Finite Element Solution of the Poisson equation with Dirichlet Boundary Conditions in a rectangular domain

by
Lawrence Agbezuge,
Visiting Associate Professor,
Rochester Institute of Technology, Rochester, NY

Abstract

The basic concepts taught in an introductory course in Finite Element Analysis are utilized to solve a steady state heat conduction problem in a rectangular domain with given Dirichlet boundary conditions. This problem was given to graduate students as a project for the final examination. Although most of the students received extensive help in solving the problem, the exercise involved in solving the problem helped the students to better understand the basic concepts of Finite Element Analysis.
The concepts utilized in solving the problem are (a) weak formulation of the Poisson Equation, (b) creation of a Finite Element Model on the basis of an assumed approximate solution, (c) creation of *4-node* rectangular elements by using interpolation functions of the Lagrange type, (d) assembly of element equations, (e) solution and post-processing of the results.
The method of solution permits *h-mesh* refinement in order to increase the accuracy of the numerical solution. The method of *p-mesh* refinement that requires the use of higher order elements, although it is familiar to the students, is not considered in this paper.
To validate the Finite Element solution of the problem, a Finite Difference solution was obtained and compared with the Finite Element solution.

Introduction

The basic concepts of Finite Element Analysis involve certain basic steps that should be taught in an introductory course. They are the following:

(a) Weak formulation of the governing differential equation, so as to reduce the continuity requirements on the Primary Variable (PV), which is the dependent variable (or variables) of the problem. The weak formulation also defines a Secondary Variable (SV), which is defined by a weighting function in the boundary terms of the weak formulation.
(b) The construction of a Finite Element model for an arbitrary n-node element in terms of interpolation functions. For one dependent variable, the interpolation functions are of the Lagrange type, that is, they could be constructed from Lagrange interpolating polynomials [1,2]. For 2 dependent variables, the interpolation functions could be constructed in terms of Hermite polynomials [3].
(c) The construction of interpolation functions for the type of elements that are used for solving the problem. In this paper, only 4-node rectangular elements are considered.
(d) Assembly of element equations for the problem, in order to ensure that continuity requirements in the Primary Variables, flux terms and source terms are satisfied between elements.

(e) Creation of the condensed equations in order to satisfy Essential Boundary Conditions on the Primary Variable, and the solution of the condensed equations.
(f) Post processing of the results and the computation of derived results.

The concepts listed above will be explained in greater detail in this paper. Because implementation of these concepts requires extensive numerical computations, the Matlab® programming language was utilized.
To validate results of the numerical solution, the Finite Difference solution of the same problem is compared with the Finite Element solution.

Problem Definition

A very simple form of the steady state heat conduction in the rectangular domain shown in Figure 1 may be defined by the Poisson Equation (all material properties are set to unity)

$$\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \qquad (1)$$

for $x = [0, a]$, $y = [0, b]$, with $a = 4$, $b = 2$.

where $u(x, y)$ is the steady state temperature distribution in the domain.
The boundary conditions are

$$u(0, y) = U_L,\ u(a, y) = U_R,\ u(x, 0) = U_B,\ u(x, b) = U_T$$

where $U_L = 100$, $U_R = 250$; imposed temperatures on the left and right boundaries,
$U_B = 50$, $U_T = 200$; imposed temperature on the bottom and top boundaries.

Note that singular points occur at the corners of the domain. Either the higher of two specified values of u may be used at the corner, or the average of the two specified values may be used.
The difference between the two solutions obtained by handling the singular points differently become insignificant as a more refined mesh is used to solve the problem.

