Chapter 7
COMPUTER IMPLEMENTATION

7.1 NUMERICAL INTEGRATION

7.1.1 Background

The finite element method, like most numerical methods, converts a continuum problem to a discrete one (i.e., converting a system with an infinite number of degrees of freedom into one with a finite number of degrees of freedom). The finite element model of a system ultimately represents a set of algebraic equations among the values of the dependent variables of the system at the selected nodes of the domain. The coefficients of the algebraic equations are typically integrals of approximation functions multiplied by the data of the problem. Exact evaluation of these integrals is not always possible because of the algebraic complexity of the data $a(x)$, $b(x)$, $c(x)$, and $f(x)$ in the mathematical model. In such cases, it is natural to seek numerical evaluation of these integral expressions. Numerical evaluation of the coefficient matrices is also useful in problems with constraints, where reduced integration techniques are used (e.g., the reduced integration element of the Timoshenko beam theory in Section 5.3).

Numerical evaluation of integrals, called *numerical integration* or *numerical quadrature,* involves approximation of the integrand by a polynomial of sufficient degree, because the integral of a polynomial can be evaluated exactly. For example, consider the integral,

$$I = \int_{x_a}^{x_b} F(x)\, dx \tag{7.1.1}$$

We approximate the function $F(x)$ by a polynomial

$$F(x) \approx \sum_{I=1}^{N} F_I \psi_I(x) \tag{7.1.2}$$

where F_I denotes the value of $F(x)$ at the Ith point of the interval $[x_a, x_b]$ and $\psi_I(x)$ are polynomials of degree $N-1$. The representation (7.1.2) can be viewed as the finite element interpolation of $F(x)$, where F_I is the value of the function at the Ith node. The interpolation can be of the Lagrange type or the Hermite type.

Substitution of (7.1.2) into (7.1.1) and evaluation of the integral gives an approximate value of I. For example, suppose that we choose linear interpolation of $F(x)$. Then $N=2$, $\psi_1 = (x_b - x)/h$, $\psi_2 = (x - x_a)/h$, and

$$I = \frac{1}{2}h(F_1 + F_2), \quad F_1 = F(x_a), \quad F_2 = F(x_b) \tag{7.1.3}$$

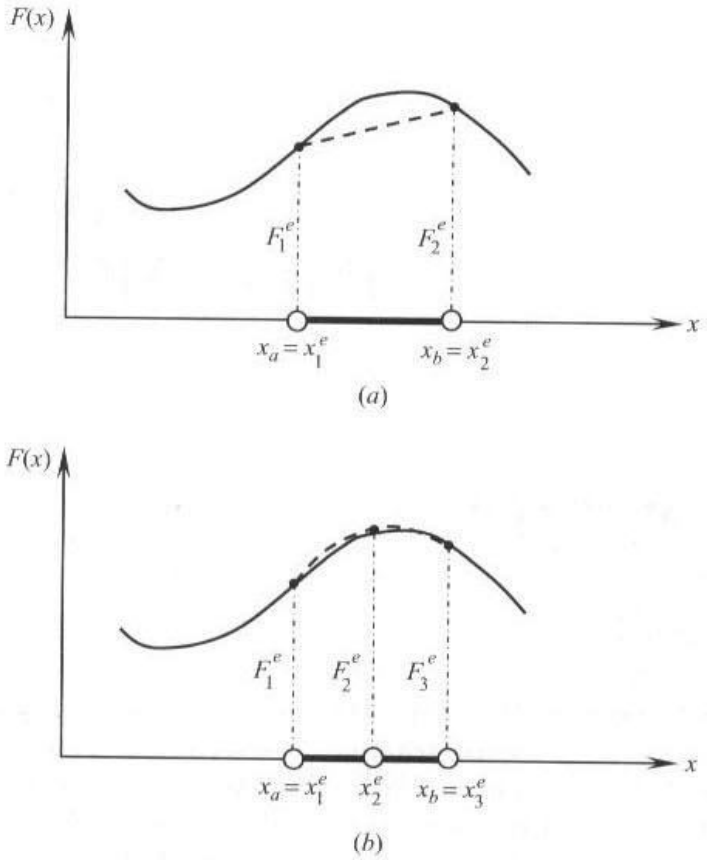


Figure 7.1.1 Approximate evaluation of an integral using the trapezoidal rule (*a*) two-point formula and (*b*) three-point formula.

Thus, the value of the integral is given by the area of a trapezoid used to approximate the area under the function $F(x)$ (see Fig. 7.1.1). Equation (7.1.3) is known as the *trapezoidal rule* of numerical integration. If we use the Lagrange quadratic interpolation of $F(x)$, we obtain

$$I = \frac{1}{6}h(F_1 + 4F_2 + F_3), \quad F_1 = F(x_a), \quad F_2 = F(x_a + 0.5h), \quad F_3 = F(x_b) \tag{7.1.4}$$

which is known as *Simpson's one-third rule*.

Equations (7.1.3) and (7.1.4) represent forms of numerical quadrature formulae. In general, a quadrature formula has the form

$$I = \int_{x_a}^{x_b} F(x)\, dx \approx \sum_{I=1}^{r} F(x_I) W_I \tag{7.1.5}$$

where x_I are called the *quadrature points* and W_I are the *quadrature weights*. These formulae require functional evaluations, multiplications, and additions to obtain the numerical value of the integral. They yield exact values of the integral whenever $F(x)$ is a polynomial of order $r - 1$.

In this section, we describe several numerical integration techniques and formulations in which the geometry as well as the dependent variables are approximated using different degrees of polynomials. We begin with the discussion of a local coordinate system.

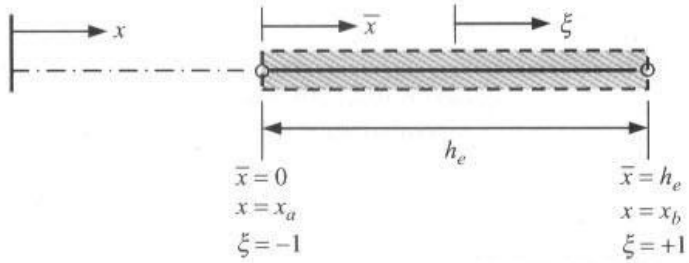


Figure 7.1.2 Global coordinate x, local coordinate $\bar{x}$, and normalized coordinate ξ.

7.1.2 Natural Coordinates

Of all the quadrature formulae, as will be discussed in the subsequent sections, the Gauss–Legendre one is the most commonly used. The details of the method itself will be discussed shortly. The formula requires the integral to be cast as one to be evaluated over the interval $[-1, 1]$. This requires the transformation of the problem coordinate x to a local coordinate ξ such that (see Fig. 7.1.2):

$$\text{when} \quad x = x_a, \quad \xi = -1; \quad \text{when} \quad x = x_b, \quad \xi = 1$$

The transformation between x and ξ can be represented by the linear "stretch" transformation

$$x = a + b\xi$$

where a and b are constants to be determined such that the above conditions hold:

$$x_a = a + b(-1), \quad x_b = a + b(1)$$

Solving for a and b, we obtain

$$b = \frac{1}{2}(x_b - x_a) = \frac{1}{2}h_e, \quad a = \frac{1}{2}(x_b + x_a) = x_a + \frac{1}{2}h_e$$

Hence, the transformation takes the form

$$x(\xi) = x_a + \frac{1}{2}h_e(1 + \xi) \tag{7.1.6}$$

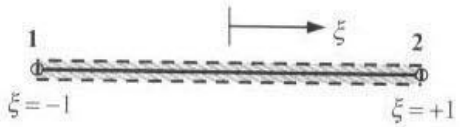
where x_a denotes the global coordinate of the left-end node of the element Ω_e and h_e is the element length (see Fig. 7.1.2).

The local coordinate ξ is called the *normal coordinate* or *natural coordinate*, and its values always lie between -1 and 1, with its origin at the center of the element. The local coordinate ξ is useful in two ways: It is (a) convenient in constructing the interpolation functions and (b) required in numerical integration using Gauss–Legendre quadrature.

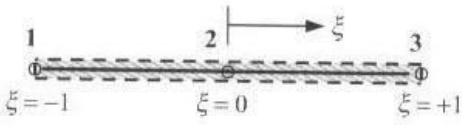
The derivation of the Lagrange family of interpolation functions in terms of the natural coordinate ξ is made easy by the following interpolation property of the approximation functions:

$$\psi_i(\xi_j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \tag{7.1.7}$$

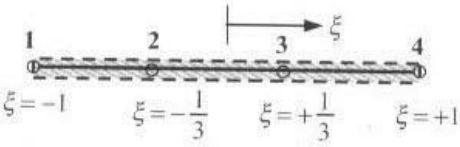
where ξ_j is the ξ coordinate of the jth node in the element. For an element with n nodes, ψ_i $(i = 1, 2, \ldots, n)$ are polynomials of degree $n - 1$. To construct ψ_i satisfying (7.1.7), we



$$\psi_1(\xi) = \tfrac{1}{2}(1-\xi), \qquad \psi_2(\xi) = \tfrac{1}{2}(1+\xi)$$



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



$$\psi_1(\xi) = -\tfrac{9}{16}(1-\xi)(\tfrac{1}{9}-\xi^2), \quad \psi_2(\xi) = \tfrac{27}{16}(1-\xi^2)(\tfrac{1}{3}-\xi)$$
$$\psi_3(\xi) = \tfrac{27}{16}(1-\xi^2)(\tfrac{1}{3}+\xi), \quad \psi_4(\xi) = -\tfrac{9}{16}(1+\xi)(\tfrac{1}{9}-\xi^2)$$

Figure 7.1.3 Lagrange family of one-dimensional interpolation functions in terms of the normalized coordinate.

proceed as follows: For each ψ_i, we form the product of $n-1$ linear functions $\xi - \xi_j$ ($j = 1, 2, \ldots, i-1, i+1, \ldots, n$; $j \neq i$):

$$\psi_i = c_i(\xi - \xi_1)(\xi - \xi_2)\cdots(\xi - \xi_{i-1})(\xi - \xi_{i+1})\cdots(\xi - \xi_n)$$

Note that ψ_i is zero at all nodes except the ith. Next, we determine the constant c_i such that $\psi_i = 1$ at $\xi = \xi_i$:

$$c_i = [(\xi_i - \xi_1)(\xi_i - \xi_2)\cdots(\xi_i - \xi_{i-1})(\xi_i - \xi_{i+1})\cdots(\xi_i - \xi_n)]^{-1}$$

Thus, the interpolation function associated with node i is

$$\psi_i(\xi) = \frac{(\xi - \xi_1)(\xi - \xi_2)\cdots(\xi - \xi_{i-1})(\xi - \xi_{i+1})\cdots(\xi - \xi_n)}{(\xi_i - \xi_1)(\xi_i - \xi_2)\cdots(\xi_i - \xi_{i-1})(\xi_i - \xi_{i+1})\cdots(\xi_i - \xi_n)} \tag{7.1.8}$$

The linear, quadratic, and cubic Lagrange interpolation functions in terms of the natural coordinate (for equally spaced nodes) are shown in Fig. 7.1.3.

7.1.3 Approximation of Geometry

We wish to use the Gauss–Legendre quadrature to numerically evaluate all integrals in the finite element method. The integrals are generally expressed in terms of the coordinate appearing in the problem description (like x or r). We shall call x and r as the *problem coordinates* or global coordinates. The Gauss–Legendre quadrature requires us to express the integral in terms of ξ over the interval -1 to $+1$. We assume a relation (or transformation) between the problem coordinate x and natural coordinate ξ in the form

$$x = f(\xi) \tag{7.1.9}$$

where f is assumed to be a one-to-one transformation. An example of $f(\xi)$ is provided by (7.1.6):

$$f(\xi) = x_a + \frac{1}{2}h_e(1+\xi)$$

In this case, $f(\xi)$ is a linear function of ξ. Hence, a straight line is transformed into a straight line.

It is natural to think of approximating the geometry in the same way as we approximated a dependent variable. In other words, the transformation $x = f(\xi)$ can be written as

$$x = \sum_{i=1}^{m} x_i^e \hat{\psi}_i^e(\xi) \tag{7.1.10}$$

where x_i^e is the global coordinate of the ith node of the element Ω_e and $\hat{\psi}_i^e$ are the Lagrange interpolation functions of degree $m-1$. When $m=2$, we have a linear transformation, and Eq. (7.1.10) is exactly the same as (7.1.6). When $m=3$, Eq.(7.1.10) expresses a quadratic relation between x and ξ. The functions $\hat{\psi}_i^e$ are called *shape functions* because they are used to express the geometry or shape of the element. When the element is a straight line, the mapping is linear (because two points, x_1^e and x_2^e, are sufficient to define a line).

The transformation (7.1.10) allows us to rewrite integrals involving x as those in terms of ξ:

$$\int_{x_a}^{x_b} F(x)\,dx = \int_{-1}^{1} \hat{F}(\xi)\,d\xi, \quad \hat{F}(\xi)\,d\xi = F(x(\xi))\,dx \tag{7.1.11}$$

so that the Gauss–Legendre quadrature can be used to evaluate the integral over $[-1, 1]$. The differential element dx in the global coordinate system x is related to the differential element $d\xi$ in the natural coordinate system ξ by

$$dx = \frac{dx}{d\xi} d\xi = J_e\, d\xi$$

where J_e is called the *Jacobian* of the transformation. We have

$$J_e = \frac{dx}{d\xi} = \frac{d}{d\xi}\left(\sum_{i=1}^{m} x_i^e \hat{\psi}_i^e\right) = \sum_{i=1}^{m} x_i^e \frac{d\hat{\psi}_i^e}{d\xi} \tag{7.1.12}$$

For a linear transformation [i.e., when $m=2$ in (7.1.10)], we have

$$\hat{\psi}_1^e = \frac{1}{2}(1-\xi), \quad \hat{\psi}_2^e = \frac{1}{2}(1+\xi)$$

$$J_e = x_1^e\left(-\frac{1}{2}\right) + x_2^e\left(\frac{1}{2}\right) = \frac{1}{2}(x_2^e - x_1^e) = \frac{1}{2}h_e \tag{7.1.13}$$

It can be shown that $J_e = \frac{1}{2}h_e$ whenever the element is a straight line, irrespective of the degree of interpolation used in the transformation (7.1.10).

7.1.4 Isoparametric Formulations

Recall that a dependent variable u is approximated in an element Ω_e by expressions of the form

$$u(x) = \sum_{j=1}^{n} u_j^e \psi_j^e(x) \tag{7.1.14}$$

In general, the degree of approximation used to describe the coordinate transformation (7.1.10) is not equal to the degree of approximation (7.1.14) used to represent a dependable

variable, i.e., $\hat{\psi}_i^e \neq \psi_i^e$. In other words, two independent meshes of elements may be used in the finite element formulation of a problem: one for the approximation of the geometry x and the other for the interpolation of the dependent variable u. Depending on the relationship between the degree of approximation used for the coordinate transformation and that used for the dependent variable, the finite element formulations are classified into three categories:

1. Subparametric formulations: $m < n$
2. Isoparametric formulations: $m = n$ (7.1.15)
3. Superparametric formulations: $m > n$

In subparametric formulations, the geometry is represented by lower-order elements than those used to approximate the dependent variable. An example of this category is provided by the Euler–Bernoulli beam element, where the Hermite cubic functions are used to approximate the deflection $w(x)$ and linear interpolation can be used, when straight beams are analyzed, to represent the geometry. In isoparametric formulations (which are the most common in practice), the same element is used to approximate the geometry as well as the dependent unknowns: $\psi_i(x) = \hat{\psi}_i(\xi)$. In the superparametric formulations, the geometry is represented with higher-order elements than those used to approximate the dependent variables. The superparametric formulation is seldom used in practice.

7.1.5 Numerical Integration

As discussed in the introduction, the evaluation of integrals of the form

$$\int_a^b F(x)\, dx \tag{7.1.16}$$

by exact means is either difficult or impossible owing to the complicated form of the integrand F. Numerical integration is also required when the integrand is to be evaluated inexactly (as in the Timoshenko beam element) or when the integrand is known only at discrete points (e.g., experimentally obtained data).

The basic idea behind all numerical integration techniques is to find a function $P(x)$, often a polynomial, that is both a suitable approximation of $F(x)$ and simple to integrate. The interpolating polynomials of degree n, denoted by P_n, which interpolate the integrand at $n + 1$ points of the interval $[a, b]$, often produce a suitable approximation and possess the desired property of simple integrability. An illustration of the approximation of the function $F(x)$ by the polynomial $P_4(x)$ that exactly matches the function $F(x)$ at the indicated base points is given in Fig. 7.1.4(a). The exact value of (7.1.16) is given by the area under the solid curve, while the approximate value

$$\int_a^b P_4(x)\, dx$$

is given by the area under the dashed curve. It should be noted that the difference (i.e., the error in the approximation) $E = F(x) - P_4(x)$ is not always of the same sign, and therefore the overall integration error may be small (because positive errors in one part cancel negative errors in other parts), even when P_4 is not a good approximation of F.

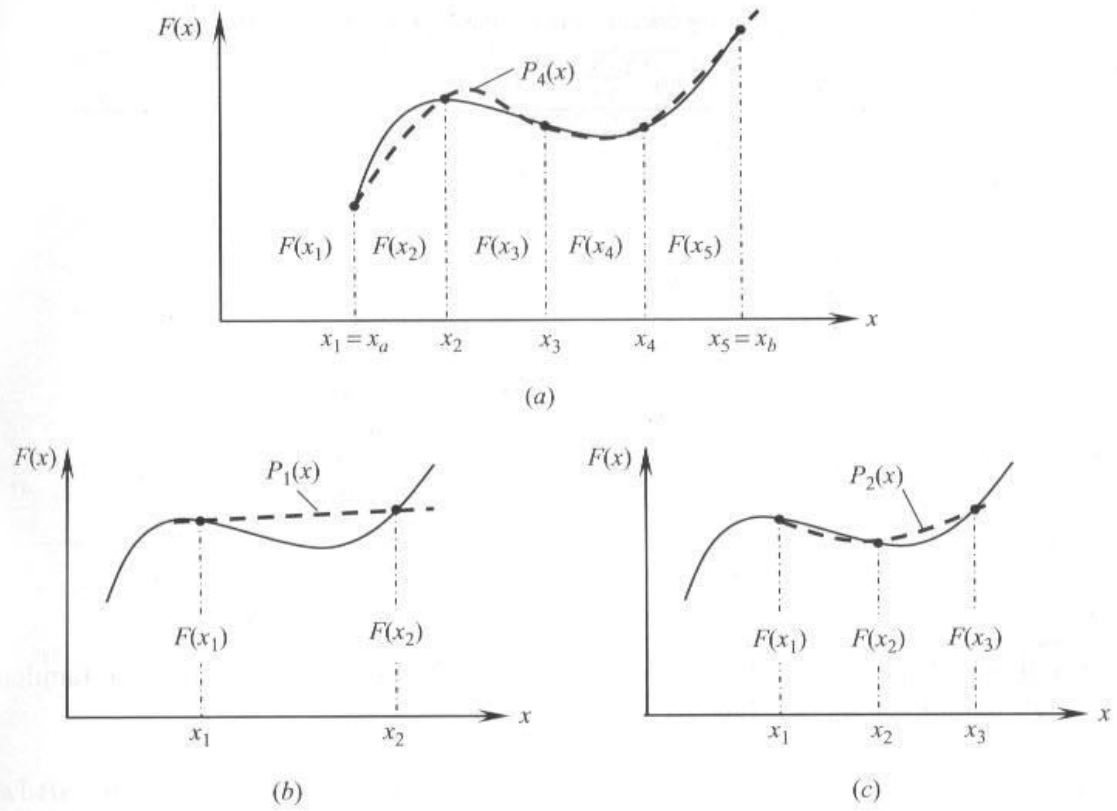


Figure 7.1.4 Numerical integration by the Newton–Cotes quadrature: (*a*) approximation of a function by $P_4(x)$; (*b*) the trapezoidal rule; and (*c*) Simpson's rule.

The commonly used numerical integration methods can be classified into two groups:

1. The Newton–Cotes formulae that employ values of the function at equally spaced points
2. The Gauss–Legendre quadrature formula that employs unequally spaced points

These two methods are described here.

The Newton–Cotes Quadrature. For r equally spaced base points, the *Newton–Cotes closed integration formula* is given by

$$\int_a^b F(x)\,dx = (b-a)\sum_{I=1}^{r} F(x_I)w_I \tag{7.1.17}$$

where w_I are the *weighting coefficients*, x_I are the *base points* that are equally spaced, and r is the number of base points (or $r-1$ is the number of intervals). Note that $r=1$ is a special case in which the number of base points as well as the number intervals is the same; in this case Eq. (7.1.17) gives the rectangle formula. For $r=2$, (7.1.17) gives the familiar *trapezoidal rule*, in which the required area under the solid curve in Fig. 7.1.4(*b*) is approximated by the area under the dotted straight line [i.e., $F(x)$ is approximated by $P_1(x)$]:

$$\int_{a=x_1}^{b=x_2} F(x)\,dx = \frac{b-a}{2}[F(x_1)+F(x_2)],\quad E=O(h^3),\quad h=b-a \tag{7.1.18}$$

where E denotes the error in the approximation and h is the uniform spacing between two base points. The notation $O(h)$, read as "order of h," is used to indicate the order of the

Table 7.1.1 Weighing coefficients for the Newton–Cotes formula.

r	w_1	w_2	w_3	w_4	w_5	w_6	w_7
1	1						
2	$\frac{1}{2}$	$\frac{1}{2}$					
3	$\frac{1}{6}$	$\frac{4}{6}$	$\frac{1}{6}$				
4	$\frac{1}{8}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{1}{8}$			
5	$\frac{7}{90}$	$\frac{32}{90}$	$\frac{12}{90}$	$\frac{32}{90}$	$\frac{7}{90}$		
6	$\frac{19}{288}$	$\frac{75}{288}$	$\frac{50}{288}$	$\frac{50}{288}$	$\frac{75}{288}$	$\frac{19}{288}$	
7	$\frac{41}{840}$	$\frac{216}{840}$	$\frac{27}{840}$	$\frac{272}{840}$	$\frac{27}{840}$	$\frac{216}{840}$	$\frac{41}{840}$

error in terms of the spacing h. For $r = 3$ (i.e., two intervals), (7.1.17) gives the familiar *Simpson's one-third rule* [see Fig. 7.1.4(*c*)]:

$$\int_{a=x_1}^{b=x_3} F(x)\,dx = \frac{b-a}{6}[F(x_1) + 4F(x_2) + F(x_3)], \quad E = O(h^5), \quad h = 0.5(b-a) \tag{7.1.19}$$

The weighting coefficients for $r = 1, 2, \ldots, 7$ are given in Table 7.1.1. Note that $\sum_{I=1}^{r} w_I = 1$. The base point location for $r = 1$ is $x_1 = a + \frac{1}{2}(b-a) = \frac{1}{2}(a+b)$. For $r > 1$, the base point locations are

$$x_1 = a, \quad x_2 = a + \Delta x, \ldots, \quad x_r = a + (r-1)\Delta x = b$$

and $\Delta x = (b-a)/(r-1)$. We note that when r is *odd* (i.e., when there is an even number of intervals or an odd number of base points), the formula is exact when $F(x)$ is a polynomial of degree r or less; when r is *even*, the formula is exact for a polynomial of degree $r-1$ or less. Odd-point formulas are frequently used because of their high order of accuracy [see Carnahan, Luther, and Wilkes (1969)].

