Chapter 15

LINEAR FLOWS

15.1 Introduction

There are several classes of linear flow models that can be cast as a finite element model. Other flows are highly nonlinear and require much more advanced finite element methods and nonlinear equation solvers. Probably the most common example is the solution of the Navier- Stokes equations for fluid flow. There are several successful finite element formulations of computational fluid dynamics (CFD). Since they are nonlinear and require an iterative solution several implementations begin with a linear flow approximation that satisfies the continuity equation (mass conservation). Thus, there continues to be a need to have efficient linear flow models. Linear flow problems include a wide range of applications like potential flow, flow through porous media, lubrication flow, creeping viscous flows, all of which are elliptic in nature, and other classes like transonic potential flow that changes from elliptic to hyperbolic in nature as the Mach number increases. The subsonic flow of an ideal gas reduces to a nonlinear Poisson equation but it is often reduced to a linearized theory that gives a near singular Laplace equation usually known as the Prandl-Glauert equation. Here we will review some of the common examples of linear flows solved by finite element methods.

15.1 Potential Flow

A common class of problem which can be formulated in terms of the Poisson equation is that of potential flow of ideal fluids. That is, we wish to model an invisid, irrotational, incompressible, steady state flow. Potential flow can be formulated in terms of the velocity potential, ϕ, or the stream function, ψ. The latter sometimes yields simpler boundary conditions, but ϕ will be utilized here since it can be easily extended to three dimensions while the stream function is quite difficult to generalize to three dimensions. For the velocity potential formulation the diffusion coefficients, K_x and K_y, become the fluid mass density, which is assumed to be constant. The source term, G, represents a source or sink term and is usually zero. In that common case the constant density term could be divided out so that the problem of potential flow is often presented as a solution of Laplace's equation rather than the Poisson equation. However, there are practical applications that merit retaining the Poisson form. Thus, the governing equation

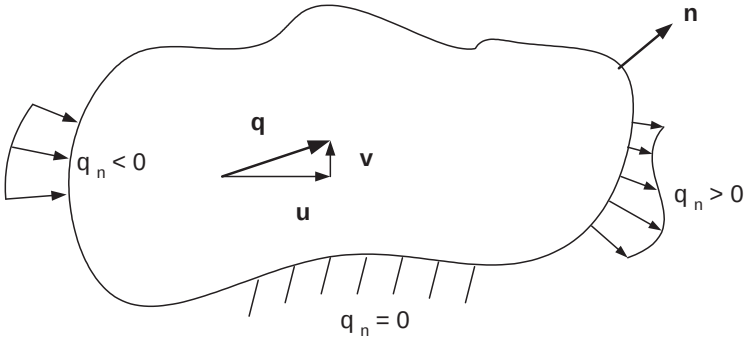

Figure 15.2.1 Defining potential flow terms

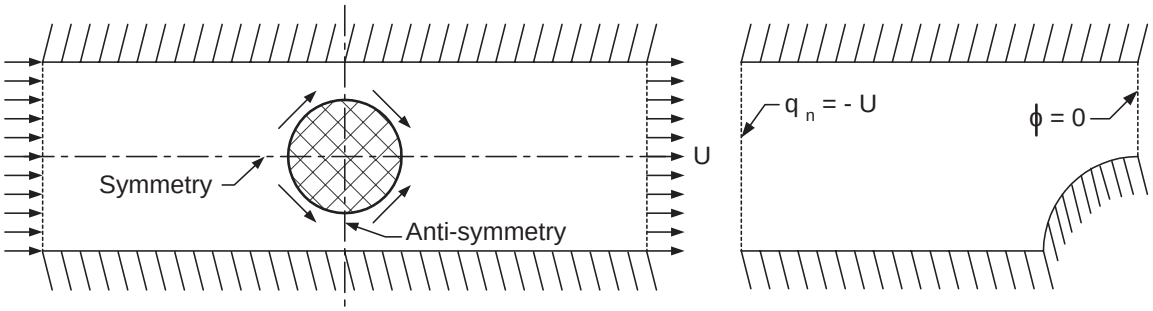

Figure 15.2.2 Typical boundary condition considerations

is

$$\rho\left(\frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2}\right) = Q. \tag{15.1}$$

The velocity potential, ϕ is usually of secondary interest and the analyst generally requires information on the velocity components. They are defined by the global derivatives of ϕ:

$$u \equiv \frac{\partial\phi}{\partial x}, \qquad v \equiv \frac{\partial\phi}{\partial y} \tag{15.2}$$

where u and v denote the x- and y-components of the velocity vector, **q**, in the plane of analysis, as shown in Fig. 15.2.1. Thus, although the program will yield the nodal values of ϕ one must also calculate the above global derivatives. This can be done economically since these derivative quantities must be generated at each quadrature point during construction of the element square matrix, **S**. Hence, one can simply store this derivative information, i.e., matrix **DGH**, and retrieve it later for calculating the global derivatives of ϕ. Another item of interest in Fig. 15.2.1 is the normal boundary flow into or out of the domain, $q_n = \partial\phi / \partial n$. The natural boundary condition is zero flux which represents an impervious wall. The use of these items to establish the boundary conditions in a simple flow domain is illustrated in Fig. 15.2.2. There on the vertical line of anti-

symmetry we see that the vertical component of velocity, v, will be zero so $\partial\phi/\partial y = 0$ along the line of x constant, so we can set ϕ to an arbitrary constant.

Patch Test

We wish to test this Poisson equation solver by running a patch test. Originally based on engineering judgement, the *patch test* has been proven to be a mathematically valid convergence test [7, 8]. Consider a patch (or sub-assembly) of finite elements containing at least one internal node. An internal node is one completely surrounded by elements. Let the problem be formulated by an integral statement containing derivatives of order n. Assume an arbitrary function, $P(x)$, whose n^{th} order derivatives are constant. Use this function to prescribe the dependent variable on all external nodes of the patch (i.e., $\phi_e = P(x_e)$). Solve for the internal nodal values of the dependent variable, ϕ_I, and its n^{th} order derivatives in each element. To be a convergent formulation:

1. The internal nodal values must agree with the assumed function evaluated at the internal points (i.e., $\phi_I = P(x_I)$?); and
2. The calculated n^{th} order derivatives *must* agree with the assumed constant values. In this application that means that all the velocity vectors are identical. Clearly that represents some inclined uniform flow state, as seen in Fig. 15.2.3.

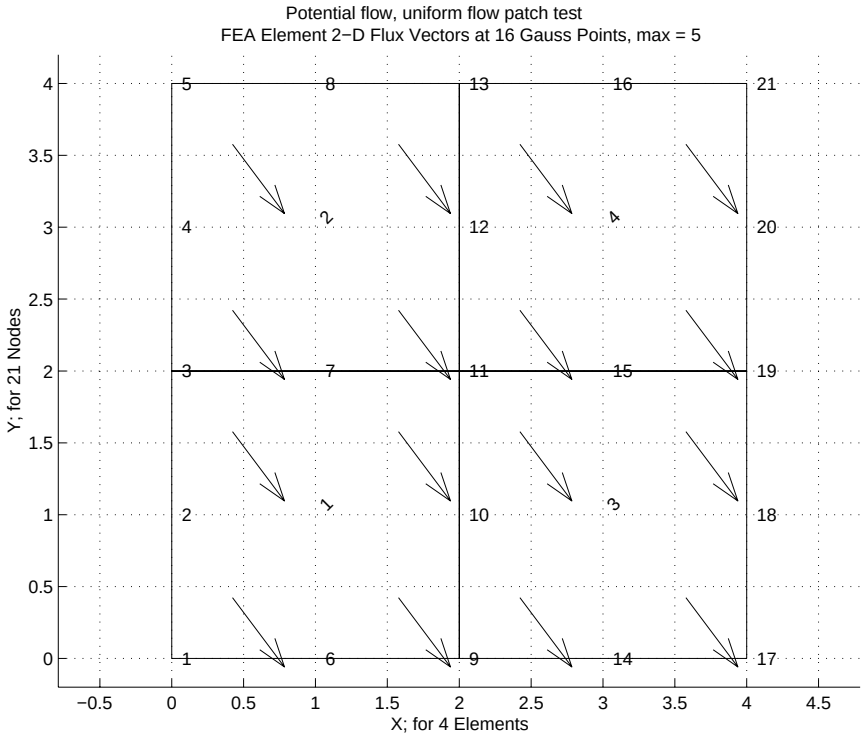

Figure 15.2.3 A uniform flow patch test

It has been found that some non-conforming elements will yield convergent solutions for only one particular mesh pattern. The patch mesh should be completely arbitrary for a valid numerical test. The patch test is very important from the engineering point of view since it can be executed numerically. Thus, one obtains a numerical check of the entire program used to formulate the patch test.

