9

Finite-Element Analysis

This chapter remedies the major problem when a partial-differential equation is solved through replacing the derivatives with finite-difference quotients. In that technique, nodes must be in rectangular arrays. In finite-element analysis (often abbreviated FEA), the topic of this chapter, nodes can be spaced in any desired orientation so that a region of any shape can be accommodated. The method is also called the finite-element method (FEM).

In particular, curved boundaries can be approximated by closely spaced nodes. It is not difficult to place modes closer together in subregions where the function is changing rapidly, thus improving the accuracy. A program to carry out FEA is not as simple as for the finite-difference method but software is available to define the region, set up the equations for all types of boundary conditions, and then get the solution. We will describe one of these programs, that from MATLAB in its PDE Toolbox.* This program is most user-friendly—a graphical user interface even lets the user draw a 2-D region on the computer screen.

The basis of FEA is to break up the region of interest into small subregions, the elements. With a 2-D region, elements can be triangles (the most common) or rectangles, even "triangles" or "rectangles" with curved sides. In 3-D, they may be pyramids or bricks. Once the region and its elements are defined, the equations for the system are set up and solved. The equation must, of course, incorporate the boundary conditions, which can be of any type.

The problems that can be solved with FEA include all three types of partial-differential equations, and other problems such as eigenvalue problems, which we do not discuss.

In this chapter, we develop the background for finite elements from a branch of mathematics called the *calculus of variations,* which offers three solution methods that do not use finite elements.

---

* This toolbox is not a part of the student edition.

Contents of This Chapter

9.1 Mathematical Background

Gives a description of three methods: the Rayleigh–Ritz method, the collocation method, and the Galerkin method. The first of these optimizes a so-called *functional* to get the solution to a boundary-value problem. The other two methods also solve such problems and are more directly used in establishing the equations for the finite-element method in later sections

9.2 Finite Elements for Ordinary-Differential Equations (ODE)

Applies the Galerkin method to the elements of the region to arrive at a system of linear equations whose solution is an approximation to the solution of an ordinary-differential equation. Several steps are used in the development. Any type of boundary values can be accommodated.

9.3 Finite Elements for Partial-Differential Equations

Uses a different approach to setting up the system of equations for the finite element solution. The development is made for all three type of PDEs: elliptic, parabolic, and hyperbolic. Simple regions are used to illustrate the method. Examples with a more complex region are solved with MATLAB's Toolbox.

9.1 Mathematical Background

Finite-element analysis is based on some elegant mathematics. We begin the discussion with the *Rayleigh–Ritz method* for solving boundary-value problems. The method comes from that part of mathematics called the *calculus of variations.*

In the Rayleigh–Ritz method, we solve a boundary-value problem by approximating the solution with a finite linear combination of *basis* functions. (We define basis functions and the requirements that are placed on them a little later.) In the calculus of variations, we seek to minimize a special class of functions called *functionals*. The usual form for a functional in problems with one independent variable is

$$I[y] = \int_a^b F\left(x, y, \frac{dy}{dx}\right) dx. \tag{9.1}$$

Observe that $I[y]$ is not a function of x because x disappears when the definite integral is evaluated. The argument y of $I[y]$ is not a simple variable but a function, $y = y(x)$. The square brackets in $I[y]$ emphasize this fact. A functional can be thought of as a "function of functions." The value of the right-hand side of Eq. (9.1) will change as the function $y(x)$ is varied, but when $y(x)$ is fixed, it evaluates to a scalar quantity (a constant). We seek the $y(x)$ that minimizes $I[y]$.

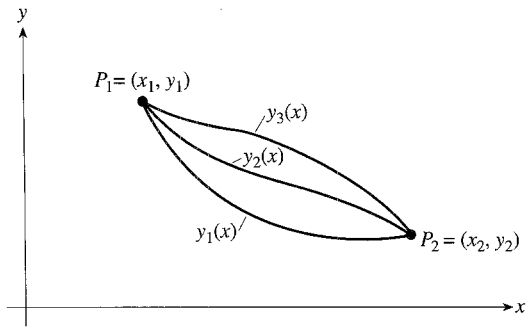


Figure 9.1

Let us illustrate this concept by a very simple example where the solution is obvious in advance—find the function $y(x)$ that minimizes the distance between two points. Although we know what $y(x)$ must be, let's pretend we don't. Figure 9.1 suggests that we are to choose from among the set of curves $y_i(x)$ of which $y_1(x)$, $y_2(x)$, and $y_3(x)$ are representative. In this simple case, the functional is the integral of the distance along any of these curves:

$$I[y] = \int_{x_1}^{x_2} \sqrt{(dx)^2 + (dy)^2} = \int_{x_1}^{x_2} \sqrt{1 + \left(\frac{dy}{dx}\right)^2}\, dx.$$

To minimize $I[y]$, just as in calculus, we set its derivative to zero. There are certain restrictions on all the curves $y_i(x)$. Obviously, each must pass through the points (x_1, y_1) and (x_2, y_2). In addition, for the optimal trajectory, the Euler–Lagrange equation must be satisfied:

$$\frac{d}{dx}\left[\frac{\partial}{\partial y'} F(x, y, y')\right] = \frac{\partial}{\partial y} F(x, y, y'). \tag{9.2}$$

Applying this to the functional for shortest distance, we have

$$F(x, y, y') = (1 + (y')^2)^{1/2},$$

$$\frac{\partial F}{\partial y} = 0,$$

$$\frac{\partial F}{\partial y'} = \frac{1}{2}(1 + (y')^2)^{-1/2}(2y'),$$

$$\frac{d}{dx}\frac{\partial F}{\partial y'} = \frac{d}{dx}\left(\frac{y'}{\sqrt{1 + (y')^2}}\right) = \frac{\partial F}{\partial y} = 0.$$

[The last comes from Eq. (9.2).]

From this, it follows that

$$\frac{y'}{\sqrt{1 + (y')^2}} = c.$$

Solving for y' gives

$$y' = \sqrt{\frac{c^2}{1 - c^2}} = \text{a constant} = b,$$

and, on integrating,

$$y = bx + a.$$

As stated, $y(x)$ must pass through P_1 and P_2; this condition is used to evaluate the constants a and b.

Let us advance to a less trivial case. Consider this second-order linear boundary-value problem over $[a, b]$:*

$$y'' + Q(x)y = F(x), \qquad y(a) = y_0, \qquad y(b) = y_n. \tag{9.3}$$

(An equation that has y = constant at the endpoints is said to be subject to Dirichlet conditions.) It turns out that the functional that corresponds to Eq. (9.3) is

$$I[u] = \int_a^b \left[\left(\frac{du}{dx} \right)^2 - Qu^2 + 2Fu \right] dx. \tag{9.4}$$

(If the boundary equations involve a derivative of y, the functional must be modified.)

We can transform Eq. (9.4) to Eq. (9.3) through the Euler–Lagrange conditions, so optimizing Eq. (9.4) gives the solution to Eq. (9.3). Observe carefully the benefit of operating with the functional rather than the original equation: We now have only first-order instead of second-order derivatives. This not only simplifies the mathematics but also permits us to find solutions even when there are discontinuities that cause y not to have sufficiently high derivatives.

If we know the solution to our differential equation, substituting it for u in Eq. (9.4) will make $I[u]$ a minimum. If the solution isn't known, perhaps we can approximate it by some (almost) arbitrary function and see whether we can minimize the functional by a suitable choice of the parameters of the approximation. The Rayleigh–Ritz method is based on this idea. We let $u(x)$, which is the approximation to $y(x)$ (the exact solution), be a sum:

$$u(x) = c_0 v_0(x) + c_1 v_1(x) + \cdots + c_n v_n(x) = \sum_{i=0}^{n} c_i v_i(x). \tag{9.5}$$

There are two conditions on the v's in Eq. (9.5): They must be chosen such that $u(x)$ meets the boundary conditions, and the individual v's must be linearly independent (meaning that no one v can be obtained by a linear combination of the others). We call the v's *trial functions;* the c's and v's are to be chosen to make $u(x)$ a good approximation to the true solution to Eq. (9.3).

If we have some prior knowledge of the true function, $y(x)$, we may be able to choose the v's to closely resemble $y(x)$. Most often we lack such knowledge, and the usual choice then is to use polynomials. We must find a way of getting values for the c's to force $u(x)$ to be close to $y(x)$. We will use the functional of Eq. (9.4) to do this.

---

* This equation is a prototype of many equations in applied mathematics. Equations for heat conduction, elasticity, electrostatics, and so on in a one-dimensional situation are of this form.

If we substitute $u(x)$ as defined by Eq. (9.5) into the functional, Eq. (9.4), we get

$$I(c_0, c_1, \ldots, c_n) = \int_a^b \left[\left(\frac{d}{dx} \Sigma c_i v_i \right)^2 - Q(\Sigma c_i v_i)^2 + 2F \Sigma c_i v_i \right] dx. \tag{9.6}$$

We observe that I is an ordinary function of the unknown c's after this substitution, as reflected in our notation. To minimize I, we take its partial derivatives with respect to each unknown c and set to zero, resulting in a set of equations in the c's that we can solve. This will define $u(x)$ in Eq. (9.5).

We now substitute the $u(x)$ of Eq. (9.5) into the functional. If we partially differentiate with respect to, say, c_i where this is one of the unknown c's, we will get

$$\frac{\partial I}{\partial c_i} = \int_a^b 2 \left(\frac{du}{dx} \right) \frac{\partial}{\partial c_i} \left(\frac{du}{dx} \right) dx - \int_a^b 2Qu \left(\frac{\partial u}{\partial c_i} \right) dx + \int_a^b 2F \left(\frac{\partial u}{\partial c_i} \right) dx, \tag{9.7}$$

where we have broken the integral into three parts.

An example will clarify the procedure.

EXAMPLE 9.1 Solve the equation $y'' + y = 3x^2$, with boundary points (0, 0) and (2, 3.5). (Here $Q = 1$ and $F = 3x^2$.) Use polynomial trial functions up to degree 3. If we define $u(x)$ as

$$u(x) = \frac{7x}{4} + c_2(x)(x - 2) + c_3(x^2)(x - 2), \tag{9.8}$$

we have linearly independent v's. The boundary conditions are met by the first term, and because the other terms are zero at the boundaries, $u(x)$ also meets the boundary conditions. [It is customary to match the boundary conditions with the initial term(s) of $u(x)$ and then make the succeeding terms equal zero at the boundaries, as we have done here.]

Examination of Eq. (9.7) shows that we need these quantities:

$$\begin{aligned} \frac{du}{dx} &= \frac{7}{4} + c_2(2x - 2) + c_3(3x^2 - 4x), \\ \frac{\partial}{\partial c_2}\left(\frac{du}{dx} \right) &= 2x - 2, \qquad \frac{\partial}{\partial c_3}\left(\frac{du}{dx} \right) = 3x^2 - 4x, \\ \frac{\partial u}{\partial c_2} &= x(x - 2), \qquad \frac{\partial u}{\partial c_3} = x^2(x - 2). \end{aligned} \tag{9.9}$$

We now substitute from Eq. (9.9) into Eq. (9.7). Note that we have two equations, one for the partial with respect to c_2 and the other from the partial with respect to c_3. The results from this step are:

$$\begin{aligned} \frac{\partial I}{\partial c_2}: \quad 0 = &\int_0^2 2 \left[\frac{7}{4} + c_2(2x - 2) + c_3(3x^2 - 4x) \right] (2x - 2)\, dx \\ &- \int_0^2 2(1) \left[\frac{7x}{4} + c_2(x^2 - 2x) + c_3(x^3 - 2x^2) \right] (x^2 - 2x)\, dx \\ &+ 2 \int_0^2 (3x^2)(x^2 - 2x)\, dx, \end{aligned} \tag{9.10}$$

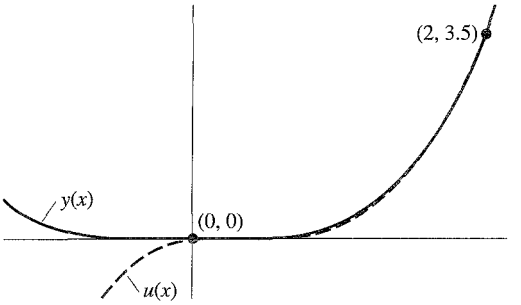


Figure 9.2

$$\frac{\partial I}{\partial c_3}: \quad 0 = \int_0^2 2\left[\frac{7}{4} + c_2(2x - 2) + c_3(3x^2 - 4x)\right](3x^2 - 4x)\,dx$$
$$- \int_0^2 2(1)\left[\frac{7x}{4} + c_2(x^2 - 2x) + c_3(x^3 - 2x^2)\right](x^3 - 2x^2)\,dx \quad (9.11)$$
$$+ 2\int_0^2 (3x^2)(x^3 - 2x^2)\,dx.$$

We now carry out the integrations. Although there are quite a few of them, all are quite simple in our example. With a more complicated $Q(x)$ and $F(x)$, this might require numerical integrations. The result of this step is the pair of equations

$$\frac{16}{5}c_2 + \frac{16}{5}c_3 = \frac{74}{15}, \quad (9.12)$$
$$\frac{16}{5}c_2 + \frac{128}{21}c_3 = \frac{36}{5},$$

which we solve to get the coefficients in our $u(x)$. On expanding, we find that

$$u(x) = \left(\frac{119}{152}\right)x^3 - \left(\frac{46}{57}\right)x^2 + \left(\frac{53}{228}\right)x. \quad (9.13)$$

Figure 9.2 shows that our $u(x)$ agrees well with the exact solution, which is $6\cos(x) + 3(x^2 - 2)$, over the interval [0, 2]. Table 9.1 compares computed values and the error of $u(x)$. The largest errors occur near the middle of the interval.

The Collocation Method

There are other ways to approximate $y(x)$ in Example 9.1. The *collocation method* is what is called a "residual method." We begin by defining the residual, $R(x)$, as equal to the left-hand side of Eq. (9.3) minus the right-hand side:

$$R(x) = y'' + Qy - F. \quad (9.14)$$

Table 9.1

x	$y(x)$	$u(x)$	Error	x	$y(x)$	$u(x)$	Error
0.00	0.000	0.000	0.000	1.10	0.352	0.321	0.031
0.10	0.000	0.016	−0.016	1.20	0.494	0.470	0.024
0.20	0.000	0.020	−0.020	1.30	0.675	0.658	0.017
0.30	0.002	0.018	−0.016	1.40	0.900	0.892	0.008
0.40	0.006	0.014	−0.008	1.50	1.174	1.175	−0.001
0.50	0.015	0.012	−0.003	1.60	1.505	1.513	−0.008
0.60	0.032	0.018	0.014	1.70	1.897	1.909	−0.012
0.70	0.059	0.036	0.023	1.80	2.357	2.370	−0.013
0.80	0.100	0.070	0.030	1.90	2.890	2.898	−0.008
0.90	0.160	0.126	0.034	2.00	3.500	3.500	0.000
1.00	0.242	0.208	0.034				

We approximate $y(x)$ again with $u(x)$ equal to a sum of trial functions, usually chosen as linearly independent polynomials, just as for the Rayleigh–Ritz method. We substitute $u(x)$ into $R(x)$ and attempt to make $R(x) = 0$ by a suitable choice of the coefficients in $u(x)$. Of course, normally we cannot do this everywhere in the interval $[a, b]$, so we select several points at which we make $R(x) = 0$. [The number of points where we do this must equal the number of unknown coefficients in $u(x)$.] An example will clarify the procedure.

EXAMPLE 9.2 Solve the same equation as in Example 9.1, but this time use collocation.

The equation we are to solve is

$$y'' + y = 3x^2, \qquad y(0) = 0, \qquad y(2) = 3.5. \tag{9.15}$$

We take $u(x)$ as before to satisfy the boundary conditions:

$$u(x) = \frac{7x}{4} + c_2(x)(x - 2) + c_3(x^2)(x - 2). \tag{9.16}$$

The residual is, after substituting $u(x)$ for $y(x)$.

$$R(x) = u'' + u - 3x^2, \tag{9.17}$$

which becomes, when we differentiate u twice to get u'',

$$R(x) = c_2(2) + c_3(6x - 4) + \frac{7x}{4} + c_2(x^2 - 2x) + c_3(x^3 - 2x^2) - 3x^2. \tag{9.18}$$

Because there are two unknown constants, we can force $R(x)$ to be zero at two points in $[0, 2]$. We do not know which two points will be the best choices, so we arbitrarily take them as $x = 0.7$ and $x = 1.3$. (These points are more or less equally spaced in the interval.)