The Gauss–Legendre Quadrature. In the Newton–Cotes quadrature, the base point locations have been specified. If the x_I are not specified, then there will be $2r+2$ undetermined parameters, $r+1$ weights w_I and $r+1$ base points x_I, which define a polynomial of degree $2r+1$. The Gauss–Legendre quadrature is based on the idea that the base points x_I and the weights w_I can be chosen so that the sum of the $r+1$ appropriately weighted values of the function yields the integral exactly when $F(x)$ is a polynomial of degree $2r+1$ or less. The Gauss–Legendre quadrature formula is given by

$$\int_a^b F(x)\,dx = \int_{-1}^{1} \hat{F}(\xi)\,d\xi \approx \sum_{I=1}^{r} \hat{F}(\xi_I) w_I \tag{7.1.20}$$

where w_I are the weight factors, ξ_I are the base points [roots of the Legendre polynomial $P_{r+1}(\xi)$], and $\hat{F}$ is the transformed integrand

$$\hat{F}(\xi) = F(x(\xi))J(\xi), \quad dx = J\,d\xi \tag{7.1.21}$$

Table 7.1.2 Weights and Gauss points for the Gauss–Legendre quadrature.

$$\int_{-1}^{1} F(\xi)d\xi = \sum_{i=1}^{r} F(\xi_i)w_i$$

Points, $\xi_i^{\dagger}$	r	Weights, w_i
0.0000000000	1	2.0000000000
±0.5773502692	2	1.0000000000
0.0000000000	3	0.8888888889
±0.7745966692		0.5555555555
±0.3399810435	4	0.6521451548
±0.8611363116		0.3478548451
0.0000000000	5	0.5688888889
±0.5384693101		0.4786286705
±0.9061798459		0.2369268850
±0.2386191861	6	0.4679139346
±0.6612093865		0.3607615730
±0.9324695142		0.1713244924

†Note that $0.57735\ldots = 1/\sqrt{3}$, $0.77459\ldots = \sqrt{3/5}$, and $0.888\ldots = 8/9$, and $0.555\ldots = 5/9$.

where J is the Jacobian of the transformation between x anbd ξ. The weight factors and Gauss points for the Gauss–Legendre quadrature (7.1.20) are given for $r = 1, \ldots, 6$ in Table 7.1.2.

The Gauss–Legendre quadrature is more frequently used than the Newton–Cotes quadrature because it requires fewer base points (hence, a saving in computation) to achieve the same accuracy. The error in the approximation is zero if the $(2r + 2)$th derivative of the integrand vanishes. In other words, a polynomial of degree p is integrated exactly by employing $r = \frac{1}{2}(p + 1)$ Gauss points. When $p + 1$ is odd, one should pick the nearest larger integer:

$$r = \left[\frac{1}{2}(p+1)\right] \tag{7.1.22}$$

In finite element formulations, we encounter integrals whose integrands F are functions of x, $\psi_i(x)$, and derivatives of $\psi_i(x)$ with respect to x. For the Gauss–Legendre quadrature, we must transform $F(x)dx$ to $\hat{F}(\xi)d\xi$ in order to use the formula (7.1.20). For example, consider the integral

$$K_{ij}^e = \int_{x_a}^{x_b} a(x)\frac{d\psi_i^e}{dx}\frac{d\psi_j^e}{dx}\,dx \tag{7.1.23}$$

Using the chain rule of differentiation, we have

$$\frac{d\psi_i^e(x)}{dx} = \frac{d\psi_i^e(\xi)}{d\xi}\frac{d\xi}{dx} = J^{-1}\frac{d\psi_i^e(\xi)}{d\xi} \tag{7.1.24}$$

Therefore, the integral in (7.1.23) can be written, with the help of (7.1.10), as

$$K_{ij}^e = \int_{-1}^{1} a(x(\xi))\frac{1}{J}\frac{d\psi_i^e}{d\xi}\frac{1}{J}\frac{d\psi_j^e}{d\xi}\,J\,d\xi \tag{7.1.25}$$

$$\approx \sum_{I=1}^{r} \hat{F}_{ij}^e(\xi_I)w_I \tag{7.1.26}$$

where

$$\hat{F}^e_{ij} = a\frac{1}{J}\frac{d\psi^e_i}{d\xi}\frac{d\psi^e_j}{d\xi}, \quad J = \sum_{i=1}^{m} x^e_i \frac{d\hat{\psi}^e_i}{d\xi} \tag{7.1.27}$$

For the isoparametric formulation, we take $\psi^e_i = \hat{\psi}^e_i$. As noted earlier, the Jacobian matrix will be the same ($J_e = \frac{1}{2}h_e$) when the element is a straight line, even if the coordinate transformation is quadratic or cubic. However, when the element is curved, the Jacobian is a function of ξ for transformations other than linear. The transformation from x to ξ is not required in the Newton–Cotes quadrature.

It is possible to determine the exact number of Gauss points required to evaluate the following element coefficients:

$$\begin{aligned}
K^e_{ij} &= \int_{x_a}^{x_b} \frac{d\psi^e_i}{dx}\frac{d\psi^e_j}{dx}\,dx = \int_{-1}^{+1} \frac{d\psi^e_i}{d\xi}\frac{d\psi^e_j}{d\xi}(J)^{-2}\,J\,d\xi \equiv \int_{-1}^{+1} G^K_{ij}(\xi)\,d\xi \\
&\approx \sum_{I=1}^{N^K} G^K_{ij}(\xi_I)\,W_I \\
M^e_{ij} &= \int_{x_a}^{x_b} \psi^e_i \psi^e_j\,dx = \int_{-1}^{+1} \psi^e_i(\xi)\,\psi^e_j(\xi)\,J\,d\xi \equiv \int_{-1}^{+1} G^M_{ij}(\xi)\,d\xi \\
&\approx \sum_{I=1}^{N^M} G^M_{ij}(\xi_I)\,W_I \\
f^e_i &= \int_{x_a}^{x_b} \psi^e_i\,dx = \int_{-1}^{+1} \psi^e_i(\xi)\,J\,d\xi \equiv \int_{-1}^{+1} G^F_{ij}(\xi)\,d\xi \\
&\approx \sum_{I=1}^{N^F} G^F_{ij}(\xi_I)\,W_I
\end{aligned} \tag{7.1.28}$$

when linear, quadratic, and cubic interpolation functions are used. For linear interpolation functions, the integrand of K^e_{ij} is constant, requiring only one-point Gauss–Legendre quadrature ($N^K = 1$). The integrand of the mass matrix M^e_{ij} is quadratic ($p = 2$), requiring [$r = \frac{1}{2}(p+1) = \frac{3}{2}$], the two-point quadrature ($N^M = 2$). The coefficients f^e_i are evaluated exactly by one-point quadrature ($N^F = 1$). Similarly, for quadratic and cubic elements, we can estimate the number of Gauss points needed to evaluate K^e_{ij}, M^e_{ij}, and f^e_i exactly. The results are summarized in Table 7.1.3. Note that, in estimating the quadrature points, it is assumed that the Jacobian J is a constant, which holds true when the element is a straight line.

If the matrices in (7.1.28) have variable coefficients or the elements are curved [and hence $J_e = J_e(\xi)$], the degree of the variation of the integrands changes and the number of Gauss points needed to exactly evaluate the integral changes. If the elements are straight and the coefficients $a = a(x)$ and $c = c(x)$ together with $f = f(x)$ are no more than linear in x, then the number of Gauss points for evaluating the coefficients

$$K^e_{ij} = \int_{x_a}^{x_b} a\frac{d\psi^e_i}{dx}\frac{d\psi^e_j}{dx}\,dx, \quad M^e_{ij} = \int_{x_a}^{x_b} c\psi^e_i\psi^e_j\,dx \tag{7.1.29}$$

Table 7.1.3 The number of Gauss–Legendre quadrature points required to evaluate K_{ij}^e, M_{ij}^e, and f_i^e of Eq. (7.1.28).

Element type	N^K	N^M	N^F
Linear	1	2	1
Quadratic	2	3	2
Cubic	3	4	2

remain the same as listed in the above table. However, the evaluation of f_i^e requires one point more than before. Conversely, the two-point quadrature for linear elements, three-point quadrature for quadratic elements, and four-point quadrature for cubic elements would exactly evaluate K_{ij}^e with a quadratic variation of $a(x)$, M_{ij}^e with linear variation of $c(x)$, and f_i^e with quadratic variation of $f(x)$.

The use of Gauss–Legendre quadrature in (7.1.28) yields the following values (exact up to the fifth decimal place) when the element is straight and the isoparametric formulation is used:

Quadratic Element (Three-Point Formula)

$$[K] = \frac{1}{h_e}\begin{bmatrix} 2.33333 & -2.66667 & 0.33333 \\ -2.66667 & 5.33333 & -2.66667 \\ 0.33333 & -2.66667 & 2.33333 \end{bmatrix}$$

$$[M] = \frac{h_e}{10}\begin{bmatrix} 1.33333 & 0.66667 & 0.33333 \\ 0.66667 & 5.33333 & 0.66667 \\ 0.33333 & 0.66667 & 1.333433 \end{bmatrix} \tag{7.1.30}$$

$$\{f\} = h_e \begin{Bmatrix} 0.166667 \\ 0.666667 \\ 0.166667 \end{Bmatrix}$$

Cubic Element (Four-Point Formula)

$$[K] = \frac{1}{h_e}\begin{bmatrix} 3.700 & -4.725 & 1.350 & -0.325 \\ -4.725 & 10.800 & -7.425 & 1.350 \\ 1.350 & -7.425 & 10.800 & -4.725 \\ -0.325 & 1.350 & -4.725 & 3.700 \end{bmatrix}$$

$$[M] = \frac{h_e}{10}\begin{bmatrix} 0.761905 & 0.589286 & -0.214286 & 0.113095 \\ 0.589286 & 3.857143 & -0.482143 & -0.214286 \\ -0.214286 & -0.482143 & 3.857143 & 0.589286 \\ 0.113095 & -0.214286 & 0.589286 & 0.761905 \end{bmatrix} \tag{7.1.31}$$

$$\{f\} = h_e \begin{Bmatrix} 0.125 \\ 0.375 \\ 0.375 \\ 0.125 \end{Bmatrix}$$

Example 7.1.1

We wish to evaluate the following integrals using the Newton–Cotes and Gauss–Legendre quadratures:

$$K_{12}=\int_{x_a}^{x_b}(x_0+x)\frac{d\psi_1}{dx}\frac{d\psi_2}{dx}\,dx,\quad G_{12}=\int_{x_a}^{x_b}(x_0+x)\psi_1\psi_2\,dx$$

where ψ_i are the linear interpolation functions

$$\psi_1=\frac{x-x_a}{x_b-x_a}=\frac{1}{2}(1-\xi),\quad \psi_2=\frac{x_b-x}{x_b-x_a}=\frac{1}{2}(1+\xi)$$

and x_0 is a constant.

Note that the integrand $F(x)$ in the integral of K_{12} is a linear polynomial (i.e., the degree is $r=1$). Hence, we expect the one-point Newton–Cotes or Gauss–Legendre quadrature to yield the exact value of K_{12}. On the other hand, the integrand of G_{12} is a cubic polynomial (i.e., the degree is $r=3$). Hence, we expect the three-point Newton–Cotes or two-point Gauss–Legendre quadrature to yield the exact value of G_{12}. The exact values are

$$K_{12}=-\frac{1}{x_b-x_a}\left[x_0+\frac{1}{2}(x_b+x_a)\right],\quad G_{12}=\frac{x_b-x_a}{6}\left[x_0+\frac{1}{2}(x_b+x_a)\right]$$

Newton–Cotes Quadrature: For various number of base points r, we have the following results:

$r=1,\ [x_1=0.5(x_a+x_b)]$:

$$\begin{aligned}K_{12}&=\int_{x_a}^{x_b}(x_0+x)\frac{d\psi_1}{dx}\frac{d\psi_2}{dx}\,dx=\int_{x_a}^{x_b}(x_0+x)\left(-\frac{1}{x_b-x_a}\right)\left(\frac{1}{x_b-x_a}\right)dx\\&=(x_b-x_a)\left[-\frac{1}{(x_b-x_a)^2}(x_0+x_1)\right]=-\frac{1}{x_b-x_a}\left[x_0+\frac{1}{2}(x_b+x_a)\right]\\G_{12}&=\int_{x_a}^{x_b}(x_0+x)\psi_1\psi_2\,dx=\int_{x_a}^{x_b}(x_0+x)\left(\frac{x-x_a}{x_b-x_a}\right)\left(\frac{x_b-x}{x_b-x_a}\right)dx\\&=(x_b-x_a)(x_0+x_1)\left(\frac{x_1-x_a}{x_b-x_a}\right)\left(\frac{x_b-x_1}{x_b-x_a}\right)\\&=\frac{x_b-x_a}{4}\left[x_0+\frac{1}{2}(x_b+x_a)\right]\end{aligned}$$

$r=2,\ (x_1=x_a, x_2=x_b)$:

$$\begin{aligned}K_{12}&=-\frac{1}{2(x_b-x_a)}[(x_0+x_1)+(x_0+x_2)]\\&=-\frac{1}{2(x_b-x_a)}(2x_0+x_a+x_b)=-\frac{1}{x_b-x_a}\left[x_0+\frac{1}{2}(x_b+x_a)\right]\\G_{12}&=\frac{x_b-x_a}{2}\left[(x_0+x_1)\left(\frac{x_1-x_a}{x_b-x_a}\right)\left(\frac{x_b-x_1}{x_b-x_a}\right)\right.\\&\qquad\left.+(x_0+x_2)\left(\frac{x_2-x_a}{x_b-x_a}\right)\left(\frac{x_b-x_2}{x_b-x_a}\right)\right]=0\end{aligned}$$

$r=3$, $[x_1=x_a, x_2=0.5(x_a+x_b), x_3=x_b]$:

$$K_{12} = -\frac{1}{6(x_b-x_a)}[(x_0+x_1)+4(x_0+x_2)+(x_0+x_3)]$$

$$= -\frac{1}{x_b-x_a}\left[x_0+\frac{1}{2}(x_b+x_a)\right]$$

$$G_{12} = \frac{x_b-x_a}{6}\left[(x_0+x_1)\left(\frac{x_1-x_a}{x_b-x_a}\right)\left(\frac{x_b-x_1}{x_b-x_a}\right)\right.$$

$$+4(x_0+x_2)\left(\frac{x_2-x_a}{x_b-x_a}\right)\left(\frac{x_b-x_2}{x_b-x_a}\right)$$

$$\left.+(x_0+x_3)\left(\frac{x_3-x_a}{x_b-x_a}\right)\left(\frac{x_b-x_3}{x_b-x_a}\right)\right]$$

$$= \frac{x_b-x_a}{6}\left[x_0+\frac{1}{2}(x_b+x_a)\right]$$

Gauss–Legendre Quadrature: The coefficient K_{12} for this case takes the form $[dx=0.5(x_b-x_a)d\xi$, $\psi_1=0.5(1-\xi)$, $\psi_2=0.5(1+\xi)$, and $d\psi_i/dx=(1/J)d\psi_i/d\xi$ with $J=0.5(x_b-x_a)]$

$$K_{12} = \int_{x_a}^{x_b}(x_0+x)\frac{d\psi_1}{dx}\frac{d\psi_2}{dx}\,dx$$

$$= -\frac{1}{2(x_b-x_a)}\int_{-1}^{1}\left[x_0+x_a+\frac{x_b-x_a}{2}(1+\xi)\right]d\xi$$

The polynomial order is $p=1$; hence, the one-point Gauss rule ($r=1$) would yield the exact value:

$$K_{12} = -\frac{1}{2(x_b-x_a)}\left[x_0+x_a+\frac{x_b-x_a}{2}(1+0.0)\right]\cdot 2$$

$$= -\frac{1}{x_b-x_a}\left[x_0+\frac{1}{2}(x_b+x_a)\right]$$

The coefficient G_{12} takes the form

$$G_{12} = \int_{x_a}^{x_b}(x_0+x)\psi_1\psi_2\,dx$$

$$= \frac{x_b-x_a}{8}\int_{-1}^{1}\left[x_0+x_a+\frac{x_b-x_a}{2}(1+\xi)\right](1-\xi^2)d\xi$$

The polynomial order is $p = 3$. Therefore, the two-point Gauss rule ($r = 2$) would yield the exact value

$$G_{12} = \frac{x_b - x_a}{8}\left\{\left[x_0 + x_a + \frac{x_b - x_a}{2}\left(1 - \frac{1}{\sqrt{3}}\right)\right]\left(1 - \frac{1}{3}\right) + \left[x_0 + x_a + \frac{x_b - x_a}{2}\left(1 + \frac{1}{\sqrt{3}}\right)\right]\left(1 - \frac{1}{3}\right)\right\} = \frac{x_b - x_a}{6}\left[x_0 + \frac{1}{2}(x_b + x_a)\right]$$

7.2 COMPUTER IMPLEMENTATION

7.2.1 Introductory Comments

Chapters 3–6 were devoted to the finite element formulations of two *classes* of boundary value, initial value, or eigenvalue problems in one dimension:

1. Second-order differential equations (e.g., heat transfer, fluid mechanics, one-dimensional elasticity, bars, and the Timoshenko beam theory)
2. Fourth-order differential equations governing the Euler–Bernoulli beam theory

The frame element, obtained by superposing the bar element and the beam element, was discussed in Chapter 5.

By now, it should be clear to the reader that the steps involved in the finite element analysis of a general class of problems (e.g., single second-order, single fourth-order, and a pair of second-order equations) are systematic, and once the finite element formulation is completed, the model can be implemented on a digital computer. Indeed, the success of the finite element method is largely due to the ease with which the analysis of a class of problems, without regard to the geometry and boundary conditions, can be implemented on a digital computer. A particular class of problems (say that described by the model equation of Chapter 3) can be solved by simply supplying the required input data to the program. For example, if we develop a general computer program to solve equations of the form

$$c_1 \frac{\partial u}{\partial t} + c_2 \frac{\partial^2 u}{\partial t^2} - \frac{\partial}{\partial x}\left(a \frac{\partial u}{\partial x}\right) + \frac{\partial^2}{\partial x^2}\left(b \frac{\partial^2 u}{\partial x^2}\right) + cu = f \tag{7.2.1}$$

then all physical problems described by Eqs. (3.1.1) and (5.1.1) and their time-dependent versions can be solved for any compatible boundary and initial conditions.

The purpose of this section is to discuss the basic steps involved in the development of a computer program for second- and fourth-order one-dimensional differential equations studied in the preceding chapters. The ideas presented here are used in the development of the model program **FEM1D,** and they are meant to be illustrative of the steps used in a typical finite element program development. One can make use of the ideas presented here and implement them using **FEM1D** to develop a program of one's own. The discussion here focuses on the finite element computations, and no attempt is made to discuss the Gauss elimination

procedure used to solve the resulting system of algebraic equations (a solver is provided with **FEM1D**; see Appendix 1 located on the book's website at www.mhhe.com/reddy3e).

7.2.2 General Outline

A typical finite element program consists of three basic units (see Fig. 7.2.1):

1. Preprocessor
2. Processor
3. Postprocessor

In the preprocessor part of the program, the input data of the problem are read in and/or generated. This includes the geometry (e.g., length of the domain and boundary conditions), the data of the problem (e.g., coefficients in the differential equation), finite element mesh information (e.g., element type, number of elements, element length, coordinates of the nodes, and connectivity matrix), and indicators for various options (e.g., print, no print, type of field problem analyzed, static analysis, eigenvalue analysis, transient analysis, and degree of interpolation).

In the processor part, all the steps in the finite element method discussed in the preceding chapter, except for postprocessing, are performed. The major steps of the processor are:

1. Generation of the element matrices using numerical integration
2. Assembly of element equations
3. Imposition of the boundary conditions
4. Solution of the algebraic equations for the nodal values of the primary variables

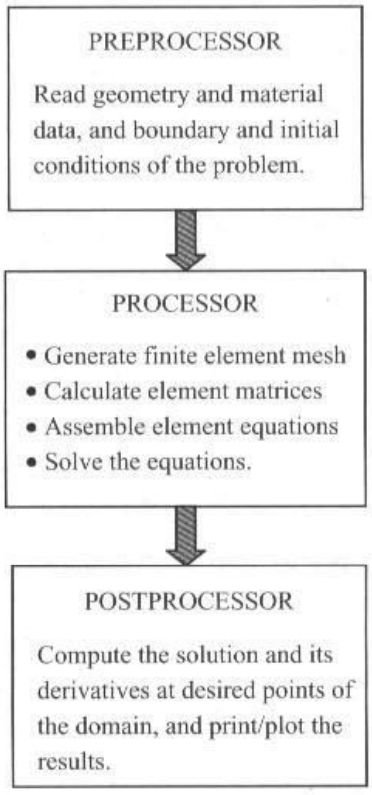


Figure 7.2.1 The three main functional units of a finite element program.

In the postprocessor part of the program, the solution is computed by interpolation at points other than nodes, secondary variables that are derivable from the solution are computed, and the output data are processed in a desired format for printout and/or plotting.

The preprocessor and postprocessors can be a few Fortran statements to read and print pertinent information, simple subroutines (e.g., subroutines to generate mesh and compute the gradient of the solution), or complex programs linked to other units via disk and tape files. The processor, where typically large amounts of computing time are spent, can consist of several subroutines, each having a special purpose (e.g., a subroutine for the calculation of element matrices, a subroutine for the imposition of boundary conditions, and a subroutine for the solution of equations). The degree of sophistication and the complexity of a finite element program depend on the general class of problems being programmed, the generality of the data in the equation, and the intended user of the program. It is always desirable to describe, through comment statements, all variables used in the computer program. A flow chart of the computer program **FEM1D** is presented in Fig. 7.2.2. The objective of each of the subroutines listed in the flow chart is described below.

ASSEMBLE: Subroutine for the assembly of element equations. The equations are assembled in upper-banded form for static and transient problems, and in full matrix form for eigenvalue problems.

EIGNSLVR: Subroutine for the solution of the eigenvalue problem $[A]\{X\} = \lambda[B]\{X\}$.