The patch of elements shown in Fig. 15.2.3 was utilized. It was assumed that

$$\phi(x, y) \equiv 1 + 3x - 4y \tag{15.3}$$

such that the derivatives $\phi_{,x} = 3$, $\phi_{,y} = -4$ are constant everywhere. All 16 points on the exterior boundary were assigned values by substituting their coordinates into the above relation. that is, the boundary conditions that $\phi_1 \equiv \phi(x_1, y_1)$, etc., were applied on the exterior boundary. Then the problem was solved numerically to determine the value of ϕ at the interior points (7, 10, 11, 12, 15) and the values of its global derivatives at each integration point. The output results of this patch test are shown in Fig. 15.2.4. The output shows clearly that the global derivatives at all integration points have the assumed values. It is also easily verified that all the interior nodal values of ϕ are consistent with the assumed form. Thus, the patch test is satisfied and the subroutines pass a necessary numerical test. It is also reassuring that the error estimator indicates that even this crude mesh does not need refinement.

Example — Flow Around a Cylinder

Martin and Carey [9] were among the first to publish a numerical example of a finite element potential flow analysis. This same example is also discussed by others [3, 5]. The problem considers the flow around a cylinder in a finite rectangular channel with a uniform inlet flow. The geometry is shown in Fig. 15.2.5. The present mesh is compared with that of Martin and Carey in Fig. 15.2.6. By using centerline symmetry and midstream antisymmetry it is possible to employ only one fourth of the flow field. The stream function boundary conditions are discussed by Martin and Carey [9] and Chung [3]. For the velocity potential one has four sets of Neumann (boundary flux) conditions and one set of Dirichlet (nodal parameter) conditions. The first involve zero normal flow, $q_n = \phi_{,n} \equiv 0$, along the centerline *ab* and the solid surfaces *bc* and *de*, and a uniform unit inflow, $q_n \equiv -1$, along *ad*. At the mid-section, *ce*, antisymmetry requires that $v = 0$. Thus, $\phi_{,y} = 0$ so that $\phi = \phi(x)$, but in this special case x is constant along that line so we can set ϕ to any desired constant, say zero, along *ce*. The output results are illustrated in Fig. 15.2.7. The velocity vector plots are shown in Fig. 15.2.8a. By changing only the specified inlet flux conditions, other problems can be considered.

As before, the presence of a boundary flux, q_n, makes it necessary to evaluate the flux column matrix

$$\mathbf{C}^b = \int_{l^b} \mathbf{H}^{bT} q_n \, ds. \tag{15.4}$$

The variation of q_n along the boundary segment is assumed to be defined by the nodal (input) values and the segment interpolation equations, i.e.,


```
TITLE: "Potential flow patch test, uniform flow"          ! 1
                                                          ! 2
***  REACTION RESULTANTS  ***                             ! 3
PARAMETER,     SUM          POSITIVE     NEGATIVE         ! 4
 DOF_1,      3.5527E-15   2.6000E+01  -2.6000E+01         ! 5
                                                          ! 6
***  OUTPUT OF RESULTS IN NODAL ORDER  ***                ! 7
    NODE,  X-Coord,    Y-Coord,    DOF_1,                ! 8
       1  0.0000E+00  0.0000E+00  1.0000E+00             ! 9
    . . .                                                 !10
       7  1.0000E+00  2.0000E+00 -4.0000E+00             !11
       8  1.0000E+00  4.0000E+00 -1.2000E+01             !12
       9  2.0000E+00  0.0000E+00  7.0000E+00             !13
      10  2.0000E+00  1.0000E+00  3.0000E+00             !14
      11  2.0000E+00  2.0000E+00 -1.0000E+00             !15
      12  2.0000E+00  3.0000E+00 -5.0000E+00             !16
      13  2.0000E+00  4.0000E+00 -9.0000E+00             !17
      14  3.0000E+00  0.0000E+00  1.0000E+01             !18
      15  3.0000E+00  2.0000E+00  2.0000E+00             !19
    . . .                                                 !20
      21  4.0000E+00  4.0000E+00 -3.0000E+00             !21
                                                          !22
*** FLUX COMPONENTS AT ELEMENT INTEGRATION POINTS ***     !23
ELEMENT, PT, X-Coord,   Y-Coord,    FLUX_1,    FLUX_2,    !24
      1  1  4.2265E-1  4.2265E-1  3.0000E+0 -4.0000E+0    !25
      1  2  1.5774E+0  4.2265E-1  3.0000E+0 -4.0000E+0    !26
      1  3  4.2265E-1  1.5774E+0  3.0000E+0 -4.0000E+0    !27
      1  4  1.5774E+0  1.5774E+0  3.0000E+0 -4.0000E+0    !28
    . . .                                                 !29
      4  1  2.4226E+0  2.4226E+0  3.0000E+0 -4.0000E+0    !30
      4  2  3.5774E+0  2.4226E+0  3.0000E+0 -4.0000E+0    !31
      4  3  2.4226E+0  3.5774E+0  3.0000E+0 -4.0000E+0    !32
      4  4  3.5774E+0  3.5774E+0  3.0000E+0 -4.0000E+0    !33
                                                          !34
*** SUPER_CONVERGENT AVERAGED NODAL FLUXES ***            !35
NODE,   X-Coord,    Y-Coord,    FLUX_1,     FLUX_2,      !36
     1  0.0000E+00  0.0000E+00  3.0000E+00 -4.0000E+00    !37
     7  1.0000E+00  2.0000E+00  3.0000E+00 -4.0000E+00    !38
    . . .                                                 !39
     8  1.0000E+00  4.0000E+00  3.0000E+00 -4.0000E+00    !40
     9  2.0000E+00  0.0000E+00  3.0000E+00 -4.0000E+00    !41
    10  2.0000E+00  1.0000E+00  3.0000E+00 -4.0000E+00    !42
    11  2.0000E+00  2.0000E+00  3.0000E+00 -4.0000E+00    !43
    12  2.0000E+00  3.0000E+00  3.0000E+00 -4.0000E+00    !44
    13  2.0000E+00  4.0000E+00  3.0000E+00 -4.0000E+00    !45
    14  3.0000E+00  0.0000E+00  3.0000E+00 -4.0000E+00    !46
    15  3.0000E+00  2.0000E+00  3.0000E+00 -4.0000E+00    !47
    . . .                                                 !48
    21  4.0000E+00  4.0000E+00  3.0000E+00 -4.0000E+00    !49
                                                          !50
 ***  S_C_P  ENERGY  NORM  ERROR  ESTIMATE  DATA ***      !51
              ERROR IN        % ERROR IN      REFINEMENT  !52
 ELEMENT,     ENERGY_NORM,    ENERGY_NORM,    PARAMETER   !53
       1      4.5626E-15      2.2813E-14      4.5626E-14  !54
       2      7.2788E-15      3.6394E-14      7.2788E-14  !55
       3      6.5704E-15      3.2852E-14      6.5704E-14  !56
       4      8.2159E-15      4.1080E-14      8.2159E-14  !57
```