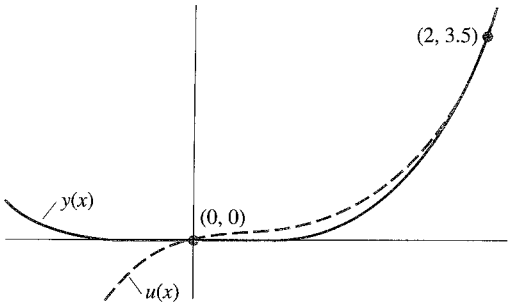


Figure 9.3

Setting $R(x) = 0$ for these choices gives a pair of equations in the c's:

$$\begin{aligned} \text{From } x = 0.7: \quad & \frac{1090c_2 - 437c_3 - 245}{1000} = 0, \\ \text{From } x = 1.3: \quad & \frac{1090c^2 + 2617c_3 - 2795}{1000} = 0. \end{aligned} \tag{9.19}$$

When these are solved for the c's, we get, for $u(x)$,

$$u(x) = \left(\frac{425}{509}\right)x^3 - \left(\frac{61607}{55481}\right)x^2 + \left(\frac{140023}{221924}\right)x, \tag{9.20}$$

in which the coefficients are quite different than in Eq. (9.13). Figure 9.3 shows that this approximation is not as good as that obtained by the Rayleigh–Ritz technique. (But the amount of arithmetic is certainly less! We could improve the approximation by using more terms in $u(x)$.) Table 9.2 compares the approximation with the exact solution.

Table 9.2

x	$y(x)$	$u(x)$	Error	x	$y(x)$	$u(x)$	Error
0.00	0.00	0.00	0.00	1.10	0.35	0.46	−0.11
0.10	0.00	0.05	−0.05	1.20	0.49	0.60	−0.11
0.20	0.00	0.09	−0.09	1.30	0.67	0.78	−0.10
0.30	0.00	0.11	−0.11	1.40	0.90	1.00	−0.10
0.40	0.01	0.13	−0.12	1.50	1.17	1.27	−0.09
0.50	0.02	0.14	−0.13	1.60	1.50	1.59	−0.08
0.60	0.03	0.16	−0.13	1.70	1.90	1.97	−0.07
0.70	0.06	0.18	−0.12	1.80	2.36	2.41	−0.05
0.80	0.10	0.22	−0.12	1.90	2.89	2.92	−0.03
0.90	0.16	0.28	−0.12	2.00	3.50	3.50	0.00
1.00	0.24	0.36	−0.11				

The Galerkin Method

The Galerkin method is widely used, especially in the very popular technique that we will describe in Section 9.2. It is important to know the Galerkin method because of its widespread application.

Like collocation, Galerkin is a "residual method" that uses the $R(x)$ of Eq. (9.14), except that now we multiply $R(x)$ by weighting functions, $W_i(x)$. The $W_i(x)$ can be chosen in many ways, but Galerkin showed that using the individual trial functions, v_i, of Eq. (9.5) is an especially good choice.

Once we have selected the v's for Eq. (9.5), we compute the unknown coefficients by setting the integral over $[a, b]$ of the weighted residual to zero:

$$\int_a^b W_i(x)R(x)\,dx = 0, \qquad i = 0, 1, \ldots, n, \tag{9.21}$$

where $W_i(x) = v_i$. (Observe that using Dirac delta functions for the $W_i(x)$ gives the collocation method.)

Let us use the Galerkin method on the same example as before.

EXAMPLE 9.3 Solve

$$y'' + y = 3x^2, \qquad y(0) = 0, \qquad y(2) = 3.5$$

by the Galerkin method. Use the same $u(x)$ as before:

$$u(x) = \frac{7x}{4} + c_2(x)(x - 2) + c_3(x^2)(x - 2),$$

so that $v_2 = x(x - 2)$ and $v_3 = x^2(x - 2)$.

The residual is

$$R(x) = y'' + y - 3x^2,$$

which becomes, after substituting u'' and u for y'' and y, respectively,

$$R(x) = c_2(2) + c_3(6x - 4) + \frac{7x}{4} + c_2(x)(x - 2) + c_3(x^2)(x - 2) - 3x^2. \tag{9.22}$$

We now carry out two integrations (because there are two unknown c's):

$$\text{Using } v_2 \text{ as a } W_i\text{:} \qquad \int_0^2 [x(x - 2)] * R(x)\,dx = 0,$$

$$\text{Using } v_3 \text{ as a } W_i\text{:} \qquad \int_0^2 [x^2(x - 2)] * R(x)\,dx = 0,$$

which gives two equations in the c's:

$$\begin{aligned} -\frac{24c_2 + 24c_3 - 37}{15} &= 0, \\ -\frac{2(84c_2 + 160c_3 - 180)}{105} &= 0. \end{aligned} \tag{9.23}$$

Solving Eqs. (9.23) for c_2 and c_3 gives

$$u(x) = \left(\frac{101}{152}\right)x^3 - \left(\frac{103}{228}\right)x^2 - \left(\frac{1}{128}\right)x. \tag{9.24}$$

Although Eq. (9.24) looks different from Eq. (9.13), between $x = 0$ and $x = 2$ it gives values for $u(x)$ that have errors about twice those from the Rayleigh–Ritz technique. Equation (9.24) differs from the analytical solution by little more than does the Rayleigh–Ritz equation. [The maximum error of Eq. (9.24) is 0.058; for Eq. (9.13), it is 0.034.]

Although the Rayleigh–Ritz method is slightly more accurate in this example, the Galerkin method is much easier and we never have to find the variational form.

9.2 Finite Elements for Ordinary-Differential Equations

The disadvantages of the methods of the previous section are twofold: Finding good trial functions [the v's in Eq. (9.5)] is not easy, and polynomials [the usual choice when we have no prior knowledge of the behavior of $y(x)$] may interpolate poorly. [We can think of $u(x)$ as an interpolation function between the boundary conditions that also obeys the differential equation.] This is especially true when the interval $[a, b]$ is large.

The remedy to this problem is based on the observation in Chapters 3 and 5 that a function can be approximated by even low-degree polynomials if the polynomial fits the function at values that are closely spaced. We then hope that we can get the solution to a boundary-value problem by applying the Galerkin method to subintervals of $[a, b]$, the boundaries of the equation. It turns out that our hope is fulfilled.

The method that we now describe is called *finite-element analysis (FEA),* also called the *finite-element method (FEM).* The strategy is as follows:

1. Subdivide $[a, b]$ into n subintervals, called *elements,* that join at $x_1, x_2, \ldots, x_{n-1}$. Add to this array $x_0 = a$ and $x_n = b$. We call the x_i the *nodes* of the interval. Number the elements from 1 to n where element (i) runs from x_{i-1} to x_i. The x_i need not be evenly spaced.
2. Apply the Galerkin method to each element separately to interpolate (subject to the differential equation) between the end nodal values, $u(x_{i-1})$ and $u(x_i)$, where these u's are approximations to the $y(x_i)$'s that are the true solution to the differential equation. [These nodal values are actually the c's in our adaptation of Eq. (9.5), the equation for $u(x)$.]
3. Use a low-degree polynomial for $u(x)$. Our development will use a first-degree polynomial, although quadratics or cubics are often used. (The development for these higher-degree polynomials parallels what we will do but is more complicated.)
4. The result of applying Galerkin to element (i) is a pair of equations in which the unknowns are the nodal values at the ends of element (i), the c's. When we have done this for each element, we have equations that involve all the nodal values, which we

combine to give a set of equations that we can solve for the unknown nodal values. (The process of combining the separate *element equations* is called *assembling the system.*)

5. These equations are adjusted for the boundary conditions and solved to get approximations to $y(x)$ at the nodes; we get intermediate values for $y(x)$ by linear interpolation.

We now begin the development. Although it involves several steps, each step is straightforward. The differential equation that we will solve is

$$y'' + Q(x)y = F(x) \qquad \text{subject to boundary conditions at } x = a \text{ and } x = b. \tag{9.25}$$

(We will specify the boundary conditions later.)

Step 1 Subdivide $[a, b]$ into n elements, as discussed. Focus attention on element (i) that runs between x_{i-1} and x_i. To simplify the notation, call the left node L and the right node R.

Step 2 Write $u(x)$ for element (i):

$$\begin{aligned} u(x) &= c_L N_L + c_R N_R = c_L \frac{x - R}{L - R} + c_R \frac{x - L}{R - L} \\ &= c_L \frac{x - R}{-h_i} + c_R \frac{x - L}{h_i}, \end{aligned} \tag{9.26}$$

Recognize that the N's in Eq. (9.26) are really first-degree Lagrangian polynomials. When we use such linear interpolation, the shape functions are often called *hat functions.* (*Chapeau functions,* from the French, is another name.) The reason for this name will become apparent.

Figure 9.4 sketches N_L and N_R within element (i). Because the values of the N's vary (from unity to zero) as x goes from x_L to x_R, they are functions of x. Note also that the c's in Eq. (9.26) are independent of x.

The reason that our N's are called "hat functions" is clear when we look at a sketch of the N's for several adjacent elements in Figure 9.5. Observe that we combine the $N_R(x)$ and $N_L(x)$ of Figure 9.4 that join at x_i into a quantity that we call N_i.

Step 3 Apply the Galerkin method to element (i). The residual is

$$R(x) = y'' + Qy - F = u'' + Qu - F, \tag{9.27}$$

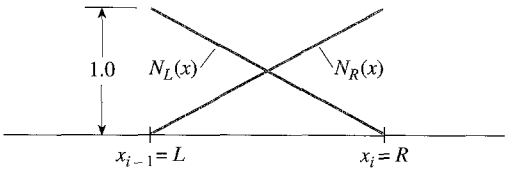


Figure 9.4
N_L and N_R within element (i)

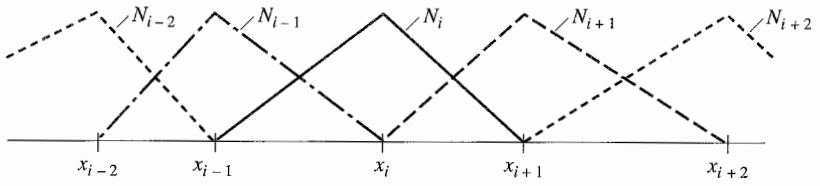


Figure 9.5

where we have substituted $u(x)$ for $y(x)$. The Galerkin method sets the integral of R weighted with each of the N's (over the length of the element) to zero:

$$\int_L^R N_L R(x)\,dx = 0,$$
$$\int_L^R N_R R(x)\,dx = 0. \tag{9.28}$$

Now expand Eq. (9.28):

$$\int_L^R u'' N_L\,dx + \int_L^R Q u N_L\,dx - \int_L^R F N_L\,dx = 0. \tag{9.29}$$

$$\int_L^R u'' N_R\,dx + \int_L^R Q u N_R\,dx - \int_L^R F N_R\,dx = 0. \tag{9.30}$$

Step 4 Transform Eqs. (9.29) and (9.30) by applying integration by parts* to the first integral. In the second integral, we will take Q out from the integrand as Q_{av}, an average value within the element. We also take F outside the third integral. When this is done, Eq. (9.29) becomes

$$-\int_L^R \left(\frac{du}{dx}\right)\left(\frac{dN_L}{dx}\right) dx + Q_{\text{av}}\int_L^R u N_L\,dx - F_{\text{av}}\int_L^R N_L\,dx + N_L\frac{du}{dx}\bigg|_{x=R} - N_L\frac{du}{dx}\bigg|_{x=L} = 0. \tag{9.31}$$

In the last two terms of Eq. (9.31), $N_L = 1$ at L and is zero at R, so the equation can be simplified:

$$-\int_L^R \left(\frac{du}{dx}\right)\left(\frac{dN_L}{dx}\right) dx + Q_{\text{av}}\int_L^R u N_L\,dx - F_{\text{av}}\int_L^R N_L\,dx - \frac{du}{dx}\bigg|_{x=L} = 0. \tag{9.32}$$

Doing similarly with Eq. (9.30) gives

$$-\int_L^R \left(\frac{du}{dx}\right)\left(\frac{dN_R}{dx}\right) dx + Q_{\text{av}}\int_L^R u N_R\,dx - F_{\text{av}}\int_L^R N_R\,dx + \frac{du}{dx}\bigg|_{x=R} = 0. \tag{9.33}$$

---

* From $d(UV) = U\,dV + V\,dU$, we have

$$\int_a^b U\,dV = -\int_a^b V\,dU + UV\Big|_a^b, \quad \text{so} \quad \int_L^R u'' N_i\,dx = -\int_L^R u'(dN_i/dx) + N_i u'\Big|_L^R.$$

Step 5 Change signs in Eqs. (9.32) and (9.33); substitute from Eq. (9.26) for u, du/dx, dN_L/dx, and dN_R/dx; and carry out the integrations. We show this separately for each term in Eq. (9.32):

$$\int_L^R \left(\frac{du}{dx}\right)\left(\frac{dN_L}{dx}\right) dx = \int_L^R \left[\frac{-c_L}{h_i} + \frac{c_R}{h_i}\right]\left(\frac{-1}{h_i}\right) dx = \left(\frac{c_L}{h_i^2} - \frac{c_R}{h_i^2}\right)\int_L^R dx \tag{9.34a}$$
$$= \left(\frac{1}{h_i}\right) c_L - \left(\frac{1}{h_i}\right) c_R.$$

$$\begin{aligned} -Q_{\text{av}}\int_L^R (c_L N_L + c_R N_R) N_L \, dx &= -c_L Q_{\text{av}} \int_L^R N_L^2 \, dx - c_R Q_{\text{av}} \int_L^R N_R N_L \, dx \\ &= -c_L Q_{\text{av}} \int_L^R \left(\frac{x - R}{h_i}\right)^2 dx \\ &\quad -c_R Q_{\text{av}} \int_L^R \left(\frac{x - L}{h_i}\right)\left(\frac{x - R}{-h_i}\right) dx \\ &= -\left(Q_{\text{av}} * \frac{h_i}{3}\right) c_L - \left(Q_{\text{av}} * \frac{h_i}{6}\right) c_R. \end{aligned} \tag{9.34b}$$

$$F_{\text{av}} \int_L^R N_L \, dx = F_{\text{av}} \int_L^R \frac{(x - R)}{-h_i} \, dx = F_{\text{av}} * \frac{h_i}{2}. \tag{9.34c}$$

Doing the same with Eq. (9.33) gives

$$\int_L^R \left(\frac{du}{dx}\right)\left(\frac{dN_R}{dx}\right) dx = -\left(\frac{1}{h_i}\right) c_L + \left(\frac{1}{h_i}\right) c_R. \tag{9.35a}$$

$$-Q_{\text{av}} \int_L^R (c_L N_L + c_R N_R) N_R \, dx = -\left(Q_{\text{av}} * \frac{h_i}{6}\right) c_L - \left(Q_{\text{av}} * \frac{h_i}{3}\right) c_R. \tag{9.35b}$$

$$F_{\text{av}} \int_L^R N_R \, dx = F_{\text{av}} * \frac{h_i}{2}. \tag{9.35c}$$

Step 6 Substitute the result of step 5 [Eqs. (9.34) and (9.35) into Eqs. (9.32) and (9.33), and rearrange to give two linear equations in the unknown c_L and c_R:

$$\begin{aligned} \left(\frac{1}{h_i} - \frac{Q_{\text{av}} h_i}{3}\right) c_L + \left(\frac{-1}{h_i} - \frac{Q_{\text{av}} h_i}{6}\right) c_R &= \frac{-F_{\text{av}} h_i}{2} - \left.\frac{du}{dx}\right|_{x=L}, \\ \left(\frac{-1}{h_i} - \frac{Q_{\text{av}} h_i}{6}\right) c_L + \left(\frac{1}{h_i} - \frac{Q_{\text{av}} h_i}{3}\right) c_R &= \frac{-F_{\text{av}} h_i}{2} + \left.\frac{du}{dx}\right|_{x=R}. \end{aligned} \tag{9.36}$$

We call the pair of equations in (9.36) the *element equations*. We can do the same for each element to get n such pairs.