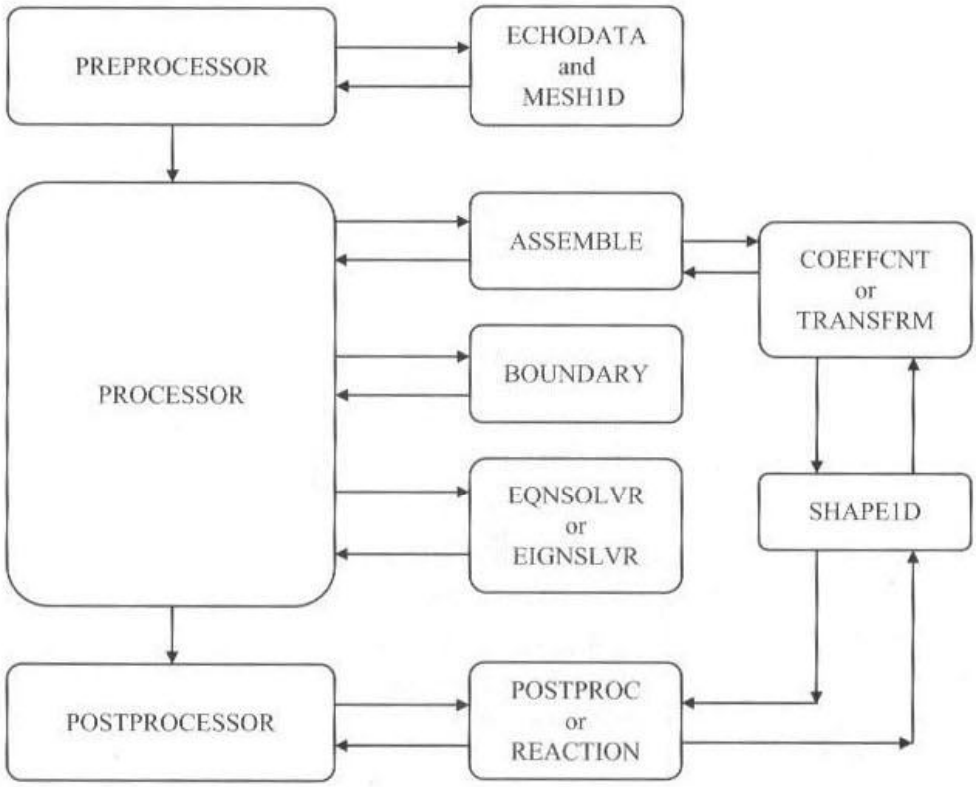


Figure 7.2.2 Flow chart of the computer program **FEM1D**.

BOUNDARY: Subroutine for imposition of specified boundary conditions (Dirichlet, Neumann, and Newton type).

COEFFCNT: Subroutine for computing element matrices $[K^e]$, $[M^e]$, and $\{f^e\}$ for all model problems except for truss and frame elements.

ECHODATA: Subroutine to echo the input to the program.

MESH1D: Subroutine to generate the mesh (coordinates of the global nodes and the connectivity array).

POSTPROC: Subroutine to postprocess the solution for all model problems except for truss and frame elements.

REACTION: Subroutine to calculate the reactions (i.e., generalized forces) for truss and frame elements.

SHAPE1D: Subroutine to compute the approximation functions and their derivatives.

EQNSOLVR: Subroutine for solving banded symmetric system of algebraic equations.

TRANSFRM: Subroutine to compute element stiffness matrix and force vector for truss and frame elements.

In the following sections, a discussion of the basic components of a typical finite element program is presented, and then the ideas are illustrated via FORTRAN statements.

7.2.3 Preprocessor

The preprocessor unit reads the input data, generates the finite element mesh, and prints the data and mesh information. The input data to a finite element program consist of element type IELEM (i.e., Lagrange element or Hermite element), number of elements used (NEM), specified boundary conditions on primary and secondary variables (number of boundary conditions, global node number and degree of freedom, and specified values of the degrees of freedom), the global coordinates of global nodes, and element properties (e.g., coefficients $a(x)$, $b(x)$, $c(x)$, $f(x)$, etc.). If a uniform mesh is used, the length of the domain should be read in, and global coordinates of the nodes can be generated in the program.

The preprocessor portion that deals with the generation of finite element mesh (when not supplied by the user) can be separated into a subroutine (**MESH1D**), depending on the convenience and complexity of the program. Mesh generation includes computation of the global coordinates X_I and the connectivity array $[B] = \text{NOD}$. Recall that the connectivity matrix describes the relationship between element nodes to global nodes as

$$\text{NOD}(I, J) = \textit{global node number} \text{ corresponding to the } J\text{th (local) node of element } I$$

This array is used in the assembly procedure as well as to transfer information from the element to the global system and vice versa. For example, to extract the vector ELX of global coordinates of element nodes from the vector GLX of global coordinates of global nodes, we can use the matrix NOD as follows: The global coordinate $x_i^{(n)}$ of the ith node of the nth element is the same as the global coordinate X_I of the global node I, where $I = \text{NOD}(n, i)$:

$$\{x_i^{(n)}\} = \{X_I\}, \quad I = \text{NOD}(n, i) \rightarrow \text{ELX}(i) = \text{GLX}(\text{NOD}(n, i))$$

The arrays ELX and GLX are used in **FEM1D** to denote $\{x^{(n)}\}$ and $\{X\}$, respectively.

7.2.4 Calculation of Element Matrices (Processor)

The most significant function of a processor is to generate element matrices. The element matrices are computed in various subroutines (**COEFFCNT** and **TRANSFRM**), depending on the type of problem being solved. These subroutines typically involve numerical evaluations of the element matrices $[K^e]$ and $[M^e]$ (program variables ELK and ELM) and the element vector $\{f^e\}$ (program variable ELF) for various field problems. The Gauss–Legendre quadrature described in Section 7.1.5 is used to evaluate element matrices and vectors, and the arrays are assembled as soon as they are computed. Thus, a loop on the number of elements in the mesh (NEM) is used to compute element matrices and assemble them (subroutine **ASSEMBLE**). It is here that the connectivity array NOD plays a crucial role. By putting one element matrix into global locations at a time, we avoid the computation of all element matrices at once.

Element matrices for different model equations (MODEL) and type of problem (NTYPE) are generated by assigning values as discussed next. Governing equations are listed for the static case. The variables used have the following meaning: $H =$ thickness of the beam/plate; $B =$ width of a beam; $E =$ Young's modulus; $G = E/[2(1+\nu)] =$ shear modulus; $\nu =$ Poisson's ratio; $D =$ bending stiffness ($D = EI = EBH^3/12$ for beams and $D = EH^3/[12(1-\nu^2)]$ for plates); $A =$ cross-sectional area; $K_s =$ shear correction factor; and $c_f =$ foundation modulus.

1. MODEL = 1, NTYPE = 0 All field problems described by the model equations (3.2.1) and (3.4.1), including radially symmetric heat-transfer-type problems:

$$-\frac{d}{dx}\left(a\frac{du}{dx}\right) + cu - f = 0;\ AX = a(x),\ CX = c(x),\ FX = f(x)$$

$$-\frac{1}{r}\frac{d}{dr}\left(ra\frac{du}{dr}\right) + cu - f = 0;\ AX = ra(r),\ CX = rc(r),\ FX = rf(r)$$

2. MODEL = 1, NTYPE = 1 Radially symmetric deformation of polar orthotropic disks (see Problem 4.37):

$$-\frac{d}{dr}\left[H\left(c_{11}r\frac{du}{dr} + c_{12}u\right)\right] + H\left(c_{22}\frac{u}{r} + c_{12}\frac{du}{dr}\right) = rf(r)$$

$$c_{11} = \frac{E_1}{1-\nu_{12}\nu_{21}},\quad c_{12} = \frac{\nu_{12}E_2}{1-\nu_{12}\nu_{21}},\quad c_{22} = \frac{E_2}{1-\nu_{12}\nu_{21}}$$

where $f(r)$ is the distributed force per unit volume (i.e., $Hf(r) = \hat{f}$ is the distributed force per unit area). For the isotropic case, $E_1 = E_2 = E$ and $\nu_{12} = \nu_{21} = \nu$.

3. MODEL = 1, NTYPE = 2 Radially symmetric deformation of cylinders:

$$-\frac{d}{dr}\left\{c\left[(1-\nu)r\frac{du}{dr} + \nu u\right]\right\} + c\left[(1-\nu)\frac{u}{r} + \nu\frac{du}{dr}\right] = rf(r)$$

$$c = \frac{E}{(1+\nu)(1-2\nu)}$$

4. MODEL = 2, NTYPE = 0 (reduced integration element; RIE) or MODEL = 2, NTYPE = 2 (consistent interpolation element; CIE) Bending of straight beams using

the Timoshenko beam theory:

$$-\frac{d}{dx}\left[GAK_s\left(\Psi+\frac{dw}{dx}\right)\right]+c_f w=q$$

$$-\frac{d}{dx}\left(EI\frac{d\Psi}{dx}\right)+GAK_s\left(\Psi+\frac{dw}{dx}\right)=0$$

$$AX=GAK_s,\quad CX=c_f(x),\quad FX=q(x),\quad BX=EI$$

5. MODEL = 2, NTYPE = 1 (RIE) or MODEL = 2, NTYPE = 3 (CIE) Axisymmetric bending of circular plates using the shear deformation plate theory:

$$-\frac{1}{r}\left[\frac{d}{dr}(rM_{rr})-M_{\theta\theta}\right]+Q_r=0,\quad -\frac{1}{r}\frac{d}{dr}(rQ_r)-q=0$$

$$M_{rr}=D\left(\frac{d\Psi}{dr}+\nu\frac{\Psi}{r}\right),\quad M_{\theta\theta}=D\left(\nu\frac{d\Psi}{dr}+\frac{\Psi}{r}\right),\quad Q_r=K_sGh\left(\Psi+\frac{dw}{dr}\right)$$

6. MODEL = 3, NTYPE = 0 Bending of straight beams using the Euler–Bernoulli beam theory:

$$\frac{d^2}{dx^2}\left(EI\frac{d^2w}{dx^2}\right)+c_f w=q(x),\quad BX=EI,\quad CX=c_f(x),\quad FX=q(x)$$

7. MODEL = 3, NTYPE = 1 Axisymmetric bending of circular plates using the classical plate theory:

$$D\frac{1}{r}\frac{d}{dr}\left\{r\frac{d}{dr}\left[\frac{1}{r}\frac{d}{dr}\left(r\frac{dw}{dr}\right)\right]\right\}+c_f w=q$$

8. MODEL = 4, NTYPE = 0 Two-node truss element.
9. MODEL = 4, NTYPE = 1 Two-node Euler–Bernoulli frame element.
10. MODEL = 4, NTYPE = 2 Two-node Timoshenko frame element.

The time-dependent option is exercised through variable ITEM:

ITEM = 0 static analysis
ITEM = 1 first-order time derivative (i.e., parabolic) problems
ITEM = 2 second-order time derivative (i.e., hyperbolic) problems

The element matrices are evaluated using the Gauss–Legendre quadrature except for MODEL = 4, where the explicit forms of element coefficients are programmed in the interest of computational efficiency.

The element shape functions SF and their derivatives GDSF are evaluated at the Gauss points in subroutine **SHAPE1D**. The gaussian weights and points associated with two-, three-, four-, and five-point integration are stored in arrays GAUSWT and GAUSPT, respectively. The nth column of GAUSWT, for example, contains the weights corresponding to the n-point Gauss–Legendre quadrature rule:

GAUSPT $(i, n) = i$th Gauss point corresponding to the n-point Gauss rule

The variable NGP is used to denote the number of Gauss points, which is selected to achieve good accuracy. As noted earlier, the linear, quadratic and cubic interpolation

functions require two, three, and four quadrature points, respectively, to evaluate the element coefficients exactly. Thus, if IELEM is the element type,

$$\text{IELEM} = \begin{cases} 1 & \text{linear} \\ 2 & \text{quadratic (Lagrange elements)} \\ 3 & \text{cubic} \end{cases}$$

then NGP = IELEM +1 would evaluate K_{ij}^e, M_{ij}^e, and f_i^e [see (7.1.29)] exactly when $c(x)$ is linear, and $a(x)$, $b(x)$, and $f(x)$ are quadratic functions. The Hermite cubic element is identified with IELEM = 0, in which case NGP is taken to be 4.

The coefficients $a(x) = \text{AX}$, $b(x) = \text{BX}$, and $c(x) = \text{CX}$, together with $f(x) = \text{FX}$ in the differential equation (7.2.1) are assumed to vary with x as follows:

$$\begin{array}{ll} \text{AX=AX0+AX1*X} & (a = a_0 + a_1 x) \\ \text{BX=BX0+BX1*X} & (b = b_0 + b_1 x) \\ \text{CX=CX0+CX1*X} & (c = c_0 + c_1 x) \\ \text{FX=FX0+FX1*X+FX2*X*X} & (f = f_0 + f_1 x + f_2 x^2) \end{array} \tag{7.2.2}$$

For radially symmetric elasticity problems, (AX0, AX1) [or (BX0, BX1) for circular plates] are used to input Young's modulus E and Poisson's ratio ν.

The Gauss–Legendre quadrature formula (7.1.20) can be implemented in the computer as follows: Consider K_{ij}^e of the form

$$K_{ij}^e = \int_{x_a}^{x_b} \left[a(x) \frac{d\psi_i^e}{dx} \frac{d\psi_j^e}{dx} + c(x) \psi_i^e \psi_j^e \right] dx \tag{7.2.3}$$

We use the following program variables for the quantities in (7.2.3):

$$\text{ELK(I, J)} = K_{ij}^e, \quad \text{SF(I)} = \psi_i^e, \quad \text{GDSF(I)} = \frac{d\psi_i^e}{dx}$$

$$\text{AX} = a(x), \quad \text{CX} = c(x), \quad \text{ELX(I)} = x_i^e$$

$$\text{NPE} = n \quad \text{(the number of nodes in the element)}$$

After transforming x to ξ [or $x = x_a + 0.5 h_e (1 + \xi)$]

$$x = \sum_{i=1}^{n} x_i^e \psi_i^e \tag{7.2.4}$$

the coefficients K_{ij}^e in (7.2.3) can be written as

$$K_{ij}^e = \int_{-1}^{1} \left[a(\xi) \frac{1}{J} \frac{d\psi_i^e}{d\xi} \frac{1}{J} \frac{d\psi_j^e}{d\xi} + c(\xi) \psi_i^e \psi_j^e \right] J \, d\xi \equiv \int_{-1}^{1} G_{ij}^e(\xi) J \, d\xi \tag{7.2.5a}$$

$$= \sum_{I=1}^{\text{NGP}} G_{ij}^e(\xi_I) J W_I \tag{7.2.5b}$$

where G_{ij}^e denotes the expression in the square brackets in (7.2.5a), J is the Jacobian, and (ξ_I, W_I) are the Ith Gauss point and weight.

Examination of (7.2.5b) shows that there are three free indices: i, j, and I. We take the Gauss-point loop on I as the outermost one. Inside this loop, we evaluate G_{ij}^e at the Gauss point ξ_I for each i and j, multiply with the Jacobian $J = \frac{1}{2} h_e$ and the weights W_I, and sum

over the range of I:

$$\text{ELK}(i, j) = \text{ELK}(i, j) + G_{ij}^{e}(\xi_I) J W_I \tag{7.2.6}$$

Since $[K^e]$, $[M^e]$, $\{f^e\}$ are evaluated for $e = 1, 2, \ldots,$ NEM, where NEM denotes the number of elements in the mesh, we must initialize all arrays that are being evaluated using the Gauss–Legendre quadrature. The initialization must be made outside of the Gauss–Legendre quadrature loop.

The computation of coefficients K_{ij}^{e} in (7.2.5a) requires evaluation of a, c, ψ_i, and $d\psi_i/d\xi$ at the Gauss point ξ_I. Hence, inside the loop on I, we call subroutine **SHAPE1D** to evaluate ψ_i, $d\psi_i/dx = (d\psi_i/d\xi)/J$. Fortran statements to evaluate $[K^e]$ and $\{f^e\}$ are given below.

```
      DO 100 NI = 1,NGP
      XI = GAUSPT(NI,NGP)
C     Call subroutine SHP1D to evaluate the interpolation functions
C     (SF) and their global derivatives (GDSF) at the Gauss point XI
      CALL SHP1D(XI,NPE,SF,GDSF,GJ)
      CONST = GJ*GAUSWT(NI,NGP)
      DO 20 I = 1,NPE
20    X = X + SF(I)*ELX(I)
      AX = AX0 + AX1*X
      CX = CX0 + CX1*X
      DO 30 J = 1,NPE
      ELF(J) = ELF(J) + CONST*SF(J)*FX
      DO 30 I = 1,NPE
30    ELK(I,J) = ELK(I,J) + CONST*(AX*GDSF(I)*GDSF(J)
                 + CX*SF(I)*SF(J))
```

In the same way, all the other coefficients (e.g., M_{ij}^{e} and f_i^e) can be evaluated. Recall that the element properties (i.e., K_{ij}^{e}, M_{ij}^{e}, and f_i^e) are calculated by calling a suitable subroutine (**COEFFCNT** or **TRANSFRM**) for the field problem being analyzed within a loop with a counter based on the number of elements in the mesh (NEM).

7.2.5 Assembly of Element Equations (Processor)

The assembly of element equations should be carried out as soon as the element matrices are computed, rather than waiting till the element coefficients of all the elements are computed. The latter requires storage of the element coefficients of each element. In the former case, we can perform the assembly in the same loop in which a subroutine is called to calculate element matrices.

A feature of the finite element equations that enables us to save storage and computing time is the assembly of element matrices in upper-banded form. When element matrices are symmetric, the resulting global (or assembled) matrix is also symmetric, with many zeros away from the main diagonal. Therefore, it is sufficient to store only the upper *half-band* of

the assembled matrix. The half bandwidth of a matrix is defined as follows: Let N_i be the number of matrix elements between the diagonal element and the last nonzero element in the ith row, after which all elements in that row are zero; the half-bandwidth is the maximum of $(N_i + 1) \times$ NDF, where NDF is the number of degrees of freedom per node

$$b_I = \max_{1 \le i \le n} [(N_i + 1) \times \text{NDF}]$$

and n is the number of rows in the matrix (or equations in the problem). General-purpose equation solvers are available for such banded systems of equations.

The half-bandwidth NHBW of the assembled (i.e., global) finite element matrix can be determined in the finite element program itself. The local nature of the finite element interpolation functions (i.e., ψ_i^e are defined to be nonzero only over the element Ω_e) is responsible for the banded character of the assembled matrix. If two global nodes do not belong to the same element, then the corresponding entries in the global matrix are zeros:

$$K_{IJ} = 0 \quad \text{if global nodes } I \text{ and } J \text{ do not correspond to local nodes of the same element}$$

This property enables us to determine the half-bandwidth NHBW of the assembled matrix

$$\text{NHBW} = \max_{\substack{1 \le N \le \text{NEM} \\ 1 \le I,\ J \le \text{NPE}}} \{\text{abs}[\text{NOD}(N,\ I) - \text{NOD}(N,\ J)] + 1\} \times \text{NDF} \tag{7.2.7a}$$

where

$$\begin{aligned} \text{NEM} &= \text{number of elements in the mesh} \\ \text{NPE} &= \text{number of nodes per element} \\ \text{NDF} &= \text{number of degrees of freedom per element} \end{aligned} \tag{7.2.7b}$$

For example, for one-dimensional problems with elements connected in series, the maximum difference between the nodes of an element is equal to NPE − 1. Hence,

$$\text{NHBW} = [(\text{NPE} - 1) + 1] \times \text{NDF} = \text{NPE} \times \text{NDF} \tag{7.2.8}$$

Of course, NHBW is always less than or equal to the number of primary degrees of freedom in the mesh, i.e., the number of equations, NEQ.

The logic for assembling the element matrices K_{ij}^e into the upper-banded form of the global coefficients K_{ij} is that the assembly can be skipped whenever $J < I$ and $J >$ NHBW. The main diagonal, $I = J$, of the assembled square matrix (i.e., full storage form), becomes the first column of the assembled banded matrix (i.e., banded storage form), as shown in Fig. 7.2.3. The upper diagonals (parallel to the main diagonal) take the position of respective columns in the banded matrix. Thus, the banded matrix has dimension NEQ × NHBW, where NEQ denotes the total number of equations in the problem.

The element coefficients K_{ij}^n and f_i^n of a typical element Ω_n are to be assembled into the global coefficients matrix $[K]$ and source vector $\{F\}$, respectively. If the ith node of the element is equal to the Ith global node and the jth node of the element is equal to the Jth global node, we have

$$K_{IJ} = K_{ij}^n, \quad F_I = F_i^n \quad (\text{for NDF} = 1) \tag{7.2.9a}$$

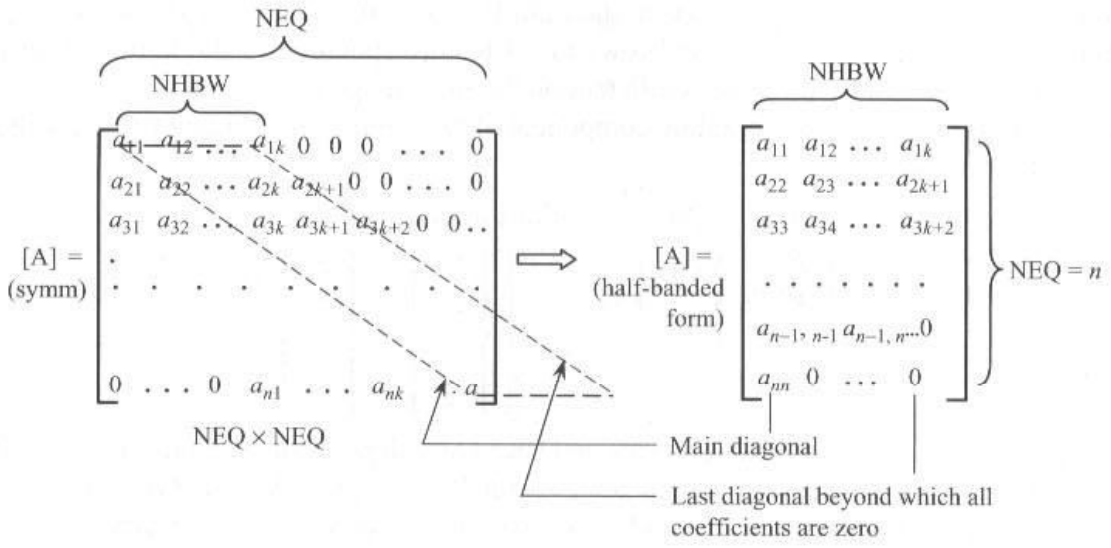


Figure 7.2.3 Finite element coefficient matrix storage in upper-half-banded form.

The values of I and J can be obtained with the help of array NOD:

$$I = \text{NOD}(n, i), \quad J = \text{NOD}(n, j) \tag{7.2.9b}$$

Recall that it is possible that the same I and J may correspond to a pair of i and j of some other element Ω^m. In that case, K_{ij}^m will be added to existing coefficients K_{IJ} during the assembly. For NDF > 1, the logic still holds, with the change

$$K_{(\text{NR})(\text{NC})} = K^n_{(i+p-1)(j+q-1)} \quad (p, q = 1, 2, \ldots, \text{NDF}) \tag{7.2.10a}$$

where

$$\text{NR} = (I - 1) \times \text{NDF} + p, \quad \text{NC} = (J - 1) \times \text{NDF} + q \tag{7.2.10b}$$

and I and J are related to i and j by (7.2.9b). These ideas are implemented in subroutine **ASSEMBLE**.