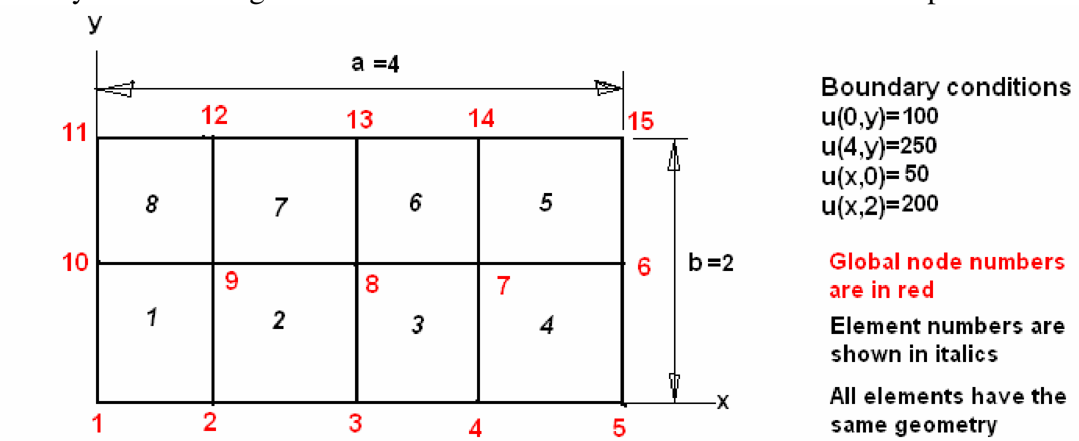


Figure 1

Weak Formulation

The continuity requirements on $u(x, y)$ are relaxed by creating a weak formulation of the governing equation. Multiply equation (1) by an arbitrary weight function,

$w(x, y)$, and integrate over an arbitrary domain, Ω^e, whose boundary is Γ^e. The arbitrary domain could represent an *n-node* element within the solution domain, Ω, with boundary, Γ, as shown in Figure 2.

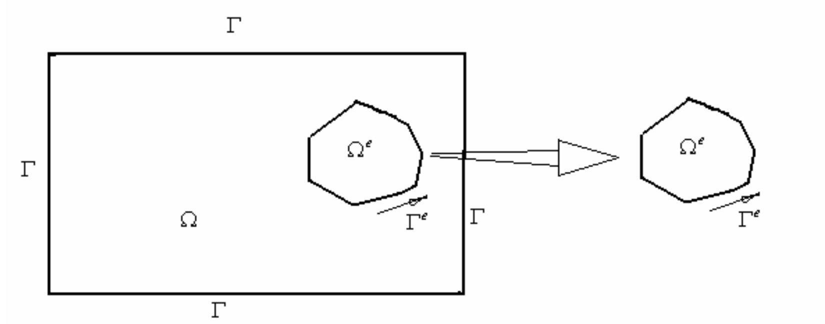


Figure 2

The equation obtained is

$$\int_{\Omega^e} w(x,y)\left[\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right]dxdy = 0 \tag{2}$$

Since $\frac{\partial}{\partial x}\left(w\frac{\partial u}{\partial x}\right)=\frac{\partial w}{\partial x}\frac{\partial u}{\partial x}+w\frac{\partial^2 u}{\partial x^2}$, and $\frac{\partial}{\partial y}\left(w\frac{\partial u}{\partial y}\right)=\frac{\partial w}{\partial y}\frac{\partial u}{\partial y}+w\frac{\partial^2 u}{\partial y^2}$, we obtain

$$\int_{\Omega^e}\left[\frac{\partial}{\partial x}\left(w\frac{\partial u}{\partial x}\right)+\frac{\partial}{\partial y}\left(w\frac{\partial u}{\partial y}\right)-\frac{\partial w}{\partial x}\frac{\partial u}{\partial x}-\frac{\partial w}{\partial y}\frac{\partial u}{\partial y}\right]dxdy = 0 \tag{3}$$

At a boundary with outward normal, $\hat{n}$, curvilinear coordinate, s, and direction cosines, $n_x = \cos(\hat{n},\hat{i})$, $n_y = \cos(\hat{n},\hat{j})$, the Gauss Divergence Theorem may be used to convert the area integral to a line integral, so that

$$\int_{\Omega^e}\frac{\partial}{\partial x}\left(w\frac{\partial u}{\partial x}\right)dxdy = \oint_{\Gamma^e} w\frac{\partial u}{\partial x}n_x ds, \text{ and } \int_{\Omega^e}\frac{\partial}{\partial y}\left(w\frac{\partial u}{\partial y}\right)dxdy = \oint_{\Gamma^e} w\frac{\partial u}{\partial y}n_y ds$$

Define the flux term, q_n, as $q_n \equiv n_x\frac{\partial u}{\partial x}+n_y\frac{\partial u}{\partial y}$, so that equation (3) may be written as

$$\int_{\Omega^e}\left[\frac{\partial w}{\partial x}\frac{\partial u}{\partial x}+\frac{\partial w}{\partial y}\frac{\partial u}{\partial y}\right]dxdy = \oint_{\Gamma^e} wq_n ds \tag{4}$$

