Water seepage through a dam: A finite element approach

Ashley B. Pitcher
Student Number 250098269
AM466b Final Project (Undergraduate)
University of Western Ontario
April 15, 2005

Abstract

We consider the problem of water seepage through a dam where water enters the dam at a vertical height of 4.00m, and the water level upon exiting the dam is at 3.00m. The basic physics of groundwater flow is discussed and it is shown that the governing equation for groundwater flow in a homogeneous, isotropic aquifer under steady-state conditions is Laplace's equation. We compute the total head at each point in a two-dimensional cross-section of the dam using a numerical finite element method using both a three-node triangular element mesh and a six-node triangular element mesh. The bottom of the dam rests on impermeable bedrock, making the lower boundary a no flow boundary. The top boundary is also assumed to be a no flow boundary. However, the vertical location of the upper boundary is unknown. Total head is the the sum of pressure head and elevation head. Thus, on the upper boundary, total head must equal elevation since pressure is zero. Using this information, the location of the upper boundary is then determined iteratively using Dupuit assumptions to form an initial guess. The approximations using both the three-node triangular element mesh and the six-node triangular element mesh show the same general resulting pattern of total head over the domain of the problem. As well, results are compared and match reasonably well with those presented by Wang and Anderson (1982). Possible improvements/extensions of the model are also discussed.

1 Background: The physics of groundwater flow

Groundwater flows in the direction of decreasing potential energy caused by differences in pressure and elevation. A common measure of this potential energy is the total head, h, which is simply the sum of pressure head and elevation head:

$$h = \frac{P}{\rho_w g} + z \tag{1}$$

where P is the pressure acting on a unit mass of water, ρ_w is the density of water, g is the acceleration due to gravity, and z is the elevation of the water. Equation (1) assumes that water is an incompressible fluid (the density is the same at all pressures).

As water flows down the head gradient, energy is lost due to friction from the boundary and decreasing elevation. The volume rate of flow per unit area is directly proportional to the rate of change of head as given by the differential form of *Darcy's Law*:

$$\boldsymbol{q} = -K\boldsymbol{\nabla} h \tag{2}$$

where $\boldsymbol{q}$ is the volume rate of flow per unit area, known as the *specific discharge* or *Darcy velocity*, and K is the *hydraulic conductivity* of the liquid.

For steady-state conditions, continuity requires that the amount of water flowing into a representative elemental volume be equal to the amount flowing out. The amount of water is measured by volume, which is equivalent to measuring mass since we have already assumed that water is an incompressible fluid. As well, the elemental volume cannot contain any sources or sinks. Thus, factors such as precipitation and evaporation are ignored. With these assumptions, continuity requires that the following equation holds:

$$\frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y} + \frac{\partial q_z}{\partial z} = 0. \tag{3}$$

In other words, the sum of the net change in the discharge rate of all component directions must equal zero.

Substituting the components of equation (2) into equation (3), we have

$$\frac{\partial}{\partial x}\left(-K\frac{\partial h}{\partial x}\right) + \frac{\partial}{\partial y}\left(-K\frac{\partial h}{\partial y}\right) + \frac{\partial}{\partial z}\left(-K\frac{\partial h}{\partial z}\right) = 0. \tag{4}$$

Now, if we assume that the hydraulic conductivity, K, is independent of x, y, and z (which is true under homogeneous, isotropic conditions) then equation (4) becomes

$$\frac{\partial^2 h}{\partial x^2} + \frac{\partial^2 h}{\partial y^2} + \frac{\partial^2 h}{\partial z^2} = 0 \tag{5}$$

which is Laplace's equation in three dimensions - the governing equation for groundwater flow through an isotropic, homogeneous aquifer under steady-state conditions.