7.2.6 Imposition of Boundary Conditions (Processor)

Imposition of boundary conditions on the primary and secondary global degrees of freedom can be carried out through a subroutine **(BOUNDARY)**, which remains unchanged for two-dimensional or three-dimensional problems. There are three types of boundary conditions for any problem:

1. Essential boundary conditions, i.e., boundary conditions on primary variables (Dirichlet boundary conditions).
2. Natural boundary conditions, i.e., boundary conditions on secondary variables (Neumann boundary conditions).
3. Mixed boundary conditions (Newton boundary conditions).

The procedure for implementing the boundary conditions on the primary variables involves modifying the assembled coefficient matrix (GLK) and right-hand column vector (GLF) by three operations:

Step 1. Moving the known products to the right-hand column of the matrix equation.
Step 2. Replacing the columns and rows of GLK corresponding to the known primary variable by zeros, and setting the coefficient on the main diagonal to unity.
Step 3. Replacing the corresponding component of the right-hand column by the specified value of the variable.

Consider the following N algebraic equations in full matrix form:

$$\begin{bmatrix} K_{11} & K_{12} & K_{13} & \cdots \\ K_{21} & K_{22} & K_{23} & \cdots \\ K_{31} & K_{32} & K_{33} & \cdots \\ \vdots & \vdots & \vdots & \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ \vdots \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ \vdots \end{Bmatrix}$$

where U_I and F_I are the global primary and secondary degrees of freedom, respectively, and K_{IJ} are the assembled coefficients. Suppose that $U_S = \hat{U}_S$ is specified. Recall that when the primary degree of freedom at a node is known, the corresponding secondary degree of freedom is unknown, and vice versa. Set $K_{SS} = 1$ and $F_S = \hat{U}_S$; further, set $K_{SI} = K_{IS} = 0$ for $I = 1, 2, \ldots, N$ and $I \neq S$. For $S = 2$, the modified equations are

$$\begin{bmatrix} K_{11} & 0 & K_{13} & K_{14} & \cdots & K_{1n} \\ 0 & 1 & 0 & 0 & \cdots & 0 \\ K_{31} & 0 & K_{33} & K_{34} & \cdots & K_{3n} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ K_{n1} & 0 & K_{n3} & K_{n4} & \cdots & K_{nn} \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ \vdots \\ U_n \end{Bmatrix} = \begin{Bmatrix} \hat{F}_1 \\ \hat{U}_2 \\ \hat{F}_3 \\ \vdots \\ \hat{F}_n \end{Bmatrix}$$

where

$$\hat{F}_i = F_i - K_{i2}\hat{U}_2 \quad (i = 1, 3, 4, 5, \ldots, n;\ i \neq 2)$$

Thus, in general, if $U_S = \hat{U}_S$ is known, we have

$$K_{SS} = 1, \quad F_S = \hat{U}_S; \quad \hat{F}_i = F_i - K_{iS}\hat{U}_S; \quad K_{Si} = K_{iS} = 0$$

where $i = 1, 2, \ldots, S-1, S+1, \ldots, n$ $(i \neq S)$. This procedure is repeated for every specified primary degree of freedom. It enables us to retain the original order of the matrix, and the specified boundary conditions on the primary degrees of freedom are printed as part of the solution. Of course, the logic should be implemented for a banded system of equations.

The specified secondary degrees of freedom (Q_i) are implemented directly by adding their specified values to the computed values. Suppose that the point source corresponding to the Rth secondary degree of freedom is specified to be $\hat{F}_R$. Then

$$F_R = f_R + \hat{F}_R$$

where f_R is the contribution due to the distributed source $f(x)$; f_R is computed as a part of the element computations and assembled.

Mixed-type boundary conditions are of the form

$$a\frac{du}{dx} + k(u - \bar{u}) = 0 \quad (\bar{u} \text{ and } k \text{ are known constants}) \tag{7.2.11}$$

which contains both the primary variable u and the secondary variable $a\ du/dx$. Thus, $a\ du/dx$ at the node p is replaced by $-k_p(u_p - \bar{u}_p)$:

$$Q_p = -k_p(U_p - \bar{U}_p)$$

This amounts to modifying K_{pp} by adding k_p to its existing value,

$$K_{pp} \leftarrow K_{pp} + k_p$$

and adding $k_p \bar{U}_p$ to F_p,

$$F_p \leftarrow F_p + k_p \bar{U}_p$$

All three types of boundary conditions are implemented in the subroutine **BOUNDARY** for boundary, initial, and eigenvalue problems. The following variable names are used in the subroutine **BOUNDARY:**

NSPV	Number of specified primary variables
NSSV	Number of specified secondary variables
NNBC	Number of Newton boundary conditions
VSPV	Column of the specified values $\bar{U}_S$ of primary variables
VSSV	Column of the specified values $\bar{F}_R$ of secondary variables
VNBC	Column of the specified values k_p
UREF	Column of the specified values $\bar{U}_p$
ISPV	Array of the global node and degree of freedom at the node that is specified [ISPV(I,1)=global node of the Ith boundary condition, ISPV(I,2)=degree of freedom specified at the global node, ISPV(I,1)]

Similar definitions are used for ISSV and INBC arrays.

7.2.7 Solving Equations and Postprocessing

Subroutine **EQNSOLVR** is used to solve a banded system of equations, and the solution is returned in array GLF. The program performs the gaussian elimination and back-substitution to compute the solution. For a discussion of the gaussian elimination used to solve a set of linear algebraic equations, the reader is referred to the book by Carnhan, Luther, and Wilkes (1969).

Postprocessing involves computation of the solution and its gradient at preselected points of the domain. The preselected points are the end points of the element, the midpoint, and three evenly spaced points between the end points and the midpoint. Subroutine **POSTPROC** is used to evaluate the solution and its derivatives at the preselected points of an element:

$$u^e(x) = \sum_{j=1}^{n} u_j^e \psi_j^e(x), \quad \left(\frac{du^e}{dx}\right)\bigg|_x = \sum_{j=1}^{n} u_j^e \left(\frac{d\psi_j^e}{dx}\right)\bigg|_x \tag{7.2.12}$$

for the Lagrange elements and

$$w^e(x) = \sum_{j=1}^{4} u_j^e \phi_j^e(x), \quad \frac{d^m w^e}{dx^m} = \sum_{j=1}^{4} u_j^e \left(\frac{d^m \phi_j^e}{dx^m}\right) \quad (m = 1, 2, 3) \tag{7.2.13}$$

for the Hermite cubic elements. The nodal values u_j^e of the element Ω_e are deduced from the global nodal values U_I as follows:

$$u_j^e = U_I, \quad I = \text{NOD}(e, j)$$

Table 7.3.1 Meaning of the program variables for various field problems. Subscripts 0, 1, and 2 on the variables denote the constant, linear, and quadratic coefficients of the variables, respectively, (e.g., f_0, f_1, f_2 denotes the coefficients in $f(x) = f_0 + f_1 \cdot x + f_2 \cdot x^2$).

	Field problem	MODEL	NTYPE	ITEM†	AX0	AX1	BX0	BX1	CX0	CX1	FX0	FX1	FX2	CT0‡	CT1†
1.	Plane wall	1	0	1	k_0	k_1	0.0	0.0	0.0	0.0	f_0	f_1	f_2	ρ_0	ρ_1
2.	Heat Exchanger fin	1	0	1	$(kA)_0$	$(kA)_1$	0.0	0.0	c_0	c_1	f_0	f_1	f_2	ρ_0	ρ_1
3.	Radially symmetric heat transfer	1	0	1	0.0	k_1	0.0	0.0	0.0	0.0	0.0	f_1	f_2	0.0	ρ_1
4.	Viscous flow through channels	1	0	1	μ_0	μ_1	0.0	0.0	0.0	0.0	f_0	f_1	f_2	ρ_0	ρ_1
5.	Viscous flow through pipes	1	0	1	0.0	μ_1	0.0	0.0	0.0	0.0	0.0	f_1	f_2	0.0	ρ_1
6.	Unidirectional seepage	1	0	1	μ_0	μ_1	0.0	0.0	0.0	0.0	f_0	f_1	f_2	ρ_0	ρ_1
7.	Radially symmetric seepage (ground water flow)	1	0	1	0.0	μ_1	0.0	0.0	0.0	0.0	0.0	f_1	f_2	0.0	ρ_1
8.	Axial deformation of a bar	1	0	2	$(AE)_0$	$(AE)_1$	0.0	0.0	c_0	c_1	f_0	f_1	f_2	$(\rho A)_0$	$(\rho A)_1$
9.	Radially symmetric deformation of a disk (plane stress)	1	1	2	E_1	E_2	v_{12}	H	c_0	c_1	f_0	f_1	f_2	$(\rho A)_0$	$(\rho A)_1$
10.	Radially symmetric deformation of a cylinder (plane strain)	1	2	2	E_1	E_2	v_{12}	H	c_0	c_2	f_0	f_1	f_2	$(\rho A)_0$	$(\rho A)_1$
11.	Euler–Bernoulli beam theory	3	0	2	0.0	0.0	$(EI)_0$	$(EI)_1$	c_0	c_1	f_0	f_1	f_2	ρA	ρ_I

Table 7.3.1 *Continued.*

	Field problem	MODEL	NTYPE	ITEM†	AX0	AX1	BX0	BX1	CX0	CX1	FX0	FX1	FX2	CT0‡	CT1‡
12.	Euler–Bernoulli theory for circular plates	3	1	2	E_1	E_2	v_{12}	H	c_0	c_2	f_0	f_1	f_2	ρ_A	ρ_I
13.	Timoshenko beam theory (RIE)*	2	0	2	$(SK)_0$	$(SK)_1$	$(EI)_0$	$(EI)_1$	c_0	c_1	f_0	f_1	f_2	ρ_A	ρ_I
14.	Timoshenko beam theory (CIE)*	2	2	2	$(SK)_0$	$(SK)_1$	$(EI)_0$	$(EI)_1$	c_0	c_1	f_0	f_1	f_2	ρ_A	ρ_I
15.	Timoshenko theory of circular plates (RIE)	2	1	2	E_1	E_2	v_{12}	H	c_0	c_1	f_0	f_1	KG_{13}	ρ_A	ρ_I
16.	Timoshenko theory of circular plates (CIE)	2	3	2	E_1	E_2	v_{12}	H	c_0	c_1	f_0	f_1	KG_{13}	ρ_A	ρ_I
17.	Plane truss	4	0	0											
18.	The Euler–Bernoulli frame element	4	1	0	For field problems 17–19, these parameters are not read; instead, E =SE, A =SA, L =SL, and so on are read for each member of the structure, where SE=modulus E, SA=cross-sectional area A, SI=moment of inertia I, SL=length L of the member, CN= $\cos\alpha$, SN= $\sin\alpha$, etc. (see Table 7.3.2).										
19.	The Timoshenko frame element	4	2	0											

* $S = GA$ and K is the shear correction factor ($K = 5/6$).

† For time-dependent problems only; when steady-state solution is required, set ITEM=0.

‡ For transient analysis only: the transient analysis option is not available in **FEM1D** for truss and frame problems.

when NDF = 1. For NDF > 1, I is given by $I = [\text{NOD}(e, j) - 1] \times \text{NDF}$ and

$$u^e_{j+p} = U_{I+p} \quad (p = 1, 2, \ldots, \text{NDF})$$

The values computed using the derivatives of the solution are often inaccurate because the derivatives of the approximate solution become increasingly inaccurate with increasing order of differentiation. For example, the shear force computed in the Euler–Bernoulli beam theory (EBT)

$$V = -\frac{d}{dx}\left(b\frac{d^2w}{dx^2}\right) = \sum_{j=1}^{n} u^e_j \frac{d}{dx}\left(b\frac{d^2\phi^e_j}{dx^2}\right) \tag{7.2.14}$$

will be in considerable error in addition to being discontinuous across the elements. The accuracy increases, rather slowly, with mesh refinement and higher-order elements. The derivatives computed using (7.2.14) are more accurate if they are computed at the Gauss points. When accurate values of the secondary variables are desired at the nodes, it is recommended that they be computed from the element equations:

$$Q^e_i = \sum_{j=1}^{n} K^e_{ij} u^e_j - f^e_i \quad (i = 1, 2, \ldots, n) \tag{7.2.15}$$

However, this requires recomputation or saving of element coefficients K^e_{ij} and f^e_i. Recall that the nodal values of generalized forces are exact at the nodes when computed using (7.2.15) for certain problems as discussed before.

7.3 APPLICATIONS OF PROGRAM FEM1D

7.3.1 General Comments

The computer program **FEM1D**, which embodies the ideas presented in the previous section, is intended to illustrate the use of the finite element models developed in Chapters 3–6 to a variety of one-dimensional field problems, some of them are not discussed in this book. The program **FEM1D** is developed as a learning computational tool for students of a first course on the finite element method (see Appendix 1 for a listing of the program located on the book's website at www.mhhe.com/reddy3e). In the interest of simplicity and ease of understanding, only the model equations discussed in this book and their immediate extensions are included in the program.

Table 7.3.1 contains a summary of the definitions of coefficients of various model problems and their corresponding program variables. The table can be used as a ready reference to select proper values of AX0, AX1, and so on for different problems.

7.3.2 Illustrative Examples

Here we revisit some of the example problems considered earlier to illustrate the use of **FEM1D** in their solution. Only certain key observations concerning the input data are made, but complete listings of input files for each problem are given. In the interest of brevity, the complete output files for most problems are not included.

A description of the input variables to program **FEM1D** is presented in Table 7.3.2. In Table 7.3.2, "skip" means that the input data is omitted (i.e., no data is required). In the "free

Table 7.3.2 Description of the input variables to the program **FEM1D**

• **Data Card 1**	
TITLE	Title of the problem being solved (80 characters)
• **Data Card 2**	
MODEL	Model equation being solved (see below)
NTYPE	Type of problem solved (see below) MODEL = 1, NTYPE = 0: A problem of MODEL EQUATION (3.2.1) MODEL = 1, NTYPE = 1: A circular DISK (PLANE STRESS) MODEL = 1, NTYPE > 1: A circular DISK (PLANE STRAIN) MODEL = 2, NTYPE = 0: A Timoshenko BEAM (RIE) problem MODEL = 2, NTYPE = 1: A Timoshenko PLATE (RIE) problem MODEL = 2, NTYPE = 2: A Timoshenko BEAM (CIE†) problem MODEL = 2, NTYPE > 2: A Timoshenko PLATE (CIE) problem MODEL = 3, NTYPE = 0: A Euler–Bernoulli BEAM problem MODEL = 3, NTYPE > 0: A Euler–Bernoulli circular plate MODEL = 4, NTYPE = 0: A plane TRUSS problem MODEL = 4, NTYPE = 1: A Euler–Bernoulli FRAME problem MODEL = 4, NTYPE = 2: A Timoshenko (CIE) FRAME problem
ITEM	Indicator for transient analysis ITEM = 0, Steady-state solution ITEM = 1, Transient analysis of PARABOLIC equations ITEM = 2, Transient analysis of HYPERBOLIC equations ITEM = 3, Eigenvalue analysis
• **Data Card 3**	
IELEM	Type of finite element IELEM = 0, Hermite cubic finite element IELEM = 1, Linear Lagrange finite element IELEM = 2, Quadratic Lagrange finite element
NEM	Number of elements in the mesh
• **Data Card 4**	
ICONT	Indicator for continuity of data for the problem ICONT = 1, Data (AX,BX,CX,FX and mesh) is continuous ICONT = 0, Data is element dependent
NPRNT	Indicator for printing of element/global matrices NPRNT = 0, Not to print element or global matrices but postprocess the solution and print NPRNT = 1, Print Element 1 coefficient matrices only but postprocess the solution and print NPRNT = 2, Print Element 1 and global matrices but NOT postprocess the solution NPRNT > 2, Not to print element or global matrices and NOT postprocess the solution

Skip Cards 5–15 for TRUSS/FRAME problems (MODEL=4), and read Cards 5–15 only if MODEL $\neq 4$.
SKIP cards 5–9 if data is discontinuous (ICONT = 0).

• **Data Card 5**	
DX(I)	Array of element lengths. DX(1) denotes the global coordinate of Node 1 of the mesh; DX(I) (I = 2, NEM1) denotes the length of the (I - 1)st element, where NEM1 = NEM + 1, and NEM denotes the number of elements in the mesh.

Cards 6–9 define the coefficients in the model equations. All coefficients are expressed in terms of GLOBAL coordinate x. See Table 7.2.1 for the meaning of the coefficients.

(**Table 7.3.2** continued)

- **Data Card 6**

AX0	Constant term of the coefficient $[a(x) =]$ AX
AX1	Linear term of AX

- **Data Card 7**

BX0	Constant term of the coefficient $[b(x) =]$ BX
BX1	Linear term of the coefficient BX

- **Data Card 8**

CX0	Constant term of the coefficient $[c(x) =]$ CX
CX1	Linear term of the coefficient CX

SKIP Card 9 for eigenvalue problems (i.e., when ITEM = 3)

- **Data Card 9**

FX0	Constant term of the source $[f(x) =]$ FX
FX1	Linear term of FX
FX2	Quadratic term of FX

SKIP Cards 10–15 if data is continuous (ICONT $\neq 0$). Cards 10–15 are read for each element (i.e., NEM times). All coefficients are with respect to the LOCAL coordinate $\bar{x}$.

- **Data Card 10**

NNM	Number of global nodes in the mesh

- **Data Card 11**

NOD	Connectivity of the element: NOD(N,I) = Global node number corresponding to the Ith node of Element N (I=1, NPE) where NPE denotes the Number of nodes Per Element
GLX(I)	Length of the Ith element

- **Data Card 12**

DCAX	Constant and linear terms of the coefficient AX

- **Data Card 13**

DCBX	Constant and linear terms of the coefficient BX

- **Data Card 14**

DCCX	Constant and linear terms of the coefficient CX

- **Data Card 15**

DCFX	Constant, linear and quadratic terms of FX

READ Cards 16–23 only for TRUSS/FRAME problems (MODEL = 4); otherwise SKIP.

- **Data Card 16**

NNM	Number of nodes in the finite element mesh

SKIP Cards 17–19 for TRUSS problems (NTYPE = 0)

- **Data Card 17 (Read for each element)**

PR	Poisson's ratio of the material (not used in EBT)
SE	Young's modulus of the material
SL	Length of the element
SA	Cross-sectional area of the element
SI	Moment of inertia of the element
CS	Cosine of the angle of orientation of the element
SN	Sine of the angle of orientation of the element; the angle is measured clockwise from the global x axis

- **Data Card 18 (Read for each element)**

HF	Intensity of the horizontal distributed force
VF	Intensity of the transversely distributed force
PF	Point load on the element
XB	Distance from node 1, along the length of the element to the point of load application, PF
CNT	Cosine of the angle of orientation of the load PF
SNT	Sine of the angle of orientation of the load PF; the angle is measured clockwise from the element x axis.

(**Table 7.3.2** continued)

- **Data Card 19**

NOD	Connectivity of the element: NOD(N,I) = global node number corresponding to the Ith node of element N (I = 1,NPE)

READ Cards 20 and 21 only for TRUSS problems (NTYPE = 0).

- **Data Card 20 (Read for each element)**

SE	Young's modulus of the material
SL	Length of the element
SA	Cross-sectional area of the element
CS	Cosine of the angle of orientation of the element
SN	Sine of the angle of orientation of the element Angle is measured counterclockwise from x axis
HF	Intensity of the horizontal distributed force

- **Data Card 21**

NOD(N,I)	Connectivity of the element: NOD(N,I) = global node number corresponding to the Ith node of element N (I = 1,NPE)

- **Data Card 22**

NCON	Number of inclined support conditions

SKIP Card 23 if no inclined support conditions are specified (NCON=0).

- **Data Card 23 (I = 1 to NCON)**

ICON(I)	Global node number of the support
VCON(I)	Angle (in degrees) between the normal and the global x-axis

- **Data Card 24**

NSPV	Number of specified PRIMARY degrees of freedom

SKIP Card 25 if no primary variables is specified (NSPV=0).

- **Data Card 25 (I = 1 to NSPV)**

ISPV(I,1)	Node number at which the PV is specified
ISPV(I,2)	Specified local primary degree of freedom (DOF) at the node
VSPV(I)	Specified value of the primary variable (PV) (will not read for eigenvalue problems)

SKIP Card 26 for eigenvalue problems (i.e., when ITEM = 3).

- **Data Card 26**

NSSV	Number of specified (nonzero) SECONDARY variables

SKIP Card 27 if no secondary variables is specified (NSSV=0); repeat Card 27 NSSV times.

- **Data Card 27 (I = 1 to NSSV)**

ISSV(I,1)	Node number at which the SV is specified
ISSV(I,2)	Specified local secondary DOF at the node
VSSV(I)	Specified value of the secondary variable (SV)

- **Data Card 28**

NNBC	Number of the Newton (mixed) boundary conditions

SKIP Card 29 if no mixed boundary condition is specified (NNBC = 0). The mixed boundary condition is assumed to be of the form:

SV+VNBC *(PV − UREF) = 0. Repeat Card 29 NNBC times.

- **Data Card 29 (I = 1 to NNBC)**

INBC(I,1)	Node number at which the mixed B.C. is specified
INBC(I,2)	Local DOF of the PV and SV at the node
VNBC(I)	Value of the coefficient of the PV in the B.C.
UREF(I)	Reference value of the PV

(**Table 7.3.2** continued)

- **Data Card 30**

NMPC	Number of multipoint constraints (solid mechanics)

SKIP Card 31 if no multipoint conditions are specified (NMPC = 0). The multipoint condition is assumed to be of the form:
VMPC(.,1)*PV1+VMPC(.,2)*PV2=VMPC(.,3). Repeat Card 31 NMPC times.

- **Data Card 31 (I = 1 to NMPC)**

IMC1(I,1)	Node number associated with PV1
IMC1(I,2)	Local DOF of PV1
IMC2(I,1)	Node number associated with PV2
IMC2(I,2)	Local DOF of PV2
VMPC(I)	Values of the coefficients of the constraint equation
VMPC(4)	Value of the force applied at the node of PV1 or PV2

Skip Card 32 if ITEM = 0 (read only for time-dependent or eigenvalue problems).

- **Data Card 32**

CT0	Constant part of CT = CT0 + CT1*X
CT1	Linear part of CT = CT0 + CT1*X

Skip remaining cards if steady-state or eigenvalue analysis is to be performed (ITEM = 0 or ITEM = 3).

- **Data Card 33**

DT	Time increment (uniform)
ALFA	Parameter in the time approximation scheme
GAMA	Parameter in the time approximation scheme*
	GAMA (not used when ITEM = 1: parabolic equation).

Give GAMA= 10^{-6} when centered difference is used (formulation in Problem 6.23 is the correct way to implement the centered difference scheme).

- **Data Card 34**

INCOND	Indicator for initial conditions
	INCOND = 0, Homogeneous (zero) initial conditions
	INCOND > 0, Nonhomogeneous initial conditions
NTIME	Number of time steps for which solution is sought
INTVL	Time step intervals at which solution is to be printed

Skip Cards 35 and 36 if initial conditions are zero (INCOND = 0).