Equation (4) represents the weak formulation of equation (1). The continuity requirement on $u(x, y)$, the Primary Variable (PV), is relaxed and shared equally with the weight function $w(x, y)$. The weight function, since it occurs in a boundary term, defines the Secondary Variable (SV) as q_n, the flux term. The PV satisfies Essential Boundary Conditions (EBC) for the given boundary value problem. The SV satisfies Natural Boundary Conditions (NBC) during the weak formulation of the problem. The weight function represents a variation of the PV, and satisfies the homogeneous form of EBCs at the boundary of the domain.

The left hand side of equation (4) is bilinear in (u, w) and linear in w.
More detailed discussion of the weak formulation may be found in standard textbooks on Finite Element Analysis [1,4,5].

Finite Element Model

The assumed solution of equation (4) for an arbitrary, *n-node* element is defined by

$$u^e(x, y) = \sum_{j=1}^{n} u_j^e \psi_j^e(x, y) \tag{5}$$

where u_j^e =nodal value for $u(x, y)$ at node j for the element, and

$\psi_j^e(x, y)$ =interpolation function for $u(x, y)$ at node j within the element.

The interpolation functions must satisfy the conditions

$$\psi_j^e(x_k, y_k) = \delta_{jk} \quad \text{at all nodes } (j,k) = 1,2,...,n \quad \text{and} \quad \sum_{j=1}^{n} \psi_j^e(x, y) = 1 \text{ within } \Omega^e .$$

The *Kronecker Delta* is defined as

$\delta_{jk} = 1$, for $j = k$ and

$\delta_{jk} = 0$ for $j \neq k$.

On substituting equation (5) into equation (4), we obtain

$$\int_{\Omega^e} \left[\frac{\partial w}{\partial x} \sum_{j=1}^{n} u_j^e \frac{\partial \psi_j^e}{\partial x} + \frac{\partial w}{\partial y} \sum_{j=1}^{n} u_j^e \frac{\partial \psi_j^e}{\partial y} \right] dxdy = \oint_{\Gamma^e} w q_n ds \tag{6}$$

Because the weight function represents a variation of the PV, it takes on the nodal values $w_i = \psi_i^e$, $i = 1,2,...,n$. Therefore equation (6), after exchanging the order of integration and summation, yields the matrix equation

$$\sum_{j=1}^{n} K_{ij}^e u_j^e = Q_i^e, \; i = 1,2,...,n \tag{7}$$

where $K_{ij}^e = \int_{\Omega^e} \left[\frac{\partial \psi_i^e}{\partial x} \frac{\partial \psi_j^e}{\partial x} + \frac{\partial \psi_i^e}{\partial y} \frac{\partial \psi_j^e}{\partial y} \right] dxdy$, and $Q_i^e = \oint_{\Gamma^e} \psi_i^e q_n ds$ (8)

If we had a source term, $f(x, y)$, in the governing equation (such as heat generation per unit area), there would have been an area integral of the form $F_i^e = \int_{\Omega^e} \psi_i^e f dxdy$.
Therefore, the finite element model for this problem, for an *n-node* element, is

$$\sum_{j=1}^{n} K_{ij}^e u_j^e = Q_i^e + F_i^e, \; i = 1,2,...,n \quad (\text{with} \quad F_i^e = 0) \tag{9}$$

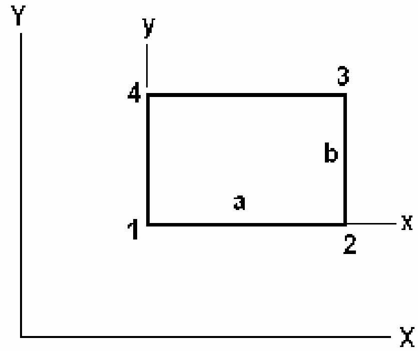
Interpolation Functions

Interpolation functions for the rectangular element are easily constructed as the matrix or tensor product of 1-dimensional Lagrange Interpolation polynomials. The *k-th* order Lagrange Interpolation polynomial, for n data points, $[x_i,\ i = 1,2,...,n]$ is the same as the 1-dimensional interpolation function

$$L_k(x) = \prod_{\substack{i=1 \\ i \neq j}}^{n} \frac{x - x_i}{x_k - x_i} = \psi_k^e(x)$$

Obviously, $L_k(x_j) = \delta_{jk}$, and $\sum_{k=1}^{n} L_k(x) = 1$ as expected.