Figure 15.2.4 Numerical results from standard patch test

$$q_n(s) \equiv \mathbf{H}^b(s)\,\mathbf{q}_n^b.$$

Therefore, the segment column matrix (for a straight segment) becomes

$$\mathbf{C}^b = \int_{l^b} \mathbf{H}^{bT}\mathbf{H}^b\, ds\, \mathbf{q}_n^b = \frac{l^b}{30}\begin{bmatrix} 4 & 2 & -1 \\ 2 & 16 & 2 \\ -1 & 2 & 4 \end{bmatrix} \mathbf{q}_n^b . \tag{15.5}$$

15.3 Axisymmetric Plasma Equilibria *

Nuclear fusion is being developed as a future source of energy. The heart of the fusion reactors will be a device for confining the reacting plasma and heating it to thermonuclear temperatures. This confinement problem can be solved through the use of magnetic fields of the proper geometry which generate a so-called "magnetic bottle". The tokamak containment concept employs three magnetic field components to confine the plasma. An externally applied toroidal magnetic field, B_T, is obtained from coils through which the torus passes. A second field component is the polodial magnetic field, B_P, which is produced by a large current flowing in the plasma itself. This current is induced in the plasma by transformer action and assists in heating the plasma. Finally, a vertical (axial) field, B_V, is also applied. These typical fields are illustrated in Fig. 15.3.1. For many purposes a very good picture of the plasma behavior can be obtained by treating it as an ideal magnetohydrodynamic (MHD) media. The equations governing the steady state flow of an ideal MHD plasma are

$$grad\,\mathbf{B} = 0, \quad \nabla \mathrm{P} = \mathbf{J} \times \mathbf{B}, \quad \mathrm{curl}\,\mathbf{B} = \mu\mathbf{J} \tag{15.6}$$

where P is the pressure, **B** the magnetic flux density vector, **J** the current density vector, and μ a constant that depends on the system of units being employed. Consider an axisymmetric equilibria defined in cylindrical coordinates (r, z, θ) so that $\partial/\partial\theta = 0$.

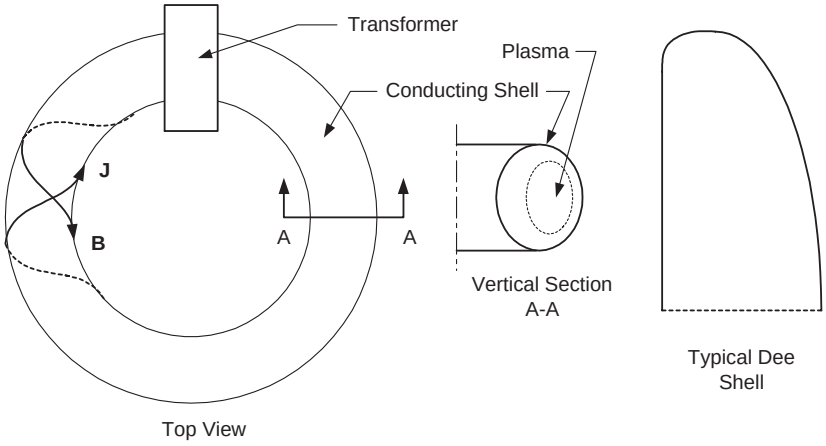

Figure 15.3.1 Schematic of tokamak fields and currents

This implies the existence of a vector potential, **A**, such that curl **A** = **B**. Assuming that **A** = **A**(r, z) and $A_\theta = \psi / r$, where ψ is a stream function, we obtain

$$B_r = -\psi_{,z}/r\,, \quad B_z = \psi_{,r}/r\,, \quad B_\theta = A_{r,z} - A_{z,r} = B_T \tag{15.7}$$

Therefore Eqn. (15.5) simplifies to

$$\frac{\partial^2\psi}{\partial r^2} - \frac{1}{r}\frac{\partial\psi}{\partial r} + \frac{\partial^2\psi}{\partial z^2} = -\mu r^2 P' - X X' = r J_\theta\,, \tag{15.8}$$

where J_θ is the plasma current, P is the pressure, $X = r B_\theta$ and where P and X are functions of ψ alone. Both **J** and **B** are vectors that lie tangent to the surfaces of constant ψ. The above is the governing equation for the steady equilibrium flow of a plasma. For certain simple choices of P and B_θ, Eqn. (15.8) will be linear but in general it is nonlinear. They are usually represented as a series in ψ as

$$P(\psi) = \alpha_0 + \alpha_1\psi + ... + \alpha_n\psi^n/n$$

$$X^2(\psi) = \beta_0 + \beta_1\psi + ... + \beta_n\psi^n/n$$

The essential boundary condition on the limiting surface, Γ_1, is

$$\psi = K + 1/2\, r^2 B_V \quad on\ \Gamma_1$$

where K is a constant and B_V is a superimposed direct current vertical (z) field. On planes of symmetry one also has vanishing normal gradients of ψ, i.e.,

$$\frac{\partial\psi}{\partial n} = 0\ on\ \Gamma_2.$$

The right-hand side of Eqn. (15.8) can often be written as

$$r J_\theta = p\psi + q \tag{15.9}$$

where, for the above special cases, $p = p(r, z)$ and $q = q(r, z)$, but where in general q is a nonlinear function of ψ, i.e., $q = q(r, z, \psi)$. Equations (15.8) and (15.9) are those for which we wish to establish the finite element model.

A finite element formulation of this problem has been presented by Akin and Wooten [1]. They recast Eqn. (15.8) in a self-adjoint form, applied the Galerkin criterion, and integrated by parts. This defines the governing variational statement

$$I = \int\int_\Omega \left[\frac{1}{2}\{(\psi_{,r})^2 + (\psi_{,z})^2 + p\psi^2\} + q\psi\right]\frac{1}{r}\,dr\,dz \tag{15.10}$$

which, for the linear problem, yields Eqn. (15.8) as the Euler equation when I is stationary, i.e., $\delta I = 0$. When $p = q = 0$, Eqn. (15.10) also represents the case of axisymmetric inviscid fluid flow. Flow problems of this type were considered by Chung [3] using a similar procedure. For a typical element with N nodes the element contributions for Eqn. (15.9) are

