Chapter 4

Two Dimensional Elastostatics

4.1 Introduction

This chapter deals with the applications of boundary elements to solve two dimensional elastostatics problems. The basic equations of elasticity are reviewed, first pointing out that boundary elements for these cases are based on the plane strain approach but can be extended to plane stress if the elastic coefficients are replaced by the corresponding equivalent values, as will be seen below.

The boundary element formulation of elastostatics is substantially different from the one for potential problems as its unknowns are in vector rather than scalar form. This means that at a particular point there are two (or three in three dimensions) components of displacements or tractions, rather than one of each as discussed in Chapter 3. The associated fundamental solution also represents a point load acting in a given direction. All this implies that the boundary element formulation for elastostatics is more complex than the one described in Chapter 2 for potential problems although it involves the same basic steps.

It seems appropriate to start by developing the necessary theory and then the computer code corresponding to the simplest case, i.e. the one with constant elements. After this the development of a quadrilateral element and associated code is described, without presenting the linear case in detail. This approach has been preferred as the linear case has already been discussed in Chapter 2 for potential problems and is comparatively simple to extend the formulation to elastostatics. The quadratic elasticity code is also more interesting from the point of view of applications, as many engineering problems are difficult to solve accurately using constant elements and the linear codes do not converge rapidly for cases involving bending, for instance. Hence while constant or linear elements can be satisfactorily applied for many potential problems they – particularly the constant ones – are generally not sufficiently accurate for stress analysis applications.

The chapter presents a series of examples describing the excellent results which can be obtained by using boundary elements in elastostatics. This chapter in conjunction with the previous one attempts to be an introduction to the use of boundary elements for stress analysis. For further developments and more complex applications the reader is advised to consult the relevant publications listed in Appendix C.

4.2 Plate Stretching – Plane Strain Problems

Two dimensional problems are divided into two types, plate stretching (sometimes

also called plane stress) and plane strain problems, depending on how the solid is restrained in the direction perpendicular to the plane under study.

To understand the difference between these two states consider the prismatic homogeneous solid shown in figure 4.1. The end surfaces are defined by the planes at $x_3 = \pm h/2$, and the cylindrical surface by $x_1 = x_1(\Gamma)$, $x_2 = x_2(\Gamma)$, where Γ is the arc length along the boundary curve.

Plate stretching The basic assumptions for plate stretching are

(i) that the body is thin, i.e. h is small by comparison with the representative dimensions along x_1 or x_2,
(ii) there are no tractions acting at the end surfaces, i.e. at $x_3 = \pm h/2$, $p_j = 0$,
(iii) the body forces are acting on $x_1 - x_2$ planes and independent of x_3, i.e. $b_3 = 0$ and b_1, b_2 are functions of $x_1 x_2$ only,
(iv) the forces acting on the cylindrical body are planar and independent of x_3, i.e. $p_3 = 0$ and $p_1 p_2$ are functions of x_1 and x_2.

Under these assumptions it is assumed that σ_{33}, σ_{31} and σ_{32} are all small in comparison with $\sigma_{11}\sigma_{22}\sigma_{12}$ and that the variation of the latter with respect to x_3 is negligible. Hence one assumes:

$$\sigma_{33} = \sigma_{31} = \sigma_{32} = 0 \tag{4.1}$$

and σ_{11}, σ_{22} and σ_{12} are functions of x_1, x_2 only.

$$\sigma_{11}(x_1, x_2); \qquad \sigma_{22}(x_1, x_2); \qquad \sigma_{12}(x_1, x_2)$$

It should be noticed however that although these assumptions are reasonable in engineering practice they are only approximate as they violate the compatibility equations.

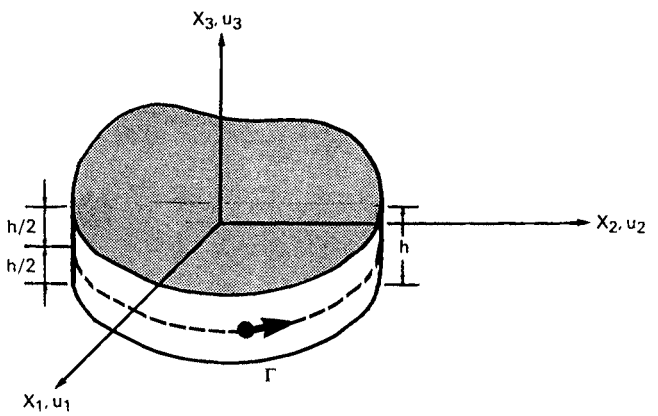


Figure 4.1 Prismatic solid

Equation (4.1) is written in function of displacements and one can have an alternative statement, i.e. the displacement components are functions of x_1x_2 only, i.e.

$$u_1 = u_1(x_1, x_2); \qquad u_2 = u_2(x_1, x_2) \tag{4.2}$$

Notice that $u_3 \neq 0$ and the ε_{33} strains can be determined in function of the σ_{11}, σ_{22}, σ_{12} stresses.

Plane Strain This case usually represents the behaviour of long structures such as tunnels and for this the displacements in the normal direction can be assumed to be zero (i.e. on the end faces).

The the plane strain assumptions are

(ii) The end faces displacements u_3 are zero as they are restrained to move normally because the thickness is large in comparison with the representative dimensions in x_1x_2 directions.

(ii) The body and surface forces acting on the cylindrical surface have no x_3 component and are independent of x_3.

In this case in addition to $u_3 = 0$, one can assume that the in-plane displacements u_1 and u_2 are independent of x_3. Hence

$$u_1 = u_1(x_1, x_2); \qquad u_2 = u_2(x_1, x_2); \qquad u_3 = 0 \tag{4.3}$$

This means that some of the resulting strains will also be zero, i.e.

$$\varepsilon_{33} = \varepsilon_{31} = \varepsilon_{32} = 0 \tag{4.4}$$

and the others independent of x_3, i.e. $\varepsilon_{11}(x_1, x_2)$, $\varepsilon_{22}(x_1, x_2)$, $\varepsilon_{12}(x_1, x_2)$.

For this case $\sigma_{33} \neq 0$ and can be determined from the value of the other components.

Constitutive Relations

One can now expand the three dimensional stress-strain relationships for an isotropic body in terms of the non-zero components for plane strain or plane stress, i.e.

$$\begin{aligned}
\varepsilon_{11} &= \frac{1}{E}(\sigma_{11} - \nu\sigma_{22} - \nu\sigma_{33}) \\
\varepsilon_{22} &= \frac{1}{E}(-\nu\sigma_{11} + \sigma_{22} - \nu\sigma_{33}) \\
\varepsilon_{33} &= \frac{1}{E}(-\nu\sigma_{11} - \nu\sigma_{22} + \sigma_{33}) \\
\varepsilon_{12} &= \frac{1}{2G}\sigma_{12}
\end{aligned} \tag{4.5}$$

The relationships for plate stretching can be obtained by setting $\sigma_{33} \equiv 0$ in (4.5) and considering only the planar components $\varepsilon_{11}, \varepsilon_{22}$ and ε_{12}, i.e.

$$\begin{aligned} \varepsilon_{11} &= \frac{1}{E}(\sigma_{11} - \nu\sigma_{22}) \\ \varepsilon_{22} &= \frac{1}{E}(-\nu\sigma_{11} + \sigma_{22}) \\ \varepsilon_{12} &= \frac{1}{2G}\sigma_{12} \end{aligned} \tag{4.6}$$

The value of ε_{33} can be obtained a posteriori from the third relationship in (4.5), i.e.

$$\varepsilon_{33} = \frac{1}{E}(-\nu\sigma_{11} - \nu\sigma_{22}) \tag{4.7}$$

Equations (4.6) can be inverted to produce,

$$\begin{aligned} \sigma_{11} &= \frac{E}{1-\nu^2}(\varepsilon_{11} + \nu\varepsilon_{22}) \\ \sigma_{22} &= \frac{E}{1-\nu^2}(\nu\varepsilon_{11} + \varepsilon_{22}) \\ \sigma_{12} &= 2G\varepsilon_{12} \end{aligned} \tag{4.8}$$

The plane strain equations can be found by first eliminating ε_{33} from (4.5), i.e.

$$\varepsilon_{33} = \frac{1}{E}(-\nu\sigma_{11} - \nu\sigma_{22} + \sigma_{33}) \equiv 0 \tag{4.9}$$

which gives

$$\sigma_{33} = \nu(\sigma_1 + \sigma_{22}) \tag{4.10}$$

Substituting for σ_{33} in equation (4.5) one obtains the following expressions,

$$\begin{aligned} \varepsilon_{11} &= \frac{1}{E}[(1-\nu^2)\sigma_{11} - \nu(1+\nu)\sigma_{22}] \\ \varepsilon_{22} &= \frac{1}{E}[-\nu(1+\nu)\sigma_{11} + (1-\nu^2)\sigma_{22}] \\ \varepsilon_{12} &= \frac{1}{2G}\sigma_{12} \end{aligned} \tag{4.11}$$

The inverse of these expressions is

$$\sigma_{11} = \frac{E}{(1+\nu)(1-2\nu)}[(1-\nu)\varepsilon_{11} + \nu\varepsilon_{22}]$$
$$\sigma_{22} = \frac{E}{(1+\nu)(1-2\nu)}[\nu\varepsilon_{11} + (1-\nu)\varepsilon_{22}] \tag{4.12}$$
$$\sigma_{12} = 2G\varepsilon_{12}$$

The value of ε_{33} if required can be obtained from (4.10).

Notice that it is sometimes more convenient to express the first two relationships above using Lamé's constant $\lambda = \nu E/[(1+\nu)(1-2\nu)]$, i.e.

$$\sigma_{11} = \frac{\lambda}{\nu}[(1-\nu)\varepsilon_{11} + \nu\varepsilon_{22}]$$
$$\sigma_{22} = \frac{\lambda}{\nu}[(1-\nu)\varepsilon_{22} + \nu\varepsilon_{11}] \tag{4.13}$$

One can pass from equations (4.8) for plate stretching to equations (4.12) for plane strain simply by replacing E and ν in the first equation by two equivalent values, E' and ν', such that

$$E' = \frac{E}{1-\nu^2}; \qquad \nu' = \frac{\nu}{1-\nu} \tag{4.14}$$

where the value of G remains the same.

This interesting relationship means that one can implement a plate stretching program and by transforming the elastic constant data in accordance to (4.14) solve also a plane strain problem. This is done in finite element analysis. Conversely in boundary elements one works with plane strain problems as the fundamental solution is known for this type of problem and then plate stretching problems can be solved using the inverse relationship to (4.14), i.e.

$$E' = (1-\nu^2)E; \qquad \nu' = \frac{\nu}{1+\nu} \tag{4.15}$$

4.3 Boundary Element Formulation

The basic relationships for boundary elements in elastostatics have been developed in Chapter 3. In what follows some of the corresponding equations will be reviewed and specialized for two dimensional cases with a view to applying them in two computer codes, one using constant element and the other quadratic elements, the latter more appropriate for elasticity problems. The development of an

elastostatics linear code if required, is left to the reader as it follows similar lines as the linear code for potential problems.

One can start with the integral expression (3.79), i.e.

$$c^i_{lk}u^i_k + \int_\Gamma u_k p^*_{lk}\, d\Gamma = \int_\Gamma p_k u^*_{lk}\, d\Gamma + \int_\Omega b_k u^*_{lk}\, d\Omega \tag{4.16}$$

where the fundamental solution has been assumed to satisfy the following equation,

$$\sigma^*_{jk,j} + \Delta^i = 0 \tag{4.17}$$

Equation (4.16) applies for points on the boundary or internal points (with $c^i_{lk} = \delta_{lk}$). Smooth surfaces give $c^i_{lk} = \frac{1}{2}\delta_{lk}$ and corners produce a different type of c^i_{lk} array as will be discussed shortly.

The fundamental solution for an isotropic material in plane strain has been given in Chapter 3 but will be repeated here for completeness.

$$\begin{aligned} u^*_{lk} &= \frac{1}{8\pi G(1-\nu)}\left[(3-4\nu)\ln\left(\frac{1}{r}\right)\delta_{lk} + \frac{\partial r}{\partial x_l}\frac{\partial r}{\partial x_k}\right] \\ p^*_{lk} &= -\frac{1}{4\pi(1-\nu)r}\left[\frac{\partial r}{\partial n}\left\{(1-2\nu)\delta_{kl} + 2\frac{\partial r}{\partial x_k}\frac{\partial r}{\partial x_l}\right\}\right. \\ &\qquad \left. -(1-2\nu)\left(\frac{\partial r}{\partial x_l}n_k - \frac{\partial r}{\partial x_k}n_l\right)\right] \end{aligned} \tag{4.18}$$

p^*_{lk} and u^*_{lk} represent the tractions and displacements in the k direction due to a unit load in the l direction acting at 'i'.

One can now write equation (4.16) in matrix form by defining the following arrays. The fundamental solution components can be written as two 2×2 matrices with elements u^*_{lk} and p^*_{lk}, i.e.

$$\mathbf{u}^* = \begin{bmatrix} u^*_{11} & u^*_{12} \\ u^*_{21} & u^*_{22} \end{bmatrix}; \qquad \mathbf{p}^* = \begin{bmatrix} p^*_{11} & p^*_{12} \\ p^*_{21} & p^*_{22} \end{bmatrix} \tag{4.19}$$

The displacement, tractions and body forces vectors are

$$\mathbf{u} = \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix}; \qquad \mathbf{p} = \begin{Bmatrix} p_1 \\ p_2 \end{Bmatrix}; \qquad \mathbf{b} = \begin{Bmatrix} b_1 \\ b_2 \end{Bmatrix} \tag{4.20}$$

Hence the basic equation becomes,

$$\mathbf{c}^i\mathbf{u}^i + \int_\Gamma \mathbf{p}^*\mathbf{u}\, d\Gamma = \int_\Gamma \mathbf{u}^*\mathbf{p}\, d\Gamma + \int_\Omega \mathbf{u}^*\mathbf{b}\, d\Omega \tag{4.21}$$

where the $\mathbf{u}^i$ defines the displacements at the - internal or boundary – point i

where the load is applied. $\mathbf{c}^i$ is a 2×2 array of constant which values depend on the type of point under consideration. If 'i' is an internal point

$$\mathbf{c}^i = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{4.22}$$

If 'i' is a boundary point on a smooth surface then,

$$\mathbf{c}^i = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \tag{4.23}$$

If 'i' is a corner, we will have

$$\mathbf{c}^i = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \tag{4.24}$$

where the c_{ik} values will depend on the type of corner under consideration as we will see shortly.

