x)$ to the approximation and written the finite element solution as

$$U(x) = \sum_{j=0}^{N} c_j \phi_j(x). \tag{1.3.6}$$

   Since, using (1.3.4b), $U(x_0) = c_0$ and $U(x_N) = c_N$, the boundary conditions are satisfied by requiring $c_0 = c_N = 0$. Thus, the representations (1.3.5) or (1.3.6) are identical; however, (1.3.6) would be useful with non-trivial boundary data.

3. The restriction of the finite element solution (1.3.5) or (1.3.6) to the element $[x_{j-1}, x_j]$ is the linear function

$$U(x) = c_{j-1}\phi_{j-1}(x) + c_j \phi_j(x), \qquad x \in [x_{j-1}, x_j], \tag{1.3.7}$$

since ϕ_{j-1} and ϕ_j are the only nonzero basis elements on $[x_{j-1}, x_j]$ (Figure 1.3.2).

Using Galerkin's method in the form (1.2.12c), we have to solve

$$\sum_{k=1}^{N-1} c_k A(\phi_j, \phi_k) = (\phi_j, f), \qquad j = 1, 2, \ldots, N-1. \tag{1.3.8}$$

Equation (1.3.8) can be evaluated in a straightforward manner by substituting replacing ϕ_k and ϕ_j using (1.3.4) and evaluating the strain energy and L^2 inner product according to (1.3.2b,c). This development is illustrated in several texts (*e.g.*, [9], Section 1.2). We'll take a slightly more complex path to the solution in order to focus on the computer implementation of the finite element method. Thus, write (1.2.12a) as the summation of contributions from each element

$$\sum_{j=1}^{N} [A_j(V, U) - (V, f)_j] = 0, \qquad \forall V \in S_0^N, \tag{1.3.9a}$$

where

$$A_j(V, U) = A_j^S(V, U) + A_j^M(V, U), \tag{1.3.9b}$$

$$A_j^S(V, U) = \int_{x_{j-1}}^{x_j} pV'U'dx, \tag{1.3.9c}$$

$$A_j^M(V, U) = \int_{x_{j-1}}^{x_j} qVUdx, \tag{1.3.9d}$$

$$(V, f)_j = \int_{x_{j-1}}^{x_j} Vfdx. \tag{1.3.9e}$$

It is customary to divide the strain energy into two parts with A_j^S arising from internal energies and A_j^M arising from inertial effects or sources of energy.

Matrices are simple data structures to manipulate on a computer, so let us write the restriction of $U(x)$ to $[x_{j-1}, x_j]$ according to (1.3.7) as

$$U(x) = [c_{j-1}, c_j] \begin{bmatrix} \phi_{j-1}(x) \\ \phi_j(x) \end{bmatrix} = [\phi_{j-1}(x), \phi_j(x)] \begin{bmatrix} c_{j-1} \\ c_j \end{bmatrix}, \qquad x \in [x_{j-1}, x_j]. \tag{1.3.10a}$$

We can, likewise, use (1.2.3b) to write the restriction of the test function $V(x)$ to $[x_{j-1}, x_j]$ in the same form

$$V(x) = [d_{j-1}, d_j] \begin{bmatrix} \phi_{j-1}(x) \\ \phi_j(x) \end{bmatrix} = [\phi_{j-1}(x), \phi_j(x)] \begin{bmatrix} d_{j-1} \\ d_j \end{bmatrix}, \qquad x \in [x_{j-1}, x_j]. \tag{1.3.10b}$$

Our task is to substitute (1.3.10) into (1.3.9c-e) and evaluate the integrals. Let us begin by differentiating (1.3.10a) while using (1.3.4a) to obtain

$$U'(x) = [c_{j-1}, c_j] \begin{bmatrix} -1/h_j \\ 1/h_j \end{bmatrix} = [-1/h_j, 1/h_j] \begin{bmatrix} c_{j-1} \\ c_j \end{bmatrix}, \qquad x \in [x_{j-1}, x_j]. \tag{1.3.11a}$$

where

$$h_j = x_j - x_{j-1}, \qquad j = 1, 2, \ldots, N. \tag{1.3.11b}$$

Thus, $U'(x)$ is constant on $[x_{j-1}, x_j]$ and is given by the *first divided difference*