$$\mathbf{S}^e = \int_{\Omega^e} [\mathbf{H}_{,r}^T\mathbf{H}_{,r} + \mathbf{H}_{,z}^T\mathbf{H}_{,z} + p(r, z)\mathbf{H}^T\mathbf{H}]\frac{1}{r}\,dr\,dz,$$


```
! .............................................................    ! 1
!    ***  ELEM_SQ_MATRIX PROBLEM DEPENDENT STATEMENTS FOLLOW ***   ! 2
! .............................................................    ! 3
!  FOR AXISYMMETRIC MHD PLASMA EQUILIBRIUM,    P,Q  /=  0  I.E.    ! 4
!  L(U) = (U,R*1/R),R + U,ZZ*1/R - P*U/R - Q/R = 0                 ! 5
!  U = STREAM FUNCTION                                            ! 6
                                                                   ! 7
!  P = -C_1 * ALPHA_0 * R**2 - 0.5 * BETA_0, = 0 IF IDEAL PLASMA   ! 8
!  Q = -C_1 * ALPHA_1 * R**2 - 0.5 * BETA_1, = 0 IF IDEAL PLASMA   ! 9
                                                                   !10
! MISC PROPERTIES 1-5 ARE: C_1, ALPHA_0, ALPHA_1, BETA_0, BETA_1   !11
                                                                   !12
 REAL(DP), PARAMETER :: FOUR_PI = TWO_PI * 2.d0                    !13
 REAL(DP)            :: DET_WT, DET, P, Q, R                       !14
 REAL(DP), SAVE      :: ALPHA_0, ALPHA_1, BETA_0, BETA_1           !15
 INTEGER             :: IP, io_1                                   !16
                                                                   !17
!-->  DEFINE ELEMENT PROPERTIES (FIRST 2 TERMS IN POWERS OF U)     !18
  IF ( IE == 1 ) THEN ! first call                                 !19
    ALPHA_0 = GET_REAL_MISC (1)                                    !20
    ALPHA_1 = GET_REAL_MISC (2)                                    !21
    BETA_0  = GET_REAL_MISC (3)                                    !22
    BETA_1  = GET_REAL_MISC (4)                                    !23
  END IF                                                           !24
                                                                   !25
  CALL REAL_IDENTITY (N_R_B, E) !   DEFAULT TO IDENTITY MATRIX     !26
                                                                   !27
!      STORE NUMBER OF POINTS FOR FLUX CALCULATIONS                !28
  CALL STORE_FLUX_POINT_COUNT ! Save LT_QP                         !29
                                                                   !30
!-->   NUMERICAL INTEGRATION LOOP                                  !31
   DO IP = 1, LT_QP                                                !32
     H   = GET_H_AT_QP (IP)   ! EVALUATE INTERPOLATION FUNCTIONS   !33
     XYZ = MATMUL (H, COORD)  ! FIND GLOBAL COORD, (ISOPARAMETRIC) !34
     R   = XYZ (1)            ! CHANGE NOTATION                    !35
     DLH = GET_DLH_AT_QP (IP) ! FIND LOCAL DERIVATIVES             !36
     AJ  = MATMUL (DLH, COORD) ! FIND JACOBIAN AT THE POINT        !37
!         FORM INVERSE AND DETERMINATE OF JACOBIAN                 !38
     CALL INVERT_JACOBIAN (AJ, AJ_INV, DET, N_SPACE)               !39
     DET_WT  = DET * WT(IP) / R                                    !40
                                                                   !41
!         EVALUATE GLOBAL DERIVATIVES, DGH == B                    !42
     DGH = MATMUL (AJ_INV, DLH)                                    !43
     B   = COPY_DGH_INTO_B_MATRIX (DGH) ! B = DGH                  !44
                                                                   !45
!      EVALUATE CONTRIBUTIONS TO SQUARE AND COLUMN MATRICES        !46
     P = - FOUR_PI * R * R * ALPHA_1 - 0.5d0 * BETA_1              !47
     Q = - FOUR_PI * R * R * ALPHA_0 - 0.5d0 * BETA_0              !48
                                                                   !49
     S = S + ( MATMUL ((MATMUL (TRANSPOSE (B), E)), B)   &         !50
             + P * OUTER_PRODUCT (H, H) ) * DET_WT                 !51
     C = C - Q * H * DET_WT                                        !52
                                                                   !53
!-->  SAVE COORDS, E AND DERIVATIVE MATRIX, FOR POST PROCESSING    !54
     CALL STORE_FLUX_POINT_DATA (XYZ, E, B)                        !55
   END DO                                                          !56
                                                                   !57
!    ***  END ELEM_SQ_MATRIX PROBLEM DEPENDENT STATEMENTS ***      !58
```