Step 7 Combine (assemble) all the element equations together to form a system of linear equations for the problem. We now recognize that point R in element (i) is precisely the same as point L in element $(i + 1)$. Renumber the c's as $c_0, c_1, \ldots, c_n$. Also notice that the

gradient (du/dx) must be the same on either side of the join of the elements—that is, $(du/dx)_{x=R}$ in element (i) equals $(du/dx)_{x=L}$ in element ($i + 1$). This means that these terms cancel when we do the assembling except in the first and last equations. (On rare occasions this is not true, but in that case the difference in the two gradients is a known value.)

The result of this step is this set of $n + 1$ equations (numbered from 0 to n),

$$[K]\{c\} = \{b\}, \tag{9.37}$$

where the diagonal elements of $[K]$ are

$$\left(\frac{1}{h_1} - Q_{\text{av},1} * \frac{h_1}{3}\right) \quad \text{in row 0,}$$

$$\left(\frac{1}{h_i} - Q_{\text{av},i} * \frac{h_i}{3}\right) + \left(\frac{1}{h_{i+1}} - Q_{\text{av},i+1} * \frac{h_{i+1}}{3}\right) \quad \text{in rows 1 to } n-1,$$

$$\left(\frac{1}{h_n} - Q_{\text{av},n} * \frac{h_n}{3}\right) \quad \text{in row } n;$$

and elements above and to the left of the diagonal in rows 1 to n are

$$\left(\frac{-1}{h_i} - Q_{\text{av},i} * \frac{h_i}{6}\right).$$

The elements of $\{c\}$ are c_i, $i = 0$ to n.

The elements of $\{b\}$ are

$$-F_{\text{av},1} * \frac{h_1}{2} - \left(\frac{du}{dx}\right)_{x=a} \quad \text{in row 0,}$$

$$-F_{\text{av},i} * \frac{h_i}{2} - F_{\text{av},i+1} * \frac{h_{i+1}}{2} \quad \text{in rows 1 to } n-1,$$

$$-F_{\text{av},n} * \frac{h_n}{2} + \left(\frac{du}{dx}\right)_{x=b} \quad \text{in row } n.$$

In the preceding equations, $Q_{\text{av},i}$ and $F_{\text{av},i}$ are values of Q and F at the midpoints of element (i).

Step 8 Adjust the set of equations from step 6 for the boundary conditions. We will handle two cases: Case (1), a Dirichlet condition is specified—$y(a)$ = constant [and/or $y(b)$ = constant]. Case (2), a Neumann condition is specified—dy/dx = constant at $x = a$ and/or $x = b$. (If $Q = 0$, we cannot have a Neumann condition at both ends, because the solution would be known only to within an additive constant.) [We leave case (3), mixed conditions, as an exercise; it is a modification of case (2).]

Case (1): Dirichlet condition. In this case, c is known at the end node. Suppose this is $y(a) = A$. Then the equation in row 0 is redundant, and so we remove it from the set of equations of step 6. In the next row, we move $k_{10} * A$ to the right-hand side (subtracting this from the element computed in step 6). If the condition is $y(b) = B$, we do the same but with the last and next to last equations.

Case (2): Neumann condition. In this case, c is not known at the end node. Suppose the condition is $dy/dx = A$ at $x = a$. We retain the equation in row 0 and substitute the

given value of dy/dx into the right-hand side. If the condition is $dy/dx = B$ at $x = b$, we do the same with the last equation.

Step 9 Solve the set of equations for the unknown c's after adjusting, in step 8, for the boundary conditions. The c's are approximations to $y(x)$ at the nodes. If intermediate values of y are needed between the nodes, we obtain them by linear interpolation.

Examples will clarify the procedure.

EXAMPLE 9.4 Solve $y'' + y = 3x^2$, $y(0) = 0$, $y(2) = 3.5$. (We solved this same equation in Section 9.1.) Subdivide into seven elements that join at $x = 0.4, 0.7, 0.9, 1.1, 1.3$, and 1.6.

Table 9.3 shows the values we need to build the system of equations.

The augmented matrix of the set of equations from step 6 is

$$\left[\begin{array}{cccccccc|c}
2.367 & -2.567 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & -0.024 \\
-2.567 & 5.600 & -3.383 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & -0.160 \\
0.000 & -3.383 & 8.167 & -5.033 & 0.000 & 0.000 & 0.000 & 0.000 & -0.328 \\
0.000 & 0.000 & -5.033 & 9.867 & -5.033 & 0.000 & 0.000 & 0.000 & -0.492 \\
0.000 & 0.000 & 0.000 & -5.033 & 9.867 & -5.033 & 0.000 & 0.000 & -0.732 \\
0.000 & 0.000 & 0.000 & 0.000 & -5.033 & 8.167 & -3.383 & 0.000 & -1.378 \\
0.000 & 0.000 & 0.000 & 0.000 & 0.000 & -3.383 & 5.600 & -2.567 & -2.890 \\
0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & -2.567 & 2.367 & -1.944
\end{array}\right] \quad (9.38)$$

To adjust for the boundary conditions, we eliminate the first and last equations and subtract $(0)(-2.567) = 0$ from the right-hand side of the top row and subtract $(3.50)(-2.567) = -8.9845$ from the right-hand side of the bottom row to get

$$\left[\begin{array}{ccccccc|c}
5.600 & -3.383 & 0.000 & 0.000 & 0.000 & 0.000 & -0.160 \\
-3.383 & 8.167 & -5.033 & 0.000 & 0.000 & 0.000 & -0.328 \\
0.000 & -5.033 & 9.867 & -5.033 & 0.000 & 0.000 & -0.492 \\
0.000 & 0.000 & -5.033 & 9.867 & -5.033 & 0.000 & -0.732 \\
0.000 & 0.000 & 0.000 & -5.033 & 8.167 & -3.383 & -1.378 \\
0.000 & 0.000 & 0.000 & 0.000 & -3.383 & 5.600 & 6.094
\end{array}\right] \quad (9.39)$$

We have shown Eqs. (9.38) and (9.39) in their full form but observe that the system is tridiagonal (and symmetric, too). It would have been better to store these as 8×4 arrays, so

Table 9.3

Element	L	R	Midpoint	h_i	Q_{av}	F_{av}
1	0	0.4	0.2	0.4	1	0.12
2	0.4	0.7	0.55	0.3	1	0.9075
3	0.7	0.9	0.8	0.2	1	1.92
4	0.9	1.1	1.0	0.2	1	3
5	1.1	1.3	1.2	0.2	1	4.32
6	1.3	1.6	1.45	0.3	1	6.3075
7	1.6	2.0	1.8	0.4	1	9.72

Table 9.4

x	$u(x)$	Anal.	Error
0.000	0.0000	0.0000	0
0.400	−0.0024	0.0064	0.0088
0.700	0.0433	0.0591	0.0158
0.900	0.1371	0.1597	0.0226
1.100	0.3232	0.3516	0.0284
1.300	0.6419	0.6750	0.0331
1.600	1.4759	1.5048	0.0289
2.000	3.5000	3.5031	0.0031

that Eq. (9.39) would be

$$\left[\begin{array}{cccc} 0 & 5.600 & -3.383 & -0.160 \\ -3.383 & 8.167 & -5.033 & -0.328 \\ -5.033 & 9.867 & -5.033 & -0.492 \\ -5.033 & 9.867 & -5.033 & -0.732 \\ -5.033 & 8.167 & -3.383 & -1.378 \\ -3.383 & 5.600 & 0 & 6.094 \end{array}\right].$$

When the system of Eq. (9.39) is solved, we get the solution as shown in Table 9.4. The table also shows the analytical solution and the errors of our computation. The table indicates that closer spacing of nodes near $x = 1$ would give better answers.

This next example solves a boundary-value problem, which has Neumann conditions at the ends of the region.

EXAMPLE 9.5 Solve $y'' - (x + 1)y = e^{-x}(x^2 - x + 2)$ subject to Neumann conditions of

$$y'(2) = 0, \qquad y'(4) = -0.036631.$$

Use four elements of equal lengths. Compare to the analytical solution

$$y(x) = e^{-x}(x - 1).$$

Table 9.5 gives values that we need to set up the equations.

Table 9.5

Element	L	R	Midpoint	h_i	Q_{av}	F_{av}
1	2	2.5	2.25	0.5	−3.25	−0.5072
2	2.5	3.0	2.75	0.5	−3.75	−0.4355
3	3.0	3.5	3.25	0.5	−4.25	−0.3611
4	3.5	4.0	3.75	0.5	−4.75	−0.2896

The initial matrix of equations is

$$\left[\begin{array}{rrrrr:r} 2.542 & -1.729 & 0.000 & 0.000 & 0.000 & 0.127 \\ -1.729 & 5.167 & -1.688 & 0.000 & 0.000 & 0.236 \\ 0.000 & -1.688 & 5.333 & -1.646 & 0.000 & 0.199 \\ 0.000 & 0.000 & -1.646 & 5.500 & -1.604 & 0.163 \\ 0.000 & 0.000 & 0.000 & -1.604 & 2.792 & 0.072 \end{array}\right].$$

After adjusting for boundary conditions, we get

$$\left[\begin{array}{rrrrr:r} 2.542 & -1.729 & 0.000 & 0.000 & 0.000 & 0.127 \\ -1.729 & 5.167 & -1.688 & 0.000 & 0.000 & 0.236 \\ 0.000 & -1.688 & 5.333 & -1.646 & 0.000 & 0.199 \\ 0.000 & 0.000 & -1.646 & 5.500 & -1.604 & 0.163 \\ 0.000 & 0.000 & 0.000 & -1.604 & 2.792 & 0.036 \end{array}\right],$$

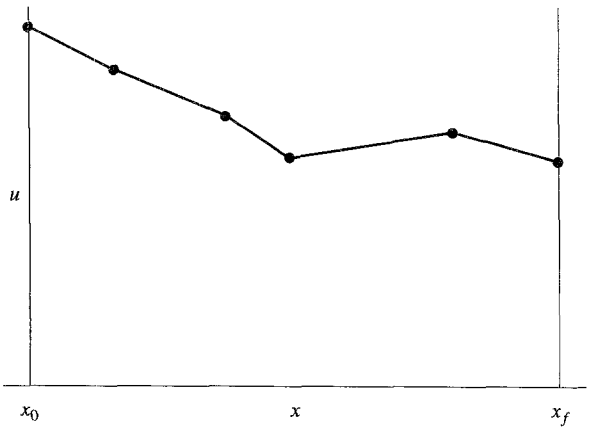
and the solution is

x	$u(x)$	Anal.	Error
2.000	0.1334	0.1353	1.896E-03
2.500	0.1228	0.1231	3.230E-04
3.000	0.0996	0.0996	−2.031E-05
3.500	0.0758	0.0755	−3.280E-04
4.000	0.0564	0.0549	−1.432E-03

We again observe that the system is tridiagonal and symmetric. Will this be true for mixed-boundary conditions?

Other Kinds of Elements

We have used the "hat" functions because they are the simplest kind of element for a 1-D problem. This may not always be adequate. This sketch shows why.



The dots in the figure represent the computed u-values at the nodes and the straight lines that connect the dots are the supposed intermediate values (because the value of u is assumed to vary linearly within the elements). This certainly does not correctly describe the function $u(x)$! (Of course, if the elements are much smaller, the broken line would be a better approximation.) How can we remedy this defect in the procedure?

The obvious way is to use a shape function that better approximates the true function. A quadratic shape function would involve three nodes: two at the ends and an intermediate one. This will force the solution to behave like a parabola that passes through the three nodes. A cubic shape function might be used; it would involve four nodes. Using such higher-order shape functions adds some modest complications to the development of the procedure, but the general approach is identical to what we have shown. When such higher-order shape functions are used, the integration of the analogs of Eqs. (9.31) or (9.5) is usually done by numerical methods. The resulting system of equations is no longer tridiagonal, but the nonzero elements of the coefficient matrix are still clustered about the main diagonal.

There is still a flaw with such higher-order elements in the 1-D problem—the curve for u is not continuous in slope at the juncture of the elements. This flaw could be eliminated if the shape function were splines, but this is not often done because that complicates the procedure significantly.

Sometimes the user of finite-element analysis wants to know the *flux* in addition to the u-values. In one dimension, the flux is $k\partial/\partial x$. (In the sketch, this is proportional to the slopes.) With the linear hat function, the flux values are discontinuous and have larger errors than $u(x)$. Higher-order shape functions help to overcome this.

Convergence Rates

A numerical analyst is always greatly concerned about the accuracy of the numerical solutions. For finite-element-method procedures, the question is "How do the errors decrease when we put nodes closer together?" It can be shown that, with linear elements, errors are of order $O(h^2)$, where h is a measure of the nodal spacing. Quadratic elements give an $O(h^3)$ accuracy; higher orders than two give even better accuracy as the mesh is refined. As we have said, the rate of decrease is a limit value that is achieved only as the h-value gets very small. (The rate of decrease in the errors with quadratic or higher-order shape functions also depends on the integration method used in formulating the system of equations.) Also, a very interesting phenomenon has been observed in studies of the effect of smaller h-values on accuracy—errors may not always decrease uniformly as the spacing is made closer. As a mesh is gradually refined, anomalous behavior can occur.

It is frequently the case that nodes are not uniformly spaced—in fact, this is one of the major advantages of the finite-element method; we can put nodes closer together where the solution $u(x)$ varies most rapidly to get better accuracy in that subregion. This imposes a problem about how best to define "h" in the order of convergence. We shall not pursue this but only remark that if a mesh is refined to improve the accuracy of the

numerical solution, we must refine it everywhere, not just in selected parts of the region.

The errors in the flux do not decrease as rapidly with smaller spacing of the nodes. For a linear shape function, errors decrease as $O(h)$.

Burnett (1987) is an excellent reference.

9.3 Finite Elements for Partial-Differential Equations

When an elliptic partial-differential equation is solved by replacing derivatives with finite-difference approximations, there are serious difficulties if the region is irregular. Analytical methods are also very awkward to apply in such cases.

The finite-element method has no such problems. As we saw in Section 9.2 for one-dimensional boundary-value problems, nodes can be placed wherever the problem solver desires with the finite-element method. This is also true for two- and three-dimensional regions. They can be placed along any boundary so as to approximate it closely. It is the method of choice for solving elliptic partial-differential equations on regions of arbitrary shape.

Although setting up the equations that solve partial-differential equations is no easy task, computer programs are available that do so. It is important to understand how this method works, although this text cannot give everything that today's scientists and engineers might want to know. Our treatment will give a basic knowledge.

The introduction to finite elements in Sections 9.1 and 9.2 is important background for what we shall do here. Recall that two ways of applying variational methods to subdivisions of the region of interest were presented: Rayleigh–Ritz and Galerkin. The first of these minimized the functional for the problem by setting partial derivatives to zero; the second by setting integrals of a weighted residual to zero. The two methods are equivalent for most problems, and both can be used for elliptic equations. We choose the former, in part to provide variety from the presentation in Section 9.2.

The elliptic equation that we will solve in this section is

$$u_{xx} + u_{yy} + Q(x, y)u = F(x, y) \tag{9.40}$$

on region R that is bounded by curve L, with boundary conditions

$$u(x, y) = u_0 \quad \text{on } L_1, \qquad \frac{\partial u}{\partial n} = \alpha u + \beta \quad \text{on } L_2,$$

where $\partial u/\partial n$ is the outward normal gradient.

Observe that we have Dirichlet conditions on some parts of the boundary and mixed boundary conditions on other parts. For our notation, we will use $u(x, y)$ as the exact solution to Eq. (9.40) and $v(x, y)$ as our approximation to $u(x, y)$. Although the finite-element method is most often used when the region is three dimensional, we will simplify the development by doing it in only two dimensions.

Here is our plan of attack:

Step 1. Find the functional that corresponds to the partial-differential equation. This is well known for a large class of problems.

Step 2. Subdivide the region into subregions (elements). Although many kinds of elements can be used, our treatment will consider only triangular elements. The elements must span the entire region and approximate the boundary relatively closely. Every node (the vertices of our triangular elements) and every side of the triangles must be common with adjacent elements except for sides on the boundaries.

Step 3. Write an interpolating relation that gives values for the dependent variable within an element based on the values at the nodes (the vertices of the triangles). We will use linear interpolation from the three nodal values for the element. We will write the interpolation function as the sum of three terms; each term involves a quantity c_i, the value of $v(x, y)$ at a node.

Step 4. Substitute the interpolating relation into the functional, and set the partial derivatives of the functional with respect to each c to zero. This gives three equations, with the c's as unknowns for each element.