$$U'(x) = \frac{c_j - c_{j-1}}{h_j}, \qquad x \in [x_{j-1}, x_j].$$

Substituting (1.3.11) and a similar expression for $V'(x)$ into (1.3.9b) yields

$$A_j^S(V,U) = \int_{x_{j-1}}^{x_j} p[d_{j-1}, d_j] \begin{bmatrix} -1/h_j \\ 1/h_j \end{bmatrix} [-1/h_j, 1/h_j] \begin{bmatrix} c_{j-1} \\ c_j \end{bmatrix} dx$$

or

$$A_j^S(V,U) = [d_{j-1}, d_j] \left(\int_{x_{j-1}}^{x_j} p \begin{bmatrix} 1/h_j^2 & -1/h_j^2 \\ -1/h_j^2 & 1/h_j^2 \end{bmatrix} dx \right) \begin{bmatrix} c_{j-1} \\ c_j \end{bmatrix}.$$

The integrand is constant and can be evaluated to yield

$$A_j^S(V,U) = [d_{j-1}, d_j] \mathbf{K}_j \begin{bmatrix} c_{j-1} \\ c_j \end{bmatrix}, \qquad \mathbf{K}_j = \frac{p}{h_j} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}. \tag{1.3.12}$$

The 2×2 matrix $\mathbf{K}_j$ is called the *element stiffness matrix*. It depends on j through h_j, but would also have such dependence if p varied with x. The key observation is that $\mathbf{K}_j$ can be evaluated without knowing c_{j-1}, c_j, d_{j-1}, or d_j and this greatly simplifies the automation of the finite element method.

The evaluation of A_j^M proceeds similarly by substituting (1.3.10) into (1.3.9d) to obtain

$$A_j^M(V,U) = \int_{x_{j-1}}^{x_j} q[d_{j-1}, d_j] \begin{bmatrix} \phi_{j-1} \\ \phi_j \end{bmatrix} [\phi_{j-1}, \phi_j] \begin{bmatrix} c_{j-1} \\ c_j \end{bmatrix} dx.$$

With q a constant, the integrand is a quadratic polynomial in x that may be integrated exactly (*cf.* Problem 1 at the end of this section) to yield

$$A_j^M(V,U) = [d_{j-1}, d_j] \mathbf{M}_j \begin{bmatrix} c_{j-1} c_j \end{bmatrix}, \qquad \mathbf{M}_j = \frac{q h_j}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}, \tag{1.3.13}$$

where $\mathbf{M}_j$ is called the *element mass matrix* because, as noted, it often arises from inertial loading.

The final integral (1.3.9e) cannot be evaluated exactly for arbitrary functions $f(x)$. Without examining this matter carefully, let us approximate it by its linear interpolant

$$f(x) \approx f_{j-1}\phi_{j-1}(x) + f_j\phi_j(x), \qquad x \in [x_{j-1}, x_j], \tag{1.3.14}$$

where $f_j := f(x_j)$. Substituting (1.3.14) and (1.3.10b) into (1.3.9e) and evaluating the integral yields

$$(V,f)_j \approx \int_{x_{j-1}}^{x_j} [d_{j-1}, d_j] \begin{bmatrix} \phi_{j-1} \\ \phi_j \end{bmatrix} [\phi_{j-1}, \phi_j] \begin{bmatrix} f_{j-1} \\ f_j \end{bmatrix} dx = [d_{j-1}, d_j]\mathbf{l}_j \tag{1.3.15a}$$

where

$$\mathbf{l}_j = \frac{h_j}{6} \begin{bmatrix} 2f_{j-1} + f_j \\ f_{j-1} + 2f_j \end{bmatrix}. \tag{1.3.15b}$$

The vector $\mathbf{l}_j$ is called the *element load vector* and is due to the applied loading $f(x)$.