Figure 15.3.2 Plasma element matrices evaluations

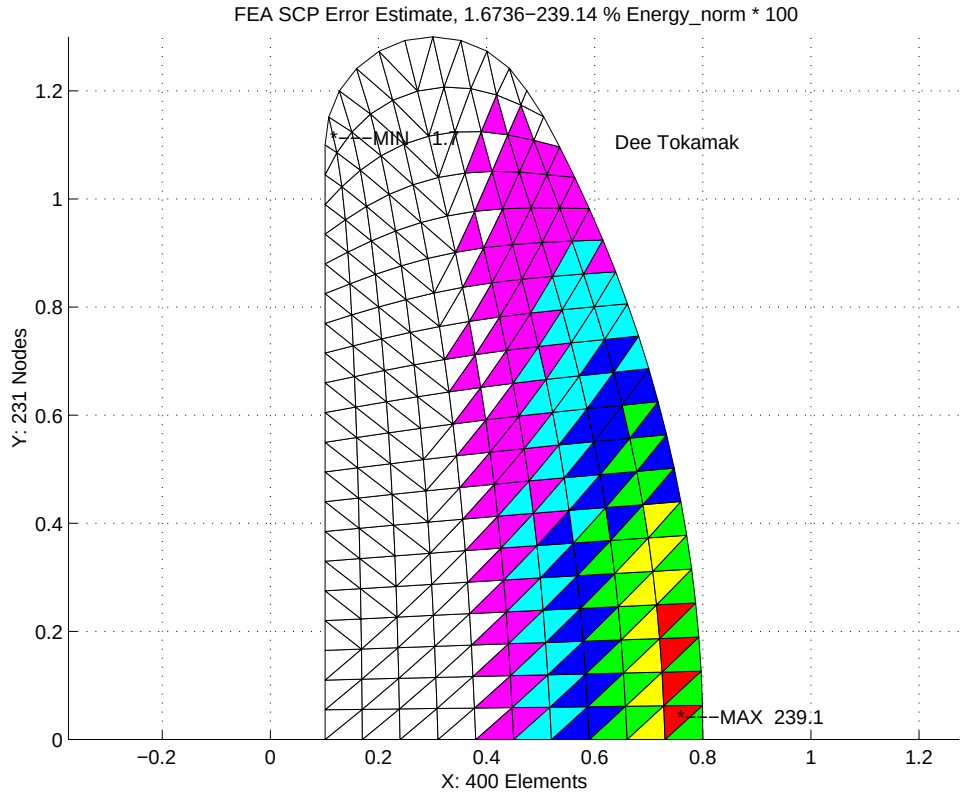

Figure 15.3.3 Initial Dee plasma mesh and error estimates

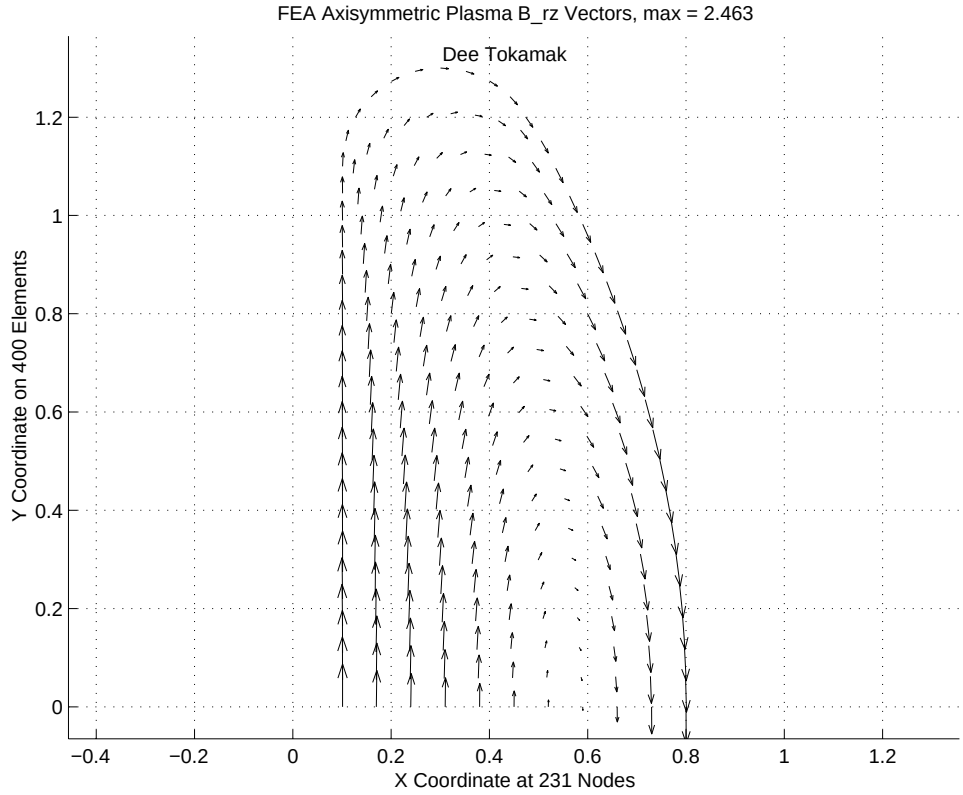

Figure 15.3.4 Initial Dee plasma planar **B** vectors

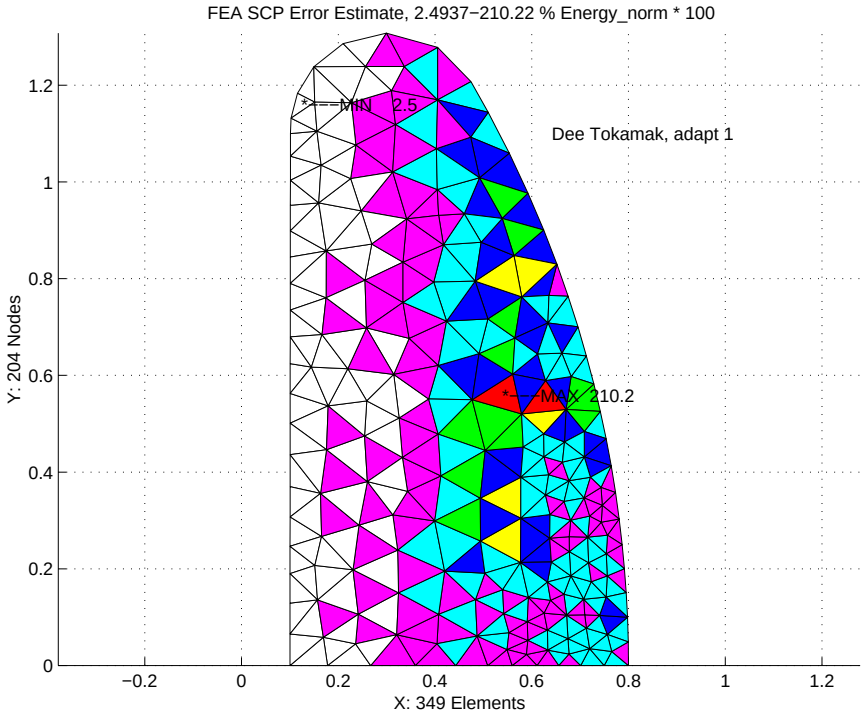

Figure 15.3.5 Plasma error estimates in first adaption

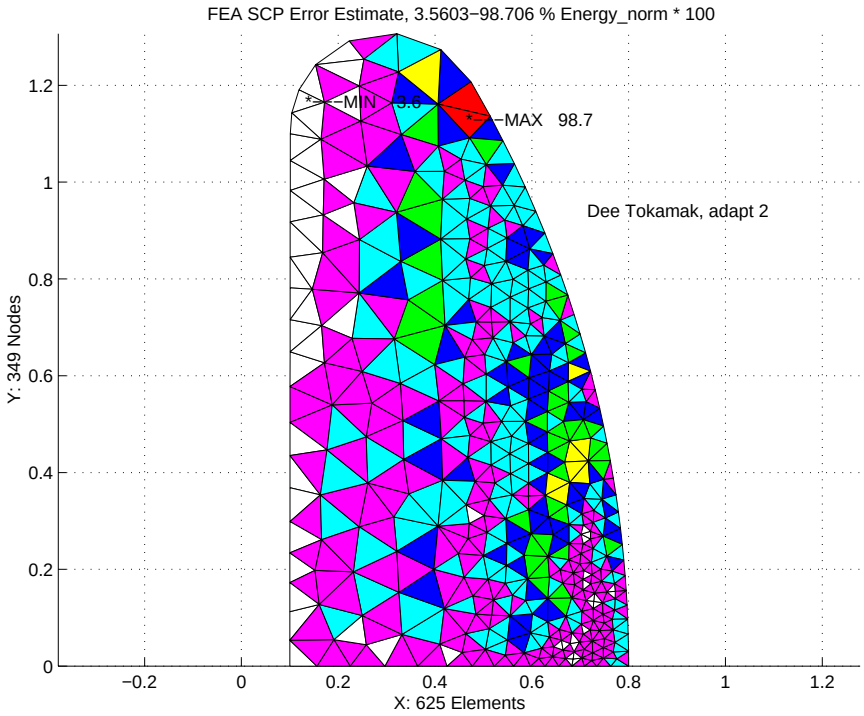

Figure 15.3.6 Plasma error estimates in second adaption

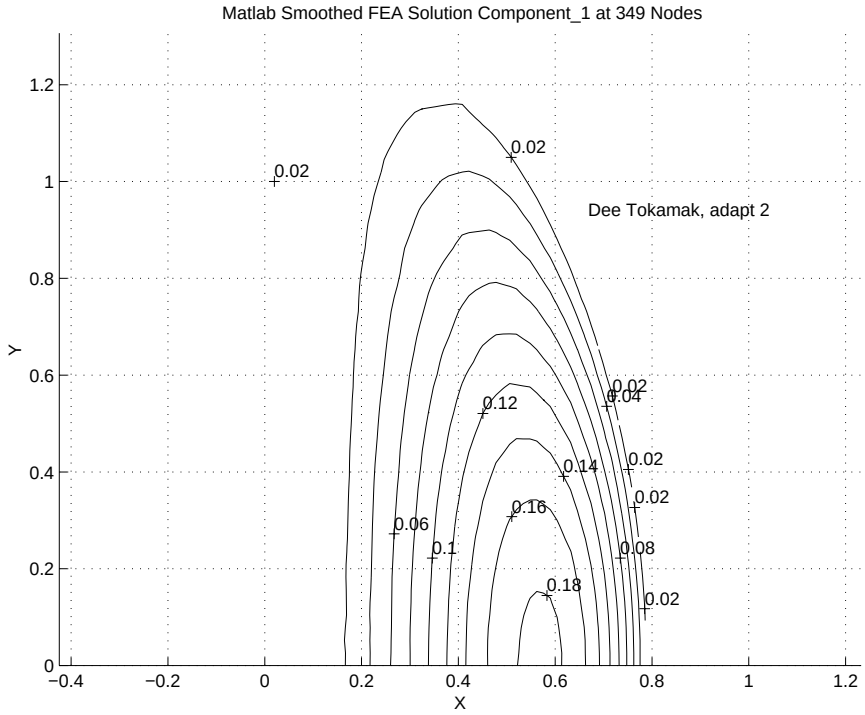

Figure 15.3.7 ψ values in second adaption

$$\mathbf{C}^e = \int_\Omega q(r,\ z)\mathbf{H}^T \frac{1}{r}\, dr\ dz, \tag{15.11}$$

