References

1. R. Jastrow, *Physical Review* **98**, 1479(1955)

2. *Geophysical Data Analysis: Discrete Inverse Theory*, William Menke, Academic Press, Inc.; (1989) ISBN 0-12-490921-3

Chapter 7

Finite Element Methods

In Chapter 3 we treated the case of the solution of a homogeneous, second order differential equation in three dimensions. We will approach some problems in modern physics involving solutions to such equations in Chapter 10. The boundary conditions for that case are indeed very simple and allow for an easy solution. There are a large number of problems for which the boundary conditions are more complicated, the equations are not homogeneous or their expansion in a complete set of functions is not convenient. For such problems one needs to look for other methods. Indeed there exist a number of techniques for solving two-dimensional problems with more-or-less arbitrary boundary conditions. Recently more attention has been given to the representation of such systems in terms of matrices. Such a development has the advantage that all points in the finite presentation of the differential equation are are available at once and hence the boundary conditions are easily specified. Of course the solution of the matrix system will take more computer resources than the point-to-point iteration, but this is an example of extending the techniques in computation to take advantage of greater computing power to solve more ambitious problems.

Finite element methods seek primarily to establish the algorithms for the solutions of boundary value problems in more than one dimension. Arbitrary meshes can be used to represent realistic shapes of practical pieces of equipment in engineering applications. These same methods, originally developed for such problems have since been used in a number of disciplines including the solution of the 3-body problem in quantum mechanics.

They proceed from general physical principles and convert what are essentially differential problems to those of linear algebra. The physics is often closely related to the equations established especially in the energy, or variational, formulation.

The designation "finite element" was first used by Clough in 1960 but the actual techniques were used earlier, in particular, Arggris (1954) and Turner (1956) used them to design aircraft. The fundamental reason for the success of the method goes back to the Rayleigh-Ritz principle and is illustrated in the variational formulation.

The method of weighted residuals, due to the work of Galerkin in the early part of the 20th century is more commonly used today, being perhaps the most direct way to determine the equations.

While there are several ways to derive the necessary equations, most of them have a common thread, the expansion of the unknown function in piecewise continuous functions. These functions are non-zero only over some finite region of space and represent the function in this small area. Thus, before we can begin the derivation of the equations we need to define and discuss a simple set of these functions.

7.1 Basis Functions – One Dimension

Since the unknown functions will be approximated in terms of piecewise continuous basis functions, we must first discuss some examples of these functions. We shall treat only the linear functions here and start in this section with functions in one dimension. Two-dimensional functions are treated in a later section. These functions are of two types called *nodal* basis functions and *element* basis functions.

Each nodal basis function has value unity at the nominal point of the function, zero at the points on either side and outside that region and is linear between the central index point and the ones on either side. These functions are shown in Figure 7.1.

For a range of x between x_0 and x_n the functions, $\phi_i(x)$ are given by:

$$\phi_i(x) = \frac{x - x_{i-1}}{x_i - x_{i-1}} = N_2^{i-1}(x) \quad \text{if} \quad x_i \geq x \geq x_{i-1} \tag{7.1}$$

$$\phi_i(x) = \frac{x_{i+1} - x}{x_{i+1} - x_i} = N_1^i(x) \quad \text{if} \quad x_{i+1} \geq x \geq x_i \tag{7.2}$$

$$\phi_i(x) = 0 \quad \text{otherwise.} \tag{7.3}$$

Note that these functions are closely related to the range of x. If $\phi_0(x)$ might seem to push the definition outside of the range of definition of the nodal points, it would only happen for Eq. 7.1. However that equation is only employed for $x < x_0$ which is outside the range of x. Similar considerations hold for $\phi_n(x)$.

Of some interest are the integrals of the basis functions and their derivatives.

$$\int \phi_i^2(x)dx = \frac{x_{i+1} - x_{i-1}}{3} \tag{7.4}$$

$$\int \phi_i(x)\phi_{i+1}(x)dx = \frac{x_{i+1} - x_i}{6} \tag{7.5}$$

Of course

$$\int \phi_i(x)\phi_{i+2}(x) = \int \phi_i(x)\phi_{i+3}(x) = \ldots = 0 \tag{7.6}$$

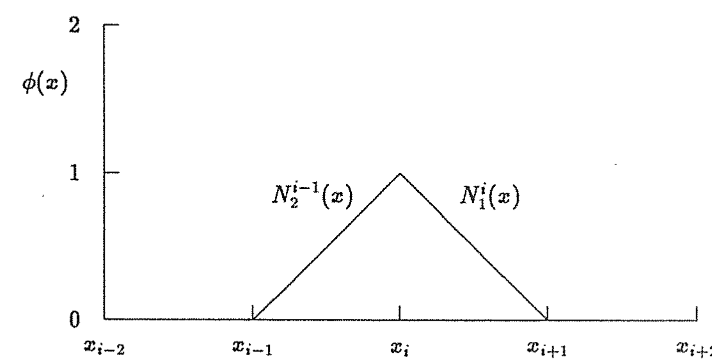

Figure 7.1: The Linear Basis Function $\phi(x)$ and the Two Element Functions. Although the figure is drawn for the case of equal intervals, they need not be (and usually are not).