- **Data Card 35**

GUO	Array of initial values of the primary variables

Skip Card 36 for parabolic equations (ITEM = 1).

- **Data Card 36**

GUI	Array of initial values of the first time derivatives of the primary variables.

format" used here, variables of each "data card" (we shall use this terminology to imply an input sequence in a single instruction) are read from the same line; if the values are not found on the same line, the computer will look for them on the next line(s). However, data required by different data cards cannot be put on single line; each data card must start with a new line. The space available after typing required data on a given line may be used to include any comments. For example, we may list the variable names on that line for ready reference but only after all of the required data are listed. The text included thereafter is *not* read by the computer (except to echo the input file).

Example 7.3.1 (Steady Heat Transfer in a Rod)

Consider the heat transfer problem of Example 4.3.3. The problem is governed by

$$-\frac{d^2\theta}{dx^2} + m^2\theta = 0 \quad \text{for} \quad 0 < x < L$$

$$\theta(0) = \theta_0, \quad \left(\frac{d\theta}{dx}\right)\bigg|_{x=L} = 0$$

where θ is the nondimensional temperature, and L, m^2, θ_0, β, and k are

$$L = 0.05 \text{ m}, \quad m^2 = 400/\text{m}^2, \quad \theta_0 = 300^\circ\text{C},$$
$$\beta = 100 \text{ W/(m}^2 \cdot {}^\circ\text{C)}, \quad k = 50 \text{ W/(m} \cdot {}^\circ\text{C)} \tag{7.3.1}$$

For this problem, we have MODEL = 1, NTYPE = 0, and ITEM = 0 (for a steady-state solution). Since $a = 1.0$ and $c = m^2 = P\beta/Ak = 400$ are the same for all elements, we set ICONT = 1, AX0 = 1.0, and CX0 = 400. All other coefficients are zero [$b = 0$ and $f = 0$] for this problem. For a uniform mesh of four linear elements (NEM = 4, IELEM = 1), the increments of the array DX(I) [where DX(1) is always the x coordinate of node 1 (see Figure 7.3.1) and $h = L/4 = 0.05/4 = 0.0125$] are: DX = {0.0, 0.0125, 0.0125, 0.0125, 0.0125}.

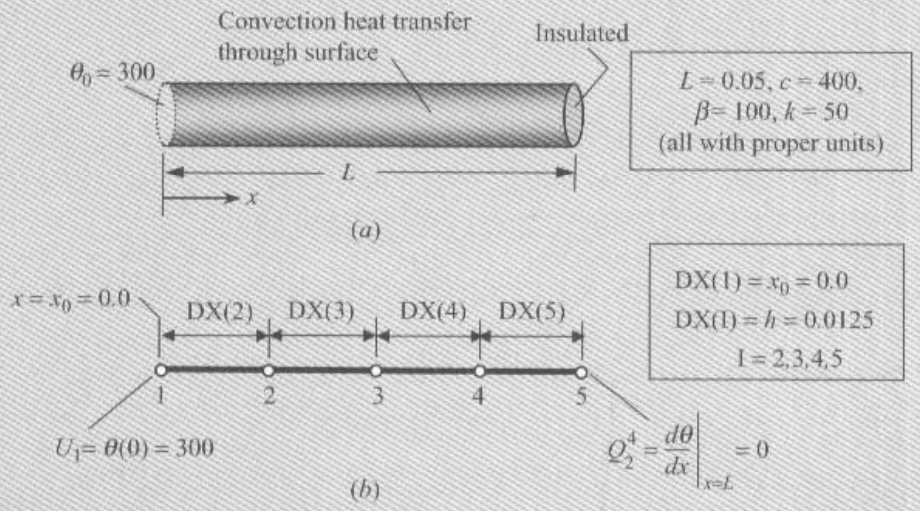


Figure 7.3.1 Finite element mesh of a rectangular fin.

The boundary conditions of the problem reduce to $U_1 = 0$ and $Q_2^4 = 0$. There is one specified boundary condition (BC) on the primary variable (NSPV = 1), and it is degree of freedom (DOF) (for heat transfer problems, there is only one DOF) at node 1 is: ISPV(1, 1) = 1 and ISPV(1, 2) = 1. The specified value is VSPV(1) = 300. Since the natural boundary condition ($Q_2^4 = 0$) is homogenous, there is no need to add a zero to the corresponding entry of the source vector (i.e., NSSV = 0). There are no mixed (i.e., convection) boundary conditions in this problem (NNBC = 0). The complete input data required to analyze the problem using **FEM1D** are presented in Box 7.3.1 and the output file is presented in Box 7.3.2. Input data and partial output for the same problem when a mesh of two quadratic elements are used are presented in Box 7.3.3.

Box 7.3.1 Input file from **FEM1D** for the problem in Example 7.3.1.

```
Example 7.3.1a: Heat transfer in a rod (4 linear elements)
  1 0 0                                    MODEL, NTYPE, ITEM
  1 4                                      IELEM, NEM
  1 1                                      ICONT, NPRNT
  0.0 0.0125  0.0125  0.0125  0.0125       DX(1)=X0; DX(2), etc. Element lengths
  1.0 0.0                                  AX0, AX1
  0.0 0.0                                  BX0, BX1
400.0 0.0                                  CX0, CX1
  0.0 0.0 0.0                              FX0, FX1,  FX2
  1                                        NSPV
  1  1   300.0                            ISPV(1, 1), ISPV(1,2), VSPV(1)
  0                                        NSSV
  0                                        NNBC
  0                                        NMPC
```

Box 7.3.2 Edited output from FEM1D for the problem in Example 7.3.1.

```
-------------------------------------------------------------

    OUTPUT from program FEM1D by J. N. REDDY

-------------------------------------------------------------

Example 7.3.1a: Heat transfer in a rod (4 linear elements)

 *** ANALYSIS OF MODEL 1, AND TYPE 0 PROBLEM ***
       (see the code below)
 MODEL=1, NTYPE=0: A problem described by MODEL EQ. 1
 MODEL=1, NTYPE=1: A circular DISK (PLANE STRESS)
 MODEL=1, NTYPE>1: A circular DISK (PLANE STRAIN)
 MODEL=2, NTYPE=0: A Timoshenko BEAM (RIE) problem
 MODEL=2, NTYPE=1: A Timoshenko PLATE (RIE) problem
 MODEL=2, NTYPE=2: A Timoshenko BEAM (CIE) problem
 MODEL=2, NTYPE>2: A Timoshenko PLATE (CIE) problem
 MODEL=3, NTYPE=0: A Euler-Bernoulli BEAM problem
 MODEL=3, NTYPE>0: A Euler-Bernoulli Circular plate
 MODEL=4, NTYPE=0: A plane TRUSS problem
 MODEL=4, NTYPE=1: A Euler-Bernoulli FRAME problem
 MODEL=4, NTYPE=2: A Timoshenko (CIE) FRAME problem

     Element type (0, Hermit e,>0, Lagrange)  ..= 1
     No. of deg. of freedom per node, NDF  ...= 1
     No. of elements in the mesh, NEM  .....= 4
     No. of total DOF in the model, NEQ  ..... = 5
     No. of specified primary DOF, NSPV  .....= 1
     No. of specified secondary DOF, NSSV....= 0
     No. of specified Newton B. C.: NNBC .....= 0
```

(**Box 7.3.2** is continued from the previous page)

```
Boundary information on primary variables:

   1   1   0.30000E+03

Global coordinates of the nodes, {GLX}:

 0.00000E+00 0.12500E-01  0.25000E-01  0.37500E-01  0.50000E-01

Coefficients of the differential equation:
     AX0 = 0.1000E+01    AX1 = 0.0000E+00
     BX0 = 0.0000E+00    BX1 = 0.0000E+00
     CX0 = 0.4000E+03    CX1 = 0.0000E+00
     FX0 = 0.6000E+01    FX1 = 0.0000E+00     FX2 =  0.0000E+00

Element coefficient matrix, [ELK]:
 0.81667E+02 -0.79167E+02
-0.79167E+02  0.81667E+02

Element source vector, {ELF}:
 0.00000E+00  0.00000E+00

SOLUTION (values of PVs) at the NODES:

 0.30000E+03  0.25152E+03  0.21892E+03  0.20016E+03  0.19403E+03
```

X	P. Variable	S. Variable
0.00000E+00	0.30000E+03	-0.38785E+04
0.12500E-01	0.25152E+03	-0.26076E+04
0.25000E-01	0.21893E+03	-0.15015E+04
0.37500E-01	0.20016E+03	-0.49018E+03

Box 7.3.3 Input and partial output using quadratic elements.

```
Example 7.3.1: Heat transfer in a rod (2 quadratic elements)

 1  0  0                    MODEL, NTYPE, ITEM
 2  2                       IELEM, NEM
 1  1                       ICONT, NPRNT
 0.0  0.025  0.025          DX(1)=X0; DX(2), etc. Ele. lengths
 1.0  0.0                   AX0,  AX1
 0.0  0.0                   BX0, BX1
400.0  0.0                  CX0,  CX1
 0.0  0.0  0.0              FX0,  FX1,  FX2
 1                          NSPV
 1  1  300.0                ISPV(1,1), ISPV(1,2), VSPV(1)
 0                          NSSV
 0                          NNBC
 0                          NMPC
```

(**Box 7.3.3** is continued from the previous page)

OUTPUT from program FEM1D by J. N. REDDY

Example 7.3.1: Heat transfer in a rod (2 quadratic elements)

Element type (0, Hermite, > 0, Lagrange) ..= 2
No. of deg. of freedom per node, NDF ....= 1
No. of elements in the mesh, NEM .....= 2
No. of total DOF in the model, NEQ .....= 5
No. of specified primary DOF, NSPV .....= 1
No. of specified secondary DOF, NSSV....= 0
No. of specified Newton B. C.: NNBC .....= 0

Global coordinates of the nodes, {GLX}:
0.00000 E+00 0.12500E-01 0.25000E-01 0.37500E-01 0.50000E-01

Element coefficient matrix, [ELK]:
0.94667E+02 -0.10600E+03 0.13000E + 02
-0.10600E+03 0.21867E+03 -0.10600E + 03
0.13000E+02 -0.10600E+03 0.94667E + 02

Element source vector, {ELF}:
0.25000E-01 0.10000E+00 0.25000E-01

SOLUTION (values of PVs) at the NODES:
0.30000E+03 0.25170E+03 0.21923E+03 0.20052E+03 0.19442E+03

X	P.Variable	S. Variable
0.00000E+00	0.30000E+03	-0.44971E+04
0.25000E-01	0.21924E+03	-0.19642E+04
0.25000E-01	0.21924E+03	-0.20014E+04
0.50000E-01	0.19443E+03	0.16471E+02

Example 7.3.2 (Steady Heat Transfer in a Composite Wall)

Here we consider the composite wall problem of Example 4.3.1. The governing equations of the problem are

$$-\frac{d}{dx}\left(kA\frac{dT}{dx}\right)=0, \quad 0<x<L \tag{7.3.2a}$$

$$T(0)=T_0, \qquad \left[kA\frac{dT}{dx}+\beta A(T-T_\infty)\right]_{x=L}=0 \tag{7.3.2b}$$

This is a problem with *discontinuous data* (ICONT = 0) because $a=kA$ is discontinuous (as well as $h_1 \neq h_2 \neq h_3$). For a three-element (nonuniform) mesh of linear elements (see Figure 7.3.2), the input data and edited output are given in Box 7.3.4.

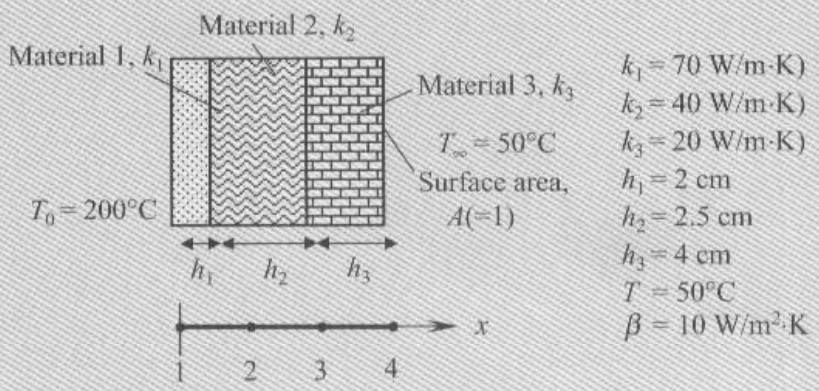


Figure 7.3.2 Heat transfer in a composite wall.

Box 7.3.4 Input for heat transfer in a composite wall (Example 7.3.2).

```
Example 7.3.2: Heat transfer in a composite wall
  1  0  0                    MODEL, NTYPE, ITEM
  1  3                       IELEM, NEM
  0  0                       ICONT, NPRNT
  4                          NNM
  1     2     0.02           NOD(1,J), GLX(1)
 70.0  0.0                   AX0, AX1
  0.0  0.0                   BX0, BX1
  0.0  0.0                   CX0, CX1
  0.0  0.0  0.0              FX0, FX1, FX2

  2     3     0.025          NOD(2,J) , GLX(2)
 40.0  0.0                   AX0, AX1
  0.0  0.0                   BX0, BX1
  0.0  0.0                   CX0, CX1
  0.0  0.0  0.0              FX0, FX1, FX2

  3     4     0.04           NOD(2,J), G LX(3)
 20.0  0.0                   AX0, AX1
  0.0  0.0                   BX0, BX1
  0.0  0.0                   CX0, CX1
  0.0  0.0  0.0              FX0, FX1, FX2
  1                          NSPV
  1  1  200.0                ISPV(1,1), ISPV(1,2), VSPV(1)
  0                          NSSV
  1                          NNBC
  4  1  10.0  50.0           INBC(1,1), INBC(1, 2), VNBC(1), UREF(1)
  0                          NMPC

OUTPUT from program  FEM1D  by J. N. REDDY

SOLUTION (values of PVs) at the NODES:
0.20000E+03  0.19958E+03  0.19867E+03  0.19576E+03
```

Data for Element 1

Data for Element 2

Data for Element 3

Example 7.3.3 (Radially Symmetric Heat Transfer in a Solid Cylinder)

Consider radially symmetric heat transfer in a solid circular cylinder of radius R_0 of Example 4.3.4. The governing equations of the problem are given by

$$-\frac{d}{dr}\left(2\pi kr\frac{dT}{dr}\right)=2\pi rg_0 \quad \text{for } 0<r<R_0 \tag{7.3.3a}$$

$$\left(2\pi kr\frac{dT}{dr}\right)\Big|_{r=0}=0, \quad T(R_0)=T_0 \tag{7.3.3b}$$

with $k=20$ W/(m·°C), $g_0=2\times10^8$ W/m^3, $T_0=100$°C, and $R_0=0.01$ m. We have MODEL = 1, NTYPE = 0, and ITEM = 0 (for a steady-state solution), and the data is continuous (ICONT = 1) in the domain for a mesh of two quadratic elements (IELEM = 2, NEM = 2). The data are

$$a=2\pi kr\rightarrow a_0=0, \quad a_1=2\pi k; \quad b=0\rightarrow b_0=0.0, \quad b_1=0.0$$

$$c=0\rightarrow c_0=0.0, \quad c_1=0.0; \quad f=2\pi g_0 r\rightarrow f_0=0.0, \quad f_1=2\pi g_0, \quad f_2=0.0$$

Thus, we have [for values $k=20$ W/(m·°C), $g_0=2\times10^8$ W/m^3]

$$\text{AX0}=0.0, \quad \text{AX1}=125.6637, \quad \text{BX0}=0.0, \quad \text{BX1}=0.0$$

$$\text{CX0}=0.0, \quad \text{CX1}=0.0, \quad \text{FX0}=0.0, \quad \text{FX1}=12.5664\times10^8, \quad \text{FX2}=0.0$$

For two-element mesh of quadratic elements, the array {DX} and boundary information are given by

$$\text{DX}=\{0.0, 0.005, 0.005\}$$

$$\text{NSPV}=1, \quad \text{ISPV}(1,1)=5, \quad \text{ISPV}(1,2)=1, \quad \text{VSPV}(1)=100$$

The input file along with the modified output from FEM1D are presented in Boxes 7.3.5 and 7.3.6, respectively. Note that the finite element solution obtained with two quadratic elements is more accurate than the solution obtained with four linear elements, and it is essentially the same as the exact solution (see Table 4.3.2 for the results).

Box 7.3.5 Input data for radially symmetric problem of Example 7.3.3.

```
Example 7.3.3: Radially symmetric heat transfer in a cylinder (2 quadratic ele)

1 0 0                          MODEL, NTYPE, ITEM
2 2                            IELEM, NEM
1 1                            ICONT, NPRNT
0.0  0.005  0.005              DX(1) = X0; DX(2), etc. Ele. lengths
0.0  125.6637                  AX0, AX1
0.0  0.0                       BX0, BX1
0.0  0.0                       CX0, CX1
0.0  12.56637E08  0.0          FX0, FX1, FX2
1                              NSPV
5 1  100.0                     ISPV(1, 1), ISPV(1, 2), VSPV(1)
0                              NSSV
0                              NNBC
0                              NMPC
```

Box 7.3.6 Partial output for radially symmetric problem of Example 7.3.3.

```
OUTPUT from program FEM1D by J. N. REDDY

Example 7.3.3: Radially symmetric heat transfer in a cylinder (2 quadratic ele)

 Element type (0, Hermite, >0, Lagrange)..=  2
      No. of deg. of freedom per node, NDF ....=  1
      No. of elements in the mesh, NEM .......=  2
      No. of total DOF in the model, NEQ ......=  5
      No. of specified primary DOF, NSPV ......=  1
      No. of specified secondary DOF, NSSV....=  0
      No. of specified Newton B. C.: NNBC .....=  0

 Element coefficient matrix, [ELK]:
 0.62832E+02 -0.83776E+02  0.20944E+02
-0.83776E+02  0.33510E+03 -0.25133E+03
 0.20944E+02 -0.25133E+03  0.23038E+03

 Element source vector, {ELF}:
 0.00000E+00   0.10472E+05   0.52360E+04

 SOLUTION (values of PVs) at the NODES:
 0.35000E+03   0.33437E+03   0.28750E+03   0.20937E+03   0.10000E+03
```

X	P. Variable	S. Variable
0.00000E-00	0.35000E+03	0.00000E+00
0.62500E-03	0.34902E+03	-0.24544E+03
0.12500E-02	0.34609E+03	-0.98175E+03
0.18750E-02	0.34121E+03	-0.22089E+04
0.25000E-02	0.33437E+03	-0.39270E+04
0.31250E-02	0.32559E+03	-0.61359E+04
0.37500E-02	0.31484E+03	-0.88357E+04
0.43750E-02	0.30215E+03	-0.12026E+05
0.50000E-02	0.28750E+03	-0.15708E+05
0.50000E-02	0.28750E+03	-0.15708E+05
0.56250E-02	0.27090E+03	-0.19880E+05
0.62500E-02	0.25234E+03	-0.24544E+05
0.68750E-02	0.23184E+03	-0.29698E+05
0.75000E-02	0.20937E+03	-0.35343E+05
0.81250E-02	0.18496E+03	-0.41479E+05
0.87500E-02	0.15859E+03	-0.48106E+05
0.93750E-02	0.13027E+03	-0.55223E+05
0.10000E-01	0.10000E+03	-0.62832E+05

Next, we consider a couple of steady-state solid mechanics problems. The reader is asked to visit the corresponding examples from Chapter 4 to have a complete background of the problems discussed here.

Example 7.3.4 (Axial Deformation of a Pier)

Here we consider the axial deformation of a pier (Example 4.5.1). The problem is governed by the equations (see Fig. 7.3.3)

$$-\frac{d}{dx}\left[\frac{1}{4}E(1+x)\frac{du}{dx}\right]=6.25(1+x),\quad 0<x<2 \tag{7.3.4a}$$

$$\left[\frac{1}{4}E(1+x)\frac{du}{dx}\right]\Big|_{x=0}=-5,\quad u(2)=0 \tag{7.3.4b}$$

For this case, the data of the problem is MODEL = 1, NTYPE = 1, and ITEM = 0 with

$$a=\frac{1}{4}E(1+x)\rightarrow a_0=0.25E,\quad a_1=0.25E;\quad b=0\rightarrow b_0=0.0,\quad b_1=0.0$$

$$c=0\rightarrow c_0=0.0,\quad c_1=0.0;\quad f=6.25(1+x)\rightarrow f_0=6.25,\quad f_1=6.25,\quad f_2=0.0$$

For $E=28\times 10^9$ N/m^2, the input data becomes

$$\text{AX0}=7\times 10^9,\quad \text{AX1}=7\times 10^9,\quad \text{BX0}=0.0,\quad \text{BX1}=0.0$$

$$\text{CX0}=0.0,\quad \text{CX1}=0.0,\quad \text{FX0}=6.25\times 10^3,\quad \text{FX1}=6.25\times 10^3,\quad \text{FX2}=0.0$$

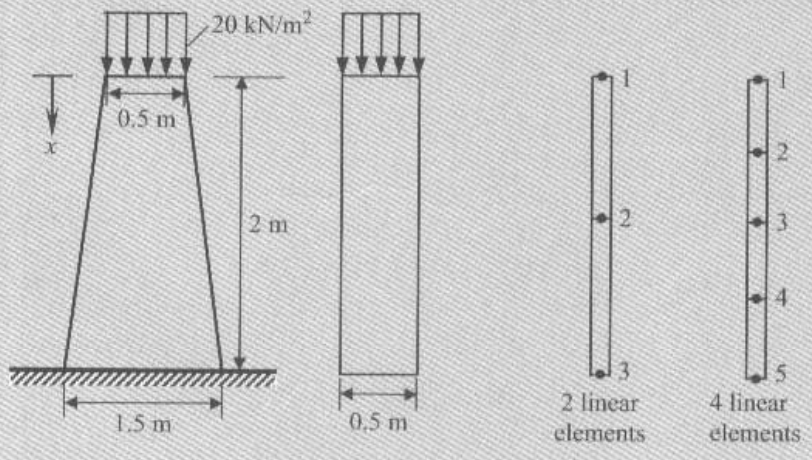


Figure 7.3.3 The concrete pier problem of Example 7.3.4.

For two-element mesh of linear elements, the array DX and boundary information are given by (NMPC = 0)

$$\text{DX} = \{0.0, 1.0, 1.0\}$$

$$\text{NSPV} = 1, \quad \text{ISPV}(1,1) = 3, \quad \text{ISPV}(1,2) = 1, \quad \text{VSPV}(1) = 0$$

$$\text{NSSV} = 1, \quad \text{ISSV}(1,1) = 1, \quad \text{ISSV}(1,2) = 1, \quad \text{VSSV}(1) = 5 \times 10^3$$

The input file and modified output of the problem are presented in Boxes 7.3.7 and 7.3.8, respectively.