The next step in the process is the substitution of (1.3.12), (1.3.13), and (1.3.15) into (1.3.9a) and the summation over the elements. Since this our first example, we'll simplify matters by making the mesh uniform with $h_j = h = 1/N$, $j = 1, 2, \ldots, N$, and summing A_j^S, A_j^M, and $(V,f)_j$ separately. Thus, summing (1.3.12)

$$\sum_{j=1}^{N} A_j^S = \sum_{j=1}^{N} [d_{j-1}, d_j] \frac{p}{h} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} c_{j-1} \\ c_j \end{bmatrix}.$$

The first and last contributions have to be modified because of the boundary conditions which, as noted, prescribe $c_0 = c_N = d_0 = d_N = 0$. Thus,

$$\sum_{j=1}^{N} A_j^S = [d_1]\frac{p}{h}[1][c_1] + [d_1, d_2]\frac{p}{h} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} + \cdots$$

$$+[d_{N-2}, d_{N-1}]\frac{p}{h} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} c_{N-2} \\ c_{N-1} \end{bmatrix} + [d_{N-1}]\frac{p}{h}[1][c_{N-1}].$$

Although this form of the summation can be readily evaluated, it obscures the need for the matrices and complicates implementation issues. Thus, at the risk of further complexity, we'll expand each matrix and vector to dimension $N-1$ and write the summation as

$$\sum_{k=1}^{N} A_j^S = [d_1, d_2, \cdots, d_{N-1}]\frac{p}{h} \begin{bmatrix} 1 & & \\ & & \\ & & \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{N-1} \end{bmatrix}$$

$$+[d_1, d_2, \cdots, d_{N-1}]\frac{p}{h}\begin{bmatrix} 1 & -1 & & \\ -1 & 1 & & \\ & & & \\ & & & \end{bmatrix}\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{N-1} \end{bmatrix}$$

$$+\cdots+[d_1, d_2, \cdots, d_{N-1}]\frac{p}{h}\begin{bmatrix} & & & \\ & & & \\ & & 1 & -1 \\ & & -1 & 1 \end{bmatrix}\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{N-1} \end{bmatrix}$$

$$+[d_1, d_2, \cdots d_{N-1}]\frac{p}{h}\begin{bmatrix} & & & \\ & & & \\ & & & \\ & & & 1 \end{bmatrix}\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{N-1} \end{bmatrix}$$

Zero elements of the matrices have not been shown for clarity. With all matrices and vectors having the same dimension, the summation is

$$\sum_{j=1}^{N} A_j^S = \mathbf{d}^T\mathbf{K}\mathbf{c}, \tag{1.3.16a}$$

where

$$\mathbf{K} = \frac{p}{h}\begin{bmatrix} 2 & -1 & & & & & \\ -1 & 2 & -1 & & & & \\ & -1 & 2 & -1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & -1 & 2 & -1 & \\ & & & & -1 & 2 & \end{bmatrix}, \tag{1.3.16b}$$

$$\mathbf{c} = [c_1, c_2, \cdots, c_{N-1}]^T, \tag{1.3.16c}$$

$$\mathbf{d} = [d_1, d_2, \cdots, d_{N-1}]^T. \tag{1.3.16d}$$

The matrix $\mathbf{K}$ is called the *global stiffness matrix*. It is symmetric, positive definite, and tridiagonal. In the form that we have developed the results, the summation over elements is regarded as an *assembly* process where the element stiffness matrices are added into their proper places in the global stiffness matrix. It is not necessary to actually extend the dimensions of the element matrices to those of the global stiffness matrix. As indicated in Figure 1.3.3, the elemental indices determine the proper location to add a local matrix into the global matrix. Thus, the 2×2 element stiffness matrix $\mathbf{K}_j$ is added to rows

$$A_1^S = d_1 \underbrace{\frac{p}{h}[1]}\, c_1 \qquad A_2^S = [d_1, d_2] \underbrace{\frac{p}{h}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$

$$A_3^S = [d_2, d_3] \underbrace{\frac{p}{h}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}} \begin{bmatrix} c_2 \\ c_3 \end{bmatrix}$$

$$\mathbf{K} = \frac{p}{h}\begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & -1 & 1 & & \\ & & & & \\ & & & & \end{bmatrix}$$

Figure 1.3.3: Assembly of the first three element stiffness matrices into the global stiffness matrix.

$j-1$ and j and columns $j-1$ and j. Some modifications are needed for the first and last elements to account for the boundary conditions.