For the *4-node* rectangular element shown in Figure 3, the interpolation functions are



$$\begin{aligned}
\psi_1^e(x,y) &= \left(1-\frac{x}{a}\right)\left(1-\frac{y}{b}\right) \\
\psi_2^e(x,y) &= \frac{x}{a}\left(1-\frac{y}{b}\right) \\
\psi_3^e(x,y) &= \frac{x}{a}\frac{y}{b} \\
\psi_4^e(x,y) &= \left(1-\frac{x}{a}\right)\frac{y}{b}
\end{aligned} \tag{10}$$

In equation (10), the interpolation functions are defined in terms of local coordinates (x, y).

Element coefficient or stiffness matrix

The element coefficient matrix, also called the element stiffness matrix, is computed from the expression

$$K_{ij}^e = \int_{\Omega^e} \left[\frac{\partial \psi_i^e}{\partial x}\frac{\partial \psi_j^e}{\partial x} + \frac{\partial \psi_i^e}{\partial y}\frac{\partial \psi_j^e}{\partial y} \right] dxdy$$

The element coefficient matrix is symmetric. Usually, *Gauss-Legendre Quadrature* [6] is used to numerically compute $[K^e]$, but the integrals are separable in x and y and easy to evaluate exactly. The computed element stiffness matrix is

$$[K^e] = \begin{bmatrix} 2(a^2+b^2) & a^2-2b^2 & -(a^2+b^2) & b^2-2a^2 \\ a^2-2b^2 & 2(a^2+b^2) & b^2-2a^2 & -(a^2+b^2) \\ -(a^2+b^2) & b^2-2a^2 & 2(a^2+b^2) & a^2-2b^2 \\ b^2-2a^2 & -(a^2+b^2) & a^2-2b^2 & 2(a^2+b^2) \end{bmatrix} \tag{11}$$

Assembled Equations and Numerical Solution

The creation of the assembled equations requires the definition of the element connectivity matrix, which defines the equivalence between local element node numbers and global node numbers. From the mesh given in Figure 1, the element connectivity matrix is

$$[B] = \begin{bmatrix} 1 & 2 & 9 & 10 \\ 2 & 3 & 8 & 9 \\ 3 & 4 & 7 & 8 \\ 4 & 5 & 6 & 7 \\ 7 & 6 & 15 & 14 \\ 8 & 7 & 14 & 13 \\ 9 & 8 & 13 & 12 \\ 10 & 9 & 12 & 11 \end{bmatrix} \tag{12}$$

There are 15 global nodes, therefore the assembled global stiffness matrix, $[K]$, will be a *[15x15]* matrix, with one degree of freedom per node. There are no flux or source term vectors to be computed. From the element connectivity matrix, it is clear that the mappings for the first two elements into the global stiffness matrix are obtained as follows.

ELEMENT #1

Local Indices of $[K^e]$ — Global Indices of $[K]$

$$\begin{bmatrix} (1,1) & (1,2) & (1,3) & (1,4) \\ (2,1) & (2,2) & (2,3) & (2,4) \\ (3,1) & (3,2) & (3,3) & (3,4) \\ (4,1) & (4,2) & (4,3) & (4,4) \end{bmatrix} \xrightarrow{\text{map to}} \begin{bmatrix} (1,1) & (1,2) & (1,9) & (1,10) \\ (2,1) & (2,2) & (2,9) & (2,10) \\ (9,1) & (9,2) & (9,9) & (9,10) \\ (10,1) & (10,2) & (10,9) & (10,10) \end{bmatrix}$$

ELEMENT #2

Local Indices of $[K^e]$ — Global Indices of $[K]$

$$\begin{bmatrix} (1,1) & (1,2) & (1,3) & (1,4) \\ (2,1) & (2,2) & (2,3) & (2,4) \\ (3,1) & (3,2) & (3,3) & (3,4) \\ (4,1) & (4,2) & (4,3) & (4,4) \end{bmatrix} \xrightarrow{\text{map to}} \begin{bmatrix} (2,2) & (2,3) & (2,8) & (2,9) \\ (3,2) & (3,3) & (3,8) & (3,9) \\ (8,2) & (8,3) & (8,8) & (8,9) \\ (9,2) & (9,3) & (9,8) & (9,9) \end{bmatrix}$$