since they have no overlap. We also have

$$c_i \equiv \int \phi_i(x)dx \tag{7.7}$$

given by

$$c_0 = \frac{x_1 - x_0}{2} \quad \text{and} \quad c_n = \frac{x_n - x_{n-1}}{2} \tag{7.8}$$

and

$$c_i = \frac{x_{i+1} - x_{i-1}}{2} \quad ; \quad i = 1, 2, \ldots, n-1 \tag{7.9}$$

Also of use are the derivatives of the basis functions.

$$\frac{d\phi_i(x)}{dx} = \frac{1}{x_i - x_{i-1}} \quad x_i \geq x \geq x_{i-1} \tag{7.10}$$

$$\frac{d\phi_i(x)}{dx} = -\frac{1}{x_{i+1} - x_i} \quad x_{i+1} \geq x \geq x_i. \tag{7.11}$$

$$\frac{d\phi_i(x)}{dx} = 0 \quad \text{elsewhere} \tag{7.12}$$

The integrals of the products of derivatives are of special interest. Defining

$$b_{ij} \equiv \int \left[\frac{d\phi_i(x)}{dx}\right] \left[\frac{d\phi_j(x)}{dx}\right] dx, \tag{7.13}$$

we have

$$b_{ii} = \frac{1}{x_i - x_{i-1}} + \frac{1}{x_{i+1} - x_i} \tag{7.14}$$

unless i is on the edge of the interval, in which case

$$b_{00} = \frac{1}{x_1 - x_0} \text{ and } b_{nn} = \frac{1}{x_n - x_{n-1}}. \tag{7.15}$$

We also have

$$b_{i,i+1} = -\frac{1}{x_{i+1} - x_i}. \tag{7.16}$$

Of course if j and i differ by more than one, $b_{ij} \equiv 0$.

While the power of the method is its ability to treat arbitrary spacing of nodes, it is useful to reduce to evenly spaced points to make contact with the more familiar methods. For equal spacing, h, of the nodes

$$c_0 = c_n = \frac{h}{2} \text{ and } c_i = h \quad i \neq 0 \text{ or } n \tag{7.17}$$

and

$$b_{00} = b_{nn} = \frac{1}{h}; \quad b_{ii} = \frac{2}{h}; \quad b_{i,i+1} = -\frac{1}{h} \tag{7.18}$$

We now wish to express an unknown function, say $F(x)$, as a series of these basis functions within the range of x from x_0 to x_n.

$$F(x) = \sum_{i=0}^{n} a_i \phi_i(x) \tag{7.19}$$

Note that, since the $\phi_i(x)$ are unity at x_i, the a_i are the values of the function at the nodes.

7.2 Establishing the System Matrix

We now wish to create a system of equations which provide a finite approximation to the differential equation (and include the boundary conditions) to be solved by the methods of linear algebra. We will often write this system of equations in matrix form. This matrix is called the system matrix and the ability to construct it is equivalent (formally) to solving the problem. Note that in defining the basis functions we have left the specification of the positions of the nodal points completely arbitrary.

There are three common methods used for establishing these systems, although the first is included for pedantic reasons.

7.2.1 Model Problem

To illustrate the three methods we will consider the problem of heat conduction in a wire. We suppose that the wire extends along the x-axis with a flow of heat at $x = 0$ such that the derivative of the temperature is given by

$$K\frac{dT}{dx} = q \text{ at } x = 0. \tag{7.20}$$

At the point $x = L$ the wire enters a heat bath with the temperature fixed at T_L. If we suppose that we might add additional heat along the length of the wire with a heating current which generates heat according to the function $Q(x)$ then the equation to be solved, with the boundary conditions given above, is

$$-K\frac{d^2T}{dx^2} = Q(x) \tag{7.21}$$

where the constant, K, is the heat conductivity of the wire.

7.2.2 The "Classical" Procedure

This method is given to make connection with the previous techniques discussed in Chapter 3. For this reason we will treat equal intervals only, that is, with equally spaced nodes at $x_i = ih$. In this method we do not use the basis functions defined earlier. Writing out Eq. 7.21 for each nodal point using the 3-point formula for the second derivative given at the beginning of Chapter 3 (and multiplying by h^2) we find:

$$\begin{array}{cccccccccccc}
a_{-1} & -2a_0 & +a_1 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & = & -\frac{Q_0}{K}h^2 \\
0 & a_0 & -2a_1 & +a_2 & 0 & \dots & 0 & 0 & 0 & 0 & = & -\frac{Q_1}{K}h^2 \\
0 & & a_1 & -2a_2 & +a_3 & \dots & 0 & 0 & 0 & 0 & = & -\frac{Q_2}{K}h^2 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & = & \vdots \\
0 & 0 & 0 & 0 & 0 & \dots & a_{n-3} & -2a_{n-2} & +a_{n-1} & 0 & = & -\frac{Q_{n-2}}{K}h^2 \\
0 & 0 & 0 & 0 & 0 & \dots & 0 & a_{n-2} & -2a_{n-1} & +a_n & = & -\frac{Q_{n-1}}{K}h^2
\end{array} \tag{7.22}$$