The summations of A_j^M and $(V,f)_j$ proceed in the same manner and, using (1.3.13) and (1.3.15), we obtain

$$\sum_{j=0}^{N} A_j^M = \mathbf{d}^T \mathbf{M} \mathbf{c}, \tag{1.3.17a}$$

$$\sum_{j=0}^{N} (V,f)_j = \mathbf{d}^T \mathbf{l} \tag{1.3.17b}$$

where

$$\mathbf{M} = \frac{qh}{6}\begin{bmatrix} 4 & 1 & & & \\ 1 & 4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 4 & 1 \\ & & & 1 & 4 \end{bmatrix}, \tag{1.3.17c}$$

$$\mathbf{l} = \frac{h}{6}\begin{bmatrix} f_0 + 4f_1 + f_2 \\ f_1 + 4f_2 + f_3 \\ \vdots \\ f_{N-2} + 4f_{N-1} + f_N \end{bmatrix}. \tag{1.3.17d}$$

The matrix $\mathbf{M}$ and the vector $\mathbf{l}$ are called the *global mass matrix* and *global load vector*, respectively.

Substituting (1.3.16a) and (1.3.17a,b) into (1.3.9a,b) gives

$$\mathbf{d}^T[(\mathbf{K}+\mathbf{M})\mathbf{c}-\mathbf{l}]=\mathbf{0}. \tag{1.3.18}$$

As noted in Section 1.2, the requirement that (1.3.9a) hold for *all* $V \in S_0^N$ is equivalent to satisfying (1.3.18) for all choices of $\mathbf{d}$. This is only possible when

$$(\mathbf{K}+\mathbf{M})\mathbf{c}=\mathbf{l}. \tag{1.3.19}$$

Thus, the nodal values c_k, $k=1,2,\ldots,N-1$, of the finite element solution are determined by solving a linear algebraic system. With $\mathbf{c}$ known, the piecewise linear finite element U can be evaluated for any x using (1.2.3a). The matrix $\mathbf{K}+\mathbf{M}$ is symmetric, positive definite, and tridiagonal. Such systems may be solved by the tridiagonal algorithm (*cf.* Problem 2 at the end of this section) in $O(N)$ operations, where an operation is a scalar multiply followed by an addition.

The discrete system (1.3.19) is similar to the one that would be obtained from a centered finite difference approximation of (1.3.1), which is [12]

$$(\mathbf{K}+\mathbf{D})\hat{\mathbf{c}}=\hat{\mathbf{l}}, \tag{1.3.20a}$$

where

$$\mathbf{D}=qh\begin{bmatrix}1&&&\\&1&&\\&&\ddots&\\&&&1\end{bmatrix},\qquad \hat{\mathbf{l}}=h\begin{bmatrix}f_1\\f_2\\\vdots\\f_{N-1}\end{bmatrix},\qquad \hat{\mathbf{c}}=\begin{bmatrix}\hat{c}_1\\\hat{c}_2\\\vdots\\\hat{c}_{N-1}\end{bmatrix}. \tag{1.3.20b}$$

Thus, the qu and f terms in (1.3.1) are approximated by diagonal matrices with the finite difference method. In the finite element method, they are "smoothed" by coupling diagonal terms with their nearest neighbors using Simpson's rule weights. The diagonal matrix $\mathbf{D}$ is sometimes called a "lumped" approximation of the *consistent mass matrix* $\mathbf{M}$. Both finite difference and finite element solutions behave similarly for the present problem and have the same order of accuracy at the nodes of a uniform mesh.

Example 1.3.1. Consider the finite element solution of

$$-u''+u=x,\qquad 0<x<1,\qquad u(0)=u(1)=0,$$

which has the exact solution

$$u(x)=x-\frac{\sinh x}{\sinh 1}.$$

Relative to the more general problem (1.3.1), this example has $p = q = 1$ and $f(x) = x$. We solve it using the piecewise-linear finite element method developed in this section on uniform meshes with spacing $h = 1/N$ for $N = 4, 8, \ldots, 128$. Before presenting results, it is worthwhile mentioning that the load vector (1.3.15) is exact for this example. Even though we replaced $f(x)$ by its piecewise linear interpolant according to (1.3.14), this introduced no error since $f(x)$ is a linear function of x.