The assembly procedure was performed by writing a short Matlab® code, whose output is shown below:

For a typical rectangular element:

$$[Ke] = \begin{bmatrix} 2/3 & -1/6 & -1/3 & -1/6 \\ -1/6 & 2/3 & -1/6 & -1/3 \\ -1/3 & -1/6 & 2/3 & -1/6 \\ -1/6 & -1/3 & -1/6 & 2/3 \end{bmatrix} \qquad \{Fe\} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

ASSEMBLED GLOBAL STIFFNESS MATRIX

$$[K]=\begin{bmatrix}
2/3 & -1/6 & 0 & 0 & 0 & 0 & 0 & 0 & -1/3 & -1/6 & 0 & 0 & 0 & 0 & 0 \\
-1/6 & 4/3 & -1/6 & 0 & 0 & 0 & 0 & -1/3 & -1/3 & -1/3 & 0 & 0 & 0 & 0 & 0 \\
0 & -1/6 & 4/3 & -1/6 & 0 & 0 & -1/3 & -1/3 & -1/3 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1/6 & 4/3 & -1/6 & -1/3 & -1/3 & -1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1/6 & 2/3 & -1/6 & -1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1/3 & -1/6 & 4/3 & -1/3 & 0 & 0 & 0 & 0 & 0 & 0 & -1/3 & -1/6 \\
0 & 0 & -1/3 & -1/3 & -1/3 & -1/3 & 8/3 & -1/3 & 0 & 0 & 0 & 0 & -1/3 & -1/3 & -1/3 \\
0 & -1/3 & -1/3 & -1/3 & 0 & 0 & -1/3 & 8/3 & -1/3 & 0 & 0 & -1/3 & -1/3 & -1/3 & 0 \\
-1/3 & -1/3 & -1/3 & 0 & 0 & 0 & 0 & -1/3 & 8/3 & -1/3 & -1/3 & -1/3 & -1/3 & 0 & 0 \\
-1/6 & -1/3 & 0 & 0 & 0 & 0 & 0 & 0 & -1/3 & 4/3 & -1/6 & -1/3 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1/3 & -1/6 & 2/3 & -1/6 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1/3 & -1/3 & -1/3 & -1/6 & 4/3 & -1/6 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1/3 & -1/3 & -1/3 & 0 & 0 & -1/6 & 4/3 & -1/6 & 0 \\
0 & 0 & 0 & 0 & 0 & -1/3 & -1/3 & -1/3 & 0 & 0 & 0 & 0 & -1/6 & 4/3 & -1/6 \\
0 & 0 & 0 & 0 & 0 & -1/6 & -1/3 & 0 & 0 & 0 & 0 & 0 & 0 & -1/6 & 2/3
\end{bmatrix}$$

ASSEMBLED SOURCE AND FLUX TERMS

{F}= [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]'   (15x1 MATRIX)

The assembled equation is of the form

$$[K]\{U\} = \{F\}.$$

Because of Essential Boundary conditions on the boundary of the domain, the nodal solution vector should be of the form

{U}= [75 50 50 50 150 250 U7 U8 U9 100 150 200 200 200 225]'   (15x1 MATRIX)

so that the unknown values of {U} occur at global nodes 7,8 and 9.
At the singular points (global nodes 1,5,15 and 11), the specified nodal values are handled by averaging. Another option is to select the higher of the two specified values.

The condensed equations are obtained by eliminating rows (and columns) 1-6 and 10-15. All known quantities are moved from the left side of the matrix equation to the right side to obtain the condensed equations

$$\begin{bmatrix} 2.6667 & -0.3333 & 0 \\ -0.3333 & 2.6667 & -0.3333 \\ 0 & -0.3333 & 2.6667 \end{bmatrix}\begin{bmatrix} U7 \\ U8 \\ U9 \end{bmatrix} = \begin{bmatrix} 375.0000 \\ 250.0000 \\ 275.0000 \end{bmatrix}$$

The solution of this equation yields

$$\begin{bmatrix} U7 \\ U8 \\ U9 \end{bmatrix} = \begin{bmatrix} 156.6532 \\ 128.2258 \\ 119.1532 \end{bmatrix}$$

Post processing and Validation of Results

To validate the results obtained from the Finite Element Analysis, a contour plot of $u(x, y)$ inside the solution domain will be generated. The solution within an element will be computed from the assumed solution

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

For example, figure 1 indicates that the computation of $u(2.5,1.5)$ should be performed within element number 6. Within the element, the global coordinate $x = 2.5$ becomes the local coordinate $x = 0.5$, and the global coordinate $y = 1.5$ becomes the local coordinate $y = 0.5$.