where the N element interpolation functions, $\mathbf{H}$, define the value of ψ within an element by interpolating from its nodal values, ψ^e. Several other applications of this model to plasma equilibria are given by Akin and Wooten [1]. The major advantage of the finite element formulation over other methods such as finite differences is that it allows the plasma physicist to study arbitrary geometries. Some feel that the fabrication of the toroidal field coils may require the use of a circular plasma, while others recommend the use of dee-shaped plasmas. The current model has been applied to both of these geometries and the following figures illustrate typical results for a dee-shape torus cross-section where the linear triangle element was employed. Biquadratic or bicubic elements would be better for some formulations which require post-solution calculations using the first and second derivatives of ψ. Figures 15.3.3, and 4 show the initial uniform mesh error estimates, and plasma **B** vectors, respectively. Note that the initial maximum error in the energy norm is about 2.4 % so adaptive refinements were taken to reduce the maximum error level to less than 1 %. The two adaptive meshes and error estimates are shown in Fig.s 15.3.5 and 6, while the final contours of ψ are in Fig 15.3.7. The initial and final surface plots of ψ are in Figs 15.3.8 and 9. The above results have assumed no external vertical **B** field so ψ was given essential boundary condition values of zero on Γ_1.

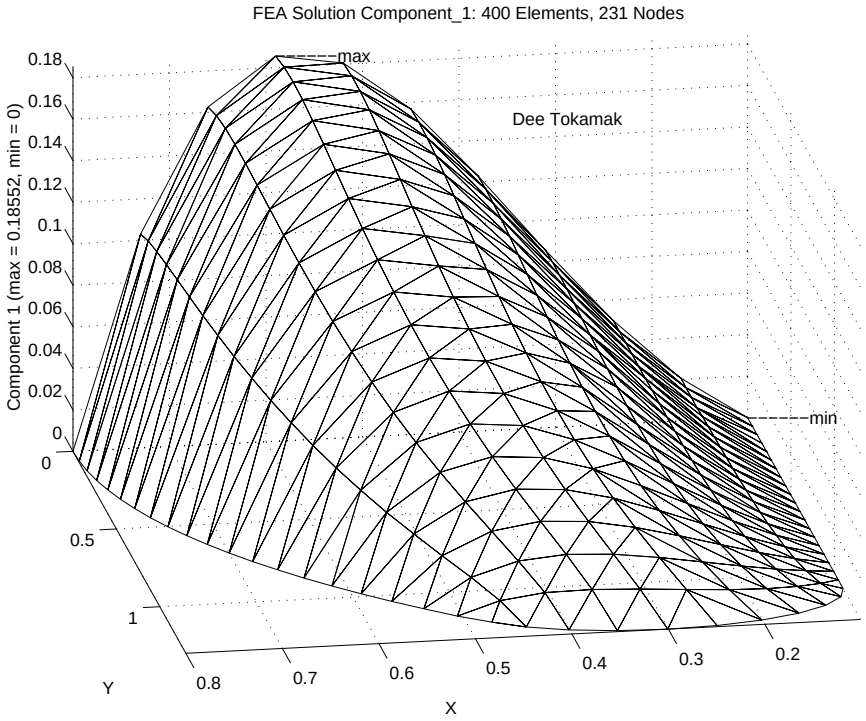

Figure 15.3.8 Initial ψ surface for half symmetry model

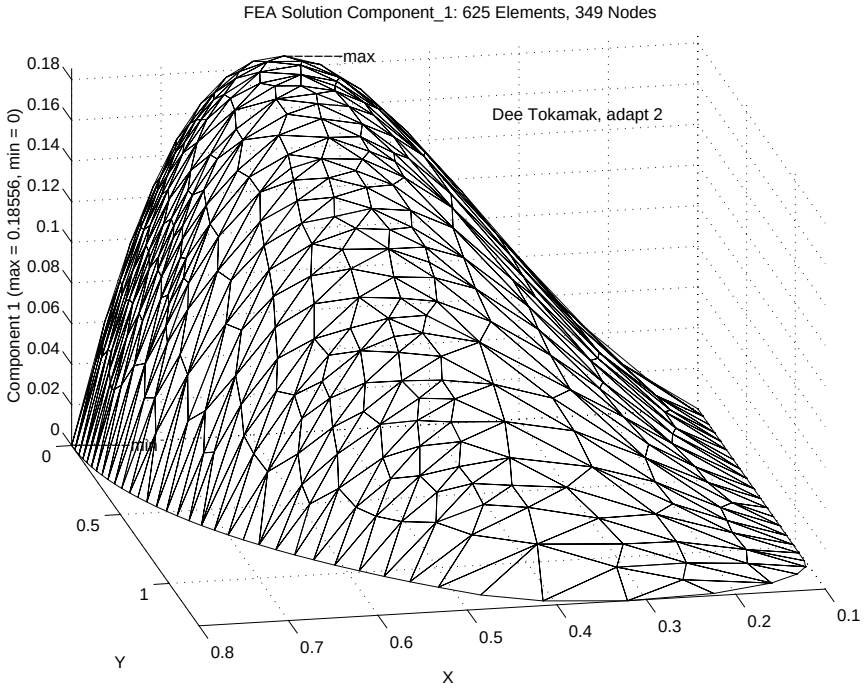

Figure 15.3.9 Final ψ surface for half symmetry model

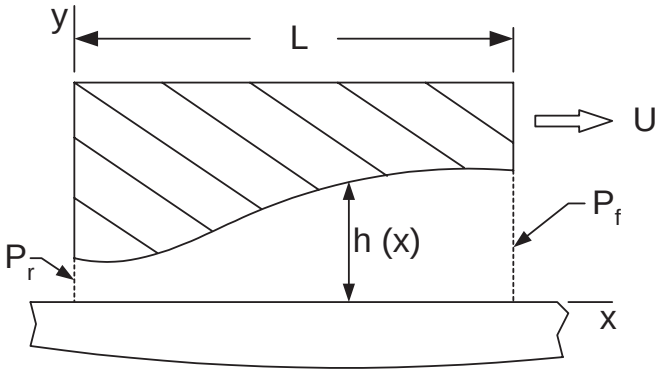

Figure 15.4.1 Thin film slider bearing notation

15.4 Slider Bearing Lubrication

Several references are available on the application of the method to lubrication problems. These include the early work of Reddi [10], a detailed analysis and computer program for the three node triangle by Allan [2], and a presentation of higher order elements by Wada and Hayashi [11]. The most extensive discussion is probably found in the text by Huebner [6]. These formulations are based on the Reynolds equation of lubrication. For simplicity a one-dimensional formulation will be presented here. Consider the slider bearing shown in Fig. 13.4.1, which is assumed to extend to infinity out of the plane of the figure. It consists of a rigid bearing and a slider moving relative to the bearing with a velocity of U. The extremely thin gap between the bearing and the slider is filled with an incompressible lubricant having a viscosity of v. For the one-dimensional case the governing Reynolds equation reduces to

$$\frac{d}{dx}\left(\frac{h^3\,dP}{6v\,dx}\right) = \frac{d}{dx}\,(Uh)\,, \tag{15.11}$$

where $P(x)$ denotes the pressure and $h(x)$ denotes the thickness of the gap. The boundary conditions are that P must equal the known external pressures (usually zero) at the two ends of the bearing. It can be shown that the variational equivalent of the one-dimensional Reynolds equation requires the minimization of the functional

$$I = \int_0^L \left[\frac{h^3}{12v}\left(\frac{dP}{dx}\right)^2 + hU\left(\frac{dP}{dx}\right)\right] dx\,. \tag{15.12}$$