4.4 Constant Element Formulation

Consider now that the surface of the boundary under study is discretized using constant elements (figure 4.2). This implies that the values of $\mathbf{u}$ and $\mathbf{p}$ are assumed to be constant on each element and equal to the value at the mid-node of the element. One can also discretize the interiors of the domain in a number of cells which are required for the integration of the body force term in (4.21). These cells are used only for numerical integration of the body force terms and in certain cases – as seen in Chapter 3 – they can be taken to the boundary. Consider here that the body Ω is discretized into N boundary elements and M internal cells, hence formula (4.21) can be written as,

$$\mathbf{c}^i\mathbf{u}^i + \sum_{j=1}^{N} \left\{ \int_{\Gamma_j} \mathbf{p}^* \, d\Gamma \right\} \mathbf{u}^j = \sum_{j=1}^{N} \left\{ \int_{\Gamma_j} \mathbf{u}^* \, d\Gamma \right\} \mathbf{p}^j + \sum_{s=1}^{M} \left\{ \int_{\Omega_s} \mathbf{u}^* \mathbf{b} \, d\Omega \right\} \tag{4.25}$$

The above equation corresponds to the particular node 'i' where the unit forces are assumed to be acting.

Notice that terms such as $\int_{\Gamma_j} \mathbf{u} \, d\Gamma$ and $\int_{\Gamma_j} \mathbf{p} \, d\Gamma$ relate node 'i' to the element or node 'j'. They produce a type of influence coefficient. After integration the integrals produce two 2×2 submatrices called $\hat{\mathbf{H}}^{ij}$ and $\mathbf{G}^{ij}$. Numerical integration of the body forces term can be carried out as follows,

$$\sum_{s=1}^{M} \left\{ \int_{\Omega_s} \mathbf{u}^* \mathbf{b} \, d\Omega \right\} = \sum_{s=1}^{M} \left\{ \sum_{p=1}^{I} (\mathbf{u}^* \mathbf{b})_p w_p \right\} \Omega_s = \sum_{s=1}^{M} \mathbf{B}^{is} \tag{4.26}$$

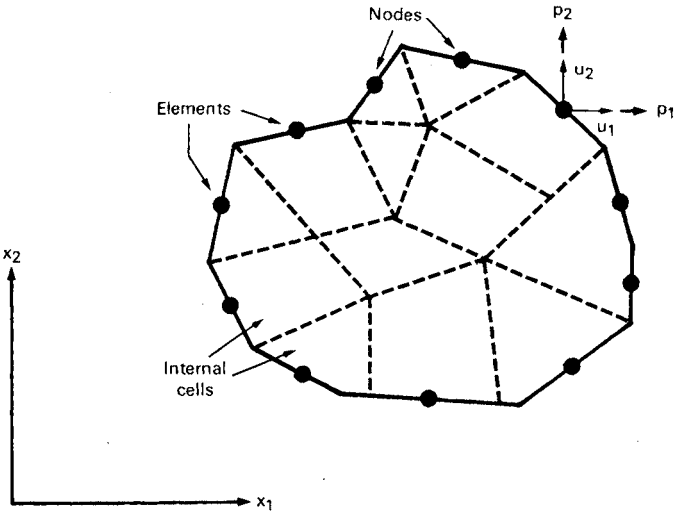


Figure 4.2 Two-dimensional body divided into boundary elements and internal cells

Notice that this produces two components of $\mathbf{B}^{is}$, i.e. B_1^{is} and B_2^{is} after the numerical integration has taken place. w_p are the weighting coefficients and Ω_s the area of the cell under consideration. The function (**u***b) has to be calculated at the p integration points, where p varies from 1 to r.

Equation (4.25) can now be written

$$\mathbf{c}^i\mathbf{u}^i + \sum_{j=1}^{N} \hat{\mathbf{H}}^{ij}\mathbf{u}^j = \sum_{j=1}^{N} \mathbf{G}^{ij}\mathbf{p}^j + \sum_{s=1}^{M} \mathbf{B}^{is} \tag{4.27}$$

This equation relates the values of **u** at node 'i' with the values of **u**'s and **p**'s at all the nodes on the boundary, including 'i'. Notice that in this case – smooth boundary – $\mathbf{c}^i$ is a 2×2 matrix with only $\frac{1}{2}$ on the diagonal.

Equation (4.27) can be written in a more compact manner if we define

$$\begin{aligned} \mathbf{H}^{ij} &= \hat{\mathbf{H}}^{ij} \qquad \text{when } i \neq j \\ \mathbf{H}^{ij} &= \hat{\mathbf{H}}^{ij} + \mathbf{c}^i \qquad \text{when } i = j \end{aligned} \tag{4.28}$$

Formula (4.27) has then the following form

$$\sum_{j=1}^{N} \mathbf{H}^{ij}\mathbf{u}^j = \sum_{j=1}^{N} \mathbf{G}^{ij}\mathbf{p}^j + \sum_{s=1}^{M} \mathbf{B}^{is} \tag{4.29}$$

If one applies (4.29) to all boundary points the result can also be written in matrix form, i.e.

$$\mathbf{HU} = \mathbf{GP} + \mathbf{B} \tag{4.30}$$

where **H** and **G** are $2N \times 2N$ matrices (N number of boundary nodes).

Equation (4.30) has to be rearranged when applying the boundary conditions. The process consists of moving to the left hand side all columns multiplied by an unknown and accumulating on the right hand side vector **F** all the values obtained by multiplying the known boundary conditions by the terms in the corresponding columns. This produces the following system of equations:

$$\mathbf{AX} = \mathbf{F} + \mathbf{B} \tag{4.31}$$

The vector **X** represents all unknowns – displacements or tractions – in the problem. Once (4.31) is solved all boundary values are found.

Results at Internal Points

Once the values of displacements and tractions are known on the boundary it is possible to calculate the displacements and stresses at any interior point. The displacements are given by formula (4.25) with $\mathbf{c}^i = \mathbf{I}$ (**I** is unit diagonal matrix), i.e.

$$\mathbf{u}^i = \int_\Gamma \mathbf{u}^* \mathbf{p}\, d\Gamma - \int_\Gamma \mathbf{p}^* \mathbf{u}\, d\Gamma + \int_\Omega \mathbf{u}^* \mathbf{b}\, d\Omega \tag{4.32}$$

This expression can be discretized

$$\mathbf{u}^i = \sum_{j=1}^{N} \left\{ \int_{\Gamma_j} \mathbf{u}^*\, d\Gamma \right\} \mathbf{p}^j - \sum_{j=1}^{N} \left\{ \int_{\Gamma_j} \mathbf{p}^*\, d\Gamma \right\} \mathbf{u}^j + \sum_{s=1}^{M} \left\{ \int_{\Omega_s} \mathbf{u}^* \mathbf{b}\, d\Omega \right\} \tag{4.33}$$

and then integrated numerically, analytically or by a combination of both techniques.

The internal stresses can be found using the formula (3.109), i.e.

$$\sigma_{kl} = \int_\Gamma \mathbf{D}_{kl} \mathbf{p}\, d\Gamma - \int_\Gamma \mathbf{S}_{kl} \mathbf{u}\, d\Gamma + \int_\Omega \mathbf{D}_{kl} \mathbf{b}\, d\Omega \tag{4.34}$$

where

$$\mathbf{D}_{kl} = [D_{1kl}, D_{2kl}]$$

$$\mathbf{S}_{kl} = [S_{1kl}, S_{2kl}]$$

Formula (4.34) can be written in discretized form as follows,

$$\sigma_{kl} = \sum_{j=1}^{N} \left\{ \int_{\Gamma_j} \mathbf{D}_{kl}\, d\Gamma \right\} \mathbf{p}^j - \sum_{j=1}^{N} \left\{ \int_{\Gamma_j} \mathbf{S}_{kl}\, d\Gamma \right\} \mathbf{u}^j + \sum_{s=1}^{M} \left\{ \int_{\Omega_s} \mathbf{D}_{kl} \mathbf{b}\, d\Omega \right\} \tag{4.35}$$

Integration

As we will see for the case of the quadratic elements, all integrals in the above expressions can be done numerically. For the case of constant elements however

it is simpler and more exact to carry out some integrations analytically, particularly those over the element with the singularity, i.e. for the case $i=j$. Everywhere else the values of the integrals in $\mathbf{H}^{ij}$ and $\mathbf{G}^{ij}$ have been computed using a four-point Gauss quadrature formula. Notice that values of the submatrices $\mathbf{H}^{ij}$ (for $i=j$) are easy to calculate using rigid body considerations as shown in 3.5. The terms in $\mathbf{G}^{ii}$ are the only coefficients to be computed analytically. The integrals are carried out for all the elements in $\mathbf{G}^{ii}$, i.e.

$$\mathbf{G}^{ii}=\begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix}^{ii} \tag{4.36}$$

Substituting the fundamental solution into the corresponding integrals one can define three terms as,

$$G_{11}=\frac{1}{8\pi\mu(1-\nu)}\left[(3-4\nu)\int_{\Gamma_i}\ln\left(\frac{1}{r}\right)d\Gamma+\int_{\Gamma_i}\left(\frac{\partial r}{\partial x_1}\right)^2 d\Gamma\right] \tag{4.37}$$

$$G_{12}=G_{21}=\frac{1}{8\pi\mu(1-\nu)}\left[\int_{\Gamma_i}\frac{\partial r}{\partial x_1}\frac{\partial r}{\partial x_2}\,d\Gamma\right] \tag{4.38}$$

$$G_{22}=\frac{1}{8\pi\mu(1-\nu)}\left[(3-4\nu)\int_{\Gamma_i}\ln\left(\frac{1}{r}\right)d\Gamma+\int_{\Gamma_i}\left(\frac{\partial r}{\partial x_2}\right)^2 d\Gamma\right] \tag{4.39}$$

Notice that Γ_i refers to element 'i' over which the singularity is acting. The derivatives of r in formulae (4.37) to (4.39) are given, for the general case, by (figure 4.3(a)).

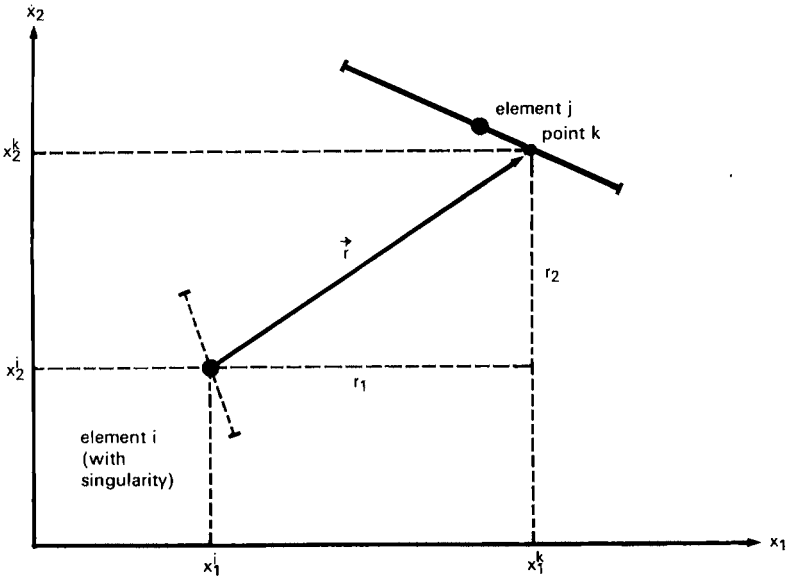
$$\begin{aligned}\frac{\partial r}{\partial x_1}&=\frac{r_1}{r}=\frac{x_1^k-x_1^i}{|r|}\\ \frac{\partial r}{\partial x_2}&=\frac{r_2}{r}=\frac{x_2^k-x_2^i}{|r|}\end{aligned} \tag{4.40}$$

where $|r|$ is the magnitude of the distance vector $\vec{r}$.

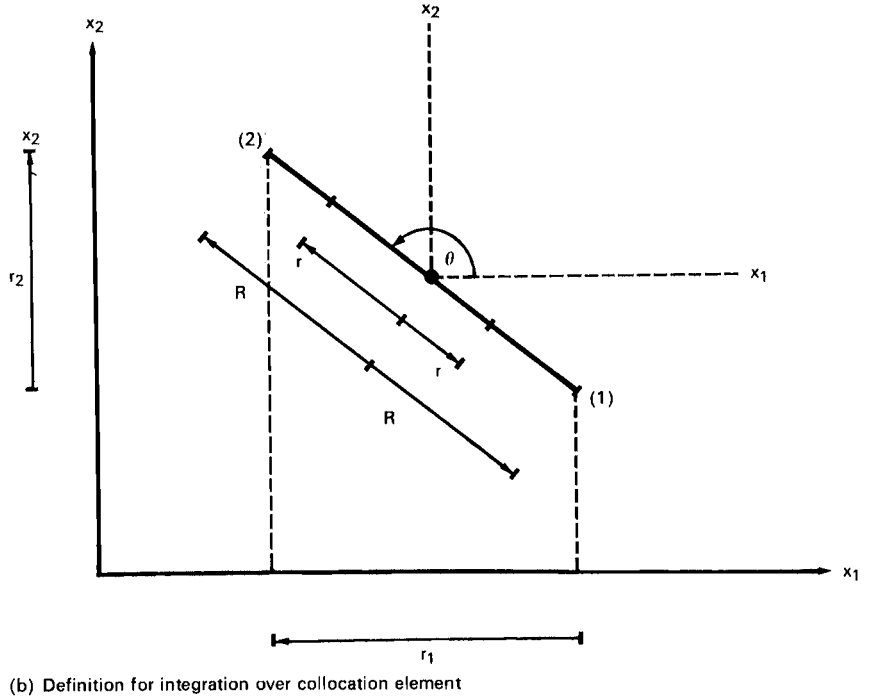
With the above definitions, one can now concentrate on the case for which the integration is carried out over the element 'i' which contains the singularity (figure 4.3(b)). Let us consider that the element starts at extreme point (1) and finishes at extreme point (2) shown in figure 4.3(a). The distances from these points to the singularity located at i in the centre of the element is R, while small r is the variable distance from i to any point over the element, i.e. r is equivalent to Γ.