The appearance of a_{-1} seems inconvenient but if we use the formula for the first derivative to express the boundary condition at $x = 0$

$$\frac{a_1 - a_{-1}}{2h} = \frac{q}{K}, \tag{7.23}$$

we can eliminate it. We also know that

$$T(x = L) = T_L = a_n \tag{7.24}$$

211 GL-M05 Wed. 10:30 am-1:00 pm

TA: Andrea Gallegos

Date:

Student	Signature	HW	HW Grade
Appelzoller, Jennifer			
Buchanan, Erica			
Bustamante, Alejandro			
Covington, Michael			
Dejnozka, Matthew			
Galaviz-Quintana, Oscar			
Grafe, Olivia			
Graham, Landyn			
Jones, William			
Kellar, Layne			
Lopez, Liliana			
Medina, Mario			
Robert, Calandria			
Sanchez, Rene			
Santillano, Pilar			
Torres, Leticia			

so we can write (transposing the n^{th} column to the right hand side of the set of equations and eliminating the -1 column)

$$\begin{array}{cccccccccc}
-2a_0 & +2a_1 & 0 & 0 & \dots & 0 & 0 & 0 & = & -\frac{Q_0}{K}h^2 + 2h\frac{q}{K} \\
a_0 & -2a_1 & +a_2 & 0 & \dots & 0 & 0 & 0 & = & -\frac{Q_1}{K}h^2 \\
0 & a_1 & -2a_2 & +a_3 & \dots & 0 & 0 & 0 & = & -\frac{Q_2}{K}h^2 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & = & \vdots \\
0 & 0 & 0 & 0 & \dots & +a_{n-3} & -2a_{n-2} & +a_{n-1} & = & -\frac{Q_{n-2}}{K}h^2 \\
0 & 0 & 0 & 0 & \dots & 0 & +a_{n-2} & -2a_{n-1} & = & -\frac{Q_{n-1}}{K}h^2 - T_L
\end{array} \tag{7.25}$$

On the right hand side we can identify three types of terms. There are those proportional to h^2 which arise from the second derivative (the original differential equation itself), those proportional to h which come from the boundary condition on the first derivative and those independent of h which are due to a boundary condition on the value of the function.

7.2.3 The Galerkin Method

In this case we can think of the original equation

$$-K\frac{d^2T(x)}{dx^2} = Q(x) \tag{7.26}$$

as a condition on a residual function defined as:

$$R(x) = -K\frac{d^2T(x)}{dx^2} - Q(x). \tag{7.27}$$

It is clear that we wish to make

$$R(x) \equiv 0.$$

If we could satisfy that condition fully we would have an exact solution for all x. We will content ourselves with the conditions that the projection of $R(x)$ on each of a set of test functions is zero. For the Galerkin method the test functions are taken to be identical to the basis functions that are used for the expansion. Thus we wish to force the integrals of the basis functions multiplied by $R(x)$ to vanish for all nodal indices.

$$-K\int_0^L \phi_i(x)\frac{d^2T}{dx^2}dx = \int_0^L \phi_i(x)Q(x)dx, \quad i = 0,\ 1,\ 2,\ \dots \tag{7.28}$$

If we try to represent $T(x)$ directly in Eq. 7.28 in terms of the basis functions, the second derivative will not be finite for the piecewise continuous linear functions we have used. Hence we first transform the equation (since at this point we have not yet

committed ourselves to the basis functions) by integrating by parts to find the set of equations

$$\int_0^L \frac{d\phi_i(x)}{dx}\frac{dT}{dx}dx - \left[\phi_i(x)\frac{dT}{dx}\right]_0^L = \frac{1}{K}\int_0^L \phi_i(x)Q(x)dx. \tag{7.29}$$

Using the expansion of $T(x)$ in nodal basis functions,

$$T(x) = \sum_{j=0}^{n} a_j\phi_j(x), \tag{7.30}$$

and remembering that

$$K\frac{dT}{dx} = q \tag{7.31}$$

at $x = 0$, we can write

$$\sum_{j=0}^{n} b_{ij}a_j - \left[\frac{dT}{dx}\right]_L \delta_{in} + \frac{q}{K}\delta_{i0} = \frac{1}{K}\int_0^L \phi_i(x)Q(x)dx \tag{7.32}$$

where we have differentiated the expansion term by term and the b_{ij} are given by Eqs. 7.14, 7.15 and 7.16. For $Q(x)$ =constant we can write out the equations.