Step 5. Combine together (assemble) the element equations of step 4 to get a set of system equations. Adjust these for the boundary conditions of the problem, then solve. This will give the values for the unknown nodal values, the c's, that are approximations to $u(x, y)$ at the nodes. We can get approximations to $u(x, y)$ at intermediate points in the region by using the interpolating relations.

We will discuss each of these five steps in turn. We will provide simple examples to illustrate some of them.

Step 1. Find the Functional

For Eq. (9.40) the functional is well known:

$$I[u] = \iint_{\text{Region}} \left[\left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial u}{\partial y} \right)^2 - Qu^2 + 2Fu \right] dx\, dy - \oint_{L_2} [\alpha u^2 + 2\beta u] dL. \qquad (9.41)$$

It is possible to develop Eq. (9.41) using the Galerkin technique. Workers in the field of structural analysis usually derive it from the principle of virtual work. We will take it as a given.

Step 2. Subdivide the Region

As stipulated, we will use triangular elements, which will be defined by our choice of nodes. The placement of nodes is, in part, an art. In general, we place nodes close together in subregions where the solution is expected to vary rapidly. It is advantageous to make the

sides run in the direction of the largest gradient. Along the curved parts of the boundary, nodes should be placed so that a side of the triangle closely approximates the boundary.

Some of these recommendations depend on knowing the nature of the solution in advance. Often, however, a better placement for the nodes can be accomplished after some preliminary computations or after preliminary trials using the finite-element method with nodes placed arbitrarily.

The chore of defining the nodes' coordinates is facilitated by computer programs that allow the user to place nodes with a pointing device on a graphical display of the region. These programs even permit rotating 3-D regions or looking at cross sections. Once the nodes have been located, the program connects them to create the elements.

Computer routines are available that can divide any given planar region into triangles automatically, but they usually do not have the expertise of an experienced engineer.

Step 3. Write the Interpolating Relations

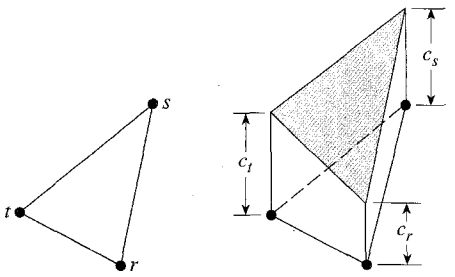
This part of the development is longer than the previous one. As stated, we will use a linear relation. Figure 9.6a is a sketch of typical element (i) whose nodes are numbered r, s, and t in counterclockwise direction. The nodal values are c_r, c_s, and c_t as indicated in Figure 9.6b. The shaded triangle shows how $v(x, y)$ varies within the element.

Within typical element (i), we write

$$v(x, y) = N_r c_r + N_s c_s + N_t c_t = \sum_{j=r,s,t} N_j c_j \tag{9.42}$$

$$= (N_r \;\; N_s \;\; N_t) \begin{Bmatrix} c_r \\ c_s \\ c_t \end{Bmatrix} = (N)\{c\},$$

where the N's (called *shape functions*) will be defined so that $v(x, y)$ at an interior point is a linear interpolation from the nodal values, the c's. We have shown in Eq. (9.42) that $v(x, y)$ can be expressed as the product of vectors (N) and $\{c\}$. (We use parentheses to enclose a row vector and curly brackets to enclose a column vector in this section.) Vector and matrix notation will be useful. We will indicate matrix M by $[M]$.



(a) **(b)**

Figure 9.6

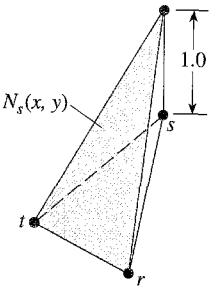


Figure 9.7

Figure 9.6b suggests that $v(x, y)$ lies on the plane above the element that passes through the nodal values. Equation (9.42) does not define $v(x, y)$ outside of element (i); there will be similar expressions for the other elements, but their N's and c's will differ.

A sketch of the entire region would not show $v(x, y)$ as a plane. Instead, it would be a surface composed of planar facets, each in a plane above an element. $v(x, y)$ for the entire region is continuous, but $v'(x, y)$ is not. (This is one of the flaws in our choice of element. Some other element definitions do not have this flaw.)

Another name for the N's of Eq. (9.42) is *pyramid function.* The reason for this name is illustrated in Figure 9.7, where N_s of Figure 9.6 is drawn. Its height at node s is unity and zero at the other nodes. It looks like an unsymmetrical pyramid whose base is the element with its apex directly above node s. The other two N's are similar. It is obvious that the N's are functions of x and y and that the c's are independent of x and y. We now develop expressions for the N's.

Because $v(x, y)$ varies linearly with position within the element, an alternative way to write the linear relation is

$$v(x, y) = a_1 + a_2x + a_3y = (1 \quad x \quad y)\{a\}, \tag{9.43}$$

which must agree with the nodal values when $(x, y) = (x_j, y_j), j = r, s, t$. Hence

$$\begin{aligned} v \text{ at } r: &\quad c_r = a_1 + a_2x_r + a_3y_r, \\ v \text{ at } s: &\quad c_s = a_1 + a_2x_s + a_3y_s, \\ v \text{ at } t: &\quad c_t = a_1 + a_2x_t + a_3y_t. \end{aligned}$$

This is a system of equations

$$[M]\{\mathrm{a}\} = \{c\} \qquad \text{(curly brackets show a column vector)}, \tag{9.44}$$

where

$$[M] = \begin{bmatrix} 1 & x_r & y_r \\ 1 & x_s & y_s \\ 1 & x_t & y_t \end{bmatrix}, \qquad \{a\} = \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \end{Bmatrix}, \qquad \{c\} = \begin{Bmatrix} c_r \\ c_s \\ c_t \end{Bmatrix}.$$

Solving for $\{a\}$:

$$\{a\} = [M^{-1}]\{\mathrm{c}\}.$$

The inverse of M is not difficult to find

$$[M^{-1}] = \frac{1}{2(\text{Area})}\begin{bmatrix} (x_s y_t - x_t y_s) & (x_t y_r - x_r y_t) & (x_r y_s - x_s y_r) \\ (y_s - y_t) & (y_t - y_r) & (y_r - y_s) \\ (x_t - x_s) & (x_r - x_t) & (x_s - x_r) \end{bmatrix}, \tag{9.45}$$

with $2(\text{Area}) = \det(M)$. The value of the determinant is the sum of the elements in row 1 of Eq. (9.45) within the brackets. Area is the area of the triangular element.* You should verify that $[M^{-1}][M] = [I]$ to ensure that Eq. (9.45) truly gives the inverse matrix.

To apply the interpolating function to the minimization of the quadratic functional, Eq. (9.41), we prefer to write $v(x, y)$ in terms of the shape functions of Eq. (9.42). This task is easy. We have, from Eqs. (9.42) and (9.43),

$$\begin{aligned} v(x, y) &= a_1 + a_2 x + a_3 y = (1 \quad x \quad y)\{a\} \\ &= (1 \quad x \quad y)[M^{-1}]\{c\}. \end{aligned}$$

However, in terms of N (from Eq. 9.42),

$$v(x, y) = (N)\{c\}. \tag{9.46}$$

Comparing the two expressions, we have

$$(N) = (1 \quad x \quad y)[M^{-1}], \tag{9.47}$$

where M^{-1} is given by Eq. (9.45). Observe carefully that Eq. (9.47) says that each N is a linear function of x and y of the form

$$N_j = A_j + B_j x + C_j y, \qquad j = r, s, t \tag{9.48}$$

and that the coefficients are in column j of $[M^{-1}]$.

We have found the expressions for the N's. Before we go on, we digress to show an example that will clarify this step.

EXAMPLE 9.6 For the triangular element shown in Figure 9.8 with nodes r, s, and t in counterclockwise order, find $\{a\}$, $\{N\}$, and $v(0.8, 0.4)$.

Node	x	y	c
r	0	0	100
s	2	0	200
t	0	1	300

Before we do any computations, we can find $v(0.8, 0.4)$ by inspection. (See Fig. 9.8.) Point 1 is at (0, 0.4), so v there is 180 by linear interpolation between nodes r and t. Similarly, v at point (2) is 240. The point (0.8, 0.4) is $\frac{2}{3}$ of the distance from points 1 and 2,

---

* That Area $= \frac{1}{2}$ det (M) is shown in most books on vectors where the cross product is explained.

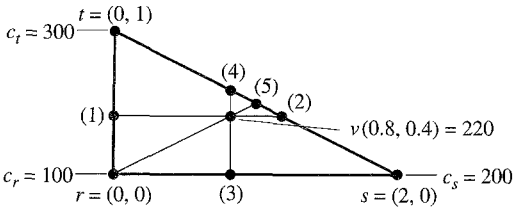


Figure 9.8

so $v(0.8, 0.4) = 180 + \frac{2}{3}(240 - 180) = 220$. We get the same result by interpolating between points 3 and 4, and between node r and (5).

To get $\{a\}$ we first compute $[M^{-1}]$:

$$[M] = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 2 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \qquad [M^{-1}] = \begin{bmatrix} 1 & 0 & 0 \\ -0.5 & 0.5 & 0 \\ -1 & 0 & 1 \end{bmatrix}.$$

Then we compute

$$\{a\} = [M^{-1}]\{c\} = \begin{bmatrix} 1 & 0 & 0 \\ -0.5 & 0.5 & 0 \\ -1 & 0 & 1 \end{bmatrix} \begin{Bmatrix} 100 \\ 200 \\ 300 \end{Bmatrix} = \begin{Bmatrix} 100 \\ 50 \\ 200 \end{Bmatrix},$$

giving $v(x, y) = 100 + 50x + 200y$. (You should confirm that this gives the correct values at each of the nodes.) If we substitute $x = 0.8$, $y = 0.4$, we get $v = 220$, as we should.

From Eq. (9.47),

$$(N) = (1 \quad x \quad y)[M^{-1}] = (1 - 0.5x - y, \quad 0.5x, \quad y).$$

In other words, we have

$$N_r = 1 - 0.5x - y,$$
$$N_s = 0.5x,$$
$$N_t = y.$$

(You should confirm that these also have the proper values at each of the nodes.) It is important to notice that the coefficients of the N's [the A_i, B_i, and C_i of Eq. (9.48)] can be read directly from the columns of $[M^{-1}]$.

In what follows we will need the partial derivatives of the N's with respect to x and to y. From $N_j = A_j + B_j x + C_j y$, we see that these are constants that can be read from rows 2 and 3 of $[M^{-1}]$ in column j.

At this point we know how to write $v(x, y)$ within the single triangular element (i) as $v(x, y) = (N^{(i)})\{c^{(i)}\}$. (The superscripts (i) tell which element is being considered whenever

this is necessary.) We now stipulate that $(N^{(i)}) \equiv (0)$ everywhere outside of element (i). Therefore we can write

$$v(x, y) = \sum_{i = \text{all elements}} (N^{(i)})\{c^{(i)}\}.$$

This is a mathematical statement of the previous observation that $v(x, y)$ is a surface composed of joined planar facets.

We are now ready for step 4 of our plan. This too is lengthy but each portion is easy.

Step 4. Substitute $v(x, y)$ into the Functional and Minimize

We continue to work with typical element (i) whose nodes are r, s, and t. Repeating Eq. (9.46), our $v(x, y)$ is

$$v(x, y) = (N)\{c\} = N_r c_r + N_s c_s + N_t c_t,$$

where the N's are given by Eqs. (9.47) and (9.45). Recall that each N is $A_j + B_j x + C_j y$ with the coefficients given by the elements in column j of $[M^{-1}]$.

Our objective is to develop a set of three equations for element (i), which is, in matrix form,

$$[K]\{c\} = \{b\},$$

and which is a prototype of similar equations for all other elements.

When we substitute $v(x, y)$ for element (i) into the functional of Eq. (9.41), we get

$$I(c_r, c_s, c_t) = \iint_{\text{element (i)}} \left[\left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 - Qv^2 + 2Fv\right] dx\, dy - \oint_{L_2} [\alpha v^2 + 2\beta v^2]\, dL. \tag{9.49}$$

[I is now an ordinary function of the c's. The integral is only over the area of element (i) because the N's that define $v(x, y)$ in element (i) are zero outside of (i). The last term appears only if element (i) has a side on the boundary. Actually, we will postpone handling this last term for now and handle it as an adjustment to the equations after they have been developed.]

We minimize I by setting the three partials (with respect to each of the three c's) to zero. We now develop expressions for these partials. First consider $\partial I/\partial c_r$.

For the first term in the integrand:

$$\frac{\partial}{\partial c_r}\left[\left(\frac{\partial v}{\partial x}\right)^2\right] = 2\left(\frac{\partial v}{\partial x}\right)\left(\frac{\partial}{\partial c_r}\left(\frac{\partial v}{\partial x}\right)\right).$$

However,

$$\frac{\partial v}{\partial x} = \left(\frac{\partial N_r}{\partial x}\right) c_r + \left(\frac{\partial N_s}{\partial x}\right) c_s + \left(\frac{\partial N_t}{\partial x}\right) c_t$$
$$= B_r c_r + B_s c_s + B_t c_t,$$

by virtue of Eq. (9.48). (The B's come from row 2 of $[M^{-1}]$.)

Also, $\partial/\partial c_r(\partial v/\partial x) = B_r$ because c_s and c_t are independent of c_r. Hence

$$\frac{\partial}{\partial c_r}\left[\left(\frac{\partial v}{\partial x}\right)^2\right] = 2(B_r^2 + B_rB_s + B_rB_t),$$

The result for the second term is similar:

$$\frac{\partial}{\partial c_r}\left[\left(\frac{\partial v}{\partial y}\right)^2\right] = 2(C_r^2 + C_rC_s + C_rC_t),$$

where the C's come from row 3 of $[M^{-1}]$.

We next consider the Q term. Q is independent of c_r, so

$$\frac{\partial}{\partial c_r}(-Qv^2) = -Q\left[2v\left(\frac{\partial v}{\partial c_r}\right)\right] - 2Q(N_rc_r + N_sc_s + N_tc_t)(N_r).$$

Finally we work with the F term. F is independent of c_r, so

$$\frac{\partial}{\partial c_r}(2Fv) = 2F\left(\frac{\partial v}{\partial c_r}\right) = 2F(N_r).$$

Putting all this together, we have

$$\begin{aligned}\frac{\partial I}{\partial c_r} = 0 = &\iint_{(i)} 2[B_r^2c_r + B_rB_sc_s + B_rB_tc_t]\,dx\,dy \\ &+ \iint_{(i)} 2[C_r^2c_r + C_rC_sc_s + C_rC_tc_t]\,dx\,dy \\ &- \iint_{(i)} 2Q[N_r^2c_r + N_rN_sc_s + N_rN_tc_t]\,dx\,dy \\ &+ \iint_{(i)} 2FN_r\,dx\,dy.\end{aligned} \tag{9.50}$$

Equation (9.50) really is a formulation with the c's unknown:

$$K_{rr}c_r + K_{rs}c_s + K_{rt}c_t = b_r, \tag{9.51}$$

where

$$\begin{aligned}K_{rr} &= \iint_{(i)} 2B_r^2\,dx\,dy + \iint_{(i)} 2C_r^2\,dx\,dy - \iint_{(i)} 2QN_r^2\,dx\,dy, \\ K_{rs} &= \iint_{(i)} 2B_rB_s\,dx\,dy + \iint_{(i)} 2C_rC_s\,dx\,dy - \iint_{(i)} 2QN_rN_s\,dx\,dy, \\ K_{rt} &= \iint_{(i)} 2B_rB_t\,dx\,dy + \iint_{(i)} 2C_rC_t\,dx\,dy - \iint_{(i)} 2QN_rN_t\,dx\,dy, \\ b_r &= -\iint_{(i)} 2FN_r\,dx\,dy.\end{aligned}$$

[Remember, we postpone handling the last part of Eq. (9.49).]

Now we recognize that the B's and C's of Eq. (9.51) are constants, so we can bring them out from under the integral sign. If we use average values for Q and F within the element, we can also bring these out as their average values. (The best average value to use is the value of Q and F at the centroid of the triangular element.)