Using the $\theta-r$ system defined in figure 4.3(b) one finds the following relationship for (4.40)

$$\begin{aligned}\frac{\partial r}{\partial x_1}&=\frac{r_1}{r}=\cos\theta\\ \frac{\partial r}{\partial x_2}&=\frac{r_2}{r}=\sin\theta\end{aligned} \tag{4.41}$$



(a) Geometrical definition



(b) Definition for integration over collocation element

Figure 4.3

One can now write formulae (4.37) to (4.39) taking limits around the singularity. (Notice that the integrals from i to (1) or (2) are the same.)

$$G_{11} = \lim_{\varepsilon \to 0} \left\{ \frac{1}{8\pi\mu(1-\nu)} \left[2(3-4\nu) \int_{\varepsilon}^{R} \ln \frac{1}{r}\, dr + 2 \int_{\varepsilon}^{R} \cos^2\theta\, dr \right] \right\}$$

$$= \lim_{\varepsilon \to 0} \left\{ \frac{1}{8\pi\mu(1-\nu)} \left[2(3-4\nu) \left\{ \left[r \ln \frac{1}{r} \right]_{\varepsilon}^{R} + [r]_{\varepsilon}^{R} \right\} + 2R\cos^2\theta \right] \right\}$$

$$= \frac{2R}{8\pi\mu(1-\nu)} \left[(3-4\nu)(1-\ln R) + \frac{(r_1)^2}{4R^2} \right] \tag{4.42}$$

$$G_{12} = G_{21} = \lim_{\varepsilon \to 0} \left\{ \frac{1}{8\pi\mu(1-\nu)} \left[2 \int_{\varepsilon}^{R} \sin\theta \cos\theta\, d\Gamma \right] \right\}$$

$$= \frac{1}{8\pi\mu(1-\nu)} [2R \sin\theta\cos\theta] = \frac{2R}{8\pi\mu(1-\nu)} \left[\frac{r_1 r_2}{4R^2} \right] \tag{4.43}$$

$$G_{22} = \lim_{\varepsilon \to 0} \left\{ \frac{1}{8\pi\mu(1-\nu)} \left[2(3-4\nu) \int_{\varepsilon}^{R} \ln \frac{1}{r}\, dr + 2 \int_{\varepsilon}^{R} \sin^2\theta\, dr \right] \right\}$$

$$= \lim_{\varepsilon \to 0} \left\{ \frac{1}{8\pi\mu(1-\nu)} \left[2(3-4\nu) [r \ln \frac{1}{r}]_{\varepsilon}^{R} + [r]_{\varepsilon}^{R} + 2R\sin^2\theta \right] \right\}$$

$$= \frac{2R}{8\pi\mu(1-\nu)} \left[(3-4\nu)(1-\ln R) + \frac{(r_2)^2}{4R^2} \right] \tag{4.44}$$

4.5 Elastostatics Code using Constant Elements (ELCONBE)

This section describes a computer code for the solution of two dimensional isotropic elastostatics problems without body forces. The code has a similar organization as those described in Chapter 2 and because of this the routines are given names similar to those used previously. The code is written for plane strain but can be used for plate stretching as well, simply by giving an equivalent value of ν (notice that the shear modulus μ or G remains the same).

The variables used in the program and their meaning are listed below,

N:	Number of boundary elements (equal to number of nodes for this case).
L:	Number of internal points where the displacements and stresses are to be calculated.
M:	Number of different surfaces (maximum 5).
NC():	Number of the last node of each different surface.
GE:	Shear modulus.
XNU:	Poisson's ratio for plane strain; use a fictitious Poisson's ratio XNU $= \nu/(1+\nu)$ for plane stress, ν being the Poisson's ratio.

X,Y: One dimensional arrays with x_1 and x_2 coordinates of the extreme points of the boundary elements.

XM,YM: One dimensional arrays with the coordinates of the nodes.

G: Matrix defined in equation (4.30). After application of the boundary conditions the matrix **A** is stored in the same location (equation (4.31)).

H: Matrix defined by equation (4.30).

KODE: One dimensional array indicating the type of boundary condition at the element nodes. KODE = 0 means that a displacement is prescribed and KODE = 1 that a traction is prescribed.

FI: Vector where the prescribed values of boundary conditions are stored. Each element is associated with a value of KODE.

DFI: Right hand side vector in the global system. After solution it contains the values of the unknowns.

CX,CY: One dimensional vectors with coordinates of internal points (maximum number 20).

DSOL: Values of displacements at internal points (2 displacements per point).

SSOL: Values of stresses at internal points (3 stresses per point).

Body forces have not been introduced for simplicity.

1. Main Program

The FORTRAN listing for the main program is as follows:

```
C-----------------------------------------------------------------------
C
      PROGRAM ELCONBE
C
C  PROGRAM 28
C
C  THIS PROGRAM SOLVES TWO-DIMENSIONAL (EL)ASTIC PROBLEMS
C  USING (CON)STANT (B)OUNDARY (E)LEMENTS
C
      CHARACTER*10 FILEIN,FILEOUT
C
      COMMON/MATG/ G(100,100)
      COMMON/MATH/ H(100,100)
      COMMON N,L,NC(5),M,GE,XNU,INP,IPR
      DIMENSION X(51),Y(51),XM(50),YM(50),FI(100),DFI(100)
      DIMENSION KODE(100),CX(20),CY(20),SSOL(60),DSOL(40)
C
C  SET MAXIMUN DIMENSION OF THE SYSTEM OF EQUATIONS (NX)
C  (THIS NUMBER MUST BE EQUAL OR SMALLER THAN THE DIMENSION OF G AND H)
C
      NX=100
C
C  ASSIGN NUMBERS FOR INPUT AND OUTPUT FILES
C
      INP=5
      IPR=6
C
C  READ NAMES AND OPEN FILES FOR INPUT AND UOTPUT
C
      WRITE(*,' (A) ') ' NAME OF THE INPUT FILE (MAX. 10 CHART.)'
      READ(*,' (A) ') FILEIN
      OPEN(INP,FILE=FILEIN,STATUS='OLD')
```

```
      WRITE(*,' (A) ') ' NAME OF THE OUTPUT FILE (MAX.10 CHART.)'
      READ(*,' (A) ') FILEOUT
      OPEN(IPR,FILE=FILEOUT,STATUS='NEW')
C
C  READ DATA
C
      CALL INPUTEC(CX,CY,X,Y,KODE,FI)
C
C  COMPUTE H AND G MATRICES AND FORM SYSTEM (A X = F)
C
      CALL GHMATEC(X,Y,XM,YM,G,H,FI,DFI,KODE,NX)
C
C  SOLVE SYSTEM OF EQUATIONS
C
      NN=2*N
      CALL SLNPD(G,DFI,D,NN,NX)
C
C  COMPUTE STRESS AND DISPLACEMENT AT INTERNAL POINTS
C
      CALL INTEREC(FI,DFI,KODE,CX,CY,X,Y,SSOL,DSOL)
C
C  PRINT RESULTS AT BOUNDARY NODES AND INTERNAL POINTS
C
      CALL OUTPTEC(XM,YM,FI,DFI,CX,CY,SSOL,DSOL)
      CLOSE (INP)
      CLOSE (IPR)
      STOP
      END
```

2. Routine INPUTEC

This subroutine reads the input required by the code. A file whose name is requested from the user by the main program will contain the following lines of free format input.

(i) *Title Line* Contains the title of the problem.

(ii) *Basic parameter line* Contains the number of elements, number of surfaces, last nodes of each different surface, the shear modulus and Poisson ratio or fictitious Poisson ratio for plane stress problems $\nu/(1+\nu)$.

(iii) *Extreme Points of Boundary Elements Lines* The coordinates of the extreme point of an element are read in the counterclockwise direction for external surfaces and clockwise for internal ones. The coordinates are read in free FORMAT.

(iv) *Boundary Conditions Lines* As many lines as boundary nodes giving the values of the known variable in x_1 and x_2 directions. The variables are displacements if KODE = 0 or tractions if KODE = 1.

(v) *Internal Points Coordinate Lines* Internal nodes are defined, each with the x_1x_2 coordinates of the point. The coordinates are read in free FORMAT.

The subroutine prints first the name of the job and the basic parameters. Then the coordinates of the extreme points of the elements and the boundary conditions given by node number, codes and prescribed values are printed. The internal point coordinates will be printed in the subroutine OUTPTEC. Its listing is as follows:

```
C-----------------------------------------------------------------------
      SUBROUTINE INPUTEC(CX,CY,X,Y,KODE,FI)
C
C  PROGRAM 29
C
```

```
      CHARACTER*80 TITLE
      DIMENSION CX(1),CY(1),X(1),Y(1),KODE(1),FI(1)
      COMMON N,L,NC(5),M,GE,XNU,INP,IPR
C
C  N= NUMBER OF BOUNDARY NODES (= NUMBER OF ELEMENTS)
C  L= NUMBER OF INTERNAL POINTS WHERE DISPLACEMENT AND STRESS
C     ARE CALCULATED
C  M= NUMBER OF DIFFERENT BOUNDARIES
C  NC(I)= LAST NODE OF BOUNDARY I
C  GE= SHEAR MODULUS
C  XNU= POISSON MODULUS
C
      WRITE(IPR,100)
  100 FORMAT(/' ',79('*'))
C
C  READ JOB TITLE
C
      READ(INP,'(A)') TITLE
      WRITE(IPR,'(A)') TITLE
C
C  READ NUMBER OF NODES, INTERNAL POINTS AND DIFFERENT BOUNDARIES;
C  READ LAST NODES OF THESE BOUNDARIES AND MATERIAL PROPERTIES
C
      READ(INP,*)N,L,M,(NC(K),K=1,5),GE,XNU
      WRITE(IPR,300)N,L,GE,XNU
  300 FORMAT(//' DATA'//2X,'NUMBER OF BOUNDARY ELEMENTS =',I3/2X,
     1'NUMBER OF INTERNAL POINTS =',I3/2X,
     2'SHEAR MODULUS =',E14.7/2X,'POISSON RATIO =',E14.7)
      IF(M)40,40,30
   30 WRITE(IPR,999)M,(NC(K),K=1,M)
  999 FORMAT(2X,'NUMBER OF DIFFERENT BOUNDARIES=',I3/2X,
     1'LAST NODES OF THESE BOUNDARIES =',5(2X,I3))
C
C  READ COORDINATES OF EXTREME POINTS OF THE BOUNDARY
C  ELEMENTS IN ARRAYS X AND Y
C
   40 WRITE(IPR,500)
  500 FORMAT(//2X,'COORDINATES OF THE EXTREME POINTS OF'
     1' THE BOUNDARY ELEMENTS'//4X,'POINT',10X,'X',18X,'Y')
      READ(INP,*) (X(I),Y(I),I=1,N)
      DO 10 I=1,N
   10 WRITE(IPR,700)I,X(I),Y(I)
  700 FORMAT(5X,I3,2(5X,E14.7))
C
C  READ BOUNDARY CONDITIONS IN FI(I) VECTOR, IF KODE(I)=0 THE FI(I)
C  VALUE IS A KNOWN DISPLACEMENT; IF KODE(I)=1 THE FI(I) VALUE IS A
C  KNOWN TRACTION.
C
      WRITE(IPR,800)
  800 FORMAT(//2X,'BOUNDARY CONDITIONS'//15X,'PRESCRIBED VALUE',15X,
     1'PRESCRIBED VALUE'/5X,'NODE',9X,'X DIRECTION',8X,'CODE',8X,
     2'Y DIRECTION',8X,'CODE')
      DO 20 I=1,N
      READ(INP,*) KODE(2*I-1),FI(2*I-1),KODE(2*I),FI(2*I)
   20 WRITE(IPR,950)I,FI(2*I-1),KODE(2*I-1),FI(2*I),KODE(2*I)
  950 FORMAT(5X,I3,8X,E14.7,8X,I1,8X,E14.7,8X,I1)
C
C  READ COORDINATES OF THE INTERNAL POINTS
C
      IF(L.EQ.0) GO TO 50
      READ(INP,*) (CX(I),CY(I),I=1,L)
   50 RETURN
      END
```

3. Routine GHMATEC

This routine computes the system matrices **H** and **G** by calling the routines EXTINEC and LOCINEC, i.e.

(i) *EXTINEC* Computes the $\mathbf{H}^{ij}$ and $\mathbf{G}^{ij}$ submatrices using numerical integration for the cases $i \neq j$.

(ii) *LOCINEC* Only calculates the submatrices $\mathbf{G}^{ii}$ applying formulae (4.42) to (4.44).

Notice that the diagonal submatrices of $\mathbf{H}$ for a smooth boundary as in constant elements, are given by

$$\mathbf{H}^{ii} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix}$$

After computing $\mathbf{H}$ and $\mathbf{G}$ matrices the routine arranges the equations in accordance with the boundary conditions and forms the $\mathbf{A}$ matrix of equations (4.31). When rearranging the columns those of $\mathbf{G}$ are multiplied by the shear modulus (called GE in the code) so that all elements of A are of the same order. This helps to increase the numerical accuracy of the solution.

The right hand side vector F of equation (4.31) also found in this routine is stored in array DFI.