$$\begin{array}{cccccccccc}
b_{00}a_0 & +b_{01}a_1 & 0 & 0 & \dots & 0 & 0 & 0 & = & \frac{Q}{K}c_0 - \frac{q}{K} \\
b_{10}a_0 & +b_{11}a_1 & +b_{12}a_2 & 0 & \dots & 0 & 0 & 0 & = & \frac{Q}{K}c_1 \\
 & b_{21}a_1 & +b_{22}a_2 & +b_{23}a_3 & \dots & 0 & 0 & 0 & = & \frac{Q}{K}c_2 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & = & \vdots \\
0 & 0 & 0 & 0 & \dots & b_{n-2,n-2}a_{n-2} & +b_{n-2,n-1}a_{n-1} & 0 & = & \frac{Q}{K}c_{n-2} \\
0 & 0 & 0 & 0 & \dots & b_{n-1,n-2}a_{n-2} & +b_{n-1,n-1}a_{n-1} & +b_{n-1,n}a_n & = & \frac{Q}{K}c_{n-1} \\
0 & 0 & 0 & 0 & \dots & 0 & b_{n,n-1}a_{n-1} & +b_{n,n}a_n & = & \frac{Q}{K}c_n + \left[\frac{dT}{dx}\right]_L
\end{array} \tag{7.33}$$

with the c_i given by Eqs. 7.8 and 7.9. We note three things:

i) The matrix (corresponding to the b_{ij} on the left hand side of Eq. 7.33) is singular so we cannot solve this system of equations as it stands,

ii) we know the last value of the function, i.e. $a_n = T_L$, and

iii) we don't know the value of $\left[\frac{dT}{dx}\right]_L$, so the last equation is useless.

Hence we reduce the unknowns by one ($a_n = T_L$), reduce the equations by one (we will drop the last equation) and the system is now non-singular.

$$\begin{array}{lllllllll}
b_{00}a_0 & +b_{01}a_1 & 0 & 0 & \dots & 0 & 0 & = & \frac{Q}{K}c_0 - \frac{q}{K} \\
b_{10}a_0 & +b_{11}a_1 & +b_{12}a_2 & 0 & \dots & 0 & 0 & = & \frac{Q}{K}c_1 \\
 & b_{21}a_1 & +b_{22}a_2 & +b_{23}a_3 & \dots & 0 & 0 & = & \frac{Q}{K}c_2 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & = & \vdots \\
0 & 0 & 0 & 0 & \dots & b_{n-2,n-2}a_{n-2} & +b_{n-2,n-1}a_{n-1} & = & \frac{Q}{K}c_{n-2} \\
0 & 0 & 0 & 0 & \dots & b_{n-1,n-2}a_{n-2} & +b_{n-1,n-1}a_{n-1} & = & \frac{Q}{K}c_{n-1} - b_{n-1,n}T_L
\end{array} \tag{7.34}$$

We now have n equations in n unknowns $(a_0, a_1, \ldots, a_{n-1})$.
For the equal spacing limit we recover

$$\begin{array}{cccccccccl}
a_0 & -a_1 & 0 & 0 & \ldots & 0 & 0 & 0 & = & \frac{Q}{2K}h^2 - h\frac{q}{K} \\
-a_0 & +2a_1 & -a_2 & 0 & \ldots & 0 & 0 & 0 & = & \frac{Q}{K}h^2 \\
 & -a_1 & +2a_2 & -a_3 & \ldots & 0 & 0 & 0 & = & \frac{Q}{K}h^2 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & = & \vdots & \\
0 & 0 & 0 & 0 & \ldots & -a_{n-3} & 2a_{n-2} & -a_{n-1} & = & \frac{Q}{K}h^2 \\
0 & 0 & 0 & 0 & \ldots & 0 & -a_{n-2} & +2a_{n-1} & = & \frac{Q}{2K}h^2 + T_L
\end{array} \tag{7.35}$$

which is nearly the same as Eq. 7.25, derived in the last section. The only difference is an overall factor of -1. If Q were not constant we would replace Qc_i by

$$\int \phi_i(x)Q(x) \tag{7.36}$$

in this case, whereas the direct evaluation would always have used the values Q_i.

7.2.4 The Variational Method

The variational principle has turned out to be extremely powerful in solving computational problems. We shall see in Chapter 11 that it allows us to transform several-body problems into a solvable form. In general it allows to convert differential problems to an integral form. There are two techniques for this transformation; the Rayleigh-Ritz expression that was introduced in Chapter 5 and the use of the Euler-Lagrange equations. It is this latter method which is used for deriving system matrices for finite-element solutions.

In this method we construct a function

$$F(\dot{T}, T, x) \equiv -\frac{K}{2}\dot{T}^2 + Q(x)T(x) \tag{7.37}$$

with $\dot{T} \equiv \frac{dT(x)}{dx}$ and minimize the integral

$$I = \int_{x_0}^{x_n} F(\dot{T}, T, x)dx \tag{7.38}$$

by using the Euler-Lagrange equations. Since

$$\frac{\partial F}{\partial \dot{T}} = -K\frac{dT}{dx}; \quad \frac{\partial}{\partial x}\frac{\partial F}{\partial \dot{T}} = -K\ddot{T}; \quad \frac{\partial F}{\partial T(x)} = Q(x)$$

we have

$$\frac{\partial}{\partial x}\frac{\partial F}{\partial \dot{T}} - \frac{\partial F}{\partial T} = 0 = -K\ddot{T} - Q(x) \tag{7.39}$$

which is equivalent to the original equation.

