Jim Lambers
ENERGY 281
Spring Quarter 2006-07
Lecture 13 Notes

These notes were originally written by Tara LaForce.

1 Two-Dimensional Problems

At long last, it is time to solve some PDEs using FEM! Fortunately, all of the concepts from the previous two lectures apply (almost) directly to two-dimensional problems. We will consider problems of the form

$$-\nabla \cdot [k(x,y)\nabla u\,(x,y)] + b\,(x,y)\,u\,(x,y) = f\,(x,y)\,, \tag{1}$$

where the negative sign ensures that we are solving an elliptic equation. Once again, we will define a flux term $\sigma(x,y) = k(x,y)\nabla u\,(x,y)$, where k is the material modulus. In porous media flow, k would be the absolute permeability, and in heat loss k would be thermal conductivity. As before, these do not need to be constants, or even continuous. However, we will restrict our derivation to cases where k is either continuous, or at least nicely enough behaved to not cause discontinuities in the flux. This is exactly what we did in solving ODEs.

We will formulate the PDE as a variational boundary-value problem by multiplying by a test function v and integrating over the domain (this is now an integral in two-dimensions!)

$$\int_\Omega [-\nabla \cdot (k(x,y)\nabla u\,(x,y)) + b\,(x,y)\,u\,(x,y) - f\,(x,y)]\,v\,dx\,dy = 0 \tag{2}$$

In order to get everything in terms of first derivatives like we did in ODEs we use the product rule for differentiation to show that

$$\begin{aligned} \nabla \cdot (vk\nabla u) &= k\nabla u \cdot \nabla v + v\nabla \cdot (k\nabla u) \\ v\nabla \cdot (k\nabla u) &= \nabla \cdot (vk\nabla u) - k\nabla u \cdot \nabla v \end{aligned} \tag{3}$$

which can be inserted into (2) to give

$$\int_{\Omega} [k\nabla u \cdot \nabla v + buv - fv] \; dx\, dy - \int_{\Omega} [\nabla \cdot (vk\nabla u)] \; dx\, dy = 0 \qquad (4)$$

Using the divergence theorem we obtain

$$-\int_{\Omega} [\nabla \cdot (vk\nabla u)] \; dx\, dy = -\int_{\partial\Omega} k\nabla u \cdot nv \, ds, \qquad (5)$$

where $\partial\Omega$ is the boundary of Ω integrated counterclockwise. Now we have the final variational boundary-value problem

$$\int_{\Omega} [k\nabla u \cdot \nabla v + buv - fv] \; dx\, dy - \int_{\partial\Omega} k\frac{\partial u\,(s)}{\partial n} v \, ds = 0 \qquad (6)$$

with boundary conditions

$$-k\,(s)\,\frac{\partial u\,(s)}{\partial n} = p\,(s)\,u\,(s), \qquad (7)$$

where $s \in \partial\Omega$, the boundary of Ω.

1.1 Approximation Functions

The idea here is to represent the approximate solution $u_h(x, y)$ and test functions $v_h(x, y)$ by polynomials defined piecewise over geometrically simple subdomains of Ω. In one-dimension this consisted of dividing the line between $[0, l]$ up into parts. In two-dimensions there are many posssible choices of simple shapes that we could choose to divide up the domain into. We will only consider trianglar and rectangular elements.

1.1.1 Two-Dimensional Problems on Triangular Mesh

The simplest possible choice of shape function in two dimensions is a line $v_h(x, y) = a_1 + a_2x + a_3y$. Three constants need to be found, which means every element must have three nodes. A triangle with nodes at the corners would be the simplest and most logical way to satisfy this constraint. Moreover, if adjacent triangular elements are forced to share two nodes then this will define a continuous function across the element boundary.

Similarly if we wanted to use a quadratic shape function $v_h(x, y) = a_1 + a_2x + a_3y + a_4x^2 + a_5xy + a_6y^2$ we have six parameters and a triangular mesh with nodes at each corner and at the midpoints of each side of the tringle would be a good choice.

1.1.2 Two-Dimensional Problems on Rectangular Mesh

Suppose instead that we wanted to use bilinear shape functions $v_h(x,y) = a_1 + a_2 x + a_3 y + a_4 xy$. In this case we need to specify four nodes per element in order to find the four constants a_1 to a_4. The logical choice here would be to use rectangular elements with nodes defined at each corner. If adjacent elements are forced to share two nodes then this will define a continuous function across the element boundary.

There are infinitely many possible combinations of shape functions and elements. Sometimes it is desirable to use more complicated shapes in the mesh instead of triangular or rectangular. We will restrict our analysis to linear functions on a triangular mesh, since that is the simplest choice.