2 Introduction to the problem

We consider here the seepage through a dam problem introduced in [4] for the homogeneous, isotropic case. We wish to estimate the value of the groundwater head throughout a two-dimensional cross section of a dam as seen in Figure 1. The formulation of the problem

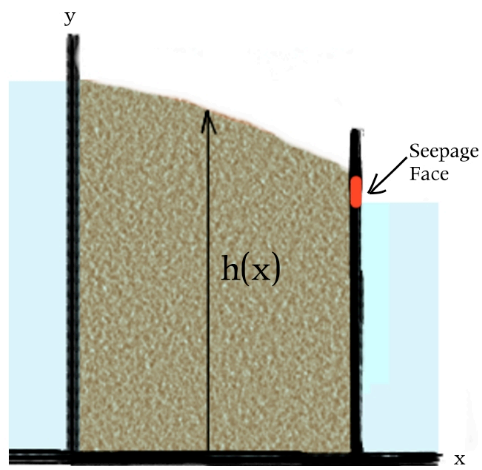

Figure 1: Two-dimensional cross section of a dam.

is as follows. The dam at $y = 0$ rests on impermeable bedrock, so the bottom boundary is a no flow boundary. The upper boundary is also a no flow boundary (no precipitation, evaporation, etc.). In addition, the total head at each point on the upper boundary must equal its elevation. The reason for this is that there is no pressure acting on the upper boundary, making total head equal to elevation (see equation (1)). The total head on the left and right boundaries are known and are 4.00m and 3.00m, respectively. Note that we do not know the vertical location of the upper boundary except that at $x = 0.00$m, $y = 4.00m$.

Since we are dealing with the homogeneous, isotropic case for the cross-section of a dam with no accumulation or loss of water from the system, the equation to be solved for this problem is Laplace's equation in two dimensions:

$$\frac{\partial^2 h}{\partial x^2} + \frac{\partial^2 h}{\partial y^2} = 0 \tag{6}$$

3 Numerical method

We will use the following assumptions to find a reasonable initial guess for the location of the upper boundary.

Dupuit assumptions

1. The exit point coincides with the tailwater (*i.e.* no seepage face - see Figure 1).

2. The hydraulic gradient is constant along a vertical line (flow is horizontal).

3. The hydraulic gradient is equal to the slope of the free surface.

Under Dupuit assumptions, flow is one-dimensional. An analytical solution can therefore be obtained by solving

$$\frac{\partial^2 h^2}{\partial x^2} = 0 \tag{7}$$

with boundary conditions $h = 4.00\text{m}$ at $x = 0.00\text{m}$ and $h = 3.00\text{m}$ at $x = 6.00\text{m}$.[4] The analytical solution of this boundary value problem is

$$h(x) = \sqrt{-1.17x + 16} \tag{8}$$

We will use equation (8) to determine a close initial guess for the location of the upper boundary (see Figure 1). We will solve the problem using the finite element method and compare the resulting head at the upper boundary with the guess given by equation (8). If they do not agree to two decimal places, we will use the newly calculated head at the upper boundary as the new vertical location (recall that head must equal elevation on the upper boundary) and repeat the procedure again each time comparing head to the vertical location until we have two decimal place agreement along the upper boundary.

The dam seepage problem presented in [4] was solved using a three-node triangular finite element mesh. We will also solve the problem using the same triangular element mesh, and then we extend the numerical method to six-node triangular elements.

4 Finite element method using three-node triangular elements

We discretize the domain into 18 three-node triangular elements as shown in Figure 2. It should be noted that the actual locations of nodes 5, 9, and 13 are not yet known and will be determined by the numerical procedure described in the previous section.