Letting

$$e(x) = u(x) - U(x) \tag{1.3.21}$$

denote the *discretization error,* in Table 1.3.1 we display the maximum error of the finite element solution and of its first derivative at the nodes of a mesh, *i.e.*,

$$|e|_\infty := \max_{0<j<N} |e(x_j)|, \qquad |e'|_\infty := \max_{1<j<N} |e'(x_j^-)|. \tag{1.3.22}$$

We have seen that $U'(x)$ is a piecewise constant function with jumps at nodes. Data in Table 1.3.1 were obtained by using derivatives from the left, *i.e.*, $x_j^- = \lim_{\epsilon\to 0} x_j - \epsilon$. With this interpretation, the results of second and fourth columns of Table 1.3.1 indicate that $|e|_\infty/h^2$ and $|e'|_\infty/h$ are (essentially) constants; hence, we may conclude that $|e|_\infty = O(h^2)$ and $|e'|_\infty = O(h)$.

N	$\|e\|_\infty$	$\|e\|_\infty/h^2$	$\|e'\|_\infty$	$\|e'\|_\infty/h$
4	0.269(-3)	0.430(-2)	0.111(0)	0.444
8	0.688(-4)	0.441(-2)	0.589(-1)	0.471
16	0.172(-4)	0.441(-2)	0.303(-1)	0.485
32	0.432(-5)	0.442(-2)	0.154(-1)	0.492
64	0.108(-5)	0.442(-2)	0.775(-2)	0.496
128	0.270(-6)	0.442(-2)	0.389(-2)	0.498

Table 1.3.1: Maximum nodal errors of the piecewise-linear finite element solution and its derivative for Example 1.3.1. (Numbers in parenthesis indicate a power of 10.)

The finite element and exact solutions of this problem are displayed in Figure 1.3.4 for a uniform mesh with eight elements. It appears that the pointwise discretization errors are much smaller at nodes than they are globally. We'll see that this phenomena, called *superconvergence*, applies more generally than this single example would imply.

Since finite element solutions are defined as continuous functions (of x), we can also appraise their behavior in some global norms in addition to the *discrete error norms* used in Table 1.3.1. Many norms could provide useful information. One that we will use quite