For $a = b = 1$, the values of the interpolating functions for local nodes 1-4 at the local coordinate $(0.5,0.5)$ are obtained as shown in Figure 3 below:

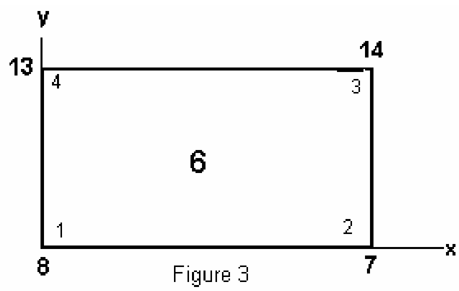


Figure 3

U7 = 156.6532
U8 = 128.2258
U13 = 200
U14 = 200

$$\psi_1^6(0.5,0.5) = (1-0.5)(1-0.5) = 0.25$$
$$\psi_2^6(0.5,0.5) = (0.5)(1-0.5) = 0.25$$
$$\psi_3^6(0.5,0.5) = (0.5)(0.5) = 0.25$$
$$\psi_4^6(0.5,0.5) = (1-0.5)(0.5) = 0.25$$
$$u(2.5,1.5) = 0.25(156.6532 + 128.2258 + 200 + 200) = 171.22$$

A Matlab® program was written to compute the solution field $u(x, y)$ inside the solution domain, and to generate the contour plot shown in Figure 4.

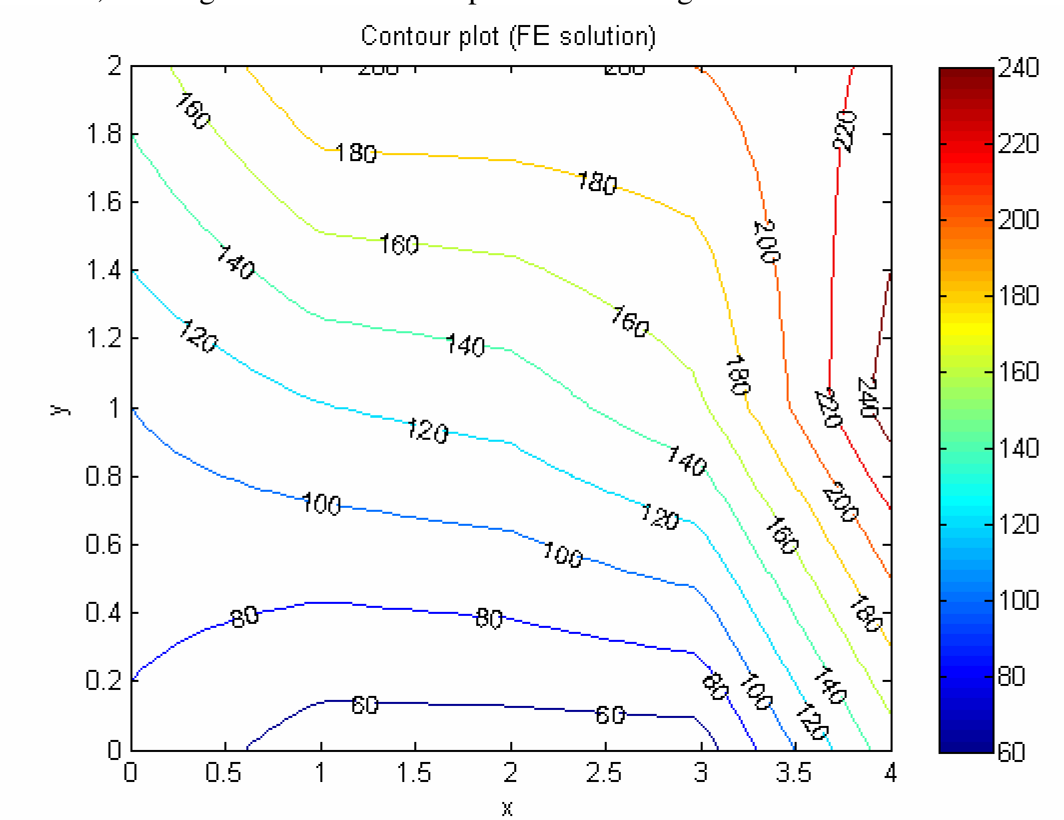
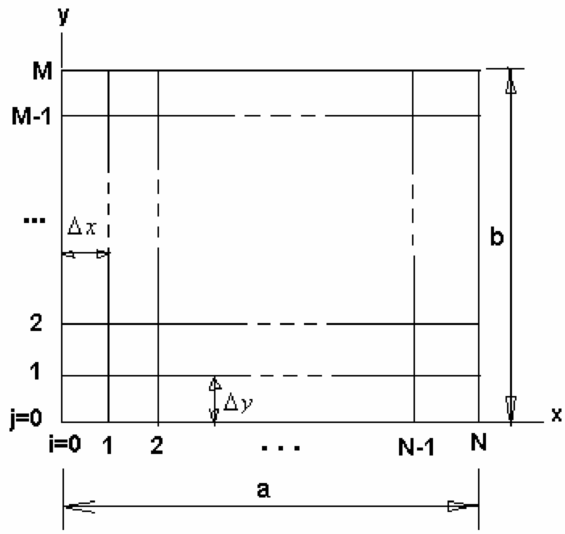