Box 7.3.7 Input data for the pier problem of Example 7.3.4.

```
Example 7.3.4: Axial deformation of a pier
1 0 0                          MODEL, NTYPE, ITEM
1 2                            IELEM, NEM
1 1                            ICONT, NPRNT
 0.0      1.0    1.0           DX(1)=X0;  DX(2), etc. Ele. lengths
 7.0E9     7.0E9               AX0, AX1
 0.0      0.0                 BX0, BX1
 0.0      0.0                 CX0, CX1
 6.25E3    6.25E3    0.0      FX0, FX1, FX2
1                              NSPV
3 1   0.0                      ISPV(1,1), ISPV(1,2), VSPV(1)
1                              NSSV
1 1   5.0E3                    ISSV(1,1), ISSV(1,2), VSSV(1)
0                              NNBC
0                              NMPC
```

Box 7.3.8 Modified output for the pier problem of Example 7.3.4.

```
OUTPUT from program FEM1D by J. N. REDDY

Example 7.3.4: Axial deformation of a pier (2 linear elements)

 Boundary information on primary variables:
     3   1   0.00000E+00

 Boundary information on secondary variables:
     1   1   0.50000E+04

 Global coordinates of the nodes, {GLX}:
 0.00000 E+00  0.10000E+01   0.20000E+01
```

(**Box 7.3.8** is continued from the previous page)

```
Coefficients of the differential equation:
     AX0=0.7000E+10      AX1=0.7000E+10
     BX0=0.0000E+00      BX1=0.0000E+00
     CX0=0.0000E+00      CX1=0.0000E+00
     FX0=0.6250E+04      FX1=0.6250E+04      FX2= 0.0000E+00

 Element coefficient matrix, [ELK]:
  0.10500E+11 -0.10500E+11
 -0.10500E+11  0.10500E+11

 Element source vector, {ELF}:
  0.41667E+04   0.52083E+04

SOLUTION (values of PVs) at the NODES:
  0.21111E-05   0.12381E-05   0.00000E+00
```

Example 7.3.5 (Deformation of a Rotating Disk)

We wish to determine the deformation and stresses in a rotating homogeneous (solid) disk of thickness H and made of isotropic material (E, ν). The governing equation of the problem is (see Problem 4.37)

$$-\frac{d}{dr}\left[c\left(r\frac{du}{dr}+\nu u\right)\right]+c\left(\frac{u}{r}+\nu\frac{du}{dr}\right)=\rho\omega^2 r, \quad 0<r<R_0 \tag{7.3.5a}$$

$$c=\frac{E}{1-\nu^2} \tag{7.3.5b}$$

For this case, we have MODEL = 1, NTYPE = 1, and ITEM = 0. For a mesh of two quadratic elements (see Figure 7.3.4) (i.e., IEL = 2 and NEM = 2), we use ICONT = 1 and

$$\text{DX} = \{0.0,\ 0.5R_0,\ 0.5R_0\}$$

where R_0 is the radius of the disk. Since we are seeking results in nondimensional form, we take

$$R_0 = 1.0, \quad E_1 = E_2 = E = 1.0, \quad \nu_{12} = \nu = 0.3, \quad \rho\omega^2 = 1$$

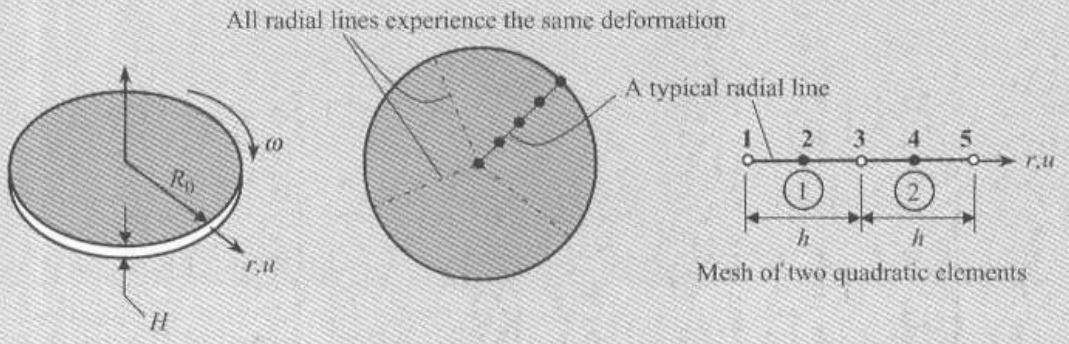


Figure 7.3.4 The rotating disk of Example 7.3.5.

where E_1 is the modulus along the radial direction and E_2 is the modulus in the circumferential direction. This information is supplied to the program through the following variables:

$$\text{AX0}(=E_1)=1.0,\ \text{AX1}(=E_2)=1.0,\ \text{BX0}(=\nu_{12})=0.3,\ \text{BX1}(=H)=1.0$$

$$\text{CX0}=0.0,\ \text{CX1}=0.0,\ \text{FX0}=0.0,\ \text{FX1}=1.0,\ \text{FX2}=0.0$$

The boundary conditions are

$$u(0)=0 \text{ (by symmetry)}, \quad Q_1^1=-(r\sigma_r)_0=0 \text{ at } r=R_0 \text{ (stress-free condition)}$$

Since the secondary variable is homogeneous, there is no need to include it in the data (NSSV=0). We have NSPV = 1, NNBC = 0, and NMPC = 0 with

$$\text{ISPV}(1,1)=1,\ \text{ISPV}(1,2)=1,\ \text{VSPV}(1)=0.0$$

The input data and modified output for this problem are presented in Boxes 7.3.9 and 7.3.10, respectively.

Box 7.3.9 Input data for the rotating disk problem of Example 7.3.5.

```
Example 7.3.5: Deformation of a circulardisk (2 quadratic elements)

 1  1  0                    MODEL, NTYPE, ITEM
 2  2                       IELEM, NEM
 1  1                       ICONT, NPRNT
  0.0  0.50   0.50          DX(I)
  1.0  1.0                  AX0, AX1
  0.3  1.0                  BX0, BX1
  0.0  0.0                  CX0, CX1
  0.0  1.0    0.0           FX0, FX1, FX2
 1                          NSPV
 1  1  0.0                  ISPV(1,1), ISPV(1, 2), VSPV(1)
 0                          NSSV
 0                          NNBC
 0                          NMPC
```

Box 7.3.10 Output for the rotating disk problem of Example 7.3.5.

```
OUTPUT from program FEM1D by J.N. REDDY

Example 7.3.5: Deformation of a circular disk (2 quadratic elements)

*** ANALYSIS OF MODEL 1, AND TYPE 1 PROBLEM ***

MODEL=1, NTYPE=0: A problem described by MODEL EQ. 1
MODEL=1, NTYPE=1: A circular DISK (PLANE STRESS)
MODEL=1, NTYPE>1: A circular DISK (PLANE STRAIN)
MODEL=2, NTYPE=0: A Timoshenko BEAM (RIE) problem
MODEL=2, NTYPE=1: A Timoshenko PLATE (RIE) problem
MODEL=2, NTYPE=2: A Timoshenko BEAM (CIE) problem
MODEL=2, NTYPE>2: A Timoshenko PLATE (CIE) problem
MODEL=3, NTYPE=0: A Euler-Bernoulli BEAM problem
MODEL=3, NTYPE>0: A Euler-Bernoulli Circular plate
```

(**Box 7.3.10** is continued from the previous page)

```
Element type (0, Hermite,>0, Lagrange) ..=  2
No. of deg. of freedom per node, NDF ....=  1
No. of elements in the mesh, NEM .......=  2
No. of total DOF in the model, NEQ ......=  5
No. of specified primary DOF, NSPV ......=  1
No. of specified secondary DOF, NSSV ...=  0
No. of specified Newton B. C.: NNBC .....=  0

Boundary information on primary variables:
   1   1   0.00000E+00

Global coordinates of the nodes, {GLX}:
 0.00000E+00   0.25000E+00   0.50000E+00   0.75000E+00  0.10000E+01

  Element coefficient matrix, [ELK]:
   0.15018E+01    0.64565E-09     0.10761E-08
   0.64565E-09    0.43956E+01    -0.21978E+01
   0.10761E-08   -0.21978E+01     0.25275E+01

  Element source vector, {ELF}:
   -0.20833E-02    0.25000E-01     0.18750E-01

 SOLUTION (values of PVs) at the NODES:
  0.00000E+00  0.70706E-01  0.13004E+00  0.16875E+00  0.17500E+00
```

X	Displacemnt	Radial Strs	Hoop Stress
0.00000E+00	0.00000E+00	0.33580E+00	0.10074E+00
0.62500E-01	0.18743E-01	0.42216E+00	0.42654E+00
0.12500E+00	0.36775E-01	0.40779E+00	0.41654E+00
0.18750E+00	0.54096E-01	0.39341E+00	0.40654E+00
0.25000E+00	0.70706E-01	0.37904E+00	0.39654E+00
0.31250E+00	0.86606E-01	0.36466E+00	0.38654E+00
0.37500E+00	0.10179E+00	0.35029E+00	0.37654E+00
0.43750E+00	0.11627E+00	0.33591E+00	0.36654E+00
0.50000E+00	0.13004E+00	0.32154E+00	0.35654E+00
0.50000E+00	0.13004E+00	0.32727E+00	0.35826E+00
0.56250E+00	0.14276E+00	0.28952E+00	0.34065E+00
0.62500E+00	0.15345E+00	0.25111E+00	0.32086E+00
0.68750E+00	0.16212E+00	0.21223E+00	0.29948E+00
0.75000E+00	0.16875E+00	0.17299E+00	0.27690E+00
0.81250E+00	0.17336E+00	0.13348E+00	0.25341E+00
0.87500E+00	0.17593E+00	0.93750E-01	0.22919E+00
0.93750E+00	0.17648E+00	0.53846E-01	0.20440E+00
0.10000E+01	0.17500E+00	0.13802E-01	0.17914E+00

The exact displacement and stresses in the disc are given by

$$u(r) = \frac{(1-\nu)R_0^3}{8E}\left[(3+\nu)\frac{r}{R_0} - (1+\nu)\left(\frac{r}{R_0}\right)^3\right]\rho\omega^2$$

$$\sigma_{rr}(r) = \frac{(3+\nu)R_0^2}{8}\left[1 - \left(\frac{r}{R_0}\right)^3\right]\rho\omega^2 \tag{7.3.6}$$

$$\sigma_{\theta\theta}(r) = \frac{(3+\nu)R_0^2}{8}\left[1 - \frac{1+3\nu}{3+\nu}\left(\frac{r}{R_0}\right)^3\right]\rho\omega^2$$

The exact displacements at $x = 0.25$, $x = 0.5$, $x = 0.75$, and $x = 1.0$ are 0.0704, 0.1302, 0.1686, and 0.1750, respectively. The maximum values of stresses occur at $r = 0$, and they are $\sigma_{rr}(0) = \sigma_{\theta\theta}(0) = 0.4125$. The displacements obtained with two quadratic elements are in good agreement with the analytical solution. The postcomputed stresses at $r = 0$ are not accurate because of the singularity there:

$$\sigma_{rr} = \frac{E}{1-\nu^2}\left(\frac{du}{dr} + \nu\frac{u}{r}\right), \quad \sigma_{\theta\theta} = \frac{E}{1-\nu^2}\left(\nu\frac{du}{dr} + \frac{u}{r}\right) \tag{7.3.7}$$

The next example deals with a multipoint constraint problem of the type discussed in Example 4.6.3.

Example 7.3.6 (Multipoint Constraint Problem)

A rigid bar AB of length $L = 1.6$ m is hinged to a support at A and supported by two vertical bars attached at points C and D, as shown in Figure 7.3.5. Both bars have the same cross-sectional

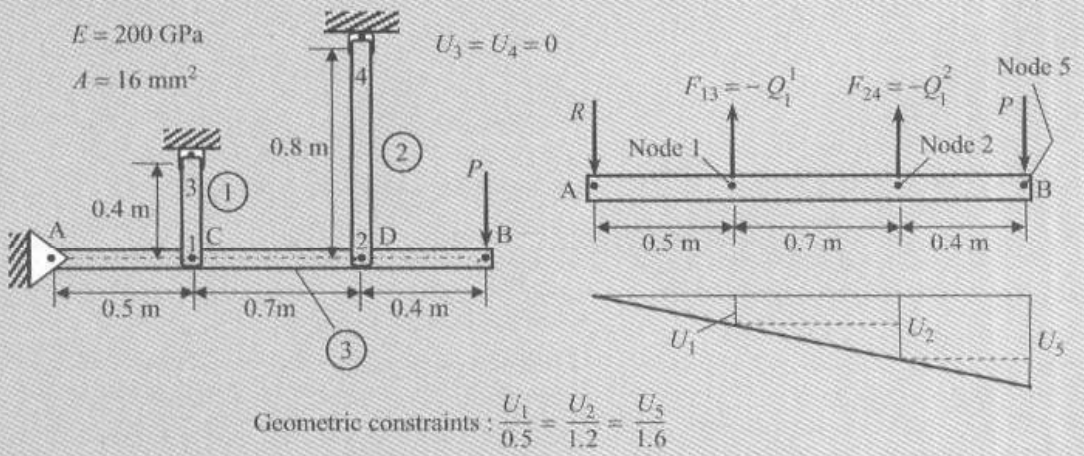


Figure 7.3.5 A rigid member supported by cables (Example 7.3.6).

area ($A = 16\ \text{mm}^2$) and are made of the same material with modulus $E = 200$ GPa. The lengths of the bars and the horizontal distances are shown in the figure. We wish to determine the elongations as well as tensile stresses in the bars.

The two geometric constraints are:

$$\frac{U_1}{0.5} = \frac{U_5}{1.6} \rightarrow 3.2U_1 - U_5 = 0, \quad \frac{U_2}{1.2} = \frac{U_5}{1.6} \rightarrow 1.333U_2 - U_5 = 0$$

Thus, we have [see Eq. (4.6.37)]

$$\beta_1^1 = 3.2,\ \beta_2^1 = -1.0,\ \beta_{12}^1 = 0.0; \quad \beta_1^2 = 1.333,\ \beta_2^2 = -1.0,\ \beta_{12}^2 = 0.0$$

In order to include the load P in the analysis, we introduce a node at B. Thus, we have NEM = 3 and NNM = 5 with $a = EA = 3.2 \times 10^6\ \text{m}^2$ for the bar elements and $a = 0$ for the third (rigid) element. It is sufficient to use linear finite elements to represent the bars. Thus, the data for the problem are MODEL = 1, NTYPE = 1, ITEM = 0, ICONT = 0. The boundary and constraint information are: NSPV = 2, ISPV(1,1) = 1, ISPV(1,2) = 1, VSPV(1) = 0, ISPV(2,1) = 2, ISPV(2, 2) = 1, VSPV(2) = 0, NSSV = 0, NNBC = 0, NMPC = 2, VMPC(1,1) = $\beta_1^1 = 3.2$, VMPC(1,2) = $\beta_2^1 = -1.0$, VMPC(1,3) = $\beta_{12}^1 = 0$, VMPC(1,4) = 0; VMPC(2,1) = $\beta_1^2 = 1.333$, VMPC(2,2) = $\beta_2^2 = -1.0$, VMPC(2,3) = $\beta_{12}^2 = 0$, and VMPC(2,4) = $P = 970$ N.

The input file and modified output of the problem are presented in Box 7.3.11. The displacements are $U_1 = 0.1$ mm, $U_2 = 0.24$ mm, and $U_5 = 0.32$ mm. The stresses are $\sigma_1 = 10^6(800/16) = 50$ MPa and $\sigma_2 = 10^6(960/16) = 60$ MPa.

Box 7.3.11 Input and edited output for the problem of Example 7.3.6.

```
Example 7.3.6: DEFORMATION OF A CONSTRAINED STRUCTURE
 1   0   0                    MODEL, NTYPE, ITEM
 1   3                        IELEM, NEM
 0   1                        ICONT, NPRNT
 5                            NNM

 1          3        0.4      NOD(1,J),GLX(1)
 3.2E6      0.0               AX0, AX1          Data for
 0.0        0.0               BX0, BX1          Element 1
 0.0        0.0               CX0, CX1
 0.0        0.0      0.0      FX0, FX1, FX2

 2          4        0.8      NOD(2,J),GLX(2)
 3.2E6      0.0               AX0, AX1          Data for
 0.0        0.0               BX0, BX1          Element 2
 0.0        0.0               CX0, CX1
 0.0        0.0      0.0      FX0, FX1, FX2

 1          5        1.6      NOD(3,J),GLX(3)
 0.0        0.0               AX0, AX1          Data for
 0.0        0.0               BX0, BX1          Element 3
 0.0        0.0               CX0, CX1
 0.0        0.0      0.0      FX0, FX1, FX2
```

(**Box 7.3.11** is continued from the previous page)

```
2                                   NS PV
3 1       0.0                       ISPV(1,1), ISPV(1,2), VSPV(1)
4 1       0.0                       ISpV(2,1), ISpV(2,2), VSpV(2)
0                                   NS SV
0                                   NN BC
2                                   NMPC
1 1       5 1       3.2             -1.0 0.0 0.0
2 1       5 1       1.33333         -1.0 0.0 970.0
                    IMC1(1,1),IMC1(1,2),IMC2(1,1),IMC2(1,2),
                                    (VMPC(1,I),I=1 to 4)
```

```
OUTPUT from program FEM1D by J. N. REDDY

Multipoint constraint information:

 1      1      5      1
0.32000E+01          -0.10000E+01          0.00000E+00          0.00000E+00
 2      1      5      1
0.13333E+01          -0.10000E+01          0.00000E+00          0.97000E+03
```

```
Properties of Element = 1

   Element length, H ..... = 0.4000E+00
   AX0 = 0.3200E+07     AX1 = 0.0000E+00
   BX0 = 0.0000E+00     BX1 = 0.0000E+00
   CX0 = 0.0000E+00     CX1 = 0.0000E+00
   FX0 = 0.0000E+00     FX1 = 0.0000E+00          FX2 = 0.0000E+00

Properties of Element = 2

   Element length, H ..... = 0.8000E+00
   AX0 = 0.3200E+07     AX1 = 0.0000E+00
   BX0 = 0.0000E+00     BX1 = 0.0000E+00
   CX0 = 0.0000E+00     CX1 = 0.0000E+00
   FX0 = 0.0000E+00     FX1 = 0.0000E+00          FX2 = 0.0000E+00

Properties of Element = 3

   Element length, H ..... = 0.1600E+01
   AX0 = 0.0000E+00     AX1 = 0.0000E+00
   BX0 = 0.0000E+00     BX1 = 0.0000E+00
   CX0 = 0.0000E+00     CX1 = 0.0000E+00
   FX0 = 0.0000E+00     FX1 = 0.0000E+00          FX2 = 0.0000E+00
```

(**Box 7.3.11** is continued from the previous page)

```
Element coefficient matrix, [ELK]:

 0.80000E+07 -0.80000E+07
-0.80000E+07  0.80000E+07

Element coefficient matrix, [ELK]:

 0.40000E+07 -0.40000E+07
-0.40000E+07  0.40000E+07

Element coefficient matrix, [ELK]:

 0.00000E+00  0.00000E+00
 0.00000E+00  0.00000E+00

SOLUTION (values of PVs) at the NODES:

 0.10000E-03   0.24000E-03   0.00000E+00   0.00000E+00   0.32001E-03

Forces at the constrained points:

 0.80001E+03   0.95999E+03
```

The next two examples deal with beams and frames of Chapter 5.

Example 7.3.7 (Clamped and Spring-Supported Beam)

Consider the spring-supported cantilever beam shown in Figure 7.3.6. We solve the problem first using the Euler–Bernoulli beam element (MODEL=3, NTYPE=0, IELEM=0) and then using the Timoshenko beam element (MODEL=2, NTYPE=0 or 2, IELEM=1,2,or 3). Since the loading is discontinuous, we set ICONT=0. A minimum of two elements are required to model the problem (i.e., NEM=2).

If we take $EI = 1.0E6$ (i.e., 10^6 lb-ft.^{2}), then

$$GAK = \frac{E}{2(1+\nu)} BH\frac{5}{6} = \frac{EI}{1+\nu}\frac{5}{H^2}$$

For $L/H = 10$, we have $H = 1.0$ because $L = 10$ ft. For the choice $\nu = 0.25$, we have

$$GAK = 4EI/H^2 = 4 \times 10^6 \text{ lb}$$

Thus, we use

$$\text{AX0} = 0.0, \quad \text{AX1} = 0.0, \quad \text{BX0}(= EI) = 1.0 \times 10^6$$

$$\text{BX1} = 0.0, \quad \text{CX0} = 0.0, \quad \text{CX1} = 0.0$$

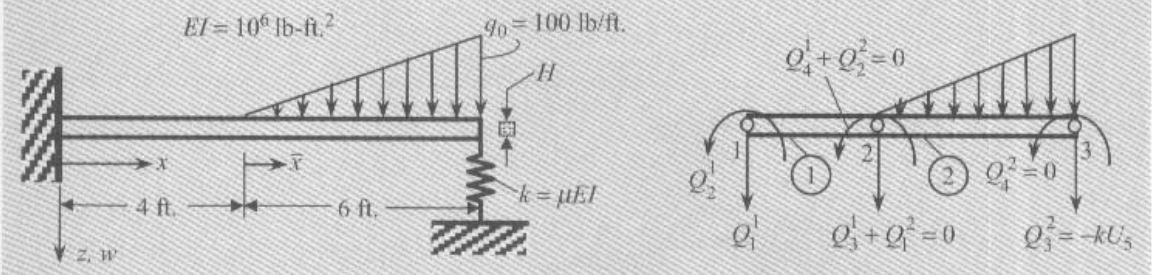


Figure 7.3.6 The spring-supported beam of Example 7.3.7.

for the Euler–Bernoulli beam and

$$\text{AX}(=GAK) = 4.0 \times 10^6, \quad \text{AX1} = 0.0, \quad \text{BX0}(=EI) = 1.0 \times 10^6$$
$$\text{BX1} = 0.0, \quad \text{CX0} = 0.0, \quad \text{CX1} = 0.0$$

for the Timoshenko beam.

The distributed transverse load is zero in element 1, and it is

$$q(\bar{x}) = \frac{100}{6}\bar{x}$$

in element 2. Hence,

$$\text{FX0} = 0.0, \quad \text{FX1} = 0.0, \quad \text{FX2} = 0.0 \text{ in element 1}$$
$$\text{FX0} = 0.0, \quad \text{FX1} = 16.666, \quad \text{FX2} = 0.0 \text{ in element 2}$$

The global coordinates of nodes and the connectivity matrix entries for each element are obvious from the geometry. For the Euler–Bernoulli beam element, the number of nodes is always equal to two (NPE = 2), whereas for the Timoshenko beam element, the number of nodes depends on the degree of interpolation (or element type) selected:

$$\text{NPE} = \text{IELEM} + 1$$

The boundary conditions for this problem are

$$w(0) = 0, \quad \left(\frac{dw}{dx}\right)\bigg|_{x=0} = 0, \quad (V + kw)\bigg|_{x=L} = 0$$

where V is the shear force. Therefore, we have (NSSV=0)

$$\text{NSPV} = 2, \quad \text{ISPV}(1,1) = 1, \quad \text{ISPV}(1,2) = 1, \quad \text{VSPV}(1) = 0.0$$
$$\text{ISPV}(2,1) = 1, \quad \text{ISPV}(2,2) = 2, \quad \text{VSPV}(2) = 0.0$$
$$\text{NNBC} = 1, \quad \text{INBC}(1,1) = 3, \quad \text{INBC}(1,2) = 1$$
$$\text{VNBC}(1)\ (=k) = 1.0 \times 10^0 \quad (\text{for } \mu = 1) \text{ and } 0.0 \ (\text{for } \mu = 0.0)$$
$$\text{UREF}(1) = 0.0$$

The inputs and modified outputs for the Euler–Bernoulli and Timoshenko elements are presented in Boxes 7.3.12 and 7.3.13, respectively. Note that the Euler–Bernoulli element is a Hermite cubic element, whereas the Timoshenko element is only a linear element (a quadratic element may be used if desired).