Thus we have recast the problem to that of minimizing the integral of Eq. 7.38. Using the expansion of the temperature in terms of the basis functions we can write the integral as

$$I = -\frac{K}{2}\sum_{k,j=0}^{n} b_{k,j}a_k a_j + \sum_{k=0}^{n} a_k \int Q(x)\phi_k(x)dx \tag{7.40}$$

Differentiating with respect to a_i, and using the fact that $b_{ij} = b_{ji}$, we have the set of equations

$$b_{ii}a_i + \sum_{j=0, j\neq i}^{n} b_{ij}a_j = \frac{1}{K}\int Q(x)\phi_i(x)dx \tag{7.41}$$

However these equations are incomplete in terms of the boundary conditions. If we wish to specify the *value* of the function on the boundary there is no problem. For example in the present model problem we can set $a_n = T_L$ as in the previous two examples, thus, removing one unknown, and eliminate the last equation. If we wish to specify the *derivative* on the boundary, however, we must modify the function to be minimized.

In the present case the function needed is:

$$I = \frac{q}{K}T(x) + \int_{x_0}^{x_n} F(\dot{T}, T, x)dx \tag{7.42}$$

which will lead to the same result as the Galerkin method found in the previous section.

7.3 Example One-dimensional Program

Let us return to the model problem introduced at the beginning of this chapter. For $Q(x)$ constant and with units such that $K \equiv 1$ we have:

$$\frac{d^2T}{dx^2} = -Q \tag{7.43}$$

and the exact solution is

$$T(x) = -\frac{1}{2}Qx^2 + \alpha x + \beta \tag{7.44}$$

where α and β are integration constants. Since $\frac{dT}{dx}$ at $x = 0$ is q, $\alpha = q$. Also we have $T(L) = T_L = -\frac{1}{2}Q * L^2 + qL + \beta$ so that the full solution is

$$T(x) = T_L + q(x - L) + \frac{1}{2}Q(L^2 - x^2). \tag{7.45}$$

Below is a simple program which sets up the system matrix and solves it for the conditions that $T_L = 0.5$, $q = 1.0$ and $Q(x) = 0.8$. This code implements Eqs. 7.34.

```
      dimension b(0:100,0:100),c(0:100),x(0:100),s(0:100,0:100)
      n=8          ! use 8 points
      al=1.        ! L=1
      h=al/n
      tl=0.5       ! T(L=1)=0.5
      sq=-1.0       ! small q
      bq=0.8       ! Q
      n1=n-1
      do 1 i=0,n  ! establish a uniform mesh
      x(i)=i*h
    1 continue
      c(0)=.5*(x(1)-x(0))  ! calculate the integrals
      c(n)=.5*(x(n)-x(n1)) ! of the basis functions
      b(0,1)=-1./(x(1)-x(0))  ! and their derivatives
      b(1,0)=b(0,1)
      b(0,0)=1./(x(1)-x(0))
      b(n,n)=1./(x(n)-x(n1))
      b(n,n1)=-1./(x(n)-x(n1))
      do 2 i=1,n1
      c(i)=.5*(x(i+1)-x(i-1))
      b(i,i+1)=-1./(x(i+1)-x(i))
      b(i,i-1)=-1./(x(i)-x(i-1))
      b(i,i)=1./(x(i)-x(i-1))+1./(x(i+1)-x(i))
    2 continue
      do i=0,n1   ! set up system matrix
      do j=0,n1
      s(i,j)=b(i,j)
      enddo
      enddo
      do 5 i=0,n1        ! set up last column in the
    5 s(i,n)=c(i)*bq     ! augmented matrix
      s(0,n)=s(0,n)-sq
      s(n1,n)=s(n1,n)-b(n1,n)*tl
      do 3 i=0,n1        ! print out the system matrix
      print 4,i,(s(i,j),j=0,n)
    4 format(i3,12f6.2)
    3 continue
      call solve(s,n,101,1)
      do 6 i=0,n1
      xx=x(i)
      print 7,x(i),s(i,n),tl+sq*(xx-al)+.5*bq*(al**2-xx**2)
```



```
    6 continue
    7 format(f6.1,3f10.5)
      end
```

Since the solution is quadratic, the result of the code will be exact.

7.4 Assembly by Elements

It is useful to be able to refer to the parts of the spatial region by elements (the space between the points) rather than the nodes themselves. To this end we define two functions, the element functions (given by Eqs. 7.2 and 7.1) which only have support within the boundaries of the element $\{x_i, x_{i+1}\}$ specified by e_i. In this case we can write the expansion of the unknown function as:

$$F(x) = \sum_{i=0}^{n} a_i\phi_i(x) = \sum_{i=0}^{n} a_i\left[N_1^{e_i}(x) + N_2^{e_{i-1}}(x)\right] \tag{7.46}$$

$$= \sum_{e=0}^{n}\left[a_e N_1^e(x) + a_{e+1}N_2^e(x)\right] \tag{7.47}$$

We now consider using the *element* basis functions for the test functions (as opposed to the *nodal* basis functions that were used before) in the Galerkin method. Thus, instead of multiplying by $\phi_i(x)$ and integrating, we multiply by $N_1^e(x)$ and $N_2^e(x)$. Note that, since element functions have support only on the region between the nodes, the element labels correspond to an orthogonal basis. The nodal basis functions are not orthogonal, since adjacent nodes overlap, while the element basis functions *are* orthogonal. They have, however, two nodal parameters contributing to each function.