This means that we have to evaluate these five integrals:

$$I_1: \quad \iint_{(i)} dx\,dy$$

$$I_2: \quad \iint_{(i)} N_r^2\,dx\,dy \qquad I_3: \quad \iint_{(i)} N_r N_s\,dx\,dy \qquad I_4: \quad \iint_{(i)} N_r N_t\,dx\,dy$$

$$I_5: \quad \iint_{(i)} N_r\,dx\,dy.$$

The first of these is easy: I_1 = Area of the element, which we already know from having computed $[M^{-1}]$. The other integrals are laborious to compute directly, but there is a useful formula for the integral of the product of powers of linear functions over a triangle:

$$\iint_{\text{(triangle)}} N_r^{\ell} N_s^{m} N_t^{n}\,dx\,dy = \frac{2\ell!m!n!}{(\ell + m + n + 2)!}\,(\text{Area}).$$

Using this with the proper values for the exponents, ℓ, m, and n, gives

$$I_2 = \frac{(\text{Area})}{6},$$

$$I_3 = \frac{(\text{Area})}{12},$$

$$I_4 = \frac{(\text{Area})}{12},$$

$$I_5 = \frac{(\text{Area})}{3}.$$

The terms in Eq. (9.51) are then

$$K_{rr}c_r + K_{rs}c_s + K_{rt}c_t + b_r, \tag{9.52}$$

where

$$K_{rr} = 2(\text{Area})\left(B_r^2 + C_r^2 - \frac{Q_{\text{av}}}{6}\right),$$

$$K_{rs} = 2(\text{Area})\left(B_r B_s + C_r C_s - \frac{Q_{\text{av}}}{12}\right),$$

$$K_{rt} = 2(\text{Area})\left(B_r B_t + C_r C_t - \frac{Q_{\text{av}}}{12}\right),$$

$$b_r = -2(\text{Area})\left(\frac{F_{\text{av}}}{3}\right).$$

If we do the same with $\partial I/\partial c_s$ and $\partial I/\partial c_t$, we get two more equations in the c's for element (i). All together we have three equations, which we call the *element equations.* We simplify these equations somewhat by omitting the common factor of 2 for each of them to get

$$[K]\begin{Bmatrix} c_r \\ c_s \\ c_t \end{Bmatrix} = \begin{Bmatrix} b_r \\ b_s \\ b_t \end{Bmatrix},$$

where

$$\text{(diagonals)} \qquad K_{jj} = \text{Area}\left[B_j^2 + C_j^2 - \frac{Q_{\text{av}}}{6}\right], \qquad j = r, s, t;$$

$$\text{(off-diagonals)} \qquad K_{jk} = \text{Area}\left[B_jB_k + C_jC_k - \frac{Q_{\text{av}}}{12}\right] \qquad \begin{cases} j \neq k, \\ j = r, s, t, \\ k = r, s, t; \end{cases}$$

$$\text{(rhs)} \qquad b_j = -\text{Area}\left[\frac{F_{\text{av}}}{3}\right], \qquad j = r, s, t.$$

Observe that $[K]$ is symmetrical: $K_{ij} = K_{ji}$.

Here is an example to clarify the formation of the element equations.

EXAMPLE 9.7 Find the element equations for the element of Example 9.6 if $Q(x, y) = (xy)/2$ and $F(x, y) = x + y$.

The nodes are $(x, y) = (0, 0)$, $(2, 0)$, and $(0, 1)$. We had, for $[M^{-1}]$,

$$[M^{-1}] = \begin{bmatrix} 1 & 0 & 0 \\ -0.5 & 0.5 & 0 \\ -1 & 0 & 1 \end{bmatrix}.$$

Area $= 1$. Centroid is at $x = (0 + 2 + 0)/3 = \frac{2}{3}$, $y = (0 + 0 + 1)/3 = \frac{1}{3}$. $Q_{\text{av}} = \left(\frac{2}{3}\right)\left(\frac{1}{3}\right)/2 = \frac{1}{9}$. $F_{\text{av}} = \frac{2}{3} + \frac{1}{3} = 1$.

Using Eq. (9.52), we find that the element equations are

$$\begin{bmatrix} 1.2315 & -0.2592 & -1.0092 \\ -0.2592 & 0.2315 & -0.0093 \\ -1.0092 & -0.0093 & 0.9815 \end{bmatrix}\begin{Bmatrix} c_r \\ c_s \\ c_t \end{Bmatrix} = \begin{Bmatrix} -0.3333 \\ -0.3333 \\ -0.3333 \end{Bmatrix}.$$

We are now ready for step 5 of the plan.

Step 5. Assemble the Equations, Adjust for Boundary Conditions, Solve

There are three separate operations in step 5: (i) assemble the equations, (ii) adjust for boundary conditions, and (iii) solve the equations.

(i) Do the Assembly As we have seen, there are three equations for every element. However, some or all nodes of element (i) are shared with other elements; the c-value for a shared node then appears in the equations of all elements that share the node. Combining all of the element equations will create a global system coefficient matrix with as many rows and columns as there are nodes in the system. We combine (assemble) the system matrix in the following way.

Suppose there are n nodes in the system. Number the nodes in order, from 1 to n. Associate the number of each node with the row and column of every element matrix where the c for that node appears on the diagonal. Also associate the node numbers with the rows and columns of the system matrix in the same way.

We get the entry in row (i) and column (j) of the system matrix by adding the values from row (i) of every element matrix that has row (i), then adding these in the columns where the column-node numbers match. We also add the b_i's from these rows to get the b_i of the system matrix. An example will clarify this operation.

EXAMPLE 9.8 Suppose there are five nodes that define three elements, as shown in Figure 9.9 with the element matrices of Eq. (9.53a, b, c) below. Construct the system matrix without adjusting for boundary conditions.

$$\text{Element [1]} \quad \begin{matrix}(1)\to\\(2)\to\\(4)\to\end{matrix}\begin{bmatrix}K_{11} & K_{12} & K_{13}\\ K_{21} & K_{22} & K_{23}\\ K_{31} & K_{32} & K_{33}\end{bmatrix}\begin{Bmatrix}c_1\\c_2\\c_4\end{Bmatrix}=\begin{Bmatrix}b_1\\b_2\\b_3\end{Bmatrix} \qquad \textbf{(9.53a)}$$
$$\begin{matrix}\uparrow & \uparrow & \uparrow\\ (1) & (2) & (4)\end{matrix}$$

$$\text{Element [2]} \quad \begin{matrix}(2)\to\\(3)\to\\(4)\to\end{matrix}\begin{bmatrix}K_{11} & K_{12} & K_{13}\\ K_{21} & K_{22} & K_{23}\\ K_{31} & K_{32} & K_{33}\end{bmatrix}\begin{Bmatrix}c_2\\c_3\\c_4\end{Bmatrix}=\begin{Bmatrix}b_1\\b_2\\b_3\end{Bmatrix} \qquad \textbf{(9.53b)}$$
$$\begin{matrix}\uparrow & \uparrow & \uparrow\\ (2) & (3) & (4)\end{matrix}$$

$$\text{Element [3]} \quad \begin{matrix}(4)\to\\(5)\to\\(1)\to\end{matrix}\begin{bmatrix}K_{11} & K_{12} & K_{13}\\ K_{21} & K_{22} & K_{23}\\ K_{31} & K_{32} & K_{33}\end{bmatrix}\begin{Bmatrix}c_4\\c_5\\c_1\end{Bmatrix}=\begin{Bmatrix}b_1\\b_2\\b_3\end{Bmatrix} \qquad \textbf{(9.53c)}$$
$$\begin{matrix}\uparrow & \uparrow & \uparrow\\ (4) & (5) & (1)\end{matrix}$$

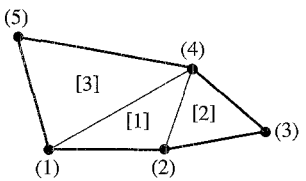


Figure 9.9

[The rows and columns of Eqs. (9.53) could have been in a different order, although we always go counterclockwise around the element in selecting the nodes.]

We construct the system matrix as follows, where the superscripts indicate the element number that provides the value:

For row 1	For row 2
col 1: $K_{11}^{[1]} + K_{33}^{[3]}$	col 1: $K_{21}^{[1]}$
col 2: $K_{12}^{[1]}$	col 2: $K_{22}^{[1]} + K_{11}^{[2]}$
col 3: 0	col 3: $K_{12}^{[2]}$
col 4: $K_{13}^{[1]} + K_{31}^{[3]}$	col 4: $K_{23}^{[1]} + K_{13}^{[2]}$
col 5: $K_{32}^{[3]}$	col 5: 0
b: $b_1^{[1]} + b_3^{[3]}$	b: $b_2^{[1]} + b_1^{[2]}$
and so forth.	

[A zero appears in column 3 of row 1 because node (3) is not in any element that includes node (1). A zero appears in column 5 of row 2 because node (5) is not in any element that includes node (2).]

Once the system matrix has been assembled, we make the adjustments for boundary conditions.

(ii) Adjust for Boundary Conditions There are two types of boundary conditions: non-Dirichlet conditions on some parts of the boundary (L_2) and Dirichlet conditions on other parts (L_1). We will always select nodes such that only one of the two types of conditions pertains to any side of the element. Hence there will always be a node at the point where the two types join. These two types of boundary conditions require two separate adjustments. We prefer to apply the adjustment for a boundary condition that involves the outward normal derivative to the system equations first and then do the adjustment for Dirichlet conditions.

Adjusting for Non-Dirichlet Conditions Non-Dirichlet conditions (those that involve the outward normal derivative) are associated, not with the nodes, but with sides of the triangular elements, sides that correspond to part of L_2 of Eq. (9.41). Consider an element that has a non-Dirichlet condition on one side that lies between nodes r and s. The effect of the boundary condition on the equations comes from differentiating the last term of Eq. (9.49) with respect to the c's. However, if we take α and β out from the integrand as average values, we see from Eq. (9.49) that they are of the same form as the Q and F terms except they are line integrals rather than area integrals. That similarity lets us immediately write the result of the differentiation with respect to c_r as

$$-2\alpha_{\text{av}} \oint (N_r^2 c_r + N_r N_s c_s)\, dL - 2\beta_{\text{av}} \oint N_r\, dL, \tag{9.54}$$

where the line integrals are along the side between nodes r and s. It is important to note that we have not included c_t because node t is not on the side we are considering. (The average values of α and β should be taken at the midpoint of the side.) When we integrate, we have

$$-2\alpha_{\text{av}} L\left(\frac{c_r}{3} + \frac{c_s}{6}\right) - 2\beta_{\text{av}}\frac{L}{2}. \tag{9.55}$$

(It is easy to evaluate the integrals when we remember that the N's are linear from 1 to 0 between the two nodes.)

Precisely the same relations result when we differentiate with respect to node c_s, except the roles of r and s are interchanged in Eqs. (9.54) and (9.55). The net result would be to add $2\beta L/2$ to the right-hand sides of the rows for c_r and c_s. We also would subtract the multipliers of c_r and c_s in Eq. (9.55) from the coefficients for c_r and c_s in row r. The similar equations from the partials with respect to c_s provide subtractions from the coefficients in row s.

Recall, however, that we canceled a 2 factor when we constructed the element equations and so we must do so here. We make this adjustment to the element equations for every element that has a derivative condition on a side.

Adjusting for Dirichlet Conditions For every node that appears on the boundary where there is a Dirichlet condition, the u-value is specified. We insert this known value in place of the c of that node in every equation where it appears and transpose to the right-hand side. (Actually, if the node number is m, all entries in column m of the matrix are multiplied by the value and subtracted from the right-hand side of the corresponding row.) We also remove the row corresponding to the number of the known node from the set of equations. (The column for this node has already been "removed" by being transferred to the right-hand side.)

Removing the rows for those nodes with a Dirichlet condition is simplified in a computer program if these rows are at the top or the bottom of the matrix. There are other ways to handle Dirichlet conditions that avoid having to remove the rows.

This completes our construction of the system equations.

(iii) Getting the Solution We solve the system in the usual way, perhaps preferring an iterative procedure if the system is large.

An example, intentionally simple, follows.

EXAMPLE 9.9 The region shown in Figure 9.10 has four nodes. It is divided into just two elements. The values for u are specified at nodes (3) and (4), and the outward normal gradient is specified on three sides as indicated. The equation we are to solve is

$$u_{xx} + u_{yy} - \left(\frac{y}{10}\right)u = \frac{x}{4} + y - 12.$$

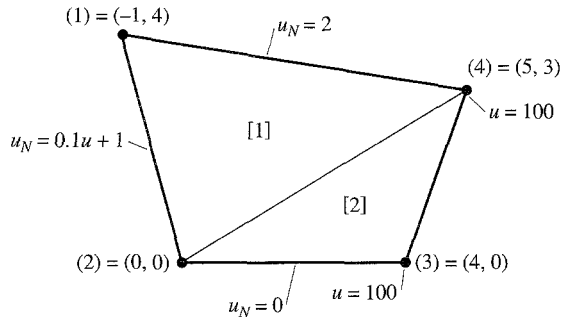


Figure 9.10

Find the solution by the finite-element method. The element-matrix inverses are:

M inverse for element 1 area is 11.5

$$\begin{bmatrix} 1.000 & 0.000 & 0.000 \\ -0.043 & 0.174 & -0.130 \\ -0.261 & 0.043 & 0.217 \end{bmatrix}$$
(2) (4) (1)

M inverse for element 2 area is 6

$$\begin{bmatrix} 1.000 & 0.000 & 0.000 \\ -0.250 & 0.250 & 0.000 \\ 0.083 & -0.417 & 0.333 \end{bmatrix}$$
(2) (3) (4)

$Q_{av} = -0.233 \quad F_{av} = -9.333 \qquad Q_{av} = -0.1 \quad F_{av} = -10.25$

From these we get these element equations:

Element equations for element 1

$$\begin{array}{c} 2 \rightarrow \\ 4 \rightarrow \\ 1 \rightarrow \end{array} \begin{bmatrix} 1.2516 & 0.0062 & -0.3633 & 35.7778 \\ 0.0062 & 0.8168 & 0.0714 & 35.7778 \\ -0.3633 & 0.0714 & 1.1864 & 35.7778 \end{bmatrix}$$
(2) (4) (1)

Element equations for element 2

$$\begin{array}{c} 2 \rightarrow \\ 3 \rightarrow \\ 4 \rightarrow \end{array} \begin{bmatrix} 0.5167 & -0.5333 & 0.2167 & 20.5000 \\ -0.5333 & 1.5167 & -0.7833 & 20.5000 \\ 0.2167 & -0.7833 & 0.7667 & 20.5000 \end{bmatrix}$$
(2) (3) (4)

These equations assemble to give this unadjusted system matrix:

$$\begin{bmatrix} 1.186 & -0.363 & 0.000 & 0.071 & 35.778 \\ -0.363 & 1.768 & -0.533 & 0.223 & 56.278 \\ 0.000 & -0.533 & 1.517 & -0.783 & 20.500 \\ 0.071 & 0.223 & -0.783 & 1.583 & 56.278 \end{bmatrix}$$

We need the lengths of sides 4–1, 1–2, and 2–3. They are

$$\text{Side 4–1: } 6.083, \qquad \text{Side 1–2: } 4.123, \qquad \text{Side 2–3: } 4.$$

We now adjust for the derivative conditions to get the modified system:

$$\begin{bmatrix} 1.049 & -0.432 & 0.000 & 0.003 & 43.922 \\ -0.432 & 1.631 & -0.533 & 0.154 & 64.422 \\ 0.000 & -0.533 & 1.517 & -0.783 & 20.500 \\ 0.003 & 0.154 & -0.783 & 1.446 & 64.422 \end{bmatrix}.$$

The second adjustment is for the known values at nodes (3) and (4), giving

$$\begin{bmatrix} 1.049 & -0.432 & 43.650 \\ -0.432 & 1.631 & 102.339 \end{bmatrix}$$

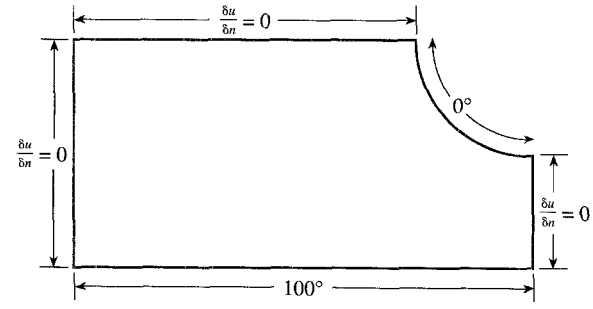
which we solve to get these estimates of u_1 and u_2:

$$u_1 = 75.71, \qquad u_2 = 82.80.$$

Solving an Elliptic Problem with MATLAB

MATLAB's professional version has a toolbox, the *Partial Differential Equation Toolbox* (not included in the student version that we use) that can solve all three types of partial-differential equations. We describe here how it solves an elliptic problem.