4.1 Shape functions

The shape functions for the three-node triangular master element are

$$\begin{aligned} N_1(u,v) =& u \\ N_2(u,v) =& v \\ N_3(u,v) =& 1-u-v. \end{aligned} \tag{9}$$

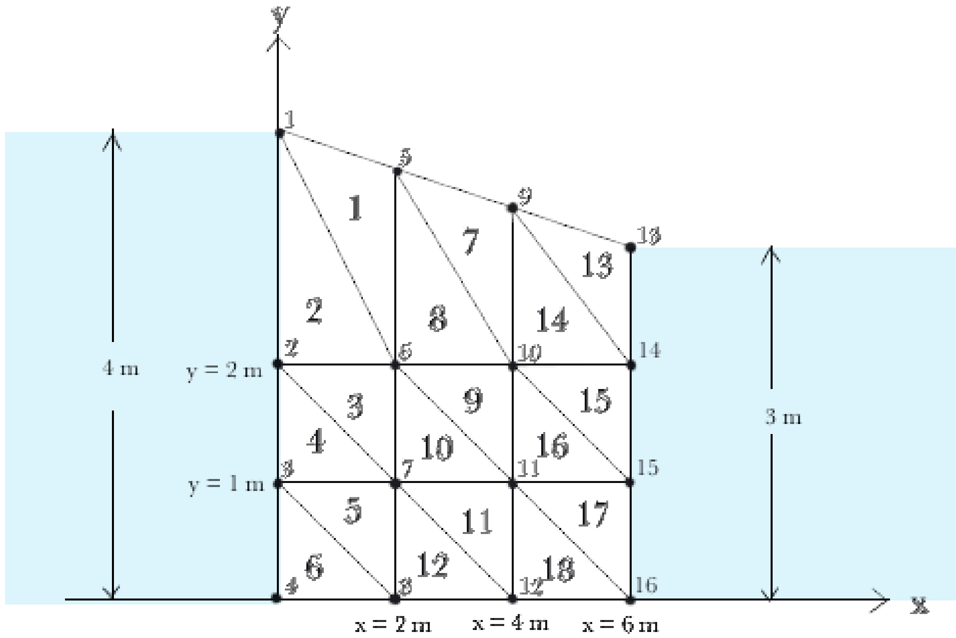

Figure 2: Three-node triangular element mesh. Note that the actual vertical locations of nodes 5, 9, and 13 are unknown.

The mapping that relates the coordinates of the master element to general element Ω_e is

$$\begin{aligned} x(u,v) =& x_{n_1^e} N_1(u,v) + x_{n_2^e} N_2(u,v) + x_{n_3^e} N_3(u,v) \\ y(u,v) =& y_{n_1^e} N_1(u,v) + y_{n_2^e} N_2(u,v) + y_{n_3^e} N_3(u,v) \end{aligned} \tag{10}$$

where $n_1^e(x_{n_1^e}, y_{n_1^e})$, $n_2^e(x_{n_2^e}, y_{n_2^e})$, $n_3^e(x_{n_3^e}, y_{n_3^e})$ are the vertices of the general three-node triangular element.

4.2 Finite element solution

The finite element approximation to this problem using a three-node triangular element mesh is given by

$$h_{approx} = \sum_{j=1}^{16} h_j w_j(x,y) \tag{11}$$

where each h_j is the total head at node j generated by the program `poisson.m` in Matlab (found in [2]) and $w_j(x,y)$ is the basis function representing node j. The basis function for a particular node j is formed using the shape functions of the elements associated with node j. See Appendix A for all computer codes and data files used by `poisson.m` to solve this problem.

4.3 Results

Only three iterations were required to achieve two decimal place agreement between head and elevation on the upper boundary. The values for the unknown node locations at each iteration are summarized in the following table. (Run 0 contains the starting values determined by equation (8).)

Node	5	9	13
Run 0	3.70	3.36	3.00
Run 1	3.73	3.42	3.10
Run 2	3.72	3.41	3.11
Run 3	3.72	3.41	3.11

From the above table, we see that we have a seepage face of 3.11m - 3.00m = 0.11m.

The following table contains the resulting head calculated at each node in the domain using the three-node triangular finite element procedure.