As a word of warning, it should be noted that, while the pressure P is continuous, the film thickness h is often discontinuous at one or more points on the bearing. Another related quantity of interest is the load capacity of the bearing. From statics one finds the resultant normal force per unit length in the z-direction, F_y, is


```
! ..................................................................... ! 1
!    ***  ELEM_SQ_MATRIX PROBLEM DEPENDENT STATEMENTS FOLLOW *** ! 2
! ..................................................................... ! 3
!     APPLICATION: LINEAR SLIDER BEARING,   Example 102          ! 4
! N_SPACE = 1,  NOD_PER_EL = 2, N_G_DOF = 1                      ! 5
! N_EL_FRE = 2, MISC_FL = 2                                      ! 6
! FLT_MISC (1) = VISCOSITY, FLT_MISC (2) = VELOCITY              ! 7
! EL_PROP (1) OR PRT_L_PT (K,1) = FILM THICKNESS                 ! 8
                                                                 ! 9
 INTEGER,   SAVE :: KALL = 1                                     !10
 REAL (DP), SAVE :: VIS, VEL, DL                                 !11
 REAL (DP)       :: THICK, CONST                                 !12
                                                                 !13
   IF ( KALL == 1 ) THEN ! GET GLOBAL REAL CONSTANTS             !14
     KALL = 0                                                    !15
     VIS = GET_REAL_MISC (1) ; VEL = GET_REAL_MISC (2)           !16
   END IF ! FIRST CALL                                           !17
                                                                 !18
!-->   DEFINE ELEMENT LENGTH AND ELEMENT THICKNESS               !19
   DL    = COORD (2, 1) - COORD (1, 1)                           !20
   THICK = 0.d0                                                  !21
   IF ( EL_REAL  > 0 ) THICK = GET_REAL_LP (1)                   !22
                                                                 !23
!      CHECK FOR ALTERNATE AVERAGE NODE THICKNESS                !24
   IF ( THICK == 0.d0 ) THEN  ! USE NODAL PROPERTY               !25
     IF ( N_NP_FLO > 0 ) THEN ! DATA EXISTS                      !26
       THICK = 0.5d0 * (PRT_L_PT (1, 1) + PRT_L_PT (2, 1) )      !27
     ELSE                                                        !28
       STOP 'NO SLIDER BEARING THICKNESS DATA'                   !29
     END IF                                                      !30
   END IF ! NODAL THICKNESS DATA                                 !31
                                                                 !32
!-->  GENERATE ELEMENT SQUARE MATRIX & COLUMN MATRIX             !33
   CONST = THICK**3 / (6.0_DP * VIS * DL)                        !34
   S (1, 1) =  CONST    ; S (2, 2) =  CONST                      !35
   S (1, 2) = -CONST    ; S (2, 1) = -CONST                      !36
   C (1) =  VEL * THICK ; C (2) = -VEL * THICK                   !37
                                                                 !38
!-->   GENERATE DATA FOR LOAD CALCULATIONS AND STORE             !39
   H_INTG (1) = 0.5_DP * DL ; H_INTG (2) = 0.5_DP * DL           !40
   IF ( N_TAPE1 > 0 ) WRITE (N_TAPE1) H_INTG                     !41
                                                                 !42
!    ***  END ELEM_SQ_MATRIX PROBLEM DEPENDENT STATEMENTS ***    !43
```

Figure 15.4.2 Slider bearing square and load matrices


```
! ..................................................................... ! 1
!  *** POST_PROCESS_ELEM PROBLEM DEPENDENT STATEMENTS FOLLOW *** ! 2
! ..................................................................... ! 3
!      APPLICATION: LINEAR SLIDER BEARING, Example 102                ! 4
! N_SPACE = 1,  NOD_PER_EL = 2, N_G_DOF = 1                           ! 5
! N_EL_FRE = 2, MISC_FL = 2                                           ! 6
                                                                      ! 7
 INTEGER,  SAVE :: KALL = 1                                           ! 8
 REAL(DP), SAVE :: FORCE, TOTAL = 0.0_DP                              ! 9
                                                                      !10
   IF ( KALL == 1 ) THEN   !    PRINT TITLES ON THE FIRST CALL        !11
     KALL = 0 ; WRITE (6, 5)                                          !12
     5 FORMAT (/, '***   E L E M E N T   L O A D S  ***',/, &         !13
                  'ELEMENT        LOAD            TOTAL')             !14
   END IF ! FIRST CALL                                                !15
                                                                      !16
!-->   CALCULATE LOADS ON THE ELEMENTS, F = H_INTG*D                  !17
   READ (N_TAPE1) H_INTG            ! RECOVER INTEGRAL OF H           !18
   FORCE = DOT_PRODUCT (H_INTG, D)  ! INTEGRAL OF PRESSURE            !19
   TOTAL = TOTAL + FORCE            ! SYSTEM UPDATE                   !20
   WRITE (6, 10) IE, FORCE, TOTAL   ! LIST RESULTS                    !21
   10 FORMAT (I5, 1PE16.5, 3X, 1PE16.5)                               !22
                                                                      !23
!  *** END POST_PROCESS_ELEM PROBLEM DEPENDENT STATEMENTS ***         !24
```

Figure 15.4.3 Element bearing load calculations

$$F_y = \int_0^L P \, dx \, .$$

This is a quantity which would be included in a typical set of post-solution calculations. As a specific example of a finite element formulation, consider a linear element with two nodes (NOD_PER_EL = 2) and one pressure per node (NG = 1). Thus, $P(x) = \mathbf{H}^e(x)\mathbf{P}^e$ where as before $\mathbf{P}^{eT} = [\, P_i \quad P_j \,]$ and the interplation functions are $\mathbf{H}^e = [\, (x_j - x) \; (x - x_i) \,]/l$, where $l = (\, x_j - x_i \,)$ is the length of the element.

Minimizing the above functional defines the element square and column matrices as