This illustrative example solves Laplace's equation to get the temperature distribution on a region that is a rectangle whose width is twice its height and that has a quarter circle removed from its upper-right corner:



Along the base of the figure, the temperature is 100°; on the arc it is 0° (Dirichlet conditions). All other edges are insulated (Neumann condition, $\partial u/\partial n = 0$). We desire the steady-state temperature distribution.

We will obtain the solution through six steps:

1. Draw the region.
2. Save the region as an M-file.
3. Set boundary conditions.
4. Create a mesh of triangular elements.
5. Define the type of equation to be solved.
6. Solve the problem.

1. Draw the Region The Toolbox provides two different ways to do this: with a Graphical User Interface (GUI), or through commands typed into the command screen.

We will use the second way. Our region will be composed of two parts: the rectangle and a circle that is subtracted from it. We begin with the rectangle. We use the command:

```
>> pderect([-1 1 -.5 .5])
```

where the parameter is a vector of the x-coordinates followed by the y-coordinates of two opposite corners. [The corners are at $(-1, -0.5)$ and $(1, 0.5)$.] After the command is entered, we see the rectangle in a separate window that we will call the "figure window."

This window has a menu bar as well as another bar that has icons; these icons are quick ways to call for many menu commands.

We now create the circle. We go back to the command window and enter

```
>> pdecirc(1, 0.5, 0.5)
```

This superimposes a circle on the rectangle with center at $x = 1$, $y = 0.5$, and radius of 0.5, which we can view in the figure window. The figure window has a box labeled "Set formula" that reads `R1 + C1`, which we change to `R1 - C1` by clicking the box and using the keyboard to make the change. The figure window does not yet reflect the change but it is in effect.

From now on, each step is done in the figure window.

2. Save the Region It is always good to save the description of the region. This permits one to retrieve it at a later time. Saving is done by invoking `File/Save As` in the figure window. We give it a name, say, FIGA, and it is added to the list of M-files.

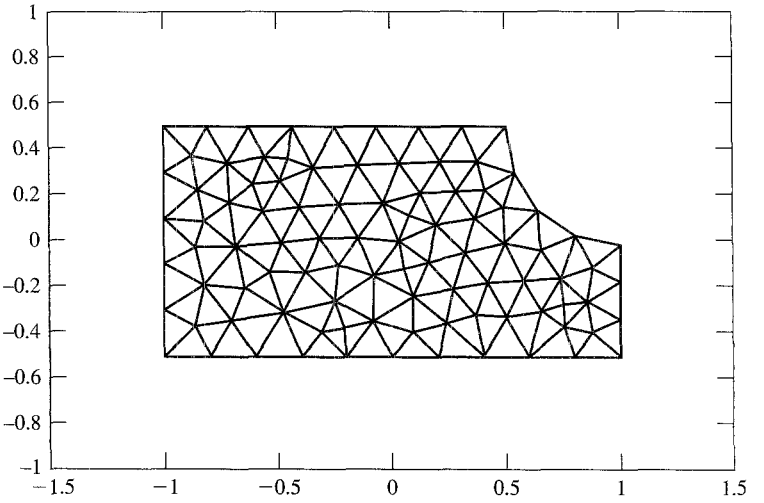
3. Set Boundary Conditions We invoke `Boundary/Boundary Mode` and see the region displayed (the distorted rectangle is now seen). Its outline is red with arrows indicating a counterclockwise ordering. We establish the boundary conditions by double-clicking on a boundary and then entering parameters into a dialog box.

We begin with the base of the figure. After double-clicking on the base, we see the dialog box. Select `Dirichlet` (actually, this is the default), and make the value of $t = 100$. Click OK and we are returned to the figure window. We double-click on the arc and select `Dirichlet`, and make $t = 0$ (both are default values).

We need to establish a Neumann condition on each of the other sides of the rectangle. This is easy to do: double-click on the side, select `Neumann`, set $g = 0$, $q = 0$ (default values), click `OK`.

The boundary conditions are now established.

4. Create a Mesh of Triangular Elements Clicking on `Mesh/Initialize Mesh` in the menu bar creates a coarse mesh of triangular elements:

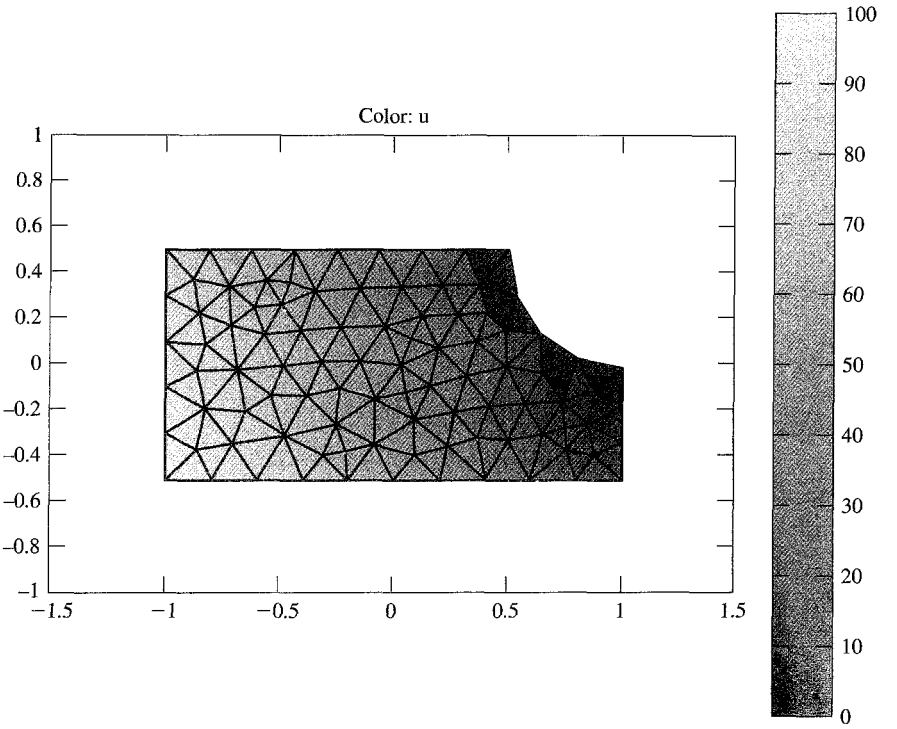


We can refine this mesh with `Mesh/Refine Mesh` but we stay with the current mesh for now.

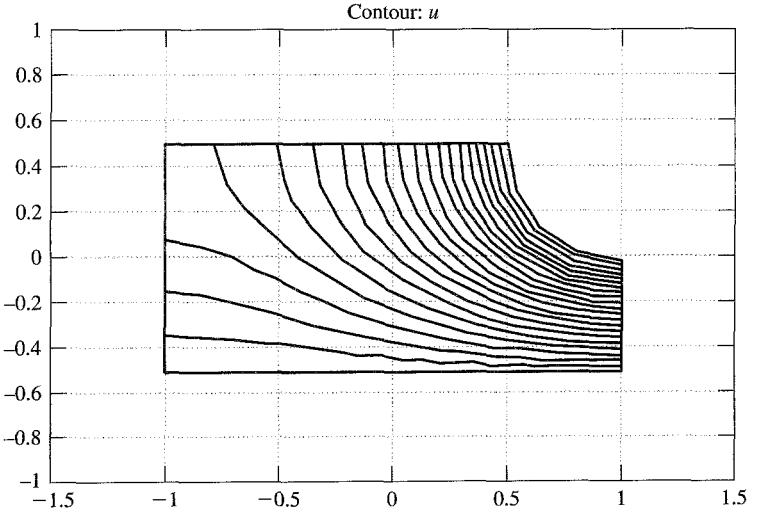
5. Define the Type of Equation to Be Solved We do this by clicking on `PDE/PDE Specification` in the menu bar. In the dialog box that appears, we select `Elliptic` (the default) and set $c = 1, a = 0, f = 0$. We then click `OK` to finish this step. (These parameters are for our equation, $\nabla^2 u = 0$.)

6. Solve the Problem Clicking `Solve/Solve PDE` in the menu bar gets the solution. The software in the toolbox sets up the equations, assembles these, adjusts for boundary conditions, and solves the system of equations.

We see a display of the region with colors indicating the temperature in each element. On the right of this is a vertical bar that shows how colors and temperatures are related. Our figure here is not in color but the output actually indicates the temperatures within each element by colors that vary from bright red (100°) to bright blue (0°). Because we use a coarse mesh, it is easy to see the temperature of each individual element by its color. This would be difficult with a fine mesh.



Another way to see the solution is to get the isotherms. Selecting only `Contour` in the `Plot Parameters` dialog box gives a plot of isotherms within the region, with $\Delta u = 5°$. This is shown in the next figure. On the computer screen, these isotherms are colored to indicate the temperatures.



The Heat Equation

As we have just seen, the finite-element method is often preferred for solving boundary-value problems. It is also the preferred method for solving the heat equation when the region of interest is not regular. You should know something about this application of finite elements, but we do not give a full treatment.

Consider the heat-flow equation in two dimensions with heat generation given by $F(x, y)$:

$$\frac{\partial u}{\partial t} = \frac{k}{c\rho}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) + F(x, y), \tag{9.56}$$

which is subject to initial conditions at $t = 0$ and boundary conditions that may be Dirichlet or may involve the outward normal gradient. Although this is really a three-variable problem (in x, y, and t), it is customary to approximate the time derivative with a finite difference and apply finite elements only to the spatial region. Doing so, we can rewrite Eq. (9.56) as

$$\frac{u^{m+1} - u^m}{\Delta t} = \frac{k}{c\rho}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) + F(x, y), \tag{9.57}$$

where we have used a forward difference as in the explicit method. (We might prefer Crank–Nicolson or the implicit method, but we will keep things simple.)

To apply finite elements to the region, we do exactly as described previously—cover the region with joined elements, write element equations for the right-hand side of Eq. (9.56), assemble these, adjust for boundary conditions, and solve. However, we must also consider the time variable. We do so by considering Eq. (9.57) to apply at a fixed point in time, t_m. Because we know the values of u everywhere within the region at $t = t_0$, we surely know the initial nodal values. We then can solve Eq. (9.57) for the u-values at $t = t_0 + \Delta t$, where the size of Δt is chosen small enough to ensure stability.

We will use the Galerkin procedure to derive the element equations to provide some variety from the above. In this procedure you will remember that we integrate the residual weighted with each of the shape functions and set them to zero. (The integrations are done over the element area.) If we stay with linear triangular elements, there are three shape functions, N_r, N_s, and N_t, where the subscripts denote the three vertices (nodes) of the element taken in counterclockwise order.

The residual for Eq. (9.56) is

$$\text{Residual} = u_t - \alpha(u_{xx} + u_{yy}) - F, \tag{9.58}$$

where we have used the subscript notation for derivatives and have abbreviated $k/c\rho$ with α.

As stated, we will use linear triangular elements; within each element we approximate u with

$$u(x, y) \approx v(x, y) = N_r c_r + N_s c_s + N_t c_t. \tag{9.59}$$

This means that Galerkin integrals are

$$\iint N_j[v_t - \alpha(v_{xx} + v_{yy}) + F]\, dx\, dy = 0, \qquad j = r, s, t. \tag{9.60}$$

If we apply integration by parts (as we did in Section 9.2) to the second derivatives of Eq. (9.60), we can reduce the order of these derivatives. Doing so and replacing v from Eq. (9.59) gives a set of three equations for each element, which we write in matrix form:

$$[C]\left\{\frac{\partial c}{\partial t}\right\} + [K]\{c\} = \{b\}. \tag{9.61}$$

The components of $\{c\}$ are the nodal temperatures of the element, of course; those of $\{\partial c/\partial t\}$ are the time derivatives. The components of the matrices of Eq. (9.61) are

$$\begin{aligned} C_{i,j} &- \int\int N_i N_j \, dx \, dy, \qquad i,j = r, s, t \\ K_{ij} &= \alpha_{\text{av}} \int\int \left[\left(\frac{\partial N_i}{\partial x}\right)\left(\frac{\partial N_j}{\partial x}\right) + \left(\frac{\partial N_i}{\partial y}\right)\left(\frac{N_j}{\partial y}\right)\right] dx \, dy, \qquad i,j = r, s, t. \\ b_i &= \int\int F N_i \, dx \, dy + \oint u_N N_i \, dL, \qquad i = r, s, t. \end{aligned} \tag{9.62}$$

In Eq. (9.62), the line integral in the b's is present only along a side of an element on the boundary of the region where the outward normal gradient u_N is specified in a boundary condition.

From the development of Eq. (9.52), we know how to evaluate all of the integrals of Eq. (9.62) when the elements are triangles. [See, for example, Burnett (1987) for the evaluations for other types of elements.]

As stated, we will use a finite-difference approximation for $\partial c/\partial t$. If this is a forward difference as suggested, we get the explicit formula

$$\frac{1}{\Delta t}[C]\{c^{m+1}\} = \frac{1}{\Delta t}[C]\{c^m\} - [K]\{c^m\} + \{b\}, \tag{9.63}$$

where all the c's on the right are nodal temperatures at $t = t_m$ and the nodal temperatures on the left in $\{c^{m+1}\}$ are at $t = t_{m+1}$.

We can put Eq. (9.63) into a more familiar iterating form by multiplying through by $\Delta t[C]^{-1}$:

$$\{c^{m+1}\} = \{c^m\} - \Delta t[C]^{-1}[K]\{c^m\} + \Delta t[C]^{-1}\{b\}. \tag{9.64}$$

[We can make Eq. (9.64) more compact by combining the multipliers of $\{c^m\}$.]

In principle, we have solved the heat-flow problem by finite elements. We construct the equations for every element from Eq. (9.64) and assemble them to get the global matrix, then adjust for boundary conditions just as before. This gives a set of equations in the unknown nodal values that we use to step forward in time from the initial point. With the explicit method illustrated here, each time step is just a matrix multiplication of the current nodal temperatures (and a vector addition) to get the next set of values. If we had used an implicit method such as Crank–Nicolson, we would have had to solve a set of equations at each step, but, unfortunately, they are not tridiagonal. We might hope for some equivalent to the A.D.I. method, but A.D.I. requires that the nodes be uniformly spaced. The conclusion is that the finite-element method in two or three dimensions is a problem that is expensive to solve. In one dimension, however, the system is tridiagonal, so that situation is not bad.

Solving a Parabolic Problem with MATLAB

The Partial Differential Equation Toolbox can solve all types of partial-differential equations. We show here how it can solve the heat equation. In the previous description of solving an elliptic problem with the toolbox, the solution is the steady-state distribution of temperatures. This is not reached instantaneously; the progress of the solution from an initial state to the steady state can be found by solving the heat equation:

$$\partial u/\partial t = k/c\rho\, \nabla^2 u.$$

MATLAB's generic form of a parabolic equation is

$$\mathbf{d}*\partial u/\partial t - \nabla(\mathbf{c}*\nabla u) + \mathbf{a}u = \mathbf{f},$$

where we have used boldface to pinpoint the parameters. For our equation, we want $\mathbf{d} = 1$, $\mathbf{c} = k/c\rho$ (the thermal diffusivity), $\mathbf{a} = 0$, and $\mathbf{f} = 0$.

Let us see how the steady state is approached as time advances for the same region and boundary conditions as before. We will take the initial temperatures within the region as 0°.

The procedure is almost exactly the same as before, only step 5 is different:

1. Define the region.
2. Define the boundary conditions.
3. Enter the values for the parameters of the equation.
4. Establish a mesh of triangular elements.
5. Enter values for the initial values for u and a list of times for which the solution is computed.
6. Solve the problem and display the results

1. Define the Region We saved the region with the file name `FIGA` so all we have to do is enter this file name as a command.

2. Define Boundary Conditions We could have saved the previous set of conditions as an M-file, but we neglected to do that so we do it again. Because several of the boundaries have the same Neumann condition, it is advantageous to do `Edit/Select All`, set the conditions to Neumann with $\partial u/\partial n = 0$, and reset the two with Dirichlet conditions afterward. If we save this with the filename 'FIG_BC,' we can do steps 1 and 2 from that file.