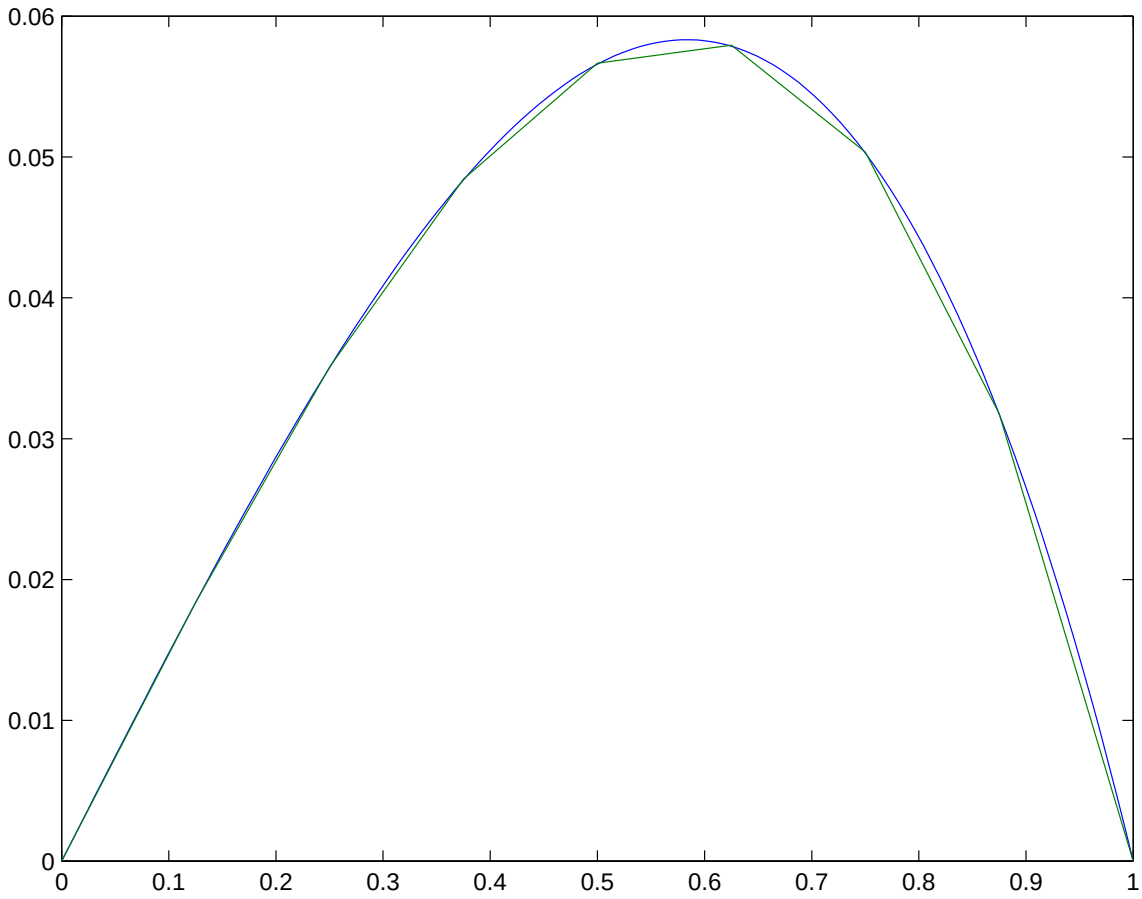

Figure 1.3.4: Exact and piecewise-linear finite element solutions of Example 1.3.1 on an 8-element mesh.

often is the square root of the strain energy of the error; thus, using (1.3.2c)

$$\|e\|_A := \sqrt{A(e,e)} = \left\{ \int_0^1 [p(e')^2 + qe^2] dx \right\}^{1/2} . \tag{1.3.23a}$$

This expression may easily be evaluated as a summation over the elements in the spirit of (1.3.9a). With $p = q = 1$ for this example,

$$\|e\|_A^2 = \int_0^1 [(e')^2 + e^2] dx.$$

The integral is the square of the norm used on the Sobolev space H^1; thus,

$$\|e\|_1 := \left\{ \int_0^1 [(e')^2 + e^2] dx \right\}^{1/2} . \tag{1.3.23b}$$

Other global error measures will be important to our analyses; however, the only one

that we will introduce at the moment is the L^2 norm

$$\|e\|_0 := \left[\int_0^1 e^2(x)dx\right]^{1/2}. \tag{1.3.23c}$$

Results for the L^2 and strain energy errors, presented in Table 1.3.2 for this example, indicate that $\|e\|_0 = O(h^2)$ and $\|e\|_A = O(h)$. The error in the H^1 norm would be identical to that in strain energy. Later, we will prove that these a priori error estimates are correct for this and similar problems. Errors in strain energy converge slower than those in L^2 because solution derivatives are involved and their nodal convergence is $O(h)$ (Table 1.3.1).

N	$\|e\|_0$	$\|e\|_0/h^2$	$\|e\|_A$	$\|e\|_A/h$
4	0.265(-2)	0.425(-1)	0.390(-1)	0.156
8	0.656(-3)	0.426(-1)	0.195(-1)	0.157
16	0.167(-3)	0.427(-1)	0.979(-2)	0.157
32	0.417(-4)	0.427(-1)	0.490(-2)	0.157
64	0.104(-4)	0.427(-1)	0.245(-2)	0.157
128	0.260(-5)	0.427(-1)	0.122(-2)	0.157

Table 1.3.2: Errors in L^2 and strain energy for the piecewise-linear finite element solution of Example 1.3.1. (Numbers in parenthesis indicate a power of 10.)

Problems

1. The integral involved in obtaining the mass matrix according to (1.3.13) may, of course, be done symbolically. It may also be evaluated numerically by Simpson's rule which is exact in this case since the integrand is a quadratic polynomial. Recall, that Simpson's rule is

$$\int_0^h \mathbf{F}(x)dx \approx \frac{h}{6}[\mathbf{F}(0) + 4\mathbf{F}(h/2) + \mathbf{F}(h)].$$

The mass matrix is

$$\mathbf{M}_j = \int_{x_{j-1}}^{x_j} \begin{bmatrix} \phi_{j-1} \\ \phi_j \end{bmatrix} [\phi_{j-1}, \phi_j]dx.$$

Using (1.3.4), determine $\mathbf{M}_j$ by Simpson's rule to verify the result (1.3.13). The use of Simpson's rule may be simpler than symbolic integration for this example since the trial functions are zero or unity at the ends of an element and one half at its center.