Box 7.3.12 Input and edited output files for the cantilever beam problem of Example 7.3.7 (Euler–Bernoulli beam element).

```
Example 7.3.7: CLAMPED AND SPRING-SUPPORTED BEAM (Euler-Bernoulli)
     3    0    0                                MODEL, NTYPE, ITEM
     0    2                                     IELEM, NEM
     0    1                                     ICONT, NPRNT
          3                                     NNM
          1    2     4.0                        NOD(1,J),GLX(1)
          0.0       0.0                        AX0, AX1        |   Data for
          1.0E6     0.0                        BX0, BX1        |   Element 1
          0.0       0.0                        CX0, CX1        |
          0.0       0.0          0.0           FX0,FX1,FX2     |
          2    3    6.0                        NOD(2,J),GLX(2)
          0.0       0.0                        AX0, AX1        |   Data for
          1.0E6     0.0                        BX0, BX1        |   Element 2
          0.0       0.0                        CX0, CX1        |
          0.0       16.666667    0.0           FX0,FX1,FX2     |
     2                                          NSPV
     1         1    0.0                        ISPV(1,J), VSPV(1)
     1         2    0.0                        ISPV(2,J), VSPV(2)
     0                                          NSSV
     1                                          NNBC (with spring)
     3         1    1.0E4          0.0         INBC(1,1),INBC(1,2),VNBC(1),UREF
     0                                          NMPC
```

OUTPUT from program FEM1D by J. N. REDDY

```
Example 7.3.7: CLAMPED AND SPRING-SUPPORTED BEAM (Euler-Bernoulli)

     *** ANALYSIS OF MODEL 3, AND TYPE 0 PROBLEM ***
                 (see the code below)

     MODEL=2, NTYPE=0: A Timoshenko BEAM (RIE) problem
     MODEL=2, NTYPE=1: A Timoshenko PLATE (RIE) problem
     MODEL=2, NTYPE=2: A Timoshenko BEAM (CIE) problem
     MODEL=2, NTYPE>2: A Timoshenko PLATE (CIE) problem
     MODEL=3, NTYPE=0: A Euler-Bernoulli BEAM problem

      Properties of Element = 1

           Element length, H ..... = 0.4000E+01
           AX0 =  0.0000E+00    AX1 = 0.0000E+00
           BX0 =  0.1000E+07    BX1 = 0.0000E+00
           CX0 =  0.0000E+00    CX1 = 0.0000E+00
           FX0 =  0.0000E+00    FX1 = 0.0000E+00    FX2 =  0.0000E+00
```

(**Box 7.3.12** is continued from the previous page)

```
Properties of Element =  2

        Element length, H . . . . . = 0.6000E+01
        AX0 =  0.0000E+00    AX1 = 0.0000E+00
        BX0 =  0.1000E+07    BX1 = 0.0000E+00
        CX0 =  0.0000E+00    CX1 = 0.0000E+00
        FX0 =  0.0000E+00    FX1 = 0.1667E+02   FX2 =  0.0000E+00

 Element coefficient matrix, [ELK]:

  0.18750E+06 -0.37500E+06 -0.18750E+06 -0.37500E+06
 -0.37500E+06  0.10000E+07  0.37500E+06  0.50000E+06
 -0.18750E+06  0.37500E+06  0.18750E+06  0.37500E+06
 -0.37500E+06  0.50000E+06  0.37500E+06  0.10000E+07

 Element coefficient matrix, [ELK]:

  0.55556E+05 -0.16667E+06 -0.55556E+05 -0.16667E+06
 -0.16667E+06  0.66667E+06  0.16667E+06  0.33333E+06
 -0.55556E+05  0.16667E+06  0.55556E+05  0.16667E+06
 -0.16667E+06  0.33333E+06  0.16667E+06  0.66667E+06

 Element source vector, {ELF}:

  0.90000E+02 -0.12000E+03  0.21000E+03   0.18000E+03

SOLUTION (values of PVs) at the NODES:
0.00000E+00  0.00000E+00  0.46272E-02  -0.19510E-02  0.16403E-01
-0.16985E-02
```

x is the local coord. if ICONT=0

x	Deflect.	Rotation	B. Moment	Shear Force
0.00000E+00	0.00000E+00	0.00000E+00	0.75969E+03	-0.13597E+03
0.50000E+00	0.92129E-04	-0.36285E-03	0.69171E+03	-0.13597E+03
0.10000E+01	0.35718E-03	-0.69171E-03	0.62372E+03	-0.13597E+03
0.15000E+01	0.77817E-03	-0.98657E-03	0.55574E+03	-0.13597E+03
0.20000E+01	0.13381E-02	-0.12474E-02	0.48775E+03	-0.13597E+03
0.25000E+01	0.20200E-02	-0.14743E-02	0.41977E+03	-0.13597E+03
0.30000E+01	0.28068E-02	-0.16672E-02	0.35178E+03	-0.13597E+03
0.35000E+01	0.36815E-02	-0.18261E-02	0.28380E+03	-0.13597E+03
0.40000E+01	0.46272E-02	-0.19510E-02	0.21582E+03	-0.13597E+03
0.00000E+00	0.46272E-02	-0.19510E-02	0.95815E+02	-0.45969E+02
0.75000E+00	0.61142E-02	-0.20099E-02	0.61338E+02	-0.45969E+02
0.15000E+01	0.76357E-02	-0.20430E-02	0.26862E+02	-0.45969E+02
0.22500E+01	0.91722E-02	-0.20502E-02	-0.76154E+01	-0.45969E+02
0.30000E+01	0.10705E-01	-0.20316E-02	-0.42092E+02	-0.45969E+02
0.37500E+01	0.12213E-01	-0.19871E-02	-0.76569E+02	-0.45969E+02
0.45000E+01	0.13679E-01	-0.19167E-02	-0.11105E+03	-0.45969E+02
0.52500E+01	0.15082E-01	-0.18205E-02	-0.14552E+03	-0.45969E+02
0.60000E+01	0.16403E-01	-0.16985E-02	-0.18000E+03	-0.45969E+02

Box 7.3.13 Input and edited output files for the cantilever beam problem of Example 7.3.7 (Timoshenko beam element).

```
Example 7.3.7: CLAMPED AND SPRING-SUPPORTED BEAM (Timoshenko)
  2   0   0                                  MODEL, NTYPE, ITEM
  1   2                                      IELEM, NEM
  0   1                                      ICONT, NPRNT
      3                                      NNM
      1   2     4.0                          NOD(1,J), GLX(1)
      4.0E6     0.0                          AX0, AX1          |   Data for
      1.0E6     0.0       (L/H = 10)         BX0, BX1          |   Element 1
      0.0       0.0                          CX0, CX1          |
      0.0       0.0       0.0                FX0, FX1, FX2     |
      2   3     6.0                          NOD(2,J), GLX(2)
      4.0E6     0.0                          AX0, AX1          |   Data for
      1.0E6     0.0                          BX0, BX1          |   Element 2
      0.0       0.0                          CX0, CX1          |
      0.0       16.666667     0.0            FX0, FX1, FX2     |
  2                                          NSPV
  1   1         0.0                          ISPV(1,J), VSPV(1)
  1   2         0.0                          ISPV(2,J), VSPV(2)
  0                                          NSSV
  1                                          NNBC (with spring)
  3   1         1.0E4           0.0          INBC(1,1),INBC(1,2),VNBC(1),UREF
  0                                          NMPC
```

OUTPUT from program FEM1D by J. N. REDDY

Example 7.3.7: CLAMPED AND SPRING-SUPPORTED BEAM (Timoshenko)

Properties of Element = 2

Element length, H . . . . . = 0.6000E+01
AX0 = 0.4000E+07 AX1 = 0.0000E+00
BX0 = 0.1000E+07 BX1 = 0.0000E+00
CX0 = 0.0000E+00 CX1 = 0.0000E+00
FX0 = 0.0000E+00 FX1 = 0.1667E+02 FX2 = 0.00

Element coefficient matrix, [ELK]:

0.66667E+06	-0.20000E+07	-0.66667E+06	-0.20000E+07
-0.20000E+07	0.61667E+07	0.20000E+07	0.58333E+07
-0.66667E+06	0.20000E+07	0.66667E+06	0.20000E+07
-0.20000E+07	0.58333E+07	0.20000E+07	0.61667E+07

Element source vector, {ELF}:

0.10000E+03	0.00000E+00	0.20000E+03	0.00000E+00

SOLUTION (values of PVs) at the NODES:

0.00000E+00	0.00000E+00	0.38273E-02	-0.18473E-02	0.16727E-01
-0.24364E-02				

(**Box 7.3.13** is continued from the previous page)

x is the the local coord.if ICONT=0

x	Deflect.	Rotation	B. Moment	Shear Force
0.00000E+00	0.00000E+00	0.00000E+00	0.46182E+03	-0.38273E+04
0.50000E+00	0.47841E-03	-0.23091E-03	0.46182E+03	-0.29036E+04
0.10000E+01	0.95682E-03	-0.46182E-03	0.46182E+03	-0.19800E+04
0.15000E+01	0.14352E-02	-0.69273E-03	0.46182E+03	-0.10564E+04
0.20000E+01	0.19136E-02	-0.92364E-03	0.46182E+03	-0.13273E+03
0.25000E+01	0.23920E-02	-0.11545E-02	0.46182E+03	0.79091E+03
0.30000E+01	0.28705E-02	-0.13855E-02	0.46182E+03	0.17145E+04
0.35000E+01	0.33489E-02	-0.16164E-02	0.46182E+03	0.26382E+04
0.40000E+01	0.38273E-02	-0.18473E-02	0.46182E+03	0.35618E+04
0.00000E+00	0.38273E-02	-0.18473E-02	0.98182E+02	-0.12109E+04
0.75000E+00	0.54398E-02	-0.19209E-02	0.98182E+02	-0.91636E+03
0.15000E+01	0.70523E-02	-0.19945E-02	0.98182E+02	-0.62182E+03
0.22500E+01	0.86648E-02	-0.20682E-02	0.98182E+02	-0.32727E+03
0.30000E+01	0.10277E-01	-0.21418E-02	0.98182E+02	-0.32727E+02
0.37500E+01	0.11890E-01	-0.22155E-02	0.98182E+02	0.26182E+03
0.45000E+01	0.13502E-01	-0.22891E-02	0.98182E+02	0.55636E+03
0.52500E+01	0.15115E-01	-0.23627E-02	0.98182E+02	0.85091E+03
0.60000E+01	0.16727E-01	-0.24364E-02	0.98182E+02	0.11455E+04

Example 7.3.8 (Analysis of a Plane Truss)

Here, we consider the three-member plane truss shown in Figure 7.3.7 (see Example 4.6.1). A plane truss problem falls into MODEL = 4 and NTYPE = 0. To obtain the solution in nondimensional form, we take $E = 1$, $A = 1$, $P = 1$, and $L = 1$. The data are discontinuous (ICONT=0). The input and partial output are presented in Box 7.3.14.

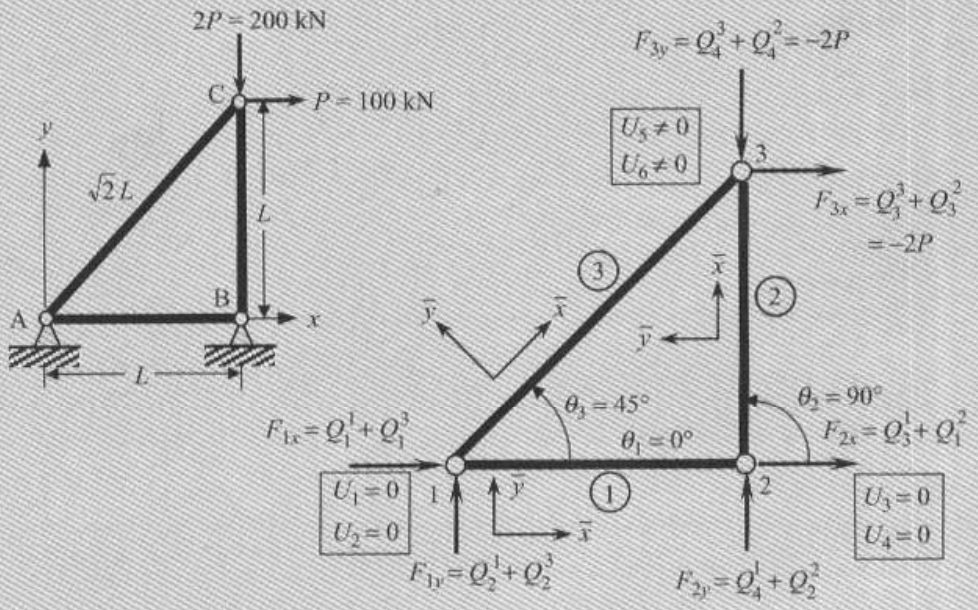


Figure 7.3.7 The plane truss problem of Example 7.3.8.

Box 7.3.14 Input and partial output for the plane truss problem of Example 7.3.8.

```
Example 7.3.8: ANALYSIS OF A PLANE TRUSS
 4 0 0                                  MODEL, NTYPE, ITEM
 0 3                                    IELEM, NEM
 0 1                                    ICONT, NPRNT
   3                                    NNM
 1.0   1.0   1.0   1.0   0.0  0.0      SE, SL, SA, CS, SN, HF
   2   3                                NOD(1, I)  (Element 1)
 1.0 1.414 1.0   0.707 0.707  0.0      same for Element 2
   1   3
 1.0 1.0   1.0   0.0   1.0
 0.0 0.0   0.0   0.0   0.0              Same for Element 3
   1   2                                0          NCON
   4                                    NSPV
   1   1   0.0
   1   2   0.0                          ISPV, VSPV
   2   1   0.0
   2   2   0.0
   2                                    NSSV
   3   1  -2.0                          ISSV, VSSV
   3   2   1.0
   0                                    NNBC
   0                                    NMPC
```

```
OUTPUT from program  FEM1D  by J. N. REDDY

SOLUTION (values of PVs) at the NODES:

 0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  -0.30000E+01  -0.58289E+01

   Generalized forces in the element coordinates
  (second line gives the results in the global coordinates)

Ele    Force, H1       Force, V1       Force, H2       Force, V2
 1   0.3000E+01      0.0000E+00     -0.3000E+01     0.0000E+00
     0.3000E+01      0.0000E+00     -0.3000E+01     0.0000E+00
 2  -0.1414E+01      0.0000E+00      0.1414E+01     0.0000E+00
    -0.1000E+01     -0.1000E+01      0.1000E+01     0.1000E+01
 3   0.0000E+00      0.0000E+00      0.0000E+00     0.0000E+00
     0.0000E+00      0.0000E+00      0.0000E+00     0.0000E+00
```

Example 7.3.9 (Analysis of a Plane Frame)

This example deals with the two-member frame structure shown in Figure 7.3.8 (see Example 5.4.1). We shall analyze the frame for displacements and member forces using first

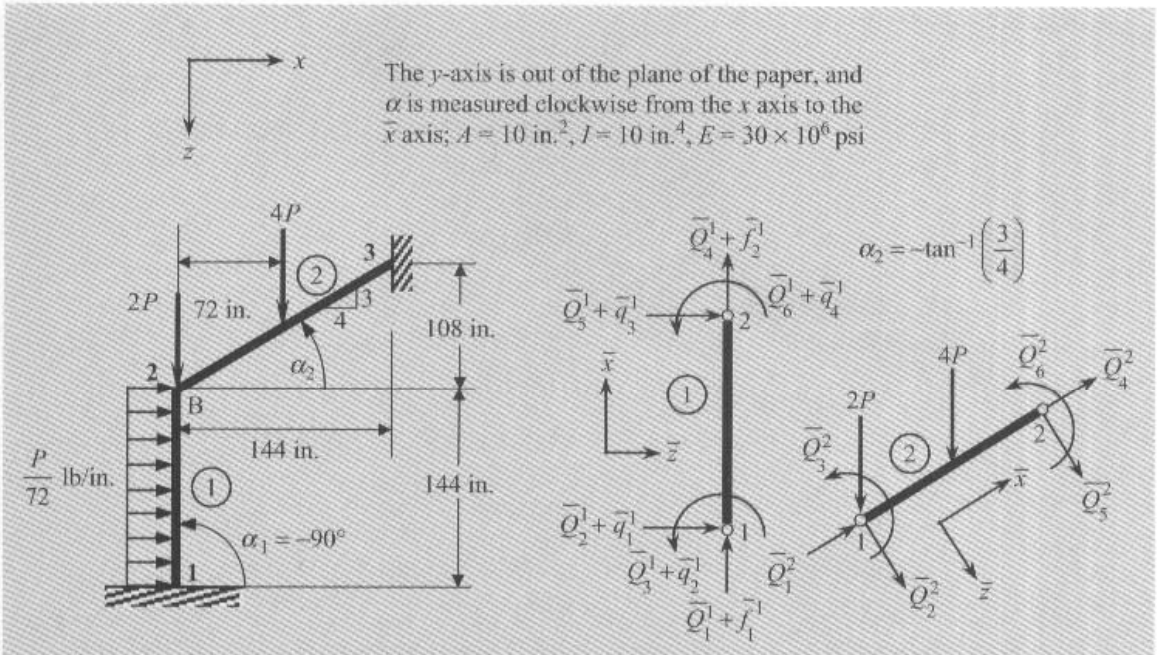


Figure 7.3.8 The plane frame problem of Example 7.3.9.

the Euler–Bernoulli frame element (MODEL = 4, NTYPE = 1) and then using the Timoshenko frame element (MODEL = 4, NTYPE = 2). The input files of the problem using two Euler-Bernoulli elements and 2 (linear) Timoshenko beam elements are presented in Box 7.3.15.

Box 7.3.15 Input data for the plane frame problem of Example 7.3.9.

```
Example 7.3.9(a): ANALYSIS OF A PLANE FRAME (2 Euler-Bernoulli elements)
 4  1  0                          MODEL, NTYPE, ITEM
 0  2                             IELEM, NEM
 0  1                             ICONT, NPRNT
 3                                NNM
 0.3  1.0E6  144.0  10.0  10.0  0.0   -1.0  PR, SE, SL, SA, SI, CS, SN
 0.0  0.0138888  0.0  0.0  0.0  0.0   HF, VF, PF, XB, CST, SNT
 1  2                                 NOD(1,J)
 0.3  1.0E6  180.0  10.0  10.0  0.8   -0.6  PR, SE, SL, SA, SI, CS, SN
 0.0  0.0  4.0  90.0  -0.6  0.8       HF, VF, PF, XB, CST, SNT
 2  3                                 NOD(2,J)
                                  0   NCON
 6                                NSPV
 1  1  0.0
 1  2  0.0
 1  3  0.0
 3  1  0.0                        ISPV, VSPV
 3  2  0.0
 3  3  0.0
 1                                NSSV
 2  2  2.0                        ISSV, VSSV
 0                                NNBC
 0                                NMPC
```

(**Box 7.3.15** is continued from the previous page)

```
Example 7.3.9(b): ANALYSIS OF A PLANE FRAME (2 TIMOSHENKO elements)
 4 2 0                                  MODEL, NTYPE, ITEM
 0 2                                    IELEM, NEM
 0 1                                    ICONT, NPRNT
 3                                      NNM
 0.3  1.0E6  144.0  10.0  10.0  0.0  -1.0   PR, SE, SL, SA, SI, CS, SN
 0.0  0.0138888  0.0  0.0  0.0  0.0         HF, VF, PF, XB, CST, SNT
 1  2                                   NOD(1,J)
 0.3  1.0E6  180.0  10.0  10.0  0.8  -0.6   PR, SE, SL, SA, SI, CS, SN
 0.0  0.0      4.0  90.0  -0.6  0.8         HF, VF, PF, XB, CST, SNT
 2  3                                   NOD(2,J)
                                0          NCON
 6                                      NSPV
 1  1  0.0
 1  2  0.0
 1  3  0.0
 3  1  0.0                              ISPV, VSPV
 3  2  0.0
 3  3  0.0
 1                                      NSSV
 2  2  2.0                              ISSV, VSSV
 0                                      NNBC
 0                                      NMPC
```

The next two examples deal with the use of **FEM1D** for eigenvalue and time-dependent problems (see Chapter 6).

Example 7.3.10 (Natural Vibration of a Cantilever Beam)

Here, we consider natural vibrations of a cantilever beam (see Example 6.1.2). The input data for all variables is the same as in the static analysis. In addition, we must input c_{i0} and c_{i1}. For the Timoshenko beam theory, c_{i0} denotes the inertia ρA, and c_{i1} denotes the rotatory inertia ρI. The eigenvalue solver used in **FEM1D** requires the matrix $[B]$ in $[A]\{x\} = \lambda[B]\{x\}$ to be positive-definite. Hence, c_{i0} and c_{i1} should be nonzero, otherwise, the mass matrix coefficients associated with Ψ will be zero. When rotary inertia is omitted in the Timoshenko beam theory, a small value for the inertia coefficient is used in order to ensure positive-definiteness of the mass matrix.

The input files for the natural vibrations of the cantilever beam by the two types of elements are given in Box 7.3.16. The reader can investigate the convergence characteristics of the elements in improving the accuracy of the fundamental frequency with the use of **FEM1D**.