Node	Head (m)
1	4.00
2	4.00
3	4.00
4	4.00
5	3.72
6	3.70
7	3.69
8	3.69
9	3.41
10	3.38
11	3.36
12	3.36
13	3.11
14	3.00
15	3.00
16	3.00

Figure 3 is a colour contour map of the domain with colours representing the total head at each point according to the colour map to the right of the plot. The values in between the nodes are interpolated using the finite element basis functions.

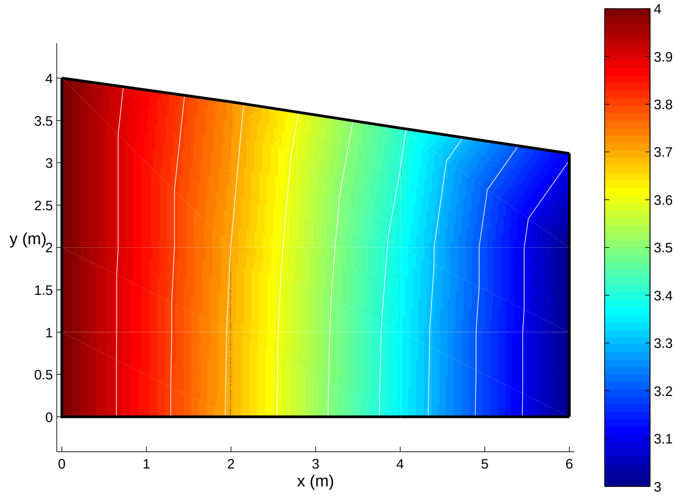

Figure 3: Colour contour map of total head over problem domain calculated using a three-node triangular element mesh (generated using the Matlab code `topo.m` found in [2]).

5 Finite element method using six-node triangular elements

We discretize the domain into 18 six-node triangular elements as shown in figure 4. It should be noted that the actual locations of nodes 8, 15, 22, 29, 36, 43, 16, 23, 30, 37, 43 and 44 are not yet known and will be determined numerically.

5.1 Shape functions

The shape functions for the six-node triangular master element are

$$\begin{aligned} N_1(u,v) =&(u+v-1)(2u+2v-1) \\ N_2(u,v) =&-4u(u+v-1) \\ N_3(u,v) =&u(2u-1) \\ N_4(u,v) =&4uv \\ N_5(u,v) =&v(2v-1) \\ N_6(u,v) =&-4v(u+v-1) \end{aligned} \tag{12}$$

The mapping that relates the coordinates of the master element to general element Ω_e is

$$\begin{aligned} x(u,v) &= \sum_{i=1}^{6} x_{n_i^e} N_i(u,v) \\ y(u,v) &= \sum_{i=1}^{6} y_{n_i^e} N_i(u,v) \end{aligned} \tag{13}$$