GHMATEC listing is as follows:

```
C-----------------------------------------------------------------------
      SUBROUTINE GHMATEC(X,Y,XM,YM,G,H,FI,DFI,KODE,NX)
C
C  PROGRAM 30
C
C  THIS SUBROUTINE COMPUTES THE G AND H MATRICES AND
C  FORMS THE SYSTEM OF EQUATIONS A X = F
C
      DIMENSION G(NX,NX),H(NX,NX)
      DIMENSION X(1),Y(1),XM(1),YM(1),FI(1)
      DIMENSION KODE(1),DFI(1)
      COMMON N,L,NC(5),M,GE,XNU,INP,IPR
C
C  COMPUTE THE NODAL COORDINATES AND STORE IN ARRAYS XM AND YM
C
      X(N+1)=X(1)
      Y(N+1)=Y(1)
      DO 10 I=1,N
      XM(I)=(X(I)+X(I+1))/2
   10 YM(I)=(Y(I)+Y(I+1))/2
      IF(M-1)15,15,12
   12 XM(NC(1))=(X(NC(1))+X(1))/2
      YM(NC(1))=(Y(NC(1))+Y(1))/2
      DO 13 K=2,M
      XM(NC(K))=(X(NC(K))+X(NC(K-1)+1))/2
   13 YM(NC(K))=(Y(NC(K))+Y(NC(K-1)+1))/2
C
C  COMPUTE THE COEFICIENTS OF G AND H MATRICES
C
   15 DO 30 I=1,N
      DO 30 J=1,N
      IF(M-1)16,16,17
   17 IF(J-NC(1))19,18,19
   18 KK=1
      GO TO 23
   19 DO 22 K=2,M
      IF(J-NC(K))22,21,22
   21 KK=NC(K-1)+1
      GO TO 23
   22 CONTINUE
   16 KK=J+1
   23 IF(I-J)20,25,20
```

```
   20 CALL EXTINEC(XM(I),YM(I),X(J),Y(J),X(KK),Y(KK),H((2*I-1),(2*J-1)),
     1H((2*I-1),(2*J)),H((2*I),(2*J-1)),H((2*I),(2*J)),G((2*I-1),
     2(2*J-1)),G((2*I-1),(2*J)),G((2*I),(2*J)))
      G((2*I),(2*J-1))=G((2*I-1),(2*J))
      GO TO 26
   25 CALL LOCINEC(X(J),Y(J),X(KK),Y(KK),G((2*I-1),(2*J-1)),
     1G((2*I-1),(2*J)),G((2*I),(2*J)))
      H((2*I-1),(2*J-1))=0.5
      H((2*I),(2*J))=0.5
      H((2*I-1),(2*J))=0.
      H((2*I),(2*J-1))=0.
      G((2*I),(2*J-1))=G((2*I-1),(2*J))
   26 CONTINUE
   30 CONTINUE
C
C  REORDER THE COLUMNS OF THE SYSTEM OF EQUATIONS IN ACCORDANCE
C  WITH THE BOUNDARY CONDITIONS AND FORM SYSTEM MATRIX A WHICH
C  IS STORED IN G
C
      NN=2*N
      DO 50 J=1,NN
      IF(KODE(J))43,43,40
   40 DO 42 I=1,NN
      CH=G(I,J)
      G(I,J)=-H(I,J)
   42 H(I,J)=-CH
      GO TO 50
   43 DO 45 I=1,NN
   45 G(I,J)=G(I,J)*GE
   50 CONTINUE
C
C  FORM THE RIGHT HAND SIDE VECTOR F WHICH IS STORED IN DFI
C
      DO 60 I=1,NN
      DFI(I)=0.
      DO 60 J=1,NN
      DFI(I)=DFI(I)+H(I,J)*FI(J)
   60 CONTINUE
      RETURN
      END
```

4. Routine EXTINEC

This routine computes the $\mathbf{G}^{ij}$ and $\mathbf{H}^{ij}$ submatrices (for $i \neq j$) using numerical integration. Notice that $\mathbf{G}^{ij}$ is symmetric but $\mathbf{H}^{ij}$ is not. The values of the coefficients are computed using four points Gaussian quadrature.

```
C-----------------------------------------------------------------------
      SUBROUTINE EXTINEC(XP,YP,X1,Y1,X2,Y2,H11,H12,H21,H22,G11,G12,G22)
C
C  PROGRAM 31
C
C  THIS SOUBROUTINE COMPUTES THE G AND H MATRICES
C  COEFFICIENTS THAT RELATE A COLLOCATION POINT WITH A DIFFERENT
C  ELEMENT USING GAUSS QUADRATURE
C
C  DIST= DISTANCE FROM THE COLOCATION POINT TO THE
C        LINE TANGENT TO THE ELEMENT
C  RA= DISTANCE FROM THE COLOCATION POINT TO THE
C      GAUSS INTEGRATION POINT AT THE BOUNDARY ELEMENT
C
      DIMENSION XCO(4),YCO(4),GI(4),OME(4)
      COMMON N,L,NC(5),M,GE,XNU,INP,IPR
      DATA GI/0.86113631,-0.86113631,0.33998104,-0.33998104/
      DATA OME/0.34785485,0.34785485,0.65214515,0.65214515/
C
      AX=(X2-X1)/2
      BX=(X2+X1)/2
```

```
      AY=(Y2-Y1)/2
      BY=(Y2+Y1)/2
      ETA1=(Y2-Y1)/(2*SQRT(AX**2+AY**2))
      ETA2=(X1-X2)/(2*SQRT(AX**2+AY**2))
C
C  COMPUTE THE DISTANCE FROM THE POINT TO THE LINE OF THE ELEMENT
C
      IF(AX)10,20,10
   10 TA=AY/AX
      DIST=ABS((TA*XP-YP+Y1-TA*X1)/SQRT(TA**2+1))
      GO TO 30
   20 DIST=ABS(XP-X1)
C
C  DETERMINE THE DIRECTION OF THE OUTWARD NORMAL
C
   30 SIG=(X1-XP)*(Y2-YP)-(X2-XP)*(Y1-YP)
      IF(SIG)31,32,32
   31 DIST=-DIST
   32 H11=0.
      H12=0.
      H21=0.
      H22=0.
      G11=0.
      G12=0.
      G22=0.
C
C  COMPUTE G AND H COEFFICIENTS
C
      DE=4*3.141592*(1-XNU)
      DO 40 I=1,4
      XCO(I)=AX*GI(I)+BX
      YCO(I)=AY*GI(I)+BY
      RA=SQRT((XP-XCO(I))**2+(YP-YCO(I))**2)
      RD1=(XCO(I)-XP)/RA
      RD2=(YCO(I)-YP)/RA
      G11=G11+((3-4*XNU)*ALOG(1./RA)+RD1**2)*OME(I)*SQRT(AX**2+AY**2)/(2
     1*DE*GE)
      G12=G12+RD1*RD2*OME(I)*SQRT(AX**2+AY**2)/(2*DE*GE)
      G22=G22+((3-4*XNU)*ALOG(1./RA)+RD2**2)*OME(I)*SQRT(AX**2+AY**2)/(2
     1*DE*GE)
      H11=H11-DIST*((1-2*XNU)+2*RD1**2)/(RA**2*DE)*OME(I)*SQRT(AX**2+AY*
     1*2)
      H12=H12-(DIST*2*RD1*RD2/RA+(1-2*XNU)*(ETA1*RD2-ETA2*RD1))*OME(I)*S
     1QRT(AX**2+AY**2)/(RA*DE)
      H21=H21-(DIST*2*RD1*RD2/RA+(1-2*XNU)*(ETA2*RD1-ETA1*RD2))*OME(I)*S
     1QRT(AX**2+AY**2)/(RA*DE)
   40 H22=H22-DIST*((1-2*XNU)+2*RD2**2)*OME(I)*SQRT(AX**2+AY**2)/(RA**2*
     1DE)
      RETURN
      END
```

5. Routine LOCINEC

It calculates the G^{ii} submatrix in accordance with formulae (4.42) to (4.44), which give the analytical results.

```
C-----------------------------------------------------------------------
      SUBROUTINE LOCINEC(X1,Y1,X2,Y2,G11,G12,G22)
C
C  PROGRAM 32
C
C  THIS SUBROUTINE COMPUTES THE VALUES OF THE MATRIX G COEFFICIENTS
C  THAT RELATE AN ELEMENT WITH ITSELF
C
      COMMON N,L,NC(5),M,GE,XNU,INP,IPR
C
      AX=(X2-X1)/2
      AY=(Y2-Y1)/2
      SR=SQRT(AX**2+AY**2)
```

```
      DE=4*3.141592*GE*(1-XNU)
      G11=SR*((3-4*XNU)*(1-ALOG(SR))+(X2-X1)**2/(4*SR**2))/DE
      G22=SR*((3-4*XNU)*(1-ALOG(SR))+(Y2-Y1)**2/(4*SR**2))/DE
      G12=(X2-X1)*(Y2-Y1)/(4*SR*DE)
      RETURN
      END
```

6. Routine SLNPD

This is the same solver as used in previous codes.

7. Routine INTEREC

This subroutine computes the displacements and stresses at internal points. First it reorders the vectors DFI and FI. Notice that the first contains the values of **X** in equation (4.31) as given by the solver and the second the boundary conditions. After reorganization the boundary displacements are all stored in FI and the tractions in DFI.

Formulae (4.33) and (4.35) are applied to compute internal stresses and displacements. This requires the integration of terms with S and D coefficients which are carried out numerically in a routine called SIGMAEC.

The FORTRAN Listing is as follows:

```
C-----------------------------------------------------------------------
      SUBROUTINE INTEREC(FI,DFI,KODE,CX,CY,X,Y,SSOL,DSOL)
C
C  PROGRAM 33
C
C  THIS SUBROUTINE COMPUTES THE VALUES OF THE STRESS AND DISPLACEMENT
C  COMPONENTS AT INTERNAL POINTS
C
      DIMENSION CX(1),CY(1),SSOL(1),DSOL(1)
      DIMENSION FI(1),DFI(1),KODE(1),X(1),Y(1)
      COMMON N,L,NC(5),M,GE,XNU,INP,IPR
C
C  REARRANGE FI AND DFI ARRAYS TO STORE ALL THE VALUES
C  OF THE DISPLACEMENT IN FI AND ALL THE VALUES OF THE TRACTIONS IN DFI
C
      NN=2*N
      DO 20 I=1,NN
      IF(KODE(I)) 15,15,10
   10 CH=FI(I)
      FI(I)=DFI(I)
      DFI(I)=CH
      GO TO 20
   15 DFI(I)=DFI(I)*GE
   20 CONTINUE
C
C  COMPUTE THE VALUES OF STRESSES AND DISPLACEMENTS
C  AT INTERNAL POINTS.
C
      IF(L.EQ.0) GO TO 50
      DO 40 K=1,L
      DSOL(2*K-1)=0.
      DSOL(2*K)=0.
      SSOL(3*K-2)=0.
      SSOL(3*K-1)=0.
      SSOL(3*K)=0.
      DO 30 J=1,N
      IF(M-1)28,28,22
   22 IF(J-NC(1))24,23,24
```

```
   23 KK=1
      GO TO 29
   24 DO 26 LK=2,M
      IF(J-NC(LK))26,25,26
   25 KK=NC(LK-1)+1
      GO TO 29
   26 CONTINUE
   28 KK=J+1
   29 CALL EXTINEC(CX(K),CY(K),X(J),Y(J),X(KK),Y(KK),H11,H12,H21,H22,
     1 G11,G12,G22)
      DSOL(2*K-1)=DSOL(2*K-1)+DFI(2*J-1)*G11+DFI(2*J)*G12-FI(2*J-1)*H11-
     1FI(2*J)*H12
      DSOL(2*K)=DSOL(2*K)+DFI(2*J-1)*G12+DFI(2*J)*G22-FI(2*J-1)*H21-FI(2
     1*J)*H22
      CALL SIGMAEC(CX(K),CY(K),X(J),Y(J),X(KK),Y(KK),D111,D211,D112,
     1 D212,D122,D222,S111,S211,S112,S212,S122,S222)
      SSOL(3*K-2)=SSOL(3*K-2)+DFI(2*J-1)*D111+DFI(2*J)*D211-FI(2*J-1)*S1
     111-FI(2*J)*S211
      SSOL(3*K-1)=SSOL(3*K-1)+DFI(2*J-1)*D112+DFI(2*J)*D212-FI(2*J-1)*S1
     112-FI(2*J)*S212
   30 SSOL(3*K)=SSOL(3*K)+DFI(2*J-1)*D122+DFI(2*J)*D222-FI(2*J-1)*S122-F
     1I(2*J)*S222
   40 CONTINUE
   50 RETURN
      END
```