2. Consider the solution of the linear system

$$\mathbf{AX} = \mathbf{F}, \tag{1.3.24a}$$

where $\mathbf{F}$ and $\mathbf{X}$ are N-dimensional vectors and $\mathbf{A}$ is an $N \times N$ tridiagonal matrix having the form

$$\mathbf{A} = \begin{bmatrix} a_1 & c_1 & & & \\ b_2 & a_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & b_{N-1} & a_{N-1} & c_{N-1} \\ & & & b_N & a_N \end{bmatrix}. \tag{1.3.24b}$$

Assume that pivoting is not necessary and factor $\mathbf{A}$ as

$$\mathbf{A} = \mathbf{LU}, \tag{1.3.25a}$$

where $\mathbf{L}$ and $\mathbf{U}$ are lower and upper bidiagonal matrices having the form

$$\mathbf{L} = \begin{bmatrix} 1 & & & & \\ l_2 & 1 & & & \\ & l_3 & 1 & & \\ & & \ddots & \ddots & \\ & & & l_N & 1 \end{bmatrix}, \tag{1.3.25b}$$

$$\mathbf{U} = \begin{bmatrix} u_1 & v_1 & & & \\ & u_2 & v_2 & & \\ & & \ddots & \ddots & \\ & & & u_{N-1} & v_{N-1} \\ & & & & u_N \end{bmatrix}. \tag{1.3.25c}$$

Once the coefficients l_j, $j = 2, 3, \ldots, N$, u_j, $j = 1, 2, \ldots, N$, and v_j, $j = 1, 2, \ldots, N-1$, have been determined, the system (1.3.24a) may easily be solved by forward and backward substitution. Thus, using (1.3.25a) in (1.3.24a) gives

$$\mathbf{LUX} = \mathbf{F}. \tag{1.3.26a}$$

Let

$$\mathbf{UX} = \mathbf{Y}, \tag{1.3.26b}$$

then,

$$\mathbf{LY} = \mathbf{F}. \tag{1.3.26c}$$

2.1. Using (1.3.24) and (1.3.25), show

$$u_1 = a_1,$$

$$l_j = b_j/u_{j-1}, \qquad u_j = a_j - l_j c_{j-1}, \qquad j = 2, 3, \ldots, N,$$

$$v_j = c_j, \qquad j = 2, 3, \ldots, N.$$

2.2. Show that $\mathbf{Y}$ and $\mathbf{X}$ are computed as

$$Y_1 = F_1,$$
$$Y_j = F_j - l_j Y_{j-1}, \qquad j = 2, 3, \ldots, N,$$
$$X_N = y_N / u_N,$$
$$X_j = (Y_j - v_j X_{j+1})/u_j, \qquad j = N-1, N-2, \ldots, 1.$$

2.3. Develop a procedure to implement this scheme for solving tridiagonal systems. The input to the procedure should be N and vectors containing the coefficients a_j, b_j, c_j, f_j, $j = 1, 2, \ldots, N$. The procedure should output the solution $\mathbf{X}$. The coefficients a_j, b_j, *etc.*, $j = 1, 2, \ldots, N$, should be replaced by u_j, v_j, *etc.*, $j = 1, 2, \ldots, N$, in order to save storage. If you want, the solution $\mathbf{X}$ can be returned in $\mathbf{F}$.

2.4. Estimate the number of arithmetic operations necessary to factor $\mathbf{A}$ and for the forward and backward substitution process.

3. Consider the linear boundary value problem

$$-pu'' + qu = f(x), \qquad 0 < x < 1, \qquad u(0) = u'(1) = 0.$$

where p and q are positive constants and $f(x)$ is a smooth function.

3.1. Show that the Galerkin form of this boundary-value problem consists of finding $u \in H_0^1$ satisfying

$$A(v, u) - (v, f) = \int_0^1 (v' p u' + v q u) dx - \int_0^1 v f dx = 0, \qquad \forall v \in H_0^1.$$

For this problem, functions $u(x) \in H_0^1$ are required to be elements of H^1 and satisfy the Dirichlet boundary condition $u(0) = 0$. The Neumann boundary condition at $x = 1$ need not be satisfied by either u or v.