Since there are twice as many element basis functions as nodal basis functions, there are twice as many equations and it is necessary to combine them to obtain the correct number of equations. Consider the equations by pairs.

$$a_e\int \dot{N}_1^e\dot{N}_1^e dx + a_{e+1}\int \dot{N}_1^e\dot{N}_2^e dx = \int N_1^e(x)Q(x)dx \tag{7.48}$$

$$a_e\int \dot{N}_2^e\dot{N}_1^e dx + a_{e+1}\int \dot{N}_2^e\dot{N}_2^e dx = \int N_2^e(x)Q(x)dx \tag{7.49}$$

or, with an obvious definition

$$m_{11}^e a_e + m_{12}^e a_{e+1} = Q_1^e \tag{7.50}$$

$$m_{21}^e a_e + m_{22}^e a_{e+1} = Q_2^e. \tag{7.51}$$

The m_{ij} are given by integrals of the derivatives of the element functions:

$$m_{11} = \int \dot{N}_1^e \dot{N}_1^e dx = \frac{1}{x_{e+1} - x_e}; \quad m_{12} = \int \dot{N}_1^e \dot{N}_2^e dx = -\frac{1}{x_{e+1} - x_e} \tag{7.52}$$

$$m_{21} = \int \dot{N}_2^e \dot{N}_1^e dx = -\frac{1}{x_{e+1} - x_e}; \quad m_{22} = \int \dot{N}_2^e \dot{N}_2^e dx = \frac{1}{x_{e+1} - x_e}. \tag{7.53}$$

Thus the full set of equations are:

$$\begin{array}{llllll}
m_{11}^0 a_0 & +m_{12}^0 a_1 & & & \cdots = & Q_1^0 \\
m_{21}^0 a_0 & +m_{22}^0 a_1 & & & \cdots = & Q_2^0 \\
 & +m_{11}^1 a_1 & +m_{12}^1 a_2 & & \cdots = & Q_1^1 \\
 & +m_{21}^1 a_1 & +m_{22}^1 a_2 & & \cdots = & Q_2^1 \\
 & & +m_{11}^2 a_2 & +m_{12}^2 a_3 & \cdots = & Q_1^2 \\
 & & +m_{21}^2 a_2 & +m_{22}^2 a_3 & \cdots = & Q_2^2 \\
\vdots & \vdots & \vdots & \vdots & \vdots \; \vdots & \vdots
\end{array}$$

Combining equations 2 and 3, 4 and 5, 6 and 7 etc.

$$\begin{array}{llllll}
m_{11}^0 a_0 & +m_{12}^0 a_1 & & & \cdots = & Q_1^0 \\
m_{21}^0 a_0 & +(m_{22}^0 + m_{11}^1) a_1 & +m_{12}^1 a_2 & & \cdots = & Q_2^0 + Q_1^1 \\
 & +m_{21}^1 a_1 & +(m_{22}^1 + m_{11}^2) a_2 & +m_{12}^2 a_3 & \cdots = & Q_2^1 + Q_1^2 \\
 & \vdots & \vdots & \vdots & \vdots \; \vdots & \vdots
\end{array}$$

which can be seen to be equivalent to

$$\begin{array}{llllll}
b_{00} a_0 & +b_{01} a_1 & & & \cdots = & Q_1^0 \\
b_{10} a_0 & +b_{11} a_1 & +b_{12} a_2 & & \cdots = & Q_2^0 + Q_1^1 \\
 & +b_{21} a_1 & +b_{22} a_2 & +b_{23} a_3 & \cdots = & Q_2^1 + Q_1^2 \\
 & \vdots & \vdots & \vdots & \vdots \; \vdots & \vdots
\end{array}$$

Since $Q_2^{i-1} + Q_1^i = \int \phi_i(x) Q(x) dx$ we find again the same result as Eq. 7.34.

7.5 Problems in Two Dimensions

7.5.1 Element Functions

For two dimensions the element basis functions will be defined over triangular elements. We will number the nodes and the elements in an arbitrary, but well defined, manner. In addition to the node and element labels, we associate a local index of

the nodes (L= 0, 1, 2) belonging to a given element. Thus there are two ways to refer to a given physical node, either by the global node index itself or by the pair of indices element and local node index. There is, of course, a correspondence between the nodal labeling scheme and the element-local index labeling which is most readily implemented in a practical program with a doubly dimensioned variable established by the programer to represent (along with the coordinate values of the nodal points) the desired mesh.