8. Routine SIGMAEC

This subroutine computes the integrals with S and D terms described in formulae (4.35) and needed to calculate the internal stresses. The integrations are carried out using a four points Gaussian integration scheme.

```
C-----------------------------------------------------------------------
      SUBROUTINE SIGMAEC(XP,YP,X1,Y1,X2,Y2,D111,D211,D112,D212,D122,
     1 D222,S111,S211,S112,S212,S122,S222)
C
C  PROGRAM 34
C
C  THIS SUBROUTINE COMPUTES THE VALUES OF THE S AND D MATRICES
C  USING GAUSS QUADRATURE IN ORDER TO COMPUTE THE STRESSES
C  AT ANY INTERNAL POINT
C
C  RA= DISTANCE FROM THE POINT TO THE GAUSS INTEGRATION POINTS
C      ON THE BOUNDARY ELEMENTS
C  DIST=   DISTANCE FROM THE POINT TO THE LINE TANGENT
C          TO THE ELEMENT
C  RD1,RD2=   DERIVATIVES OF RA
C  ETA1,ETA2= COMPONENTS OF THE UNIT NORMAL TO THE ELEMENT
C
      DIMENSION XCO(4),YCO(4),GI(4),OME(4)
      COMMON N,L,NC(5),M,GE,XNU,INP,IPR
      DATA GI/0.86113631,-0.86113631,0.33998104,-0.33998104/
      DATA OME/0.34785485,0.34785485,0.65214515,0.65214515/
C
      AX=(X2-X1)/2
      BX=(X2+X1)/2
      AY=(Y2-Y1)/2
      BY=(Y2+Y1)/2
      ETA1=(Y2-Y1)/(2*SQRT(AX**2+AY**2))
      ETA2=(X1-X2)/(2*SQRT(AX**2+AY**2))
C
C  COMPUTE THE DISTANCE FROM THE POINT TO THE LINE OF THE ELEMENT
C
      IF(AX)10,20,10
   10 TA=AY/AX
      DIST=ABS((TA*XP-YP+Y1-TA*X1)/SQRT(TA**2+1))
      GO TO 30
   20 DIST=ABS(XP-X1)
```

```
C
C  DETERMINE THE DIRECTION OF THE OUTWARD NORMAL
C
   30 SIG=(X1-XP)*(Y2-YP)-(X2-XP)*(Y1-YP)
      IF(SIG)31,32,32
   31 DIST=-DIST
   32 D111=0.
      D211=0.
      D112=0.
      D212=0.
      D122=0.
      D222=0.
      S111=0.
      S211=0.
      S112=0.
      S212=0.
      S122=0.
      S222=0.
C
C  COMPUTE D AND S COEFFICIENTS
C
      FA=1-4*XNU
      AL=1-2*XNU
      DE=4*3.141592*(1-XNU)
      DO 40 I=1,4
      XCO(I)=AX*GI(I)+BX
      YCO(I)=AY*GI(I)+BY
      RA=SQRT((XP-XCO(I))**2+(YP-YCO(I))**2)
      RD1=(XCO(I)-XP)/RA
      RD2=(YCO(I)-YP)/RA
      D111=D111+(AL*RD1+2*RD1**3)*OME(I)*SQRT(AX**2+AY**2)/(DE*RA)
      D211=D211+(2*RD1**2*RD2-AL*RD2)*OME(I)*SQRT(AX**2+AY**2)/(DE*RA)
      D112=D112+(AL*RD2+2*RD1**2*RD2)/(DE*RA)*OME(I)*SQRT(AX**2+AY**2)
      D212=D212+(AL*RD1+2*RD1*RD2**2)/(DE*RA)*OME(I)*SQRT(AX**2+AY**2)
      D122=D122+(2*RD1*RD2**2-AL*RD1)/(DE*RA)*OME(I)*SQRT(AX**2+AY**2)
      D222=D222+(AL*RD2+2*RD2**3)/(DE*RA)*OME(I)*SQRT(AX**2+AY**2)
      S111=S111+(2*DIST/RA*(AL*RD1+XNU*2*RD1-4*RD1**3)+4*XNU*ETA1*RD1**2
     1+AL*(2*ETA1*RD1**2+2*ETA1)-FA*ETA1)*2*GE/(DE*RA**2)*OME(I)*SQRT(AX
     2**2+AY**2)
      S211=S211+(2*DIST/RA*(AL*RD2-4*RD1**2*RD2)+4*XNU*ETA1*RD1*RD2+AL*2
     1*ETA2*RD1**2-FA*ETA2)*2*GE/(DE*RA**2)*OME(I)*SQRT(AX**2+AY**2)
      S112=S112+(2*DIST/RA*(XNU*RD2-4*RD1**2*RD2)+2*XNU*(ETA1*RD2*RD1+ET
     1A2*RD1**2)+AL*(2*ETA1*RD1*RD2+ETA2))*2*GE/(DE*RA**2)*OME(I)*SQRT(A
     2X**2+AY**2)
      S212=S212+(2*DIST/RA*(XNU*RD1-4*RD1*RD2**2)+2*XNU*(ETA1*RD2**2+ETA
     12*RD1*RD2)+AL*(2*ETA2*RD1*RD2+ETA1))*2*GE/(DE*RA**2)*OME(I)*SQRT(A
     2X**2+AY**2)
      S122=S122+(2*DIST/RA*(AL*RD1-4*RD1*RD2**2)+4*XNU*ETA2*RD1*RD2+AL*2
     1*ETA1*RD2**2-FA*ETA1)*2*GE/(DE*RA**2)*OME(I)*SQRT(AX**2+AY**2)
   40 S222=S222+(2*DIST/RA*(AL*RD2+2*XNU*RD2-4*RD2**3)+4*XNU*ETA2*RD2**2
     1+AL*(2*ETA2*RD2**2+2*ETA2)-FA*ETA2)*2*GE/(DE*RA**2)*OME(I)*SQRT(AX
     2**2+AY**2)
      RETURN
      END
```

9. Routine OUTPTEC

This subroutine prints the results in the following order.

(ii) Boundary nodes values with coordinates $x_1 x_2$, values of $u_1 u_2$ displacements and $p_1 p_2$ tractions.

(ii) Internal node results with coordinates $x_1 x_2$, values of $u_1 u_2$ displacements and $\sigma_{11}\sigma_{12}\sigma_{22}$ stresses.

Notice that stresses on the boundary are not produced. If required they will need to be computed using formulae (3.5) with the derivatives of displacements

tangential to the surface computed by considering the nodes of displacements and adjacent elements and applying a finite different type approach.

The listing of OUTPT is as follows:

```
C-----------------------------------------------------------------------
      SUBROUTINE OUTPTEC(XM,YM,FI,DFI,CX,CY,SSOL,DSOL)
C
C  PROGRAM 35
C
C  THIS SUBROUTINE PRINTS THE VALUES OF THE DISPLACEMENTS
C  AND TRACTIONS AT BOUNDARY NODES. IT ALSO PRINTS THE VALUES
C  OF DISPLACEMENTS AND STRESSES AT INTERNAL POINTS
C
      DIMENSION XM(1),YM(1),FI(1),DFI(1)
      DIMENSION CX(1),CY(1),SSOL(1),DSOL(1)
      COMMON N,L,NC(5),M,GE,XNU,INP,IPR
C
      WRITE(IPR,100)
  100 FORMAT(' ',79('*')//1X,'RESULTS'//2X,'BOUNDARY NODES'//6X
     1,'X',12X,'Y',9X,'DISPL. X',5X,'DISPL. Y',4X,
     2'TRACTION X',3X,'TRACTION Y'/)
      DO 10 I=1,N
   10 WRITE(IPR,200) XM(I),YM(I),FI(2*I-1),FI(2*I),DFI(2*I-1),DFI(2*I)
  200 FORMAT(6(1X,E12.5))
C
      IF(L.EQ.0.) GO TO 30
      WRITE(IPR,300)
  300 FORMAT(//2X,'INTERNAL POINTS DISPLACEMENTS'//8X,'X',15X,'Y',10X,
     1'DISPLACEMENT X',5X,'DISPLACEMENT Y')
      DO 20 K=1,L
   20 WRITE(IPR,400)CX(K),CY(K),DSOL(2*K-1),DSOL(2*K)
      WRITE(IPR,350)
  350 FORMAT(//2X,'INTERNAL POINTS STRESSES'//8X,'X',15X,'Y',12X,
     1'SIGMA X',10X,'TAU XY',9X,'SIGMA Y')
      DO 25 K=1,L
   25 WRITE(IPR,450) CX(K),CY(K),SSOL(3*K-2),SSOL(3*K-1),SSOL(3*K)
  400 FORMAT(2(2X,E14.7),2(5X,E14.7))
  450 FORMAT(5(2X,E14.7))
   30 WRITE(IPR,500)
  500 FORMAT(' ',79('*'))
      RETURN
      END
```

Example 4.1

Figure 4.4 describes a circular cavity under internal pressure in an infinite medium. Numerical results will be compared against the known analytical solution, first using constant elements and then with the quadratic elements of section 4.7 and 4.8.

The boundary is divided here into 24 constant elements and 5 internal nodes are defined at which displacements and stresses will be found.

Notice that in order to stop all rigid body movements one can suppress the displacements in the x_2 directions at node 18.

The input needed to run the ELCONBE code is as follows.

CIRCULAR CAVITY (DATA)

```
  CIRCULAR CAVITY UNDER INTERNAL PRESSURE (24 CONSTANT ELEMENTS)
24 5  0 0 0 0 0 0 94500. 0.1
-.3916 -2.9743
-1.1481 -2.7716
-1.8263 -2.3801
-2.3801 -1.8263
-2.7716 -1.1480
-2.9743 -.3916
-2.9743  .3916
```

```
-2.7716  1.1481
-2.3801  1.8263
-1.8263  2.3801
-1.1480  2.7716
-.3916   2.9743
.3916  2.9743
1.1481 2.7716
1.8263 2.3801
2.3801 1.8263
2.7716 1.1480
2.9743 .3916
2.9743 -.3916
2.7716 -1.1481
2.3801 -1.8263
1.8263 -2.3801
1.1480 -2.7716
.3916  -2.9743
1 -25.88 1 -96.59
1 -50.   1 -86.6
1 -70.71 1 -70.71
1 -86.6  1 -50.
1 -96.59 1 -25.88
1 -100.  1 0.
1 -96.59 1 25.88
1 -86.6  1 50.
1 -70.71 1 70.71
1 -50.   1 86.6
1 -25.88 1 96.59
0  0.    1 100.
1 25.88 1 96.59
1 50.   1 86.6
1 70.71 1 70.71
1 86.6  1 50.
1 96.59 1 25.88
1 100.  0 0.
1 96.59 1 -25.88
1 86.6  1 -50.
1 70.71 1 -70.71
1 50.   1 -86.6
1 25.88 1 -96.59
0  0.   1 -100.
4. 0. 2.82843 2.82843 -4. 0. 6. 0. 10. 0.
```

This produces the output given below. Notice that the results are very close to the theoretical results as shown by the values of radial stresses at internal points. Stresses and displacements decay with increasing distance from the cavity as is to be expected. Radial and hoop stresses at internal points have the same absolute value but with different signs as expected.

Table 4.1 Radial Stresses at Internal Points. 24 Constant Elements Discretization

Distance to the centre of the cavity	Constant boundary element discretization	Elasticity theory
4	-57.234	-56.250
6	-25.295	-25.000
10	-9.106	-9.000
20	-2.276	-2.250
50	-0.364	-0.360
200	-0.227×10^{-1}	-0.225×10^{-1}
1000	-0.991×10^{-3}	-0.9×10^{-3}

The computer output is as follows,

CIRCULAR CAVITY (OUTPUT)

```
*************************************************************************
 CIRCULAR CAVITY UNDER INTERNAL PRESSURE (24 CONSTANT ELEMENTS)

DATA

 NUMBER OF BOUNDARY ELEMENTS = 24
 NUMBER OF INTERNAL POINTS =  5
 SHEAR MODULUS = 0.9450000E+05
 POISSON RATIO = 0.1000000E+00

 COORDINATES OF THE EXTREME POINTS OF' THE BOUNDARY ELEMENTS

  POINT          X                Y
    1     -0.3916000E+00    -0.2974300E+01
    2     -0.1148100E+01    -0.2771600E+01
    3     -0.1826300E+01    -0.2380100E+01
    4     -0.2380100E+01    -0.1826300E+01
    5     -0.2771600E+01    -0.1148000E+01
    6     -0.2974300E+01    -0.3916000E+00
    7     -0.2974300E+01     0.3916000E+00
    8     -0.2771600E+01     0.1148100E+01
    9     -0.2380100E+01     0.1826300E+01
   10     -0.1826300E+01     0.2380100E+01
   11     -0.1148000E+01     0.2771600E+01
   12     -0.3916000E+00     0.2974300E+01
   13      0.3916000E+00     0.2974300E+01
   14      0.1148100E+01     0.2771600E+01
   15      0.1826300E+01     0.2380100E+01
   16      0.2380100E+01     0.1826300E+01
   17      0.2771600E+01     0.1148000E+01
   18      0.2974300E+01     0.3916000E+00
   19      0.2974300E+01    -0.3916000E+00
   20      0.2771600E+01    -0.1148100E+01
   21      0.2380100E+01    -0.1826300E+01
   22      0.1826300E+01    -0.2380100E+01
   23      0.1148000E+01    -0.2771600E+01
   24      0.3916000E+00    -0.2974300E+01

 BOUNDARY CONDITIONS

            PRESCRIBED VALUE                    PRESCRIBED VALUE
  NODE        X DIRECTION        CODE          Y DIRECTION        CODE
    1     -0.2588000E+02          1        -0.9659000E+02          1
    2     -0.5000000E+02          1        -0.8660000E+02          1
    3     -0.7071000E+02          1        -0.7071000E+02          1
    4     -0.8660000E+02          1        -0.5000000E+02          1
    5     -0.9659000E+02          1        -0.2588000E+02          1
    6     -0.1000000E+03          1         0.0000000E+00          1
    7     -0.9659000E+02          1         0.2588000E+02          1
    8     -0.8660000E+02          1         0.5000000E+02          1
    9     -0.7071000E+02          1         0.7071000E+02          1
   10     -0.5000000E+02          1         0.8660000E+02          1
   11     -0.2588000E+02          1         0.9659000E+02          1
   12      0.0000000E+00          0         0.1000000E+03          1
   13      0.2588000E+02          1         0.9659000E+02          1
   14      0.5000000E+02          1         0.8660000E+02          1
   15      0.7071000E+02          1         0.7071000E+02          1
   16      0.8660000E+02          1         0.5000000E+02          1
   17      0.9659000E+02          1         0.2588000E+02          1
   18      0.1000000E+03          1         0.0000000E+00          0
   19      0.9659000E+02          1        -0.2588000E+02          1
   20      0.8660000E+02          1        -0.5000000E+02          1
   21      0.7071000E+02          1        -0.7071000E+02          1
   22      0.5000000E+02          1        -0.8660000E+02          1
   23      0.2588000E+02          1        -0.9659000E+02          1
   24      0.0000000E+00          0        -0.1000000E+03          1
*************************************************************************
```