1.1.3 Shape Functions

In two dimensions there are three basic requirements for the shape functions (and they are almost the same as in the ODE):

- The approximation to u must be continuous across element boundaries.
- The shape functions ψ_i^e must each be one at exactly one node and zero at all others.
- The basis functions must be square-integrable and have square-integrable first partial derivatives.

The linear function $v_h(x,y) = a_1 + a_2 x + a_3 y$ defines a plane in space. As a consequence the approximation of u will be made up of triangular shaped segments of planes that are continuous.

Suppose that the corners of a triangular elements are given by (x_1, y_1) (x_2, y_2) and (x_3, y_3). The shape function ψ_1 that equals one at (x_1, y_1) and zero at nodes (x_2, y_2) and (x_3, y_3) is found from the plane equation evaluated at each node and is

$$\psi_1^e(\xi) = \frac{1}{2A_e}\left[(x_2 y_3 - x_3 y_2) + (y_2 - y_3)\,x + (x_3 - x_2)\,y\right] \qquad (8)$$

where $A_e = x_2y_3 + x_1y_2 + x_3y_1 - x_2y_1 - x_3y_2 - x_1y_3$ is the area of the element. Similarly, the shape functions that are one at the node (x_2, y_2) and (x_3, y_3) are ψ_2 and ψ_3 respectively.

$$\begin{aligned} \psi_2^e(\xi) &= \tfrac{1}{2A_e}\left[(x_3y_1 - x_1y_3) + (y_3 - y_1)\,x + (x_1 - x_3)\,y\right] \\ \psi_3^e(\xi) &= \tfrac{1}{2A_e}\left[(x_1y_2 - x_2y_1) + (y_1 - y_2)\,x + (x_2 - x_1)\,y\right] \end{aligned} \tag{9}$$

1.2 Finite Element Approximations

In general we are trying to find $u_h(x, y) = \sum_{j=1}^{N} u_j \phi_j(x, y)$ such that $u_j = \hat{u}_j$ at the nodes on $\partial\Omega_h$ and

$$\begin{aligned} &\textstyle\int_{\Omega_h} \left[k\left(\frac{\partial u_h}{\partial x}\frac{\partial v_h}{\partial x} + \frac{\partial u_h}{\partial y}\frac{\partial v_h}{\partial y}\right) + b u_h v_h\right] dx\,dy + \int_{\partial\Omega} p u_h v_h\,ds \\ &= \textstyle\int_{\Omega_h} f v_h\,dx\,dy + \int_{\partial\Omega_h} \gamma v_h\,ds \end{aligned} \tag{10}$$

where $\gamma = p\hat{u}$. The general boundary condition has been put into the integral equation.

As before, the stiffness matrix K is given by

$$K_{ij} = \int_{\Omega_h} \left[k\left(\frac{\partial \phi_j}{\partial x}\frac{\partial \phi_i}{\partial x} + \frac{\partial \phi_j}{\partial y}\frac{\partial \phi_i}{\partial y}\right) + b\phi_j\phi_i\right] dx\,dy + \int_{\partial\Omega} p\phi_j\phi_i\,ds \tag{11}$$

and the load vector F is

$$F_i = \int_{\Omega_h} f\phi_i\,dx\,dy + \int_{\partial\Omega_h} \gamma\phi_i\,ds. \tag{12}$$

Ignoring the boundary conditions for the moment, we can look at the structure of the linear basis functions and see that each function v_i will contribute to exactly three of the columns of K (i.e., it effects three of the u_j) in row i since there are three nodes per element. Hence the element matrix k^e is a 3×3 matrix, and the element load vector f^e is a 3×1 column vector.

Consider a generic element e. The contribution to the stiffness matrix from the first basis function $v = \psi_1$ is given by

$$
\begin{array}{l}
\int_{\Omega_h}\left[\begin{array}{l} k\left[\begin{array}{l}(\psi_{x1}\alpha_1+\psi_{x2}\alpha_2+\psi_{x2}\alpha_3)\,\psi_{x1}\\ +(\psi_{y1}\alpha_1+\psi_{y2}\alpha_2+\psi_{y2}\alpha_3)\,\psi_{y1}\end{array}\right]\\ +b\,(\psi_1\alpha_1+\psi_2\alpha_2+\psi_2\alpha_3)\,\psi_1\end{array}\right]dx\,dy=k_{11}^e\alpha_1+k_{12}^e\alpha_2+k_{13}^e\alpha_3\\
k_{11}^e=\int_{\Omega_h}\left[k\left[\psi_{x1}^2+\psi_{y1}^2\right]+b\psi_1^2\right]\,dx\,dy\\
k_{12}^e=\int_{\Omega_h}\left[k\left[\psi_{x2}\psi_{x1}+\psi_{y2}\psi_{y1}\right]+b\psi_1\psi_2\right]\,dx\,dy\\
k_{13}^e=\int_{\Omega_h}\left[k\left[\psi_{x3}\psi_{x1}+\psi_{y3}\psi_{y1}\right]+b\psi_1\psi_3\right]\,dx\,dy
\end{array}
\tag{13}
$$