Figure 4

Finite Difference Solution

To solve the governing equation by using the Finite Difference method, we define the solution over the grid shown in Figure 5 below:



$$u_{i,j} \equiv u(i\,\Delta x,\; j\,\Delta y)$$

$$\text{for} \quad i = 0,1,2,\ldots,N \qquad j = 0,1,2,\ldots,M$$

$$N = a/\Delta x$$

$$M = b/\Delta y$$

Figure 5: Finite Difference Grid

In discretized form, equation (1) is

$$\frac{[u_{i+1,j} + u_{i-1,j} - 2u_{i,j}]}{\Delta x^2} + \frac{[u_{i,j+1} + u_{i,j-1} - 2u_{i,j}]}{\Delta y^2} = 0$$

This equation may be written in a form that can be solved iteratively by the *Jacobi method* [7]:

$$u_{i,j}^{(k+1)} = \frac{1}{2(1+\alpha^2)}\left(\alpha^2[u_{i+1,j} + u_{i-1,j}]^{(k)} + [u_{i,j+1} + u_{i,j-1}]^{(k)}\right)$$

where $\alpha = \dfrac{\Delta y}{\Delta x}$

$k = 0,1,2,\ldots,$ are iteration levels

$u_{i,j}^{(k)} \equiv$ value of $u(x,y)$ at nodes (i,j) after k iterations.

An initial guess for $u_{i,j}^{(0)}$, in the solution domain: $\{\, i = 1,2,\ldots,N-1;\; j = 1,2,\ldots,M-1 \,\}$ is required to start the solution.

Notice that the solution domain should not include the boundaries, because Dirichlet boundary conditions are specified.

The boundary conditions, in discretized form are

$$u(0,y) = 100 \Rightarrow u_{0,j} = 100$$

$$u(a,y) = 250 \Rightarrow u_{N,j} = 250$$

$$u(x,0) = 50 \Rightarrow u_{i,0} = 50$$

$$u(x,b) = 200 \Rightarrow u_{i,M} = 200$$

Specified boundary values at the singular points are handled by averaging, as in the Finite Element method.

Comparison between Finite Element and Finite Difference Solutions

The solution along the line $y = 1.5$ (listed in Table 1, and shown in Figure 6) was also computed at the locations $x = 0, 0.5, 1.0, 1.5, \ldots, 4.0$ for comparison with the Finite Difference solution. Better agreement should be obtained between the two results by using a finer grid for the FD solution, and by using higher level *h-meshing* for the FE solution.