RESULTS

BOUNDARY NODES

X	Y	DISPL. X	DISPL. Y	TRACTION X	TRACTION Y
-0.76985E+00	-0.28730E+01	-0.42449E-03	-0.15842E-02	-0.25880E+02	-0.96590E+02
-0.14872E+01	-0.25759E+01	-0.82002E-03	-0.14204E-02	-0.50000E+02	-0.86600E+02
-0.21032E+01	-0.21032E+01	-0.11597E-02	-0.11597E-02	-0.70710E+02	-0.70710E+02
-0.25759E+01	-0.14872E+01	-0.14204E-02	-0.82001E-03	-0.86600E+02	-0.50000E+02
-0.28730E+01	-0.76980E+00	-0.15842E-02	-0.42449E-03	-0.96590E+02	-0.25880E+02
-0.29743E+01	0.00000E+00	-0.16401E-02	-0.73851E-08	-0.10000E+03	0.00000E+00
-0.28730E+01	0.76985E+00	-0.15842E-02	0.42448E-03	-0.96590E+02	0.25880E+02
-0.25759E+01	0.14872E+01	-0.14204E-02	0.82002E-03	-0.86600E+02	0.50000E+02
-0.21032E+01	0.21032E+01	-0.11597E-02	0.11597E-02	-0.70710E+02	0.70710E+02
-0.14872E+01	0.25759E+01	-0.82001E-03	0.14204E-02	-0.50000E+02	0.86600E+02
-0.76980E+00	0.28730E+01	-0.42449E-03	0.15842E-02	-0.25880E+02	0.96590E+02
0.00000E+00	0.29743E+01	0.00000E+00	0.16401E-02	0.14446E-02	0.10000E+03
0.76985E+00	0.28730E+01	0.42449E-03	0.15842E-02	0.25880E+02	0.96590E+02
0.14872E+01	0.25759E+01	0.82002E-03	0.14204E-02	0.50000E+02	0.86600E+02
0.21032E+01	0.21032E+01	0.11597E-02	0.11597E-02	0.70710E+02	0.70710E+02
0.25759E+01	0.14872E+01	0.14204E-02	0.82001E-03	0.86600E+02	0.50000E+02
0.28730E+01	0.76980E+00	0.15842E-02	0.42449E-03	0.96590E+02	0.25880E+02
0.29743E+01	0.00000E+00	0.16401E-02	0.00000E+00	0.10000E+03	-0.20253E-02
0.28730E+01	-0.76985E+00	0.15842E-02	-0.42448E-03	0.96590E+02	-0.25880E+02
0.25759E+01	-0.14872E+01	0.14204E-02	-0.82002E-03	0.86600E+02	-0.50000E+02
0.21032E+01	-0.21032E+01	0.11597E-02	-0.11597E-02	0.70710E+02	-0.70710E+02
0.14872E+01	-0.25759E+01	0.82001E-03	-0.14204E-02	0.50000E+02	-0.86600E+02
0.76980E+00	-0.28730E+01	0.42449E-03	-0.15842E-02	0.25880E+02	-0.96590E+02
0.00000E+00	-0.29743E+01	0.00000E+00	-0.16401E-02	-0.12447E-02	-0.10000E+03

INTERNAL POINTS DISPLACEMENTS

X	Y	DISPLACEMENT X	DISPLACEMENT Y
0.4000000E+01	0.0000000E+00	0.1204821E-02	0.7319613E-08
0.2828430E+01	0.2828430E+01	0.8519205E-03	0.8519267E-03
-0.4000000E+01	0.0000000E+00	-0.1204821E-02	-0.4833055E-08
0.6000000E+01	0.0000000E+00	0.8029994E-03	0.6300070E-08
0.1000000E+02	0.0000000E+00	0.4817977E-03	0.6126356E-08

INTERNAL POINTS STRESSES

X	Y	SIGMA X	TAU XY	SIGMA Y
0.4000000E+01	0.0000000E+00	-0.5723444E+02	-0.5717874E-03	0.5711818E+
0.2828430E+01	0.2828430E+01	-0.5605986E-01	-0.5717675E+02	-0.5633205E-
-0.4000000E+01	0.0000000E+00	-0.5723446E+02	-0.6064773E-03	0.5711818E+
0.6000000E+01	0.0000000E+00	-0.2529478E+02	-0.7735193E-04	0.2529438E+
0.1000000E+02	0.0000000E+00	-0.9106032E+01	-0.1208484E-04	0.9105983E+

****

4.6 Linear Elements

In this section we will consider the development of linear elements similar to those discussed in Chapter 2 section 5, for potential problems. The difference is that now we need to interpolate two values for u's and two for p's (figure 4.5).

$$\mathbf{u} = \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} = \begin{bmatrix} \phi_1 & 0 & \phi_2 & 0 \\ 0 & \phi_1 & 0 & \phi_2 \end{bmatrix} \begin{Bmatrix} u_1^1 \\ u_2^1 \\ u_1^2 \\ u_2^2 \end{Bmatrix}^j = \mathbf{\Phi} \mathbf{u}^j \tag{4.45}$$

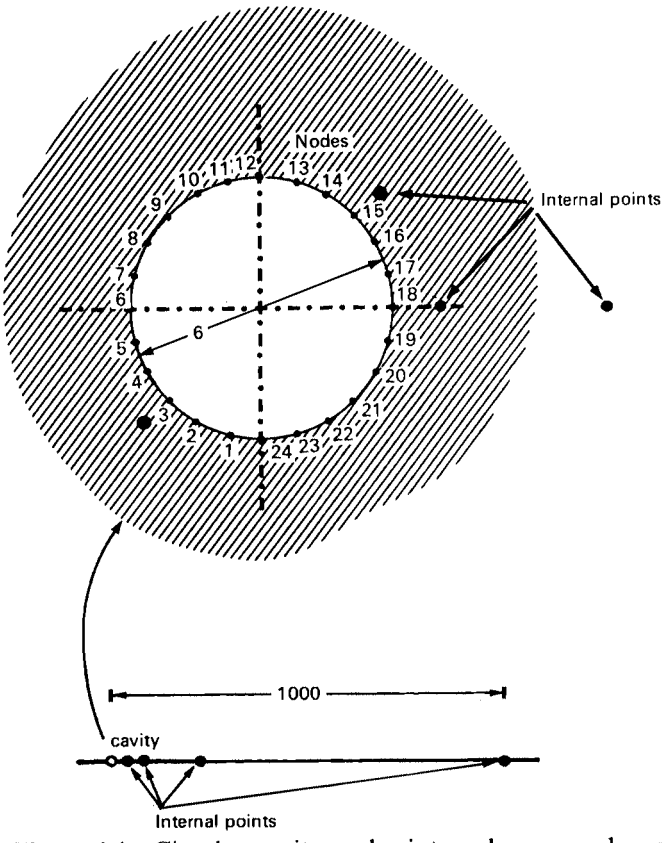


Figure 4.4 Circular cavity under internal pressure boundary element mesh and internal point description

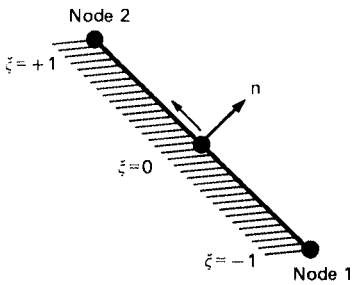


Figure 4.5 Linear element

$$\mathbf{p} = \begin{Bmatrix} p_1 \\ p_2 \end{Bmatrix} = \begin{bmatrix} \phi_1 & 0 & \phi_2 & 0 \\ 0 & \phi_1 & 0 & \phi_2 \end{bmatrix} \begin{Bmatrix} p_1^1 \\ p_2^1 \\ p_1^2 \\ p_2^2 \end{Bmatrix}^j = \mathbf{\Phi p}^j \tag{4.46}$$

where $\mathbf{u}^j$ and $\mathbf{p}^j$ refer to the nodal components of elements j. The components of these vectors are u_l^k and p_l^k where k represents the node under consideration within the element and l defines the component of displacements or tractions in the l direction.

The functions ϕ_i are linear interpolation functions, such that,

$$\begin{aligned}\phi_1 &= -\tfrac{1}{2}(\xi - 1)\\ \phi_2 &= \tfrac{1}{2}(\xi + 1)\end{aligned} \tag{4.47}$$

If we consider N linear boundary elements the governing equation after neglecting boundary forces for simplicity, becomes,

$$\mathbf{c}^i\mathbf{u}^i + \sum_{j=1}^{N}\left\{\int_{\Gamma_j}\mathbf{p}^*\boldsymbol{\Phi}\,d\Gamma\right\}\mathbf{u}^j = \sum_{j=1}^{N}\left\{\int_{\Gamma_j}\mathbf{u}^*\boldsymbol{\Phi}\,d\Gamma\right\}\mathbf{p}^j \tag{4.48}$$

Some of these integrals can be evaluated using numerical integration, others – those with the singularity – can still be computed analytically. Once they are all found, all the elements contribution can be assembled in much the same way as for the linear potential problem (section 2.5). A major difference with the linear potential problem is that now we have two unknowns per node instead of one.

Corner Points

Another important difference with the linear potential problem is the type of $\mathbf{c}^i$ coefficients (equation (4.48)) required in the elasticity solution at corners. While these coefficients in the linear potential case were associated with the value of the solid angle at the corner, those for the elasticity problems are more complicated to find.

For smooth boundaries the $\mathbf{c}^i$ is simply a diagonal matrix with $\frac{1}{2}$ on the diagonal. When the point i is at a corner however (figure 4.6) the limit of the fundamental solution tractions, i.e.

$$[I] = \lim_{\varepsilon\to 0}\left\{\int_{\Gamma_\varepsilon}\mathbf{p}^*\,d\Gamma\right\} \tag{4.49}$$

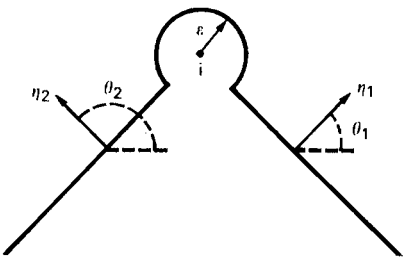


Figure 4.6 Corner point

gives a different result. For instance, for the two dimensional problems discussed here the above limit instead of a $-\frac{1}{2}$ diagonal matrix gives the following result,

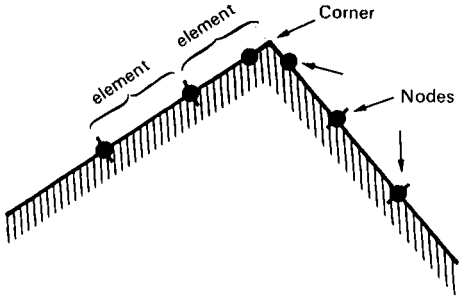
$$[I] = \frac{-1}{8\pi(1-\nu)} \begin{bmatrix} 4(1-\nu)(\pi + \theta_2 - \theta_1) + \sin 2\theta_1 - \sin 2\theta_2, & \cos 2\theta_2 - \cos 2\theta_1 \\ \cos 2\theta_2 - \cos 2\theta_1, & 4(1-\nu)(\pi + \theta_2 - \theta_1) + \sin 2\theta_2 - \sin 2\theta_1 \end{bmatrix}$$

so that,

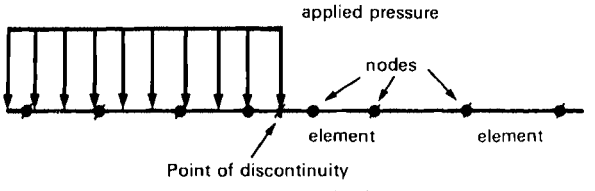
$$c_{lk} = \delta_{lk} + I_{lk} \tag{4.50}$$

It is much more complex to obtain a general expression for I_{lk} in three dimensions since the slope discontinuity may be of different types. In principle however, one may always do the integration over the corresponding portion of the spherical surface.

The simplest way of computing the diagonal submatrices of the **H** matrix, which includes the **c** submatrices is using rigid body consideration as shown in Chapter 3 for bounded domains or regions tending to infinity.



(a) Use of discontinuous elements at corners



(b) Use of discontinuous elements to simulate a sudden change in boundary conditions

Figure 4.7 Linear discontinuous elements

Boundary Conditions, Discontinuous Elements

As discussed in section 3.5 different strategies can be followed to define boundary conditions at corners or points of discontinuity. A simple way to solve this problem is by introducing the concept of discontinuous elements (figure 4.7) which implies moving the points inside the element when the number of unknowns at the corner – or point of discontinuity – is more than one in each direction. This allows the user to apply all types of boundary conditions in adjacent elements in a rather simple fashion. Furthermore the approach can be used even in the presence of high stress concentration or singularities at the corner giving in this case also acceptable results when the discretization around the singularity is sufficiently fine.

Following the same approach as in the potential case the discontinuous element formulation can be written as follows.

The displacements and tractions over a linear element are written in terms of the extreme point values as given by equations (4.45) and (4.46) respectively. If the two nodes are shifted from the two ends distances a and b, respectively (see figure 2.11) equations (4.45) and (4.76) can be particularized for those two points. For instance in the case of displacements

$$\begin{Bmatrix} u_1^a \\ u_2^a \\ u_1^b \\ u_2^b \end{Bmatrix} = \begin{bmatrix} \phi_1(\xi_a) & 0 & \phi_2(\xi_a) & 0 \\ 0 & \phi_1(\xi_a) & 0 & \phi_2(\xi_b) \\ \phi_1(\xi_b) & 0 & \phi_2(\xi_b) & 0 \\ 0 & \phi_1(\xi_b) & 0 & \phi_2(\xi_b) \end{bmatrix} \begin{Bmatrix} u_1^1 \\ u_2^1 \\ u_1^2 \\ u_2^2 \end{Bmatrix} \tag{4.51}$$

where $\xi_a = (2a/l) - 1$ and $\xi_b = 1 - (2b/l)$

Equation (4.51) can be inverted and after substitution in (4.45) yields the value of **u** at any point on the element in terms of the nodal values

$$\mathbf{u} = \begin{bmatrix} \phi_1 & 0 & \phi_2 & 0 \\ 0 & \phi_1 & 0 & \phi_2 \end{bmatrix} \mathbf{Q} \begin{Bmatrix} u_1^1 \\ u_2^1 \\ u_1^2 \\ u_2^2 \end{Bmatrix} \tag{4.52}$$

where **Q** is the inverse of the matrix in equation 4.51, i.e.

$$\mathbf{Q} = \frac{1}{l-a-b} \begin{bmatrix} l-b & 0 & -a & 0 \\ 0 & l-b & 0 & -a \\ -b & 0 & l-a & 0 \\ 0 & -b & 0 & l-a \end{bmatrix} \tag{4.53}$$

The same relation can be written for the tractions

$$\mathbf{p} = \begin{bmatrix} \phi_1 & 0 & \phi_2 & 0 \\ 0 & \phi_1 & 0 & \phi_2 \end{bmatrix} \mathbf{Q} \begin{Bmatrix} p_1^1 \\ p_2^1 \\ p_1^2 \\ p_2^2 \end{Bmatrix} \tag{4.54}$$

After discretizing the boundary into N elements equation (4.48) can be written with the only difference that for those elements which are discontinuous the integrals over the elements take the form:

$$\begin{Bmatrix} \int_{\Gamma_j} \mathbf{p}^* \boldsymbol{\Phi} \mathbf{Q} \, d\Gamma \end{Bmatrix} \mathbf{u}^j$$
$$\begin{Bmatrix} \int_{\Gamma_j} \mathbf{u}^* \boldsymbol{\Phi} \mathbf{Q} \, d\Gamma \end{Bmatrix} \mathbf{p}^j \tag{4.55}$$

when solving an elastic problem continuous and discontinuous elements can be used together in the same mesh with the only difference that for the latter the integrals in equation (4.55) should be used instead of those in (4.48). Numerical integration is used when the collocation point is not on the integration element. Otherwise the integrals for the discontinuous elements are carried at analytically after subdividing the element in two parts at both sides of the node. Notice that $\mathbf{c}^i$ matrix when the node is shifted towards the interior of the element is always that of a smooth boundary point.