where ψ_{yi} is the derivative of ψ_i with respect to y, etc. and $u^e(x,y) = \sum_{j=1}^{3}\alpha_j\psi_j$. Similarly, the contribution to the stiffness matrix from the first basis function $v=\psi_2$ is given by

$$
\begin{array}{l}
k_{21}^e=k_{12}^e\\
k_{22}^e=\int_{\Omega_h}\left[k\left[\psi_{x2}^2+\psi_{y2}^2\right]+b\psi_2^2\right]\,dx\,dy\\
k_{23}^e=\int_{\Omega_h}\left[k\left[\psi_{x3}\psi_{x2}+\psi_{y3}\psi_{y2}\right]+b\psi_2\psi_3\right]\,dx\,dy
\end{array}
\tag{14}
$$

and the contribution from ψ_3 is

$$
\begin{array}{l}
k_{31}^e=k_{13}^e\\
k_{32}^e=k_{23}^e\\
k_{33}^e=\int_{\Omega_h}\left[k\left[\psi_{x3}^2+\psi_{y3}^2\right]+b\psi_3^2\right]\,dx\,dy
\end{array}
\tag{15}
$$

the contribution to the load vector from each basis function on the element e are

$$
f_1^e=\int_{\Omega_h}[f\psi_1]\,dx\,dy,\quad f_2^e=\int_{\Omega_h}[f\psi_2]\,dx\,dy,\quad f_3^e=\int_{\Omega_h}[f\psi_3]\,dx\,dy \tag{16}
$$

The contribution of the first element to the total stiffness matrix and load vector are given by

$$
K^1=\begin{bmatrix}
k_{11}^1 & k_{12}^1 & k_{13}^1 & 0 & 0 & 0 & 0\\
k_{21}^1 & k_{22}^1 & k_{23}^1 & 0 & 0 & 0 & 0\\
k_{31}^1 & k_{32}^1 & k_{33}^1 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
F^1=\begin{bmatrix} f_1^1\\ f_2^1\\ f_3^1\\ 0\\ 0\\ 0\\ 0\end{bmatrix}
\tag{17}
$$

In the case of two-dimensional problems we can't just add each element matrix to the diagonal like we did in one dimensional problems because of the locations of the nodes. Because the mesh is triangular elements and nodes don't necessarily have a nice correspondence. For example element 6 may have as its nodes 3, 5, and 6 as in the picture. In that case, the shape functions on Ω_1 will only effect the value of u at nodes 3, 5, and 6. This means that the contribution of the sixth element to the stiffness matrix and load vector are given by

$$K^6 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & k_{11}^6 & 0 & k_{12}^6 & k_{13}^6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & k_{21}^6 & 0 & k_{22}^6 & k_{23}^6 & 0 \\ 0 & 0 & k_{31}^6 & 0 & k_{32}^6 & k_{33}^6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} F^6 = \begin{bmatrix} 0 \\ 0 \\ f_1^6 \\ 0 \\ f_2^6 \\ f_3^6 \\ 0 \end{bmatrix} \tag{18}$$

For two-dimensional problems the stiffness matrix is no longer tridiagional. If the nodes and elements are carefully ordered it can usually be written so that it is a sparse matrix with large blocks of zeros.

1.3 Boundary Conditions

The stiffness matrix and load vectors we have assembled do not include any boundary conditions. Implementation of boundary conditions is conceptually the same as in one-dimensional problems: Neumann boundary conditions are implemented by subtracting the flux term $\frac{k(s)\partial u(s)}{\partial n}$ from the load vector. Natural boundary conditions are implemented by subtracting the flux term $\frac{k(s)\partial u(s)}{\partial n}$ from the load vector and subtracting the value $p(s)u(s)$ from the components of the matrix that are on the boundary. Dirichlet boundary conditions are implemented by getting rid of the row and column for which u is known and adding the known value to the neighboring load vectors. In practice this can get quite complicated because the boundary is specified on every element that has one side on the edge of Ω. Moreover, the boundary conditions are usually different for different parts of the domain.