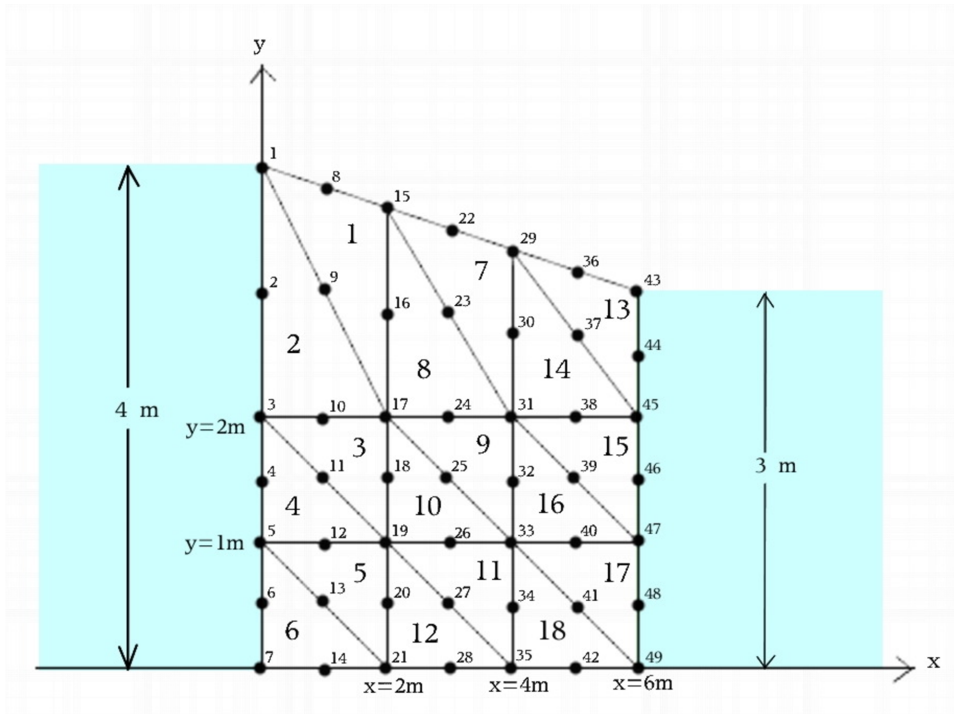

Figure 4: Six-node triangular element mesh.

where $n_i^e(x_{n_i^e}, y_{n_i^e})$, $i = 1, 2, ..., 6$ are the vertices of the general six-node triangular element.

5.2 Finite element solution

The finite element approximation to this problem is given by

$$h_{approx} = \sum_{j=1}^{49} h_j w_j(x, y) \tag{14}$$

where each h_j is the total head at node j generated by the program `steady.m` in Matlab (from [2]) and $w_j(x, y)$ is the basis function representing node j. The basis function for a particular node j is formed using the shape functions of the elements associated with node j. See Appendix B for all computer codes/data files used by `steady.m` to solve this problem.

5.3 Results

Only three iterations were required to achieve two decimal place agreement between head and elevation on the upper boundary. The values for the unknown node locations at each iteration are summarized in the following table. (Run 0 contains the starting values determined by equation (8).)

Node	8	15	16	22	23	29	30	36	37	43	44
Run 0	3.85	3.70	2.85	3.53	2.85	3.36	2.68	3.19	2.68	3.00	2.50
Run 1	3.87	3.73	2.87	3.57	2.87	3.40	2.70	3.22	2.70	3.04	2.52
Run 2	3.87	3.72	2.87	3.57	2.87	3.40	2.70	3.22	2.70	3.04	2.52
Run 3	3.87	3.72	2.87	3.57	2.87	3.40	2.70	3.22	2.70	3.04	2.52

From the above table, we see that we have a seepage face of 3.04m - 3.00m = 0.04m.

Note that the interior nodes in the above table are recalculated at each iteration as the midpoint between the new location of the adjacent boundary node and $y = 2.00$m.

Figure 5 is a colour contour map of the domain with colours representing the total head at each point according to the colour map to the right of the plot. The values in between the nodes are interpolated using the finite element basis functions.

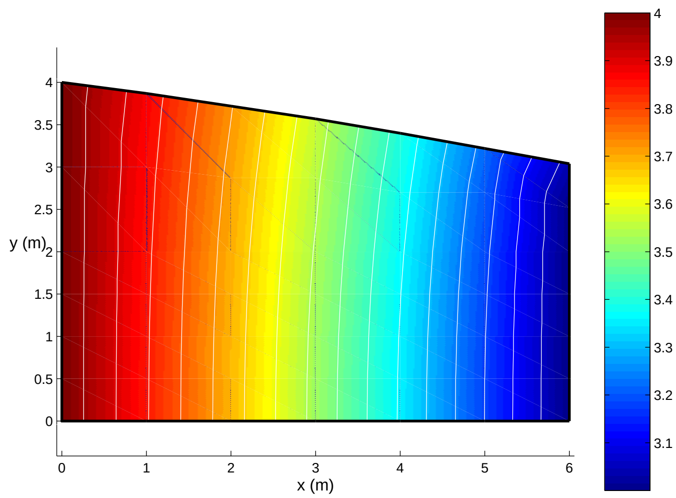

Figure 5: Colour contour map of total head over problem domain calculated using a six-node triangular element mesh (generated using the Matlab code `topo.m` found in [2]).