4.7 Quadratic Elements

Constant and linear boundary elements are well suited to solve many plane elasticity problems including those with infinite as well as finite domains and some problems which present regions of stress concentration. Their main limitation is that they can not represent properly curved geometries. Problems involving flexure also require the use of higher order elements as their deformations are difficult to model using linear elements (constant elements for this type of problem give very poor results).

The simplest and more versatile type of curved boundary element is the quadratic for which the displacements and tractions can be represented as (figure 4.8)

$$u = \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} = \begin{bmatrix} \phi_1 & 0 & \phi_2 & 0 & \phi_3 & 0 \\ 0 & \phi_1 & 0 & \phi_2 & 0 & \phi_3 \end{bmatrix} \begin{Bmatrix} u_1^1 \\ u_2^1 \\ u_1^2 \\ u_2^2 \\ u_1^3 \\ u_2^3 \end{Bmatrix}^j = \mathbf{\Phi u}^j \tag{4.56}$$

$$p = \begin{Bmatrix} p_1 \\ p_2 \end{Bmatrix} = \begin{bmatrix} \phi_1 & 0 & \phi_2 & 0 & \phi_3 & 0 \\ 0 & \phi_1 & 0 & \phi_2 & 0 & \phi_3 \end{bmatrix} \begin{Bmatrix} p_1^1 \\ p_2^1 \\ p_1^2 \\ p_2^2 \\ p_1^3 \\ p_2^3 \end{Bmatrix}^j = \mathbf{\Phi p}^j \tag{4.57}$$

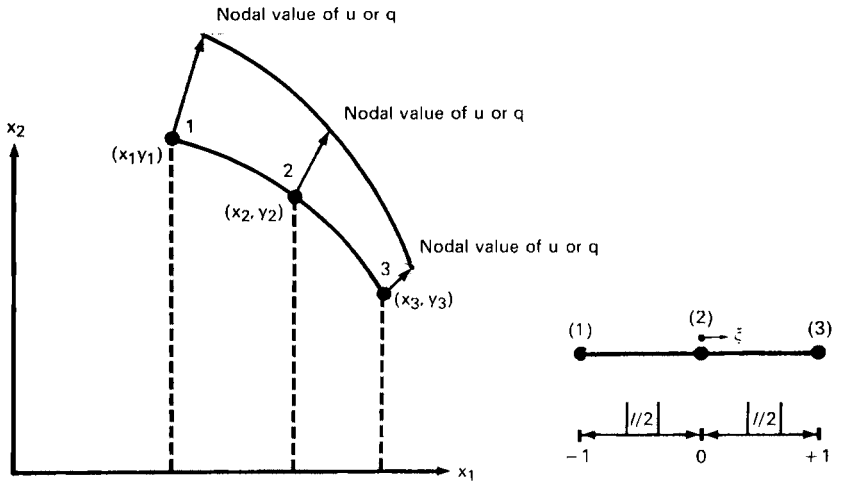
The ϕ_i are quadratic interpolation functions such that,

$$\begin{aligned} \phi_1 &= \tfrac{1}{2}\xi(\xi - 1) \\ \phi_2 &= (1 - \xi^2) \\ \phi_3 &= \tfrac{1}{2}\xi(\xi + 1) \end{aligned} \tag{4.58}$$

where ξ is the dimensionless coordinate along the element (figure 4.7).

The geometry of the element can also be considered as quadratic and is represented by the nodes coordinates and the same interpolation functions ϕ_i used for displacements and traction components, i.e.

$$\mathbf{x} = \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} = \begin{bmatrix} \phi_1 & 0 & \phi_2 & 0 & \phi_3 & 0 \\ 0 & \phi_1 & 0 & \phi_2 & 0 & \phi_3 \end{bmatrix} \begin{Bmatrix} x_1^1 \\ x_2^1 \\ x_1^2 \\ x_2^2 \\ x_1^3 \\ x_2^3 \end{Bmatrix}^j = \mathbf{\Phi x}^j \tag{4.59}$$



Node	ξ	ϕ_1	ϕ_2	ϕ_3
1	−1	1	0	0
2	0	0	1	0
3	1	0	0	1

Figure 4.8 Quadratic element

The discretized boundary integral equations can be written as follows,

$$\mathbf{c}^i\mathbf{u}^i + \sum_{j=1}^{NE} \left\{ \int_{\Gamma_j} \mathbf{p}^*\boldsymbol{\Phi}\, d\Gamma \right\} \mathbf{u}^j = \sum_{j=1}^{NE} \left\{ \int_{\Gamma_j} \mathbf{u}^*\boldsymbol{\Phi}\, d\Gamma \right\} \mathbf{p}^j \tag{4.60}$$

where NE is the number of elements.

The integrals along $d\Gamma$ need now to be transformed to the homogeneous coordinate system ξ as was done in section 2.7 for potential problems. Hence the element integrals in (4.60) become,

$$\begin{aligned} \int_{\Gamma_j} \mathbf{p}^*\boldsymbol{\Phi}\, d\Gamma &= \int_{-1}^{+1} \mathbf{p}^*\boldsymbol{\Phi} |G|\, d\xi \\ \int_{\Gamma_j} \mathbf{u}^*\boldsymbol{\Phi}\, d\Gamma &= \int_{-1}^{+1} \mathbf{u}^*\boldsymbol{\Phi} |G|\, d\xi \end{aligned} \tag{4.61}$$

where the Jacobian $|G|$ is,

$$|G| = \frac{d\Gamma}{d\xi} = \left\{ \left(\frac{dx_1}{d\xi} \right)^2 + \left(\frac{dx_2}{d\xi} \right)^2 \right\}^{1/2} \tag{4.62}$$

where $dx_1/d\xi$ and $dx_2/d\xi$ are easily obtained by differentiating (4.59)in terms of ξ.

4.8 Elastostatics Code using Quadratic Elements (ELQUABE)

In what follows a FORTRAN code for isotropic elasticity is described. The program has the same organization as the constant element code previously studied.

All integer variables in this code have the same meaning as for the constant element program (ELCONBE). The same applies to all real arrays with the exception of XM and YM which are not required now. Also FI and DFI have a slightly different form. The dimension of FI is $(2N)$, N being the number of nodes and that of DFI is $(3N)$ or what is the same $(6NE)$, where NE is the number of elements. The prescribed boundary conditions are read in DFI (six per element). FI is used as the right hand side vector for the solution of the system of equations. Vectors FI and DFI are reordered after solution in such a way that all values of displacements are stored in FI and tractions in DFI.

The program allows for the traction values at both sides of the nodes connecting two elements to be different. This is a similar treatment as was done for linear and quadratic potential elements (codes POLINBE and POQUABE, sections 2.6 and 2.9). For each particular x_1 or x_2 direction, (i) when both tractions are prescribed as different at both sides of the node the displacement is the only unknown; (ii) when the displacement and one traction is prescribed one on each side the other traction is the unknown and (iii) if only the displacement is prescribed, one value of the traction is the only unknown and this will be the same at both sides of the node.

In problems with only one region the case of having more unknowns than boundary conditions at a corner point seldom occurs and does not present difficulties. If the displacement is known along the two boundary elements that converge into a corner, the derivatives along these two directions are known as well, thus strains, stresses and tractions are known. One only needs to prescribe two variables along each direction and leave the third (displacement, traction before the node or after the node) as a check that it coincides with its known value. Only in cases of discontinuities of the stress tensor at certain corners one needs to implement the problem of corners in a different way. The same happens for non-homogeneous media. An accurate and simple procedure is the use of

discontinuous elements, i.e. elements that have one of the node shifted inside the element in such a way that a different node will be created for each element converging into the corner.

1. Main Program

This program follows the same structure as the constant elasticity code in section 4.5.

Its present dimensions allow for 50 elements and 100 nodes.

The listing is as follows:

```
C-----------------------------------------------------------------------

      PROGRAM ELQUABE
C
C  PROGRAM 36
C
C
C  THIS PROGRAM SOLVES TWO DIMENSIONAL (EL)ASTIC PROBLEMS
C  USING (QUA)DRATIC (B)OUNDARY (E)LEMENTS
C
      CHARACTER*10 FILEIN,FILEOUT
C
      COMMON/MATG/ G(100,150)
      COMMON/MATH/ H(100,100)
      COMMON N,L,GE,XNU,INP,IPR
      DIMENSION X(51),Y(51)
      DIMENSION DFI(150),FI(100),KODE(150)
      DIMENSION CX(20),CY(20),SSOL(60),DSOL(40)
C
C  SET MAXIMUN DIMENSION OF THE SYSTEM OF EQUATIONS (NX)
C  NX=  2*MAXIMUN NUMBER OF NODES= 4*MAXIMUN NUMBER OF ELEMENTS
C  NX1= 3*MAXIMUN NUMBER OF NODES= 6*MAXIMUN NUMBER OF ELEMENTS
C
      NX=100
      NX1=150
C
C  ASSIGN NUMBERS FOR INPUT AND OUTPUT FILES
C
      INP=5
      IPR=6
C
C  READ NAMES AND OPEN  FILES FOR INPUT AND OUTPUT
C
      WRITE(*,' (A) ') ' NAME OF INPUT FILE (MAX. 10 CHART.)'
      READ(*,'  (A) ')FILEIN
      OPEN(INP,FILE=FILEIN,STATUS='OLD')
      WRITE(*,' (A) ') ' NAME OF OUTPUT FILE (MAX. 10 CHART.)'
      READ(*,' (A) ')FILEOUT
      OPEN(IPR,FILE=FILEOUT,STATUS='NEW')
C
C  READ DATA
C
      CALL INPUTEQ(CX,CY,X,Y,KODE,DFI)
C
C  COMPUTE G AND H MATRICES AND FORM SYSTEM (A X = F)
C
      CALL GHMATEQ(X,Y,G,H,FI,DFI,KODE,NX,NX1)
C
C  SOLVE SYSTEM OF EQUATIONS
C
      NN=2*N
      CALL SLNPD(H,FI,D,NN,NX)
```

```
C
C  COMPUTE STRESS AND DISPLACEMENT VALUES AT INTERNAL POINTS.
C
      CALL INTEREQ(FI,DFI,KODE,CX,CY,X,Y,SSOL,DSOL)
C
C  PRINT RESULTS AT BOUNDARY NODES AND INTERNAL POINTS
C
      CALL OUTPTEQ(X,Y,FI,DFI,CX,CY,SSOL,DSOL)
C
C   CLOSE INPUT AND OUTPUT FILES
C
      CLOSE (INP)
      CLOSE (IPR)
      STOP
      END
```

2. Routine INPUTEQ

This subroutine reads all the input required by the program and requests a file from the user containing the following lines of free format input;

(i) *Title Line* Contains the title of the problem.

(ii) *Basic Parameter Lines* Contains the number of elements, number of internal points, the shear modulus and the Poisson ratio (or fictitious Poisson's ratio for plane stress problems for which $v' = v/(1 + v)$).

(iii) *Boundary Nodes Coordinates Lines* The coordinates are read counter-clockwise for external boundaries and clockwise for internal ones.

(iv) *Boundary Conditions Lines* As many lines as boundary elements. Six values of KODE and the known variables are read for each element, corresponding to the three nodes and two directions per node. In this way, a value of a traction may be prescribed for a node as part of one element and a different value as part of the other element. The displacement however must be unique for any node. The variables are displacement if KODE = 0 or traction if KODE = 1.

(v) *Internal Points Coordinate Lines* The x_1x_2 coordinates for each point are read in free format. There will be one or more lines, if necessary, i.e. if internal points have been defined.

This subroutine first prints the name of the run and the basic parameters. Then the coordinates of the nodes and the boundary conditions given by element, with codes and prescribed values, are printed. The internal point coordinates will be printed in the subroutine OUTPTEQ.