$$J_{node} = \mathrm{NN}(e, L); \quad L = 0,\ 1,\ 2 \tag{7.54}$$

Within a given triangle we define three linear functions $N_i(x, y)$, which only have support within the triangle such that

$$N_0(x_0, y_0) = 1; \quad N_0(x_1, y_1) = 0; \quad N_0(x_2, y_2) = 0 \tag{7.55}$$

$$N_1(x_0, y_0) = 0; \quad N_1(x_1, y_1) = 1; \quad N_1(x_2, y_2) = 0 \tag{7.56}$$

$$N_2(x_0, y_0) = 0; \quad N_2(x_1, y_1) = 0; \quad N_2(x_2, y_2) = 1 \tag{7.57}$$

It is easy to show that these functions are given by:

$$N_i(x, y) = \frac{a_i + b_i x + c_i y}{d} \tag{7.58}$$

where

$$a_i = x_j y_k - x_k y_j; \quad b_i = y_j - y_k; \quad c_i = x_k - x_j \tag{7.59}$$

and (i, j, k) are cyclic, i.e.

$$\begin{array}{l}
\text{if } i = 0 \text{ then } j = 1 \text{ and } k = 2 \\
\text{if } i = 1 \text{ then } j = 2 \text{ and } k = 0 \\
\text{if } i = 2 \text{ then } j = 0 \text{ and } k = 1.
\end{array}$$

and

$$d = \begin{vmatrix} 1 & x_0 & y_0 \\ 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \end{vmatrix} = \pm 2 \times \text{the area of the triangle}$$

$$= c_2 b_1 - c_1 b_2. \tag{7.60}$$

Note that

$$\frac{\partial N_i(x, y)}{\partial x} = \frac{b_i}{d} \tag{7.61}$$

$$\frac{\partial N_i(x, y)}{\partial y} = \frac{c_i}{d} \tag{7.62}$$

7.5.2 Poisson's Equation

To solve the equation

$$\frac{\partial^2 T(x,y)}{\partial x^2} + \frac{\partial^2 T(x,y)}{\partial y^2} = 0 \tag{7.63}$$

with fixed and natural boundary conditions we can expand

$$T(x,y) = \sum_{e,L'} f_{J'} N_{e,L'}(x,y) \tag{7.64}$$

where the index J' is to be associated with the element, e, and local index L'. Note that a given $f_{J'}$ can occur more than once in the sum.

Applying the Galerkin method we multiply Eq. 7.63 by each function $N^e_L(x,y)$. Performing the integration by parts necessary to reduce to first derivatives and integrating over the area of the surface we obtain, for a fixed value of e and L:

$$\sum_{L'} m^e_{L,L'} f_{J'} = 0 \tag{7.65}$$

where J' corresponds to the pair (e, L').

$$m^e_{L,L'} = \int_{element} \left[\frac{\partial N^e_L(x,y)}{\partial x}\frac{\partial N^e_{L'}(x,y)}{\partial x} + \frac{\partial N^e_L(x,y)}{\partial y}\frac{\partial N^e_{L'}(x,y)}{\partial y} \right] dxdy \tag{7.66}$$

$$= \frac{b_L b_{L'} + c_L c_{L'}}{2|d|} \tag{7.67}$$

Since we generate equations labeled by e and L, there would be too many and the system would be over determined. Hence we combine all values of e and L which correspond to a given value of J. Thus the system matrix is one indexed by the pair (J, J'). It is "almost diagonal" since the e index is diagonal and the local node index only allow a deviation of three units. Since the labeling of the nodes is arbitrary, however, it takes some practice to make the matrix nearly diagonal in a practical case.

For the specific mesh given in Figure 7.2 the nodes will be defined by an input file similar to the following specifying the (x, y) pairs of each node.

```
 12
0.0  0.0
1.0  0.0
3.0  0.0
1.0  1.0
1.4  1.0
3.0  1.0
1.4  1.8
3.0  1.8
3.6  1.8
1.4  2.6
3.0  2.6
3.6  2.6
```


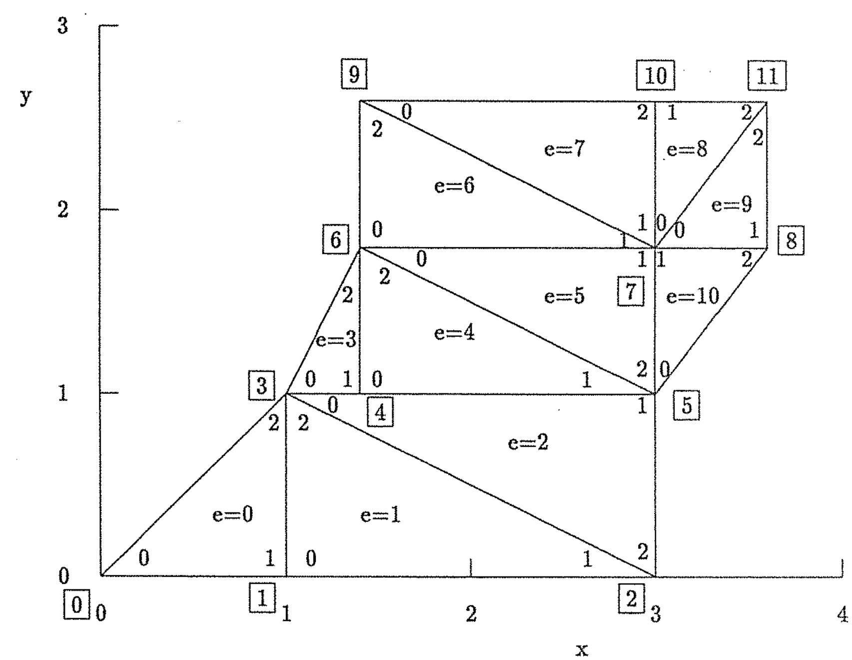

Figure 7.2: Area to be solved in the two-dimensional boundary value problem. The global node numbers are in the boxes.