6 Conclusion and discussion

The following table summarizes the results obtained by [4] using a three-node triangular element mesh.

Node	Head (m)
1	4.00
2	4.00
3	4.00
4	4.00
5	3.72
6	3.69
7	3.68
8	3.68
9	3.40
10	3.37
11	3.36
12	3.35
13	3.08
14	3.00
15	3.00
16	3.00

When compared to the results obtained in this paper for the three-node triangular element mesh, we see that the difference in computed values is at most 0.03m. Most importantly, the same overall pattern of head distribution is maintained by both the three-node triangular element approximation and the six-node triangular element approximation, and are consistent with the solution given by [4] in the above table.

Some finite element and finite difference approximations to this problem assume the Dupuit assumptions hold and therefore no seepage face exists. We see that in the results obtained by [4] as well as the results shown here using both the three-node triangular element approximation and the six-node triangular element approximation all suggest the presence of a seepage face.[1]

We conclude that our results, while not perfectly accurate, support the known findings for the distribution of total head in the water seepage through a dam problem. One can see the similar patterns in the colour maps in Figures 3 and 5.

This model could be improved by generating a finer mesh (more elements) to increase accuracy, possibly through the use of a finite element mesh generator as opposed to generating the mesh by hand, as was done in this project. One could also try four or eight-node quadrilateral elements.

Future work on this model could involve the consideration of the anisotropic case as well as the case allowing precipitation and/or evaporation. Different domain shapes could also be considered, as well as extending the model to consider flow in three dimensions.

[1] The seepage face for this problem (shown in Figure 1) is the difference in the vertical location of the node that belongs to both the right and upper boundaries and the vertical height of the water level exiting the dam (the tailwater).

References

[1] K.R. Rushton, S.C. Redshaw, Seepage and Groundwater Flow. Wiley: New York. 1979.

[2] E.G. Thompson, Introduction To The Finite Element Method: Theory, Programming, and Applications. Wiley: U.S.A. 2005.

[3] J.Toth, A Theoretical Analysis of Groundwater Flow in Small Drainage Basins. Journal of Geophysical Research. 68(16):4795-4812. 1963.

[4] H.F. Wang, M.P. Anderson, Introduction to Groundwater Modeling: Finite Difference and Finite Element Methods. W.H. Freeman and Company: San Francisco. 1982.

Appendix A

```
%----------------
%  MESHo file
%----------------
  16  % NUMNP
  18  % NUMEL
   3  % NNPE
```

```
%----------------------------
%      NP array
%----------------------------
     1    6    5
     1    2    6
     2    7    6
     2    3    7
     3    8    7
     3    4    8
     5   10    9
     5    6   10
     6   11   10
     6    7   11
     7   12   11
     7    8   12
     9   14   13
     9   10   14
    10   15   14
    10   11   15
    11   16   15
    11   12   16
%----------------------------
```

```
%          NODES
%----------------------------
%  XORD   YORD   NPcode
%----------------------------
      0      4      4
      0      2      4
      0      1      4
      0      0      4
      2      3.72   1
      2      2      0
      2      1      0
      2      0      1
      4      3.41   1
      4      2      0
      4      1      0
      4      0      1
      6      3.11   1
      6      2      3
      6      1      3
      6      0      3
%----------------------------
```

```
%-----------------
%  NWLD array
%-----------------
    13
    9
    5
    1
    14
    10
    6
    2
    15
    11
    7
    3
    16
    12
    8
    4
    5    % IB
```

```
%----------------------------------
%     INITIAL.m
%
%     Specify boundary conditions
%------------------------------------
     for I=1:NUMNP
        if NPcode(I) == 4
       NPBC(I) = 1;
       PHI(I)  = 4.00;
       Q(I)    = 0;
        elseif NPcode(I) == 3
       NPBC(I) = 1;
       PHI(I)  = 3.00;
       Q(I)    = 0;
        elseif NPcode(I) == 1
       NPBC(I) = 0;
       PHI(I)  = 0;
       Q(I)    = 0;
        end
     end
```

```
%==================================
%    COEF.m
%    Coefficients for poisson.m
%
%     RXI = k_x
%     RYI = k_Y
%     QVI = Q
%
%==================================

         RXI = 1;
         RYI = 1;
         QVI = 0;
```