3. Enter Values of Equation Parameters From `PDE/PDE Specifications`, we select `Parabolic`, and make d = 1, c = 1, a = 0, and f = 0 to match our equation.

4. Initialize the Mesh The easiest way to do this is with the triangular-shaped icon in the toolbar. We see the same mesh as before.

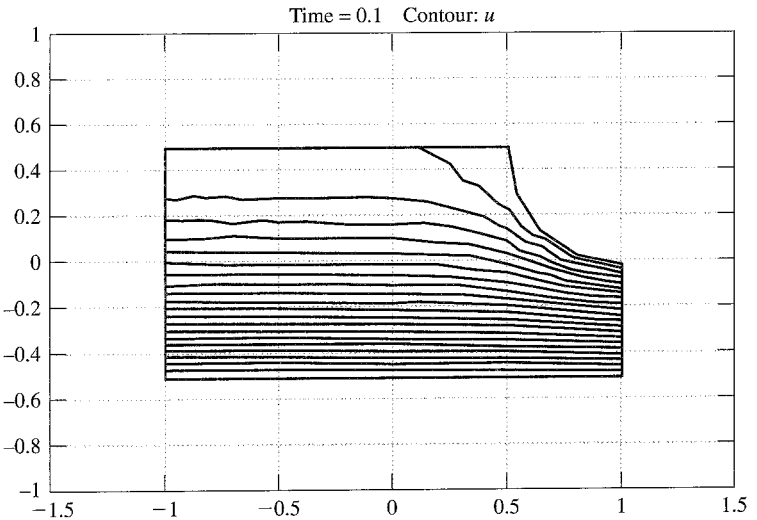
5. Enter Initial Temperature and List of Times This is done through the `Solve/Parameters/Solve Parameters` combination. We enter `u0 = 0` (the default), and enter into the time field 0:0.1:0.1 to obtain the solution after one-tenth of a second. (We will revise this after seeing this solution to find the temperatures within the object after 0.2, 0.4, 0.8 and 10.0 seconds.)

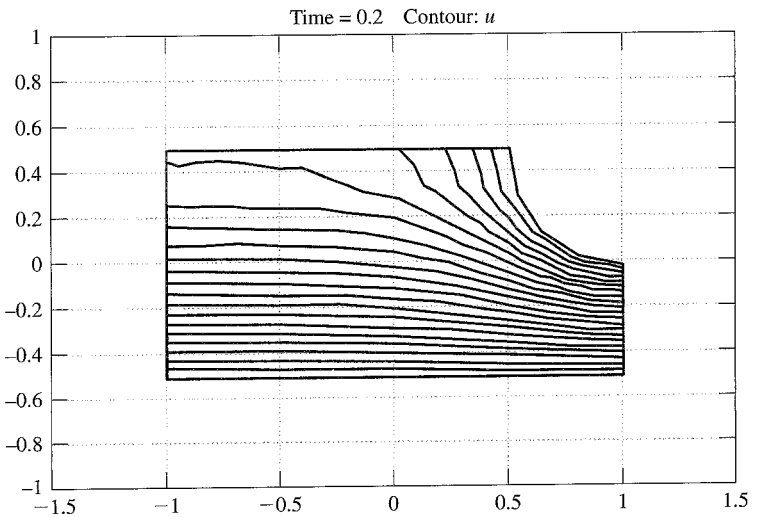
6. Solve the Equation We have many options here. Clicking on the = icon gives a color image similar to that from our elliptical example, except the temperatures are lower. Getting the isotherms is a better way to see the temperature distribution. This is accomplished by `Plot/Parameters` and then choosing only `Contour`.

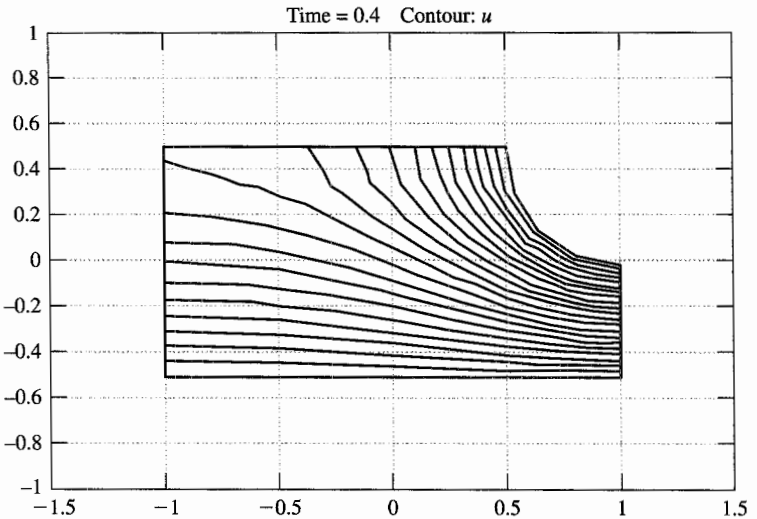
We repeated step 6 with different ending times to see how the isotherms change over time. At $t = 10.0$, the temperatures are essentially at steady state. (Smaller values for **c** in the equation delay the time to reach equilibrium.)

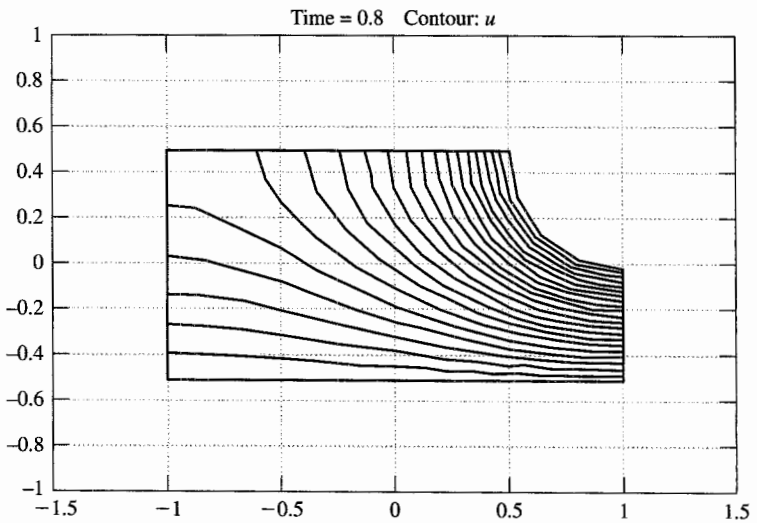
The figures show the isotherms for the sequence of ending times. By counting the number of isotherms, we estimate the temperature at the origin (0, 0) to be

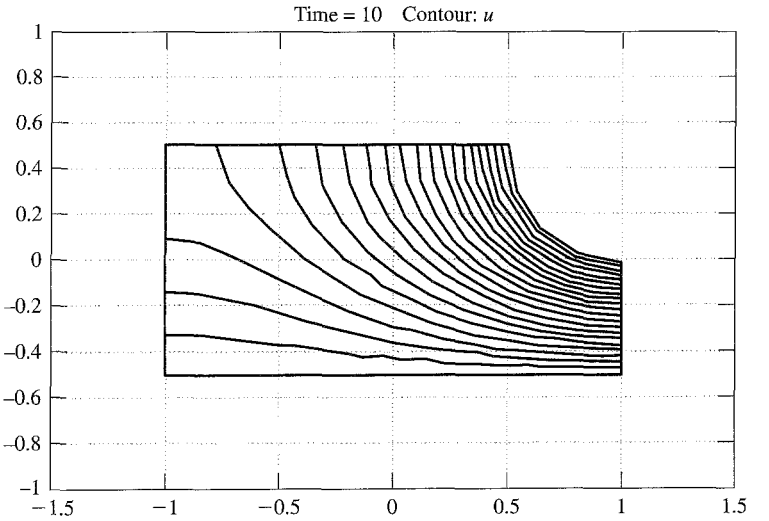
t:	0.1	0.2	0.4	0.8	10.0
temp:	28°	43°	57°	66°	68.6°






Time = 0.4 Contour: u
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
−1.5
−1
−0.5
0
0.5
1
1.5


Time = 0.8 Contour: u
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
−1.5
−1
−0.5
0
0.5
1
1.5



The Wave Equation

We will only outline how finite elements are applied to the wave equation, because this topic is too complex for full coverage here. Just as for the heat equation, finite elements are used for the space region and finite differences for time derivatives. We will develop only the vibrating string case (one dimension); two or three space dimensions are handled analogously but are harder to follow.

The equation that is usually solved is a more general case of the simple wave equation we have been discussing. In engineering applications, damping forces that serve to decrease the amplitude of the vibrations are important, and external forces that excite the system are usually involved. We therefore use, for a 1-D case, this equation for the displacement of points on the vibrating string, $y(x, t)$:

$$\frac{\partial}{\partial x}\left[T(x)\frac{\partial y}{\partial x}\right] - h(x)\frac{\partial y}{\partial t} + F(x, t) = \frac{w(x)}{g}\frac{\partial^2 y}{\partial t^2}. \tag{9.65}$$

Here T represents the tension, which is allowed to vary with x; h represents a damping coefficient that opposes motion in proportion to the velocity; F is the external force; and w/g is the mass density. There are boundary conditions (at $x = a$ and $x = b$) as well as initial conditions that specify initial displacements and velocities.

The approach is essentially identical to that used for unsteady-state heat flow: Apply finite elements to x and finite differences to the time derivatives. We will use linear one-dimensional elements, so we subdivide $[a, b]$ into portions (elements) that join at points that we call nodes. Within each element, we approximate $y(x, t)$ with $v(x, t)$,

$$y(x, t) \approx v(x, t) = N_L c_L + N_R c_R, \tag{9.66}$$

where c_L and c_R are the approximations to the displacements at the nodes at the left and right ends of a typical linear element. The N's are shape functions (in this 1-D case, we have called them "hat functions").

By using the Galerkin procedure, we can get this integral equation, which we will eventually transform into the element equations:

$$\begin{aligned}\int_{\mathrm{L}}^{\mathrm{R}} N_i\left(\frac{W}{g}\right) y_{tt}\,dx + \int_{\mathrm{L}}^{\mathrm{R}} N_i'(T) y_x\,dx + \int_{\mathrm{L}}^{\mathrm{R}} N_i(h) y_t\,dx \\ = \int_{\mathrm{L}}^{\mathrm{R}} N_i(F)\,dx + N_i(T) y_x\Big]_{x=\mathrm{R}} - N_i(T) y_x\Big]_{x=\mathrm{L}}, \qquad i = \mathrm{L}, \mathrm{R}.\end{aligned} \tag{9.67}$$

In Eq. (9.67) we have used subscript notation for the partial derivatives of y with respect to t and x and primes to represent the derivatives of the N's with respect to x (because the N's are functions of x only).

We now use Eq. (9.66) to find substitutions for y and its derivatives:

$$\begin{aligned} y(x, t) &\approx N_{\mathrm{L}} c_{\mathrm{L}} + N_{\mathrm{R}} c_{\mathrm{R}}, \\ \frac{\partial y}{\partial x} &\approx N_{\mathrm{L}}' c_{\mathrm{L}} + N_{\mathrm{R}}' c_{\mathrm{R}}, \\ \frac{\partial y}{\partial t} &\approx N_{\mathrm{L}} \dot{c}_{\mathrm{L}} + N_{\mathrm{R}} \dot{c}_{\mathrm{R}}, \\ \frac{\partial^2 y}{\partial t^2} &\approx N_{\mathrm{L}} \ddot{c}_{\mathrm{L}} + N_{\mathrm{R}} \ddot{c}_{\mathrm{R}}. \end{aligned} \tag{9.68}$$

Here we employ the dot notation for time derivatives. (The c's vary with time, of course, but the N's do not.)

We now substitute from Eqs. (9.68) into Eq. (9.67) to get a pair of equations for each element (we write them in matrix form):

$$\begin{aligned} &[M]\{\ddot{c}\} + [C]\{\dot{c}\} + [K]\{c\} = \{b\}, \\ &\left.\begin{aligned} M_{ij} &= \int_{\mathrm{L}}^{\mathrm{R}} N_i\left(\frac{w}{g}\right) N_j\,dx, \\ C_{ij} &= \int_{\mathrm{L}}^{\mathrm{R}} N_i(h) N_j\,dx, \\ K_{ij} &= \int_{\mathrm{L}}^{\mathrm{R}} N_i'(T) N_j'\,dx, \end{aligned}\right\} \quad i, j = \mathrm{L}, \mathrm{R}, \\ &b_i = \int_{\mathrm{L}}^{\mathrm{R}} N_i(F)\,dx + N_i(T)\frac{\partial y}{\partial x}\Big]_{x=\mathrm{R}} - N_i(T)\frac{\partial y}{\partial x}\Big]_{x=\mathrm{L}}, \qquad i = \mathrm{L}, \mathrm{R}. \end{aligned} \tag{9.69}$$

We will replace the time derivatives with finite differences, selecting central differences because they worked so well in the finite-difference solution to the simple wave equation. Thus we get

$$[M]\frac{\{c^{m+1} - 2c^m + c^{m-1}\}}{(\Delta t)^2} + [C]\frac{\{c^{m+1} - c^{m-1}\}}{2\Delta t} + [K]\{c^m\} = \{b^m\}. \quad (9.70)$$

Now we solve Eq. (9.70) for $\{c^{m+1}\}$:

$$\left(\frac{1}{(\Delta t)^2}[M] + \frac{1}{2\Delta t}[C]\right)\{c^{m+1}\} = \left(\frac{2}{(\Delta t)^2}[M] - [K]\right)\{c^m\} - \left(\frac{1}{(\Delta t)^2}[M] - \frac{1}{2\Delta t}[C]\right)\{c^{m-1}\} + \{b^m\}. \quad (9.71)$$

Notice that we need two previous sets of displacements to advance to the new time, t_{m+1}. We faced this identical problem when we solved the simple wave equation with finite differences, and we solve it in the same way. We use the initial velocities (given as one of initial conditions) to get $\{c^{-1}\}$to start the solution:

$$\{c^{-1}\} = \{c^1\} - 2\Delta t\{g(x)\}, \quad (9.72)$$

where $\{g(x)\}$ is the vector of initial velocities. [In view of our earlier work, we expect improved results if we use a weighted average of the g-values if the $g(x)$'s are not constants.]

We have not specifically developed the formulas for the components of the matrices and vector of Eqs. (9.69), but they are identical to those we derived when we applied finite elements to boundary-value problems in Section 9.2 because we will take out w, h, T, and F as average values within the elements. So we just copy from Section 9.2:

$$\begin{aligned}
M_{11} &= M_{22} = \left(\frac{w}{g}\right)\frac{\Delta}{3}, & M_{12} &= M_{21} = \left(\frac{w}{g}\right)\frac{\Delta}{6},\\
C_{11} &= C_{22} = h\frac{\Delta}{3}, & C_{12} &= C_{21} = h\frac{\Delta}{6},\\
K_{11} &= K_{22} = \frac{T}{\Delta}, & K_{12} &= K_{21} = -\frac{T}{\Delta},\\
b_1 &= F\frac{\Delta}{2} - \left[T\frac{\partial y}{\partial x}\right]_{x=\mathrm{L}} & b_2 &= F\frac{\Delta}{2} + \left[T\frac{\partial y}{\partial x}\right]_{x=\mathrm{R}}.
\end{aligned} \quad (9.73)$$

In this set, Δ represents the length of the element.

We now have everything we need to construct the element equations. Except for the end elements (and then only if the boundary conditions involve the gradient), the gradient terms in Eqs. (9.73) cancel between adjacent elements. Assembly in this case is very simple because there are always two elements that share each node (except at the ends).

What advantage is there to finite elements over finite differences? The major one is that we can use nodes that are unevenly spaced without having to modify the procedure. The advantage becomes really significant in two- and three-dimensional situations, but the other side of the coin is that solving the equations for each time step is not easy.