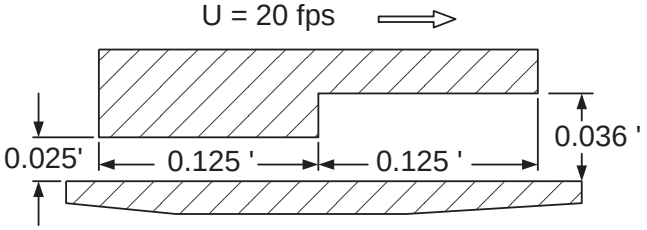

Figure 15.4.4 Step bearing example geometry


```
LINEAR SLIDER BEARING                                    ! 1
                                                         ! 2
NUMBER OF NODAL POINTS IN SYSTEM =.....    3 ! 3
NUMBER OF ELEMENTS IN SYSTEM =.........    2 ! 4
NUMBER OF NODES PER ELEMENT =..........    1 ! 5
NUMBER OF PARAMETERS PER NODE =........    2 ! 6
DIMENSION OF SPACE =...................    1 ! 7
NUMBER OF ITERATIONS TO BE RUN =.......    1 ! 8
NUMBER OF ROWS IN B MATRIX =...........    1 ! 9
ELEMENT SHAPE: LINE, TRI, QUAD, HEX, TET =...    1 !10
NUMBER OF DIFFERENT ELEMENT TYPES =.....    1 !11
STIFFNESS STORAGE MODE: SKY, BAND =.....    1 !12
NUMBER OF REAL PROPERTIES PER ELEMENT =.....    1 !13
NUMBER OF REAL MISCELLANEOUS  PROPERTIES =...    2 !14
OPTIONAL UNIT NUMBERS: N_TAPE1 =  8                      !15
*** NODAL POINT DATA ***                                 !16
NODE, CONSTRAINT FLAG, 1 COORDINATES                     !17
         1         1    0.0000                          !18
         2         0    0.1250                          !19
         3         1    0.2500                          !20
*** ELEMENT CONNECTIVITY DATA ***                        !21
ELEMENT NO., 2 NODAL INCIDENCES.                         !22
    1    1    2                                         !23
    2    2    3                                         !24
***  NODAL PARAMETER CONSTRAINT LIST  ***                !25
TYPE      EQUATIONS                                      !26
   1         2                                          !27
*** CONSTRAINT EQUATION DATA ***                         !28
CONSTRAINT TYPE ONE                                      !29
EQ. NO.   NODE1   PAR1          A1                       !30
      1      1      1   0.00000E+00                      !31
      2      3      1   0.00000E+00                      !32
***  ELEMENT  PROPERTIES   ***                           !33
ELEMENT  PROPERTY      VALUE                             !34
    1         1    2.50000E-02                          !35
    2         1    3.60000E-02                          !36
***  MISCELLANEOUS SYSTEM PROPERTIES  ***                !37
PROPERTY     VALUE                                       !38
       1     2.00000E-03                                 !39
       2     2.00000E+01                                 !40
                                                         !41
***  OUTPUT OF RESULTS  ***                              !42
NODE, 1 COORDINATES, 1 PARAMETERS.                       !43
    1  0.00000E+00  0.00000E+00                          !44
    2  1.25000E-01  5.29857E+00                          !45
    3  2.50000E-01  0.00000E+00                          !46
                                                         !47
***   E L E M E N T   L O A D S  ***                     !48
ELEMENT        LOAD            TOTAL                     !49
    1     3.31160E-01        3.31160E-01                 !50
    2     3.31160E-01        6.62321E-01                 !51
                                                         !52
*** EXTREME VALUES OF THE NODAL PARAMETERS ***           !53
PARAMETER    MAXIMUM, NODE      MINIMUM, NODE            !54
      1   5.2986E+00,   2    0.0000E+00,   3            !55
```

Figure 15.4.5 Slider bearing results

$$\mathbf{S}^e = \frac{1}{6v}\int_{x_i}^{x_j} h^3 \mathbf{H}_{,x}^{eT}\,\mathbf{H}_{,x}^{e}\,dx\,,\quad \mathbf{C}^e = -U\int_{x_i}^{x_j} h\,\mathbf{H}_{,x}^{eT}\,dx\,.$$

For the element under consideration $\mathbf{H}^e$ is linear in x so that its first derivative will be constant. That is, $\mathbf{H}_{,x}^{e} = [-1 \quad 1]/l$ so that the element matrices simplify to

$$\mathbf{S}^e = \frac{1}{6vl^2}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\int_{x_i}^{x_j} h^3\,dx \tag{15.13}$$

$$\mathbf{C}^e = \frac{U}{l}\begin{Bmatrix} 1 \\ -1 \end{Bmatrix}\int_{x_i}^{x_j} h\,dx\,.$$

Thus, it is clear that the assumed thickness variation within the element has an important effect on the complexity of the element matrices. It should also be clear that the nodal points of the mesh must be located such that any discontinuity in h occurs at a node. The simplest assumption is that h is constant over the length of the element. In this case the latter two integrals reduce to $h^3 l$ and $h\,l$, respectively. One may wish to utilize this element to approximate a varying distribution of h by a series of constant steps. In this case, one could use an average thickness of $h = (h_i + h_j)/2$, where h_i and h_j denote the thickness at the nodal points of the element. Subroutine ELEM_SQ_MATRIX for this element is shown in Figs. 15.4.2. Note that it allows for two methods of defining the film thickness, h, in each element. In the default option (keywords el_real 1 , pt_real 0) the value of h is input as a floating point element property, i.e., H = GET_REAL_LP (1). In the second option (keywords el_real 0 , pt_real 1) the thickness is specific at each node as a floating point property. Note that for the latter option, el_real 0, causes h = 0. The program checks for this occurrence and then skips to the second definition of h.

The program for performing the post-solution calculations, POST_PROCESS_ELEM, is shown in Figs. 15.4.3. It evaluates the force, F_y^e, carried by each element. The load on a typical element is $F_y^e = \mathbf{Q}^e\mathbf{P}^e$ where $\mathbf{Q}^e = [l/2 \quad l/2]$. Subroutine ELEM_SQ_MATRIX also generates and stores $\mathbf{Q}^e$ for each element. Subroutine POST_PROCESS_ELEM carries out the multiplication once the nodal pressures, $\mathbf{P}^e$, are known. In addition, it sums the force on each element to obtain the total load capacity of the bearing. It prints the element number and its load and the total load on the bearing. With the addition of a few extra post-solution calculations, one could also output the location of the resultant bearing force. In closing, recall that both U and v are constant along the entire length of the bearing. Thus, they are simply defined as floating point miscellaneous system properties (keyword reals 2).

As a numerical example consider the step bearing shown in Fig. 15.4.6, which has two constants gaps of different thicknesses but equal lengths. Select a mesh with three nodes (nodes 3) and two elements (elems 2). Let $l_1 = l_2 = 0.125$ ft, $U = 20$ ft/s, $v = 0.002$ lb s/ft^2, $h_1 = 0.025$ ft, and $h_2 = 0.036$ ft. The two boundary conditions are $P_1 = P_3 = 0$ and we desire to calculate the pressure, P_2, at the step. The calculated pressure is $P_2 = 5.299$ psf, which is the exact value, and the total force on the bearing is $F_y = 0.66$ ppf. The accuracy is not surprising since the exact solution for this problem gives a linear pressure variation over each of the two segments of the bearing. The

typical output data are shown in Fig. 15.4.5.

15.5 Exercises

1. Sketch the boundary conditions for the ideal fluid flow around a cylinder given in Fig. 15.2.2 for an approach based on the use of a) the velocity potential, b) the stream function.

15.6 References

[1] Akin, J.E. and Wooten, J.W., "Tokamak Plasma Equilibria by Finite Elements," in *Finite Elements in Fluids III*, Chapter 21, ed. R.H. Gallagher, New York: John Wiley (1978).

[2] Allan, T., "The Application of Finite Element Analysis to Hydrodynamic and Externally Pressurized Pocket Bearings," *Wear*, **19**, pp. 169−206 (1972).

[3] Chung, T.J., *Finite Element Analysis in Fluid Dynamics*, New York: McGraw-Hill (1978).

[4] Cook, R.D., Malkus, D.S., Plesha, N.E., and Witt, R.J., *Concepts and Applications of Finite Element Analysis*, New York: John Wiley (2002).

[5] Desai, C.S. and Abel, J.F., *Introduction to the Finite Element Method*, New York: Van Nostrand-Reinhold (1972).

[6] Huebner, K.H., Thornton, E.A., and Byrom, T.G., *Finite Element Method for Engineers*, New York: John Wiley (1994).

[7] Irons, B.M. and Razzaque, A., "Experience with the Patch Test for Convergence of the Finite Element Method," pp. 557−587 in *Mathematical Foundation of the Finite Element Method*, ed. A.R. Aziz, New York: Academic Press (1972).

[8] Irons, B.M. and Ahmad, S., *Techniques of Finite Elements*, New York: John Wiley (1980).

[9] Martin, H.C. and Carey, G.F., *Introduction to Finite Element Analysis*, New York: McGraw-Hill (1974).

[10] Reddi, M.M., "Finite Element Solution of the Incompressible Lubrication Problem," *J. Lubrication Technology*, **53**(3), pp. 524−532 (July 1969).

[11] Wada, S. and Hayashi, H., "Application of Finite Element Method to Hydrodynamic Lubrication Problems," *Bulletin of Japanese Soc. Mech. Eng.*, **14**(77), pp. 1222−1244 (1971).