Appendix B

```
%----------------
%  MESHo file
%----------------
   49   %  NUMNP
   18   %  NUMEL
    6   %  NNPE
```

```
%------------------------
%    NP array
%
% NP1 NP2 NP3 NP4 NP5 NP6
%------------------------
    1   9  17  16  15   8
    1   2   3  10  17   9
    3  11  19  18  17  10
    3   4   5  12  19  11
    5  13  21  20  19  12
    5   6   7  14  21  13
   15  23  31  30  29  22
   15  16  17  24  31  23
   17  25  33  32  31  24
   17  18  19  26  33  25
   19  27  35  34  33  26
   19  20  21  28  35  27
   29  37  45  44  43  36
   29  30  31  38  45  37
   31  39  47  46  45  38
   31  32  33  40  47  39
   33  41  49  48  47  40
   33  34  35  42  49  41
```

```
%--------------------
%      NODES
%
%  XORD  YORD  NPcode
%--------------------
    0      4      4
    0      3      4
    0      2      4
    0      1.5    4
    0      1      4
    0      0.5    4
    0      0      4
    1      3.87   1
    1      3      0
    1      2      0
    1      1.5    0
    1      1      0
    1      0.5    0
    1      0      1
    2      3.72   1
    2      2.87   0
    2      2      0
    2      1.5    0
    2      1      0
    2      0.5    0
    2      0      1
    3      3.57   1
    3      2.87   0
    3      2      0
    3      1.5    0
    3      1      0
    3      0.5    0
    3      0      1
    4      3.40   1
    4      2.70   0
    4      2      0
    4      1.5    0
    4      1      0
    4      0.5    0
    4      0      1
    5      3.22   1
    5      2.70   0
    5      2      0
    5      1.5    0
    5      1      0
    5      0.5    0
    5      0      1
    6      3.04   1
    6      2.52   3
    6      2      3
    6      1.5    3
    6      1      3
    6      0.5    3
    6      0      3
```

```
%------------------
%   NWLD array
%------------------
1
8
15
22
29
36
43
2
9
16
23
30
37
44
3
10
17
24
31
38
45
4
11
18
25
32
39
46
5
12
19
26
33
40
47
6
13
20
27
34
41
48
7
14
21
28
35
42
49
17   % IB
```

```
%-------------------------------
%    INITIAL.m
%
%    Set problem parameters.
%-------------------------------

      for Ix=1:NUMNP
        if NPcode(Ix) == 4
       NPBC(Ix)= 1;
       PHI(Ix) = 4.00;
        elseif NPcode(Ix) == 3
       NPBC(Ix)= 1;
       PHI(Ix) = 3.00;
        elseif NPcode(Ix) == 1
       NPBC(Ix)= 0;
       PHI(Ix) = 0;
    else
       NPBC(Ix)=0;
    end
      end
```

```
%-------------------------------------------------------*
%   COEF.m                                              *
%   d    d@    d    d@      d@     d@                   *
%   --(RX--) + --(RY--) + BX-- + BY-- + G@ + HV = 0     *
%   dx   dx    dy   dy      dx     dy                   *
%                                                       *
%-------------------------------------------------------*

    RXJ = 1;
    RYJ = 1;

    BXJ=0;
    BYJ=0;

    GVJ = 0;
    HVJ = 0;
```