Box 7.3.16 Input data files for beam vibration of Example 7.3.10.

```
Example 7.3.10(a): NATURAL VIBRATIONS OF A CANTILEVER BEAM (E-B; w/o RI)
 3 0 3                          MODEL, NTYPE, ITEM
 0 2                            IELEM, NEM
 1 1                            ICONT, NPRNT
  0.0   0.5   0.5               DX(I)
  0.0   0.0                     AX0, AX1
  1.0   0.0                     BX0, BX1
  0.0   0.0                     CX0, CX1
  2                             NSPV
 1 1                            ISPV(1, J)
 1 2                            ISPV(2, J)
 0                              NNBC
 0                              NMPC
 1.0  8.3333-06                 CT0, CT1
-------------------------------------------------------------------------
Example 7.3.10(b): NATURAL VIB. OF A CANTILEVER BEAM (TIM element w/o RI)
 2 0 3                          MODEL, NTYPE,  ITEM
 1 2                            IELEM, NEM
 1 1                            ICONT, NPRNT
 0.0   0.5   0.5                DX(I)
 4.0E4 0.0                      AX0, AX1  (nu=0.25, H=0.01)
 1.0   0.0                      BX0, BX1
 0.0   0.0                      CX0, CX1
 2                              NSPV
 1 1                            ISPV(1,J)
 1 2                            ISPV(2,J)
 0                              NNBC
 0                              NMPC
 1.0  8.33333E-06               CT0, CT1  (L/H=100)
```

Example 7.3.11 (Transient Heat Transfer)

Consider the transient heat conduction in a plane wall. The governing equation is

$$\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} = 0 \quad \text{for } 0 < x < 1 \tag{7.3.8}$$

The boundary conditions considered are

$$\textbf{Set 1:} \quad u(0,t) = 0 \quad \frac{\partial u}{\partial x}(1,t) = 0; \quad \textbf{Set 2:} \quad u(0,t) = 0, \quad \left(\frac{\partial u}{\partial x} + u\right)\bigg|_{x=1} = 0 \tag{7.3.9}$$

and the initial condition is taken to be

$$u(x,0) = 1 \tag{7.3.10}$$

We have MODEL = 1, NTYPE = 0, ICONT = 1, NSPV = 1, and NSSV = 0; NNBC = 0 for Set 1 and NNBC = 1 for Set 2. The coefficients of the differential equations are ($a = 1.0$, $b = 0.0$, $c = 0.0$, $c_t = 1.0$, and $f = 0$)

$$AX0 = 1.0, \quad AX1 = 0.0, \quad BX0 = 0.0, \quad BX1 = 0.0, \quad CX0 = 0.0, \quad CX1 = 0.0$$

$$FX0 = 0.0, \quad FX1 = 0.0, \quad FX2 = 0.0, \quad CT0 = 1.0, \quad CT1 = 0.0$$

Box 7.3.17 Input files for the transient heat conduction problem of Example 7.3.11.

```
Example 7.3.11(a): Transient heat conduction in a plane wall (Set 1)
 1 0 1                    MODEL, NTYPE, ITEM
 1 2                      IELEM, NEM
 1 1                      ICONT, NPRNT
  0.0  0.5  0.5           DX(I)
  1.0  0.0                AX0, AX1
  0.0  0.0                BX0, BX1
  0.0  0.0                CX0, CX1
  0.0  0.0  0.0           FX0, FX1, FX2
  1                       NSPV
  1 1  0.0                ISPV(I, J), VSPV(I)
  0                       NSSV
  0                       NNBC
  0                       NMPC
  1.0  0.0                CT0, CT1
  0.05 0.5  0.0           DT, ALFA, GAMA
  1    20   1             INCOND, NTIME, INTVL
  0.0  1.0  1.0           GU0(I)

Example 7.3.11(b): Transient heat conduction in a plane wall (Set 2)
 1 0 1                    MODEL, NTYPE, ITEM
 1 2                      IELEM, NEM
 1 1                      ICONT, NPRNT
  0.0  0.5  0.5           DX(I)
  1.0  0.0                AX0, AX1
  0.0  0.0                BX0, BX1
  0.0  0.0                CX0, CX1
  0.0  0.0  0.0           FX0, FX1, FX2
  1                       NSPV
  1 1   0.0               ISPV(I, J), VSPV(I)
  0                       NSSV
  1                       NNBC
  0                       NMPC
  5 1  1.0  0.0           INBC, VNBC, UREF
  1.0  0.0                CT0, CT1
  0.05 0.5  0.0           DT, ALFA, GAMA
  1   20   1              INCOND, NTIME, INTVL
  0.0  1.0  1.0           GU0(I)
```

The boundary and initial conditions (since INCOND=1) are input as

$$\text{ISPV}(1,1)=1, \quad \text{ISPV}(1,2)=1, \quad \text{VSPV}(1)=0.0; \quad \text{GU0(I)}=\{0.0, 1.0, 1.0, \ldots\}$$

From the discussions of Example 6.2.1, we use $\Delta t = 0.05$ (DT=0.05) and print the solution for every time step (i.e., INTVL=1).

The input files of the problem for the two sets of boundary conditions are presented in Box 7.3.17 for a mesh of two linear elements.

7.4 SUMMARY

In this chapter three main items have been discussed: numerical integration of finite element coefficient matrices and vectors, numerical implementation of a typical finite element program and their contents, and applications of the finite element program **FEM1D.** The numerical evaluation of the coefficients is required because of (*a*) variable coefficients of the differential equations modeled and (*b*) special evaluation of the coefficients, as was required for the Timoshenko beam element with equal interpolation. The Newton–Cotes and Gauss–Legendre integration rules have been discussed. The integration rules require transformation of the integral expressions from the global coordinate system to a local coordinate system and interpolation of the global coordinate x. Depending on the relative degrees of interpolation of the geometry and the dependent variables, the formulations are classified as subparametric, isoparametric, and superparametric.

The three logical units—preprocessor, processor, and postprocessor—have been discussed. The contents of the processor, where most finite element calculations are carried out, have been considered in detail. Computer implementation for numerical evaluation of integral expressions, assembly of element coefficients, and imposition of boundary conditions have been discussed. A description of the input variables to the finite element computer program **FEM1D** has been presented, and application of **FEM1D** to problems of heat transfer and solid mechanics has been discussed. Fluid mechanics problems are similar in mathematical structure to heat transfer problems and hence not discussed in this chapter.

PROBLEMS

Section 7.1

In Problems 7.1–7.4, use the appropriate number of integration points, and verify the results with those obtained by the exact integration.

7.1 Evaluate the integrals in Example 7.1.1 using the Newton–Cotes formula and Gauss–Legendre quadrature when ψ_i are the quadratic interpolation functions

$$\psi_1 = \left(1 - \frac{x - x_a}{x_b - x_a}\right)\left(1 - 2\frac{x - x_a}{x_b - x_a}\right) = -\frac{1}{2}\xi(1 - \xi)$$

$$\psi_2 = 4\left(\frac{x - x_a}{x_b - x_a}\right)\left(1 - \frac{x - x_a}{x_b - x_a}\right) = (1 - \xi^2)$$

$$\psi_3 = -\frac{x - x_a}{x_b - x_a}\left(1 - 2\frac{x - x_a}{x_b - x_a}\right) = \frac{1}{2}\xi(1 + \xi)$$

7.2 Use Newton–Cotes integration formula to evaluate

$$K_{11} = \int_{x_a}^{x_b} \left(\frac{d^2\phi_1}{dx^2}\right)^2 dx, \quad G_{11} = \int_{x_a}^{x_b} (\phi_1)^2\, dx$$

where ϕ_i are the Hermite cubic interpolation functions [see Eqs. (5.2.12), (5.2.13*a*), and (5.2.13*b*)]. *Answer:* $r = 2$: $K_{11} = 12/h^3$ (exact), $G_{11} = 0.398148h$.

7.3 Use Gauss–Legendre quadrature to evaluate the integrals of Problem 7.2 for the case in which the interpolation functions ϕ_i are the fifth-order Hermite polynomials of Problem 5.4. *Answer:* $K_{11} = \frac{120}{7}h^3$, $G_{11} = \frac{181}{462}h$ (exact).

7.4 Repeat Problem 7.3 for the case in which the interpolation functions ϕ_i are the fifth-order Hermite polynomials of Problem 5.5. *Answer:* $K_{11} = \frac{5092}{35h_e^3}$, $G_{11} = \frac{523h_e}{3465}$ (exact).

Section 7.2

Computer Exercises (use FEM1D)

7.5 Solve the problem

$$-\frac{d}{dx}\left(k\frac{dT}{dx}\right) = g_0$$

$$\left(-k\frac{dT}{dx}\right)_{x=0} = Q_0, \quad \left[k\frac{dT}{dx} + \beta(T - T_\infty)\right]_{x=L} = 0$$

using two and four linear elements. Compare the results with the exact solution. Use the following data: $L = 0.02$ m, $k = 20$ W/(m $\cdot$°C), $g_0 = 10^6$ W/m^2, $Q_0 = 10^2$ W, $T_\infty = 50$°C, $\beta = 500$ W/(m $\cdot$°C).

7.6 Solve Problem 7.5 using two quadratic elements.

7.7 Solve the heat transfer problem in Example 4.3.3 (Set 1), using (*a*) four linear elements and (*b*) two quadratic elements (see Table 4.3.1).

7.8 Solve the axisymmetric problem in Example 4.3.4 using four quadratic elements and compare the solution with that obtained using eight linear elements and the exact solution of Table 4.3.2.

7.9 Solve the one-dimensional flow problem of Example 4.4.1 (Set 1), for $dP/dx = -24$, using eight linear elements (see Figure 4.4.1). Compare the finite element results with the exact solution $(4.4.20)_1$.

7.10 Solve the Couette flow problem in Example 4.4.1 (Set 2) using four quadratic elements. Compare the finite element solution with the exact solution.

7.11 Solve Problem 4.10 (heat flow in a composite wall) using the minimum number linear finite elements.

7.12 Solve Problem 4.22 (axisymmetric problem of unconfined aquifer) using the minimum number of linear finite elements.

7.13 Solve Problem 4.25.

7.14 Solve Problem 4.27.

7.15 Solve Problem 4.35 using two linear elements.

7.16 Determine the forces and elongations in the wires AB and CD shown in Figure P7.16. Each wire has a cross-sectional area of $A = 0.03$ in.^{2} and modulus of elasticity $E = 30 \times 10^6$ psi.

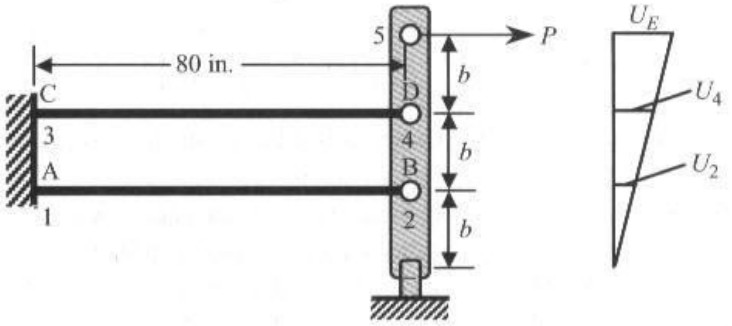


Figure P7.16

7.17 Solve the problem of axisymmetric deformation of a rotating circular disk using four linear elements (see Example 7.3.5).

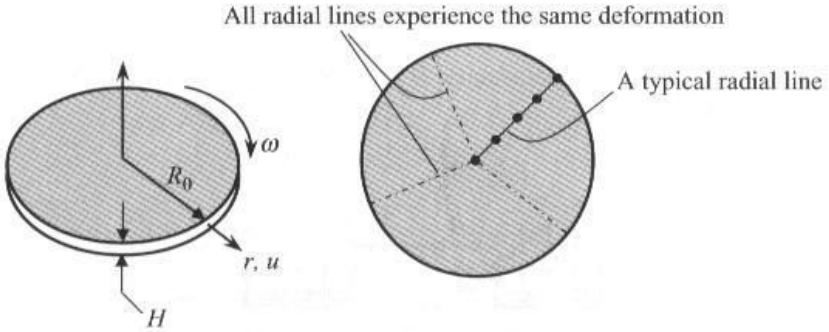


Figure P7.17

7.18–7.25 Solve Problems 5.7–5.14 using the minimum number of Euler–Bernoulli beam elements (*Note:* Numerous other beam problems can be found in books on mechanics of deformable solids).

7.26 Analyze Problem 7.22 (same as Problem 5.11) using the RIE Timoshenko element. Assume $\nu = 0.25$, $K_s = 5/6$, and $H = 0.1$ m (beam height). Use two, four and eight elements to see the convergence characteristics of the RIE (two-element model may yield results very far off from the Euler–Bernoulli beam solution).

7.27 Repeat Problem 7.26 using the CIE Timoshenko element.

7.28 Analyze a clamped circular plate under a uniformly distributed transverse load using the Euler–Bernoulli plate element. Investigate the convergence using two, four, and eight

elements by comparing with the exact solution [from Reddy (2002)]

$$w(r) = \frac{q_0 a^4}{64D}\left[1 - \left(\frac{r}{a}\right)^2\right]^2$$

where $D = EH^3/12(1-\nu^2)$, q_0 is the intensity of the distributed load, a is the radius of the plate, H is its thickness, and ν is Poisson's ratio ($\nu = 0.25$). Tabulate the center deflection.

7.29 Repeat Problem 7.28 with the RIE Timoshenko plate element for $a/H = 10$. Use four and eight linear elements and two and four quadratic elements and tabulate the center deflection. Take $E = 10^7$, $\nu = 0.25$, and $K_s = 5/6$. The exact solution is [see page 403 of Reddy (2002)]

$$w(r) = \frac{q_0 a^4}{64D}\left[1 - \left(\frac{r}{a}\right)^2\right]^2 + \frac{q_0 a^2}{4K_s GH}\left[1 - \left(\frac{r}{a}\right)^2\right]$$

7.30 Repeat Problem 7.29 with the Timoshenko plate element (CIE) (and linear elements) for $a/H = 10$.

7.31 Consider an annular plate of outer radius a, and an inner radius b, and thickness H. If the plate is simply supported at the outer edge and subjected to a uniformly distributed load q_0 (see Fig. P7.31), analyze the problem using the Euler–Bernoulli plate element. Compare the four-element solution with the analytical solution [from Reddy (2002)]

$$w = \frac{q_0 a^4}{64D}\left\{-1 + \left(\frac{r}{a}\right)^4 + \frac{2\alpha_1}{1+\nu}\left[1 - \left(\frac{r}{a}\right)^2\right] - \frac{4\alpha_2\beta^2}{1-\nu}\log\left(\frac{r}{a}\right)\right\}$$

$$\alpha_1 = (3+\nu)(1-\beta^2) - 4(1+\nu)\beta^2\kappa, \quad \alpha_2 = (3+\nu) + 4(1+\nu)\kappa$$

$$\kappa = \frac{\beta^2}{1-\beta^2}\log\beta, \quad \beta = \frac{b}{a}, \quad D = \frac{EH^3}{12(1-\nu^2)}$$

where E is the modulus of elasticity, H the thickness, and ν Poisson's ratio. Take $E = 10^7$, $\nu = 0.3$, and $b/a = 0.25$.

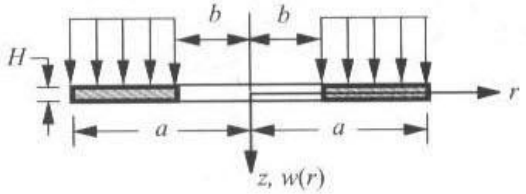


Figure P7.31

7.32 Repeat Problem 7.31 with (a) four linear elements and (b) two quadratic Timoshenko (RIE) elements for $a/H = 10$.

7.33–7.47 Analyze the truss problems in Figures P4.38–P4.44 and frame problems in Figures P5.28–P5.35.

7.48 Consider the axial motion of an elastic bar, governed by the second-order equation

$$EA\frac{\partial^2 u}{\partial x^2} = \rho A\frac{\partial^2 u}{\partial t^2} \quad \text{for } 0 < x < L$$

with the following data: length of bar $L = 500$ mm, cross-sectional area $A = 1$ mm^2, modulus of elasticity $E = 20,000$ N/mm^2, density $\rho = 0.008$ kg/mm^3, boundary conditions

$$u(0, t) = 0, \quad EA\frac{\partial u}{\partial x}(L, t) = 1$$

and zero initial conditions. Using 20 linear elements and $\Delta t = 0.002$ s, determine the axial displacement and plot the displacement as a function of position along the bar for $t = 0.8$ s.

7.49 Consider the following nondimensionalized differential equation governing the plane wall transient:

$$-\frac{\partial^2 T}{\partial x^2} + \frac{\partial T}{\partial t} = 0 \quad \text{for } 0 < x < 1$$

with boundary conditions $T(0, t) = 1$ and $T(1, t) = 0$, and initial condition $T(x, 0) = 0$. Solve the problem using eight linear elements. Determine the critical time step; solve the problem using the Crank–Nicholson method and $\Delta t = 0.002$ s.

Note: Modify program **FEM1D** to solve Problems 7.50–7.52 (solutions to these problems are not presented here for obvious reasons).

7.50 Consider a simply supported beam of length L subjected to a point load

$$P(t) = \begin{cases} P_0 \sin \frac{\pi t}{\tau} & \text{for } 0 \le t \le \tau \\ 0 & \text{for } t \ge \tau \end{cases}$$

at a distance c from the left end of the beam (assumed to be at rest at $t = 0$). The transverse deflection $w(x, t)$ is given by [see Harris and Crede (1961), pp. 8–53]

$$w(x, t) =$$

$$\begin{cases} \dfrac{2P_0L^3}{\pi^4 EI} \displaystyle\sum_{i=1}^{\infty} \frac{1}{i^4} \sin \frac{i\pi c}{L} \sin \frac{i\pi x}{L} \left[\frac{1}{1 - T_i^2/4\tau^2} \left(\sin \frac{\pi t}{\tau} - \frac{T_i}{2\tau} \sin \omega_i t \right) \right], & 0 \le t \le \omega\tau \\ \dfrac{2P_0L^3}{\pi^4 EI} \displaystyle\sum_{i=1}^{\infty} \frac{1}{i^4} \sin \frac{i\pi c}{L} \sin \frac{i\pi x}{L} \left[\frac{\frac{T_i}{\tau} \cos \frac{\pi\tau}{T_i}}{T_i^2/4\tau^2 - 1} \sin \omega_i \left(t - \frac{1}{2}\tau\right) \right], & t \ge \tau \end{cases}$$

where

$$T_i = \frac{2\pi}{\omega_i} = \frac{2L^2}{i^2\pi}\sqrt{\frac{A\rho}{EI}} = \frac{T_1}{i^2}$$

Use the data $P_0 = 1000$ lb, $\tau = 20 \times 10^{-6}$ s, $L = 30$ in., $E = 30 \times 10^6$ lb/in.^{2}, $\rho = 733 \times 10^{-6}$ lb/in.^{3}, $\Delta t = 10^{-6}$ s, and assume that the beam is of square cross section of 0.5 in by 0.5 in. Using five Euler–Bernoulli beam elements in the half-beam, obtain the finite element solution and compare with the series solution at midspan for the case $c = \frac{L}{2}$.

7.51 Repeat Problem 7.50 for $c = \frac{1}{4}L$ and eight elements in the full span.

7.52 Repeat Problem 7.50 for $P(t) = P_0$ at midspan and eight elements in the full span.

7.53 Consider a cantilevered beam with a point load P_0 at the free end (Fig. P7.53). Using the data of Problem 7.50, find the finite element solution for the transverse deflection using eight Euler–Bernoulli beam elements.

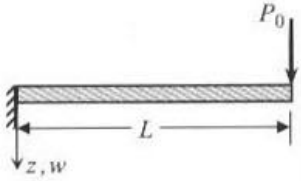


Figure P7.53

$E = 30 \times 10^6$ psi, $\nu = 0.3$,
$\rho = 733 \times 10^{-6}$ lb/in.^{3},
$L = 30$ in., $P_0 = 10^3$ lbs,
$A = 0.25$ in.^{2}, $\Delta t = 10^{-6}$ s

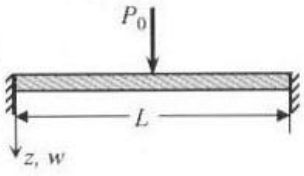


Figure P7.54

7.54 Repeat Problem 7.50 for a clamped beam with the load at the midspan (see Fig. P7.54).

7.55 Repeat Problem 7.54 using four linear Timoshenko beam elements. Use $\nu = 0.3$.

7.56 Repeat Problem 7.54 using two quadratic Timoshenko beam elements. Use $\nu = 0.3$.

REFERENCES FOR ADDITIONAL READING

Fluid mechanics

1. Bird, R. B., Stewart, W. E., and Lightfoot, E. N., *Transport Phenomena,* John Wiley, New York, 1960.
2. Duncan, W. J., Thom, A. S., and Young, A. D., *Mechanics of Fluids,* 2nd ed., Elsevier, New York, 1970.
3. Eskinazi, S., *Principles of Fluid Mechanics,* Allyn and Bacon, Boston, 1962.
4. Harr, M. E., *Groundwater and Seepage,* McGraw-Hill, New York, 1962.
5. Nadai, A., *Theory of Flow and Fracture of Solids,* vol. II, McGraw-Hill, New York, 1963.
6. Schlichting, H., *Boundary-Layer Theory* (translated by J. Kestin), 7th ed., McGraw-Hill, New York, 1979.
7. Shames, I. H., *Mechanics of Fluids,* 3rd ed., McGraw-Hill, New York, 1982.
8. Vallentine, H. R., *Applied Hydrodynamics,* Plenum Press, New York, 1967.
9. Verrujit, A., *Theory of Groundwater Flow,* Gordon and Breach, New York, 1970.

Heat transfer

10. Carslaw, H. S. and Jaeger, J. C., *Conduction of Heat in Solids,* 2nd ed., Oxford University Press, Oxford, 1986.
11. Holoman, J. P., *Heat Transfer,* 8th ed., McGraw-Hill, New York, 1996.
12. Kreith, F. and Bahn, M. S., *Principles of Heat Transfer,* 5th ed., Harper & Row, New York, 1993.
13. Myers, G. G., *Analytical Methods in Conduction Heat Transfer,* 2nd ed., AMCHT Publications, Madison, WI, 1998.
14. Ozisik, M. N., *Heat Transfer: A Basic Approach,* McGraw-Hill, New York, 1985.

Elasticity

15. Budynas, R. G., *Advanced Strength and Applied Stress Analysis,* McGraw-Hill, New York, 1977.
16. Harris, C. M. and Crede, C. E., *Shock and Vibration Handbook,* vol. 1, McGraw-Hill, New York, 1961.
17. Reddy, J. N., *Energy Principles and Variational Methods in Applied Mechanics,* John Wiley, New York, 2002.
18. Ugural, A. C. and Fenster, S. K., *Advanced Strength and Applied Elasticity,* 4th ed., Prentice Hall, Upper Saddle River, NJ, 2003.

Time approximations

19. Bathe, K. J., *Finite Element Procedures,* Prentice Hall, Englewood Cliffs, NJ, 1996.
20. Clough, R. W. and Penzien, J., *Dynamics of Structures,* 2nd ed., McGraw-Hill, New York, 1993.

21. Hughes, T. J. R., *The Finite Element Method,* Prentice Hall, Englewood Cliffs, NJ, 1987; Dover, New York, 2000.

Numerical integration

22. Carnahan, B., Luther, H. A., and Wilkes, J. O., *Applied Numerical Methods,* John Wiley, New York, 1969.
23. Loxan, A. N., Davids, N. and Levenson, A., "Table of the Zeros of the Legendre Polynomials of Order 1–16 and the Weight Coefficients for Gauss' Quadrature Formula," *Bulletin of the American Mathematical Society,* 48, pp. 739–743, 1942.
24. Stroud, A. H. and Secrest, D., *Gaussian Quadrature Formulas,* Prentice-Hall, Englewood Cliffs, NJ, 1966.
25. Zienkiewicz, O. C. and Taylor, R. L., *The Finite Element Method,* vols. 1 and 2, McGraw-Hill, 1989 and 1991.