1.4 Example

In order to clarify some of these concepts we will look at an example problem. The domain, nodes and elements are shown in Figure 4. The domain of this problem is a reservoir with a constant pressure (Dirichlet) boundary condition along the $x = 0$ axis (Γ_{74}) to model influx from an aquifer, while no-flow (Neumann) boundaries exist everywhere else. The forcing term is a single production well in element 6 with constant pressure. This can also be thought of as a boundary condition. The PDE we are going to solve is

$$-\nabla \cdot [k(x,y)\nabla u\,(x,y)] = f\,(x,y) \tag{19}$$

If we further assume that $k = 1$ is constant then this simplifies to the problem

$$\begin{array}{l} -\nabla^2 u\,(x,y) = f\,(x,y) \\ f\,(x,y) = \left\{ \begin{array}{cc} u_1 & \Omega_6 \\ 0 & \text{elsewhere} \end{array} \right\} \\ u\,(x,y) = 0 \quad \text{on} \quad \Gamma_{41} \\ \frac{\partial u(x,y)}{\partial n} = 0 \quad \text{on} \quad \Gamma_{74}, \Gamma_{12}, \Gamma_{25}, \Gamma_{56}, \Gamma_{67} \end{array} \tag{20}$$

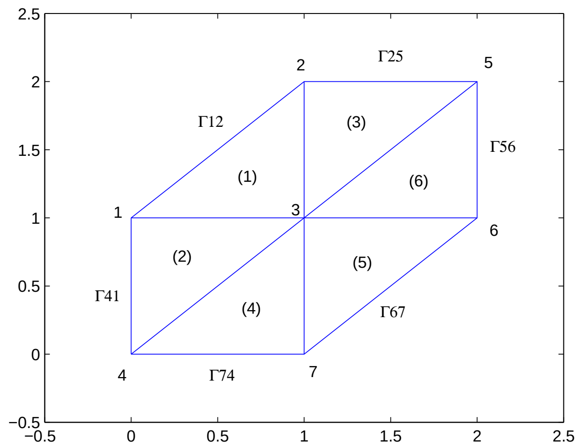

Figure 1: A six element domain with seven nodes.

The stiffness matrix and load vector for this example are given by:

$$K = \begin{bmatrix} K_{11} & K_{12} & K_{13} & K_{14} & 0 & 0 & 0 \\ K_{21} & K_{22} & K_{23} & 0 & K_{25} & 0 & 0 \\ K_{31} & K_{32} & K_{33} & K_{34} & K_{35} & K_{36} & K_{37} \\ K_{41} & 0 & K_{43} & K_{44} & 0 & 0 & K_{47} \\ 0 & K_{52} & K_{53} & 0 & K_{55} & K_{56} & 0 \\ 0 & 0 & K_{63} & 0 & K_{65} & K_{66} & K_{67} \\ 0 & 0 & K_{73} & K_{74} & 0 & K_{76} & K_{77} \end{bmatrix} \quad F = \begin{bmatrix} 0 \\ 0 \\ f_1^6 \\ 0 \\ f_2^6 \\ f_3^6 \\ 0 \end{bmatrix} \tag{21}$$

when the boundary conditions are neglected. Each of the entries K_{ij} is the sum of the contributions of hat function to u at node i. Since node 3 is a member of every element, the row $K_{3,j}$ and the column $K_{i,3}$ are both filled.

Imposing the boundary condition $u_1 = u_4 = 0$ on Γ_{41} shrinks this to a 5×5 matrix problem

$$\begin{bmatrix} K_{22} & K_{23} & K_{25} & 0 & 0 \\ K_{32} & K_{33} & K_{35} & K_{36} & K_{37} \\ K_{52} & K_{53} & K_{55} & K_{56} & 0 \\ 0 & K_{63} & K_{63} & K_{66} & K_{67} \\ 0 & K_{73} & 0 & K_{76} & K_{77} \end{bmatrix} \begin{bmatrix} u_2 \\ u_3 \\ u_5 \\ u_6 \\ u_7 \end{bmatrix} = \begin{bmatrix} F_2 \\ F_3 \\ F_5 \\ F_6 \\ F_7 \end{bmatrix} \tag{22}$$

Since the remaining boundary conditions are no-flux they don't make any contribution to the load vector. If the boundaries were constant flux then elements 1, 2, 4, 5, 6, and 7 of the load vector would have to have the known flux subtracted off of them. This is analogous to the one-dimensional problem with Neumann boundary conditions.

The above matrix problem can be solved by finding $u = K^{-1}F$. In two dimensional problems this can be much harder to do than in 1D because the matrix has filled in and is no longer tri-diagonal. In order to get a sufficiently accurate solution, the size of the matrix K for 2D problems is also generally much larger than for a 1D problem.

References

[1] Becker, E. B., G. F. Carey, and J. T. Oden, *Finite Elements: an Introduction*, Texas Institute for Computational Mechanics, UT Austin, 1981.