Solving the Wave Equation with MATLAB

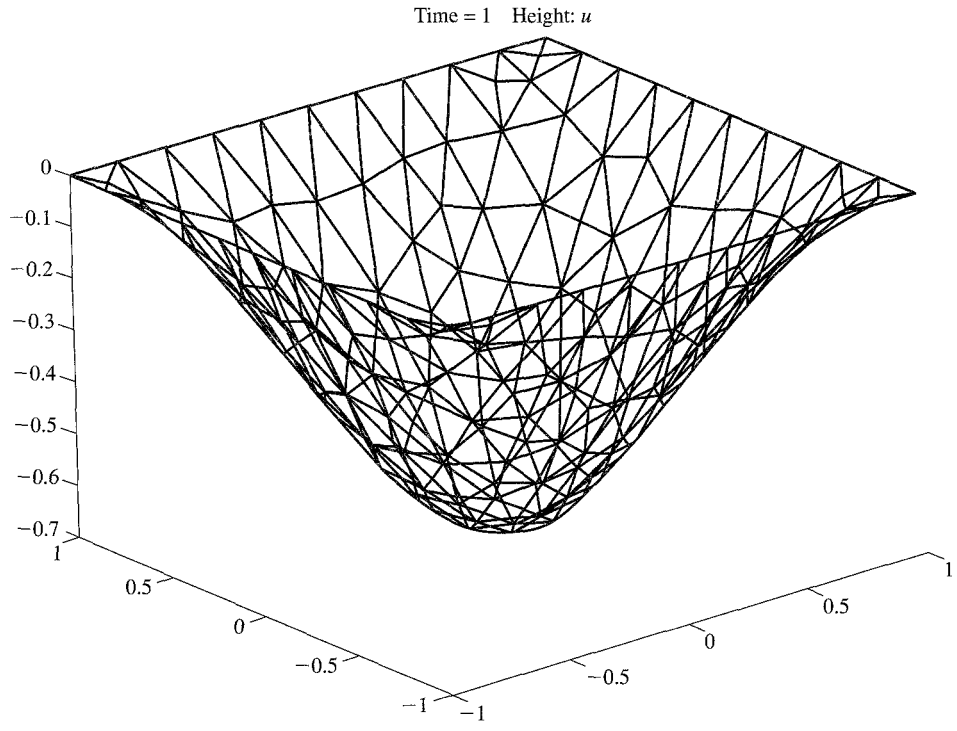
The wave equation is a hyperbolic partial-differential equation. Lets see how MATLAB's PDE Toolbox handles an example. We will solve Example 8.14 by FEM. (The vibrating string problem can be solved with `pdep`, available in the student edition.)

The steps in the procedure are identical to those for a parabolic equation except for step five:

1. Define the region.
2. Define the boundary conditions.
3. Enter the values for the parameters of the equation.
4. Establish a mesh of triangular elements.
5. Enter the initial values for u, $\partial u/\partial t$, and a list of times for which the solution is computed.
6. Solve the problem and display the results.

Example 8.14 finds the displacements of a square flexible membrane that has an initial displacement but zero initial velocity. We will put the center of the square at the origin rather than a corner. This changes the initial displacement function to $(1 - x^2)(1 - y^2)$.

1. We draw the square with `pderect ([-1 1 -1 1])` and we see the square in the figure window. It is labeled `SQ1`.
2. All boundaries are at $u = 0$. Doing `Boundary/Boundary Mode` shows the region in red. This means that the Dirichlet conditions with $u = 0$ are automatically supplied. (We can verify this by double-clicking on a side.)
3. We do `PDE/PDE Specification` and fill in the dialog box to have $c = 1$, $a = 0, f = 0$, and $d = 1$.
4. Clicking on the triangular icon creates a coarse mesh of triangles. We will stay with this coarse mesh to make it easier to see how the individual elements change with time. A finer mesh would give a more accurate solution.
5. We do `Solve/Parameters` and fill in the dialog box with `Time = 0:0.2:1` and `u(t0) = (1-x.^2).*(1-y.^2)`.
6. We are now ready for the solution. For this problem, seeing the results as a "movie" is best. So we do `Plot/Parameters` and select only `Height (3-D Plot)` and `Animation` in the dialog box. When we click on `Plot`, we see the membrane go from its initial bubblelike position to its mirror image on the other side of the (x, y) plane and back again repeatedly. The animation repeats itself several times. This figure shows the final position that is reached after one second.



Exercises

Section 9.1

1. Show that the integrand of Eq. (9.4) is equivalent to Eq. (9.3) if the Euler–Lagrange condition is used. This means that Eq. (9.4) is the functional for any second-order boundary-value problem of the form

$$y'' + Q(x)y = F(x),$$

subject to Dirichlet boundary conditions

$$y(a) = A, \qquad y(b) = B,$$

where A and B are constants.

▶ **2.** Use the Rayleigh–Ritz method to approximate the solution of

$$y'' = 3x + 1, \qquad y(0) = 0, \qquad y(1) = 0,$$

using a quadratic in x as the approximating function. Compare to the analytical solution by graphing the approximation and the analytical solution.

3. Repeat Exercise 2, but this time, for the approximating function, use

$$ax(x - 1) + bx^2(x - 1).$$

Show that this reproduces the analytical solution.

4. Another approximating function that meets the boundary condition of Exercise 3 is

$$ax(x - 1) + bx(x - 1)^2.$$

Use this to solve by the Rayleigh–Ritz technique.

5. Suppose that the boundary conditions in Exercise 3 are $y(0) = 1$, $y(1) = 3$. Modify the procedure of Exercise 3 to get a solution.

▶ **6.** Solve Exercise 2 by collocation, setting the residual to zero at $x = \frac{1}{3}$ and $x = \frac{2}{3}$. Compare this solution to that from Exercise 2.

7. Repeat Exercise 6, except now use different points within [0, 1] for setting the residual to zero. Are some pairs of points better than others?

8. Repeat Exercise 3, but now use collocation. Does it matter where within [0, 1] you set the residual to zero?

9. Use Galerkin's technique to solve Exercise 2. Is the same solution obtained?

10. Repeat Exercise 3, but now use Galerkin.

Section 9.2

11. Suppose that, in Eq. (9.25), $Q(x) = \sin(x)$ and $F(x) = x^2 + 2$. For an element that occupies [0.33, 0.45],

▶a. Find N_L and N_R of Eq. (9.26).
b. Write out the integrals of Eq. (9.28).
c. Write out the element equations (9.36).
▶d. Compute the correct average values for Q and F.

12. Repeat Exercise 11 for two adjacent elements. These occupy [0.21, 0.33) and [0.45, 0.71].

13. Assemble the three pairs of element equations of Exercises 11 and 12 to form a set of four equations with the nodal values at $x = 0.21$, $x = 0.33$, $x = 0.45$, and $x = 0.71$ as unknowns.

▶14. Solve by the finite-element method:

$$y'' + xy = x^3 - \frac{4}{x^3}, \quad y(1) = -1, \quad y(2) = 3.$$

Put nodes at $x = 1.2$, 1.5, and 1.75 well as at the ends of [1, 2]. Compare your solution to the analytical solution, which is $y = x^2 - 2/x$.

15. Repeat Exercise 14, except for the end condition at $x = 1$ of $y'(1) = 4$.

16. Repeat Exercise 14, but with more nodes. Place added nodes at $x = 1.1$, 1.3, 1.4, 1.65, and 1.9. Compare the errors with those of Exercise 14.

Section 9.3

17. Confirm that Eq. (9.45) is in fact the inverse of matrix M in Eq. (9.44).

18. Find M^{-1}, a, N, and $u(x, y)$ for these triangular elements:

a. Nodes: (1.2, 3.1), (−0.2, 4), (−2, −3); u-values at these nodes: 5, 20, 7; point where u is to be determined: (−1, 0)
b. Nodes: (20, 40), (50, 10), (5, 10); u-values at these nodes: 12.5, 6.2, 10.1; point where u is to be determined: (20, 20)
c. Nodes: (12.1, 11.3), (8.6, 9.3), (13.2, 9.3); u-values at these nodes: 121, 215, 67; point where u is to be determined: (10.6, 9.6)

19. Confirm that the sum of the entries in the first row of M^{-1} is equal to twice the area for each of the elements in Exercise 18.

▶20. Find the element equations for the element in part (c) of Exercise 18 if $Q = x^2y$ and $F = -x/y$ (these refer to Eq. 9.40). There are no derivative conditions on any of the element boundaries.

21. Solve Example 8.1 (Chapter 8) by finite elements. Place nodes at each corner and at the midpoints of the top and bottom edges, also at points 9, 12, and 14. Draw triangular elements whose vertices are at these nodes. Compare the answers at each node to those obtained with finite-difference approximations to the derivatives.

22. In Exercise 21, the temperatures in the top half of the slab are the same as those in the bottom half because of symmetry in the boundary conditions. Solve the problem for the top half only of the slab with the same nodes as in Exercise 21. (Along the horizontal midline, the gradient will be zero).

▶23. For a triangular element that has nodes at points (1.2, 3.2), (4.3, 2.7), and (2.4, 4.1), find the components of each matrix in the element equations [Eqs. (9.61) and (9.62)] if the material is aluminum.

24. For heat flow in one dimension, the governing equation is

$$k(\partial^2 u/\partial x^2) = c\rho(\partial u/\partial t).$$

Repeat the development of the analog of Eq. (9.62) for this case.

25. Use the equations that you derived in Exercise 24 to solve Exercise 31 of Chapter 8. Place the nodes exactly as those used in the finite-difference solution. Are the resulting equations the same?

26. Use finite elements to solve Exercise 34 of Chapter 8. Place interior nodes at three arbitrarily selected points (but do not make these symmetrical). Create triangular elements with these nodes and the four corner points. Set up the element equations, assemble, and solve for four time steps. Use the resulting nodal temperatures to estimate the same set of temperatures that were computed by finite differences. Compare the two methods of solving the problem.

27. Solve Example 8.6 (Chapter 8) by finite elements. Place nodes strategically along the edges and within the slab so there are a total of 14 or 15 nodes. Use triangular elements. Compare the solution to that obtained with finite-difference approximations. (You may want to take

advantage of symmetry in the boundary conditions to solve the problem with fewer elements.)

▶**28.** Rederive Eq. (9.64), but now for the Crank–Nicolson method.

29. Repeat Exercise 28, but now for the theta method.

30. Set up the finite-element equations for advancing the solution to part (a) of Exercise 50 of Chapter 8.

▶**31.** Set up the finite-element equations for starting the solution to part (a) of Exercise 50 of Chapter 8. Do this first for the analog of Eq. (8.42) and then for the analog of Eq. (8.49).

▶**32.** If we were to solve part (c) of Exercise 50 of Chapter 8, would there be an advantage to using shorter elements near the middle of the string where the displacements depart more from linearity?

33. Solve, using finite elements, Example 8.14, except with initial conditions of

$$u(x, y) = 0, \qquad u_t(x, y) = x^2(2 - x)y^2(2 - y).$$

34. Repeat Exercise 33, but with these initial conditions:

$$u(x, y) = x^2(2 - x)y^2(2 - y), \qquad u_t(x, y) = 0.$$

35. Solve Exercise 58 of Chapter 8 using finite elements. Where do you think interior nodes should be placed if there are

a. 6 of them?
b. 12 of them?

Compare the solutions from these two cases to that from the finite-difference method.

36. Solve Exercise 60 of Chapter 8 by finite elements, placing five interior nodes at points that you think are best. Justify your choice of nodal positions.

▶**37.** Using the isotherm plots from the MATLAB solution to a parabolic equation, count the isotherms (there are 20 curves) to see how the temperature at the upper-left corner varies with time. Plot these. Can you find an equation that fits?

Applied Problems and Projects

APP1. Use the Internet to find software that solves both ordinary- and partial-differential equations. Can you find any that use the finite-element method? (*Hint*: Try http://gams.nist.gov/ and search the topic: partial differential equations.)

APP2. Write a computer program that uses finite elements to solve the vibrating string problem. Test it by solving Example 8.13.

APP3. Repeat APP2, but now for the heat equation, Eq. (9.56). Test it by solving Exercises 26 and 27.

APP4. Write a computer program (using your favorite language) to solve a two-dimensional elliptic partial-differential equation. Allow for both Dirichlet and non-Dirichlet boundary conditions. Have the program read in the required data from a file. Provide function procedures to compute the values for $f(x, y)$ and $q(x, y)$. Here is a suggested data structure:

NN = the total number of nodes

NK = the number of boundary nodes with Dirichlet conditions. (NN − NK = number of nodes whose values are not specified, that is, the interior nodes and those boundary nodes whose values are not specified.)

VX (NN) = an array to hold the x-values for all nodes in the order that nodes are numbered. There is an advantage if the nodes whose u-values are specified are numbered so as to follow those nodes where the u-values must be computed.

VY (NN) = an array to hold the corresponding y-values for all nodes

M (NE, 4, 3) = an array to hold the element matrices. The first subscript indicates the element number. The second and third subscripts indicate the row and column of the matrix. The fourth row holds the node numbers for nodes in this element in counterclockwise order. There is an advantage if the unspecified nodes come before the nodes whose u-values are known.

UU (NN) = an array to hold unknown and known u-values at nodes in order of the node number. Zeros may be used as fillers for unknown u-values.

AE (NE) = an array to hold areas of the elements

F(NE) = an array to hold average f-values for each element

Q(NE) = an array to hold average q-values for each element

A(NN, NN + 1) = the system matrix

Here is what your logic might look like:

1. Read in NN, NE, NU.
2. Read in (x, y) values for the nodes, storing in VX and VY.
3. Read in node numbers for each element in turn (nodes should be in counterclockwise order), storing in the fourth row of the element matrices.
4. Read in the unknown and known u-values for each node.
5. Compute average values for f and q in each element. (You may prefer to evaluate these at the centroid of the element.) Store in F and Q.
6. Read in the known u-values, storing in UK.
7. Compute the area for each element and its inverse [Eq. (9.49), the area from the first row elements].
8. Find the element equations and add the appropriate values to the system matrix.
9. Adjust the system matrix for non-Dirichlet boundary conditions. (You may want to have the user input the a and b values for these and the node numbers at the ends of the element boundary where this applies. Alternatively, these could have been read in with the other parts of the data.)
10. Adjust the system matrix for Dirichlet conditions using values from the UU array.
11. Solve the system.
12. Display the u-values for each node.

APP5. Write and test a program that solves the vibrating membrane problem using the finite-element method.

APP6. In developing the element equations, a number of integrals must be evaluated [see Eq. (9.51)]. For triangular elements, these are very easy to get: Each is just the area divided by a number. These simple triangular elements that we have discussed are called *C^0-linear elements.*

Other types of elements besides these simple triangles are sometimes useful. For example, connecting the nodes with lines that form quadrilateral elements can cut the number of elements almost in half. For these, the integrals are not so readily evaluated.

Even if we stay with triangular elements, the accuracy of the solution is improved if we add one node within each of the three sides. Such additional nodes can even permit the "triangle" to have curved sides. Such a more elaborate triangular element is called a *C^0-quadratic element.* This idea can be extended to add more than three nodes to the triangle, and additional nodes are sometimes added to quadrilateral elements.

For all of these more elaborate elements, the shape functions no longer have a "flat top" like that sketched in Figure 9.7. The normal procedure for these is to employ Gaussian quadrature in which a weighted sum of the integrand at certain points, called *Gauss-points*, approximates the integral quite well.

For a square region with opposite corners at $(-1, -1)$ and $(1, 1)$, these Gauss-points are at $x = \pm\sqrt{3}/3$, $y = \pm\sqrt{3}/3$, as given in Table 5.13. For a region that is a triangle with vertices at (0, 0), (1,0), (0, 1), there are three Gauss-points at $\left(\frac{1}{6}, \frac{1}{6}\right)$, $\left(\frac{2}{3}, \frac{1}{6}\right)$, and $\left(\frac{1}{6}, \frac{2}{3}\right)$, each weighted with $\frac{1}{3}$. For elements that do not conform to these basic cases, they must be mapped to coincide with them. Where are the Gauss-points for

a. A triangle whose vertices are $(-1, 3)$, (7, 1), and (2, 7)?
b. A quadrilateral whose vertices are (1, 2), (5, −1), (6, 3), (3, 5)?

APP7. Use MATLAB's PDE Toolbox to solve several of the examples of Chapters 8 and 9. Define the regions both with the mouse on the graphical user interface and also by using commands.

APP8. There are other software packages that let you solve engineering and scientific problems with FEA. Two of these are ALGOR and MSC/Nastran. Find information on these and compare their capabilities with that of MATLAB's PDE Toolbox. The Internet is a good place to get some information. Your library may have books on them, too.

APP9. Search for information on finite elements with a Web browser. Write a report on what you find.