Table 1: Comparison between FE and FD solutions

x	**u(x,1.5) FE solution**	**u(x,1.5) FD solution**
0	100	100
0.50	142.3	141.7
1.00	159.6	156.1
1.50	161.8	161.7
2.00	164.1	165.8
2.50	171.2	172.0
3.00	178.3	183.8
3.50	207.9	207.0
4.00	250	250

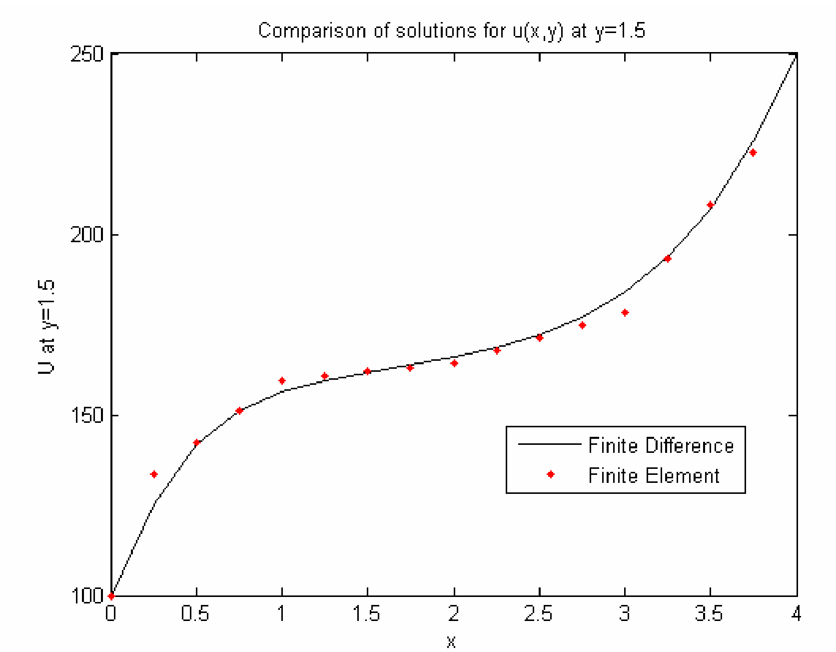


Figure 6

Conclusions

This paper provides a summary of a pedagogical approach for teaching fundamentals of Finite Element Analysis in an introductory course, and how to test the students' comprehension of the course material. The problem that the students have solved addresses most of the basic concepts of Finite Element Analysis, and demonstrates how to perform a test of reasonableness to validate the numerical results. Furthermore,

prerequisites for the course such as functional interpolation, iterative solution of a system of linear equations, and the Finite Difference solution of linear boundary value problems (from a course in Numerical Methods) are tested as well.
Other problems of this type that could be solved will use both rectangular and triangular elements, and could include source (heat generation) terms. The domain of the problem could have irregular geometries.
The boundary conditions for this problem are of the Dirichlet type because values for the PV are specified on the boundary. Other types of boundary conditions could be used to solve the same problem. Neumann boundary condition could specify convective heat transfer on the boundaries, or Mixed Boundary conditions (includes boundary conditions of both the Dirichlet and Neumann types) could be specified. The Neumann type boundary conditions become Natural Boundary conditions on the Secondary Variable, and are satisfied during the weak formulation. The assembly of the element equations will require a little more bookkeeping in order to balance the source and flux terms.
By introducing a little more complexity in formulating the Finite Element model, many useful and practical problems could be solved. The power of the Finite Element method becomes more evident, because the Finite Difference method will have much more difficulty in solving problems in a domain with complex geometries.

References

1. Hughes, Thomas, R., *The Finite Element Method (Linear Static and Dynamic Finite Element Analysis)*, Dover Publications, New York (2000), ISBN 0-486-41181-8. pages. 126-131.
2. Faires, Douglas, J. and Burden, Richard, *Numerical Methods*, 3rd Ed.,Thomson (2003), ISBN 0-534-40761-7, pages 66-72.
3. *Ibid*., pages 85-88.
4. Reddy, J.N., An Introduction to the Finite Element Method, 3rd Ed., McGraw Hill (2006), ISBN 0-07-246685-5.
5. Zienkiewicz, O.C. and Taylor, R.L., *The Finite Element Method*, 4th Ed. Vol. 1, *Basic Formulation and Linear Problems*, McGraw-Hill, London, U.K. (1989).
6. See Reference 2, pages 138-143, and Reference 4, pages 348-353.
7. See Reference 2, pages 306-311.