The key to carrying out the construction of the system matrix is the ability to establish the connection between the global node numbering and the labeling by element and local node index. This is done by a two-dimensional variable (called `NN` in the following code fragments) and constructed by the investigator. The following file gives a possible definition of the 11 elements in Figure 7.2 . Each line corresponds to an element and gives, in order, the node number for local index 0, 1, and 2.

```
11
    0    1    3
    1    2    3
    3    5    2
    3    4    6
    4    5    6
    6    7    5
    6    7    9
    9    7   10
    7   10   11
    7    8   11
    5    7    8
```

Assuming that the value of the function at the boundary is 5 at nodes 0, 1, and 2 and the values are 9, 11 and 12 at nodes 9, 10 and 11 respectively, we can construct a file for the boundary conditions giving the node number and value for each node for which a value is specified:

```
    6
    0    5.
    1    5.
    2    5.
    9    9.
   10   11.
   11   12.
```

To create a program to solve Eq. 7.63 we start by dimensioning some variables.

```
      dimension s(0:40,0:40),b(0:40),c(0:40),nn(0:100,0:2)
     1 ,ib(40),bndy(40),x(0:40),y(0:40),xl(0:2),yl(0:2)
```

Now read in the files defining the nodal values, the correspondence between the global node numbers and the element-local node numbers and then set up the system matrix.

```
      do i=0,n
      do j=0,n
      s(i,j)=0.
      enddo
      enddo
      do 1 ie=0,ne          ! start working around the diagonal of
      do l=0,2              ! the matrix.  Pick the nodes corresponding
```



```
      xl(l)=x(nn(ie,l))     ! to the 3 local nodes
      yl(l)=y(nn(ie,l))
      enddo
      do i=0,2              ! Set up the a, b and c needed to compute
      j=mod(i+1,3)          ! the integrals of the derivatives of the
      k=mod(i+2,3)          ! element functions
      b(i)=yl(j)-yl(k)      ! a is not needed in this case
      c(i)=xl(k)-xl(j)
      enddo
      di=abs(.5/(c(2)*b(1)-c(1)*b(2)))  ! delta inverse
      do lrow=0,2                  ! now build the block around the diagonal
      do lcol=0,2
      ir=nn(ie,lrow)
      ic=nn(ie,lcol)
      s(ir,ic)=s(ir,ic)+di*(b(lrow)*b(lcol)+c(lrow)*c(lcol))
      enddo
      enddo
    1 continue
      do 2 i=1,nb       ! insert boundary conditions
      nt=ib(i)          ! number of the node
      do j=0,n
      s(nt,j)=0.        ! zero row
      enddo
      s(nt,nt)=1.       ! diagonal element 1
      s(nt,n+1)=bndy(i) ! insert boundary value on rhs
    2 continue
      call solve(s,n+1,41,1)
      print 120
  120 format('     Results')
      print 121
  121 format(' node      x       y      value')
      do i=0,n
      print 122,i,x(i),y(i),s(i,n+1)
      enddo
  122 format(i6,3f8.2)
      end
```

The augmented system matrix which results has the following form:

1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	5.00
0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	5.00
0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	5.00
0.00	−1.50	0.00	2.75	−1.00	−0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	−1.00	2.50	−0.25	−1.25	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	−1.00	−0.25	−0.25	2.88	0.00	−1.38	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	−1.25	0.00	2.75	−0.50	0.00	−1.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	−1.38	−0.50	4.58	−1.33	0.00	−1.38	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	−1.33	1.71	0.00	0.00	−0.37	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	9.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	11.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	12.00

(7.68)

Because of the nearly diagonal form of this matrix, the Gauss-Seidel technique works well for its solution.

Problems

Problems marked with * are meant to be solved without the aid of a computer.

1. Set up and run the one-dimensional problem given in section 7.3. Modify the code to solve the system on an unequal mesh. Use 6 elements of size 0.05 and 7 elements of size 0.1 to span the region $x = 0$ to $x = 1$.

2. * Consider the two-dimensional element function discussed in the notes in section 7.6. Assuming that the function $N_0(x, y)$ can be written in the form

$$N_0(x, y) = \alpha + \beta x + \gamma y$$

and

$$N_0(x_0, y_0) = 1,\ N_0(x_1, y_1) = 0,\ N_0(x_2, y_2) = 0$$

find α, β and γ, i.e. derive Eqs. 7.58 and 7.59 for $i = 0$.

3. Create a code to solve the problem given in section 7.5.2. Use the code fragments given in the text and the `SOLVE` routine developed in Chapter 5