3.2. Introduce N equally spaced elements on $0 \le x \le 1$ with nodes $x_j = jh$, $j = 0, 1, \ldots, N$ $(h = 1/N)$. Approximate u by U having the form

$$U(x) = \sum_{j=1}^{N} c_k \phi_k(x),$$

where $\phi_j(x)$, $j = 1, 2, \ldots, N$, is the piecewise linear basis (1.3.4), and use Galerkin's method to obtain the global stiffness and mass matrices and the load vector for this problem. (Again, the approximation $U(x)$ does not satisfy the *natural boundary condition* $u'(1) = 0$ nor does it have to. We will discuss this issue in Chapter 2.)

3.3. Write a program to solve this problem using the finite element method developed in Part 3.2b and the tridiagonal algorithm of Problem 2. Execute your program with $p = 1$, $q = 1$, and $f(x) = x$ and $f(x) = x^2$. In each case, use $N = 4$, 8, 16, and 32. Let $e(x) = u(x) - U(x)$ and, for each value of N, compute $|e|_\infty$, $|e'(x_N)|$, and $\|e\|_A$ according to (1.3.22) and (1.3.23a). You may (optionally) also compute $\|e\|_0$ as defined by (1.3.23c). In each case, estimate the rate of convergence of the finite element solution to the exact solution.

4. The Galerkin form of (1.3.1) consists of determining $u \in H_0^1$ such that (1.3.2) is satisfied. Similarly, the finite element solution $U \in S_0^N \subset H_0^1$ satisfies (1.2.12). Letting $e(x) = u(x) - U(x)$, show

$$A(e, e) = A(u, u) - A(U, U)$$

where the strain energy $A(v, u)$ is given by (1.3.2c). We have, thus, shown that the strain energy of the error is the error of the strain energy.

Bibliography

[1] I. Babuška, J. Chandra, and J.E. Flaherty, editors. *Adaptive Computational Methods for Partial Differential Equations*, Philadelphia, 1983. SIAM.

[2] I. Babuška, O.C. Zienkiewicz, J. Gago, and E.R. de A. Oliveira, editors. *Accuracy Estimates and Adaptive Refinements in Finite Element Computations*. John Wiley and Sons, Chichester, 1986.

[3] M.W. Bern, J.E. Flaherty, and M. Luskin, editors. *Grid Generation and Adaptive Algorithms*, volume 113 of *The IMA Volumes in Mathematics and its Applications*, New York, 1999. Springer.

[4] G.F. Carey. *Computational Grids: Generation, Adaptation, and Solution Strategies*. Series in Computational and Physical Processes in Mechanics and Thermal science. Taylor and Francis, New York, 1997.

[5] K. Clark, J.E. Flaherty, and M.S. Shephard, editors. *Applied Numerical Mathematics*, volume 14, 1994. Special Issue on Adaptive Methods for Partial Differential Equations.

[6] R. Courant. Variational methods for the solution of problems of equilibrium and vibrations. *Bulletin of the American Mathematics Society*, 49:1–23, 1943.

[7] J.E. Flaherty, P.J. Paslow, M.S. Shephard, and J.D. Vasilakis, editors. *Adaptive methods for Partial Differential Equations*, Philadelphia, 1989. SIAM.

[8] A. Hrennikoff. Solutions of problems in elasticity by the frame work method. *Journal of Applied Mechanics*, 8:169–175, 1941.

[9] C. Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element method*. Cambridge, Cambridge, 1987.

[10] D.L. Logan. *A First Course in the Finite Element Method using ALGOR*. PWS, Boston, 1997.

[11] D. McHenry. A lattice analogy for the solution of plane stress problems. *Journal of the Institute of Civil Engineers*, 21:59–82, 1943.

[12] J.C. Strikwerda. *Finite Difference Schemes and Partial Differential Equations.* Chapman and Hall, Pacific Grove, 1989.

[13] M.J. Turner, R.W. Clough, H.C. Martin, and L.J. Topp. Stiffness and deflection analysis of complex structures. *Journal of the Aeronautical Sciences*, 23:805–824, 1956.

[14] R. Verfürth. *A Review of Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques.* Teubner-Wiley, Stuttgart, 1996.