The FORTRAN listing of INPUTEQ is as follows:

```
C-----------------------------------------------------------------------
      SUBROUTINE INPUTEQ(CX,CY,X,Y,KODE,DFI)
C
C  PROGRAM 37
C
C  NE= NUMBER OF BOUNDARY ELEMENTS
C  L = NUMBER OF INTERNAL POINTS
C  GE= SHEAR MODULUS
```

```
C  XNU= POISSON RATIO
C
      CHARACTER*80 TITLE
      COMMON N,L,GE,XNU,INP,IPR
      DIMENSION KODE(1),X(1),Y(1),CX(1),CY(1),DFI(1)
      WRITE(IPR,100)
  100 FORMAT(' ',79('*'))
C
C  READ JOB TITLE
C
      READ(INP,'(A)') TITLE
      WRITE(IPR,'(A)') TITLE
C
C  READ NUMBER OF BOUNDARY ELEMENTS AND INTERNAL POINTS;
C  READ MATERIAL PROPERTIES
C
      READ(INP,*)NE,L,GE,XNU
      WRITE(IPR,210)NE,L,GE,XNU
  210 FORMAT(//2X,'DATA'/2X,'NUMBER OF BOUNDARY ELEMENTS=',
     1I3/2X,'NUMBER OF INTERNAL POINTS=',I3/2X,'SHEAR MODULUS=',
     2E14.7/2X,'POISSON RATIO=',E14.7)
      N=2*NE
C
C  READ BOUNDARY NODES COORDINATES
C
      N=2*NE
      WRITE(IPR,500)
      READ(INP,*) (X(I),Y(I),I=1,N)
      DO 10 I=1,N
   10 WRITE(IPR,240) I,X(I),Y(I)
  500 FORMAT(//2X,'BOUNDARY NODES COORDINATES'///4X,
     1'NODE',10X,'X',18X,'Y'/)
  240 FORMAT(5X,I3,2(5X,E14.7))
C
C  READ BOUNDARY CONDITIONS IN DFI(I) VECTOR, IF KODE(I)=0
C  THE DFI(I) VALUE IS A KNOWN DISPLACEMENT; IF KODE(I)=1 THE
C  DFI(I) VALUE IS A KNOWN TRACTION.
C  SIX BOUNDARY CONDITIONS ARE READ PER ELEMENT.
C  NODES BETWEEN TWO ELEMENTS MAY HAVE TWO DIFFERENT VALUES
C  OF THE TRACTION BUT ONLY ONE VALUE OF THE DISPLACEMENT.
C
      WRITE(IPR,800)
  800 FORMAT(//2X,'BOUNDARY CONDITIONS'//35X,'PRESCRIBED VALUES' /
     17X,'-----FIRST NODE------',3X,'-----SECOND NODE-----',3X,
     2'-----THIRD NODE------'/
     31X,'ELE',3X,'X DIR.',2X,'C',3X,'Y DIR.',2X,'C',3X,
     4'X DIR.',2X,'C',3X,'Y DIR.',2X,'C',3X,'X DIR.',
     52X,'C',3X,'Y DIR.',2X,'C'/)
      DO 20 I=1,NE
      READ(INP,*) (KODE(6*I-6+J),DFI(6*I-6+J),J=1,6)
   20 WRITE(IPR,950)I,(DFI(6*I-6+J),KODE(6*I-6+J),J=1,6)
  950 FORMAT(1X,I3,6(F9.3,2X,I1))
C
C  READ INTERNAL POINTS COORDINATES
C
      IF(L.EQ.0) GO TO 30
      READ(INP,*) (CX(I),CY(I),I=1,L)
   30 RETURN
      END
```

3. Routine GHMATEQ

This subroutine computes the G and H system matrices by calling routines EXTINEQ and LOCINEQ.

EXTINEQ: Computes the GW and HW (2×6) submatrices which relates a collocation point with an element as defined by its three nodes.

LOCINEQ: Computes the GW (2×6) submatrix for the case when the collocation point is one of the nodes within the element under consideration (i.e. the singularity is in the same element). Notice that the corresponding HW (2×6) is computed using EXTINEQ except for that part that relates a node with itself which is computed using rigid body considerations, which results in adding row coefficients.

The resulting GW and HW submatrices are assembled in the **G** and **H** system matrices. Matrix G is now rectangular since each extreme node of an element may have different tractions, i.e. one 'before' and another 'after' the node.

Once the matrices H and G are assembled, the system of equations needs to be reordered in accordance with the boundary conditions to form

$$\mathbf{AX} = \mathbf{F}$$

where **X** is a $(2N)$ vector of unknowns, N being the number of nodes; **A** is a $(2N \times 2N)$ matrix whose columns are a combination of columns of **H** or **G** depending on the boundary conditions or of two consecutive columns of **G** when the unknown is the *unique* value of the tractions at both sides of the extreme node of an element; **F** is a known vector computed by multiplying the prescribed boundary conditions by the corresponding row terms of **G** or **H**.

At the end of the subroutine GHMATEQ and after rearranging H contains the matrix **A**, and **FI** the **F** vector.

The FORTRAN listing of GHMATEQ is as follows.

```
C-----------------------------------------------------------------------
      SUBROUTINE GHMATEQ(X,Y,G,H,FI,DFI,KODE,NX,NX1)
C
C  PROGRAM 38
C
C  THIS SUBROUTINE COMPUTES THE G AND H MATRICES AND FORMS
C  THE SYSTEM A X = F READY TO BE SOLVED
C  H IS A SQUARE MATRIX (4*NE,4*NE); G IS RECTANGULAR (4*NE,6*NE)
C
      DIMENSION X(1),Y(1),G(NX,NX1),H(NX,NX),HW(2,6),GW(2,6)
      DIMENSION FI(1),DFI(1),KODE(1)
      COMMON N,L,GE,XNU,INP,IPR
      NN=2*N
      NE=N/2
      DO 20 I=1,NN
      DO 11 J=1,NN
   11 H(I,J)=0.
      DO 12 J=1,3*N
   12 G(I,J)=0.
   20 CONTINUE
      X(N+1)=X(1)
      Y(N+1)=Y(1)
C
C  COMPUTE THE GW AND HW MATRICES FOR EACH COLLOCATION
C  POINT AND EACH BOUNDARY ELEMENT
C
      DO 40 LL=1,N
      DO 40 I=1,N-1,2
      IF((LL-I)*(LL-I-1)*(LL-I-2)*(LL-I+N-2)) 22,21,22
   21 NODO=LL-I+1
      IF((LL.EQ.1).AND.(I.EQ.N-1)) NODO=NODO+N
      CALL EXTINEQ(X(LL),Y(LL),X(I),Y(I),X(I+1),Y(I+1),X(I+2),Y(I+2),
     *HW,GW)
```

```
      CALL LOCINEQ(X(I),Y(I),X(I+1),Y(I+1),X(I+2),Y(I+2),GW,NODO)
      GO TO 34
   22 CALL EXTINEQ(X(LL),Y(LL),X(I),Y(I),X(I+1),Y(I+1),X(I+2),Y(I+2),
     *HW,GW)
C
C  PLUG THE GW AND HW MATRICES INTO THE GENERAL G AND H MATRICES.
C
   34 DO 39 K=1,2
      DO 38 J=1,6
      G(2*LL-2+K,3*I-3+J)=G(2*LL-2+K,3*I-3+J)+GW(K,J)
      IF(I-N+1) 37,35,37
   35 IF(J-5) 37,36,36
   36 H(2*LL-2+K,J-4)=H(2*LL-2+K,J-4)+HW(K,J)
      GO TO 38
   37 H(2*LL-2+K,2*I-2+J)=H(2*LL-2+K,2*I-2+J)+HW(K,J)
   38 CONTINUE
   39 CONTINUE
   40 CONTINUE
C
C  COMPUTE THE DIAGONAL COEFFICIENTS OF THE H MATRIX
C
      DO 70 I=1,N
      H(2*I-1,2*I-1)=0.
      H(2*I,2*I-1)=0.
      H(2*I-1,2*I)=0.
      H(2*I,2*I)=0.
      DO 60 J=1,N
      IF(I.EQ.J) GO TO 60
      H(2*I-1,2*I-1)=H(2*I-1,2*I-1)-H(2*I-1,2*J-1)
      H(2*I,2*I-1)=H(2*I,2*I-1)-H(2*I,2*J-1)
      H(2*I-1,2*I)=H(2*I-1,2*I)-H(2*I-1,2*J)
      H(2*I,2*I)=H(2*I,2*I)-H(2*I,2*J)
   60 CONTINUE
C
C  ADD ONE TO THE DIAGONAL COEFFICIENTS FOR
C  EXTERNAL PROBLEMS.
C
      IF(H(2*I-1,2*I-1)) 65,70,70
   65 H(2*I-1,2*I-1)=1.+H(2*I-1,2*I-1)
      H(2*I,2*I)=1.+H(2*I,2*I)
   70 CONTINUE
C
C  REORDER THE COLUMNS OF THE SYSTEM OF EQUATIONS IN ACCORDANCE
C  WITH THE BOUNDARY CONDITIONS AND FORM SYSTEM MATRIX A WHICH
C  IS STORED IN H
C
      DO 180 I=1,NE
      DO 170 J=1,6
      IF(KODE(6*I-6+J)) 110,110,170
  110 IF((I-NE).NE.0 .OR. J.LT.5) GO TO 125
      IF(KODE(J-4)) 115,115,113
  113 DO 114 K=1,NN
      CH=H(K,J-4)
      H(K,J-4)=-G(K,6*I-6+J)*GE
  114 G(K,6*I-6+J)=-CH
      GO TO 170
  115 DO 116 K=1,NN
      H(K,J-4)=H(K,J-4)-G(K,6*I-6+J)*GE
  116 G(K,6*I-6+J)=0.
      GO TO 170
  125 IF(I.EQ.1 .OR. J.GT.2 .OR. KODE(6*I-8+J).EQ.1) GO TO 130
      DO 129 K=1,NN
      H(K,4*I-4+J)=H(K,4*I-4+J)-G(K,6*I-6+J)*GE
  129 G(K,6*I-6+J)=0.
      GO TO 170
  130 DO 132 K=1,NN
      CH=H(K,4*I-4+J)
      H(K,4*I-4+J)=-G(K,6*I-6+J)*GE
  132 G(K,6*I-6+J)=-CH
  170 CONTINUE
  180 CONTINUE
C
C  FORM THE RIGHT HAND SIDE VECTOR F WHICH IS STORED IN FI
```

```
C
      DO 190 I=1,NN
      FI(I)= 0.
      DO 185 J=1,6*NE
  185 FI(I)=FI(I)+G(I,J)*DFI(J)
  190 CONTINUE
      RETURN
      END
```

4. Routine EXTINEQ

This subroutine computes using numerical integration, the (2×6) submatrices GW and HW that corresponds to an element when the collocation point is at a node other than any of those 3 in the element. The coordinates of the collocation points are XP and YP. The integrals are of the type (4.61), i.e.

$$\mathbf{HW} = \int_{\Gamma_j} \mathbf{p}^* \boldsymbol{\phi}\, d\xi = \int_{-1}^{+1} \mathbf{p}^* \boldsymbol{\phi} |G|\, d\xi \tag{4.63}$$

$$\mathbf{GW} = \int_{\Gamma_j} \mathbf{u}^* \boldsymbol{\phi}\, d\Gamma = \int_{-1}^{+1} \mathbf{u}^* \boldsymbol{\phi} |G|\, d\xi \tag{4.64}$$

They can be expanded as follows,

$$\mathbf{HW} = \int_{-1}^{+1} \begin{bmatrix} p_{11}^* & p_{12}^* \\ p_{21}^* & p_{22}^* \end{bmatrix} \begin{bmatrix} \phi_1 & 0 & \phi_2 & 0 & \phi_3 & 0 \\ 0 & \phi_1 & 0 & \phi_2 & 0 & \phi_3 \end{bmatrix} |G|\, d\xi \tag{4.65}$$

$$\mathbf{GW} = \int_{-1}^{+1} \begin{bmatrix} u_{11}^* & u_{12}^* \\ u_{21}^* & u_{22}^* \end{bmatrix} \begin{bmatrix} \phi_1 & 0 & \phi_2 & 0 & \phi_3 & 0 \\ 0 & \phi_1 & 0 & \phi_2 & 0 & \phi_3 \end{bmatrix} |G|\, d\xi \tag{4.66}$$

The Jacobians are calculated by taking derivatives of the expressions for the x_1 and x_2 coordinates, in the same way that was done in section 2.9 for potential problems using quadratic elements.

The FORTRAN listing is as follows:

```
C-----------------------------------------------------------------------
      SUBROUTINE EXTINEQ(XP,YP,X1,Y1,X2,Y2,X3,Y3,HW,GW)
C
C  PROGRAM 39
C
C  THIS SUBRUOTINE COMPUTES THE HW AND GW MATRICES
C  THAT RELATE A NODE (XP,YP) WITH A BOUNDARY
C  ELEMENT USING GAUSS QUADRATURE.
C
C  RA            = RADIUS
C  RD1,RD2,RDN   = RADIUS DERIVATIVES
C  ETA1,ETA2     = COMPONENTS OF THE UNIT NORMAL TO THE ELEMENT
C  XCO,YCO       = INTEGRATION POINT ALONG THE ELEMENT
C  XJA           = JACOBIAN
C
      COMMON N,L,GE,XNU,INP,IPR
      DIMENSION GW(2,6),HW(2,6)
      DIMENSION GI(10),OME(10)
      DATA GI/0.9739065285,-0.9739065285,0.8650633666,-0.8650633666
```

```
     @,0.6794095683,-0.6794095682,0.4333953941,-0.4333953941,
     @0.1488743389,-0.1488743389/
      DATA OME/0.0666713443,0.0666713443,0.1494513491,0.1494513491
     @,0.2190863625,0.2190863625,0.2692667193,0.2692667193,
     @0.2955242247,0.2955242247/
      DO 30 I=1,2
      DO 20 J=1,6
      HW(I,J)=0.
   20 GW(I,J)=0.
   30 CONTINUE
      A=X3-2*X2+X1
      B=(X3-X1)/2
      C=Y3-2*Y2+Y1
      D=(Y3-Y1)/2
      DE=4*3.141592*(1-XNU)
      DO 40 I=1,10
C
C  COMPUTE THE VALUES OF THE SHAPE FUNCTIONS AT THE
C  INTEGRATION POINTS
C
      F1=GI(I)*(GI(I)-1)*0.5
      F2=1.-GI(I)**2
      F3=GI(I)*(GI(I)+1)*0.5
C
C  COMPUTE GEOMETRICAL PROPERTIES AT THE INTEGRATION POINTS
C
      XCO=X1*F1+X2*F2+X3*F3
      YCO=Y1*F1+Y2*F2+Y3*F3
      XJA=SQRT((GI(I)*A+B)**2+(GI(I)*C+D)**2)
      ETA1=(GI(I)*C+D)/XJA
      ETA2=-(GI(I)*A+B)/XJA
      RA=SQRT((XP-XCO)**2+(YP-YCO)**2)
      RD1=(XCO-XP)/RA
      RD2=(YCO-YP)/RA
      RDN=RD1*ETA1+RD2*ETA2
C
C  COMPUTE GW AND HW MATRICES
C
      GW(1,1)=GW(1,1)+((3-4*XNU)*ALOG(1./RA)+RD1**2)*OME(I)*XJA*F1/
     1(2*DE*GE)
      GW(1,2)=GW(1,2)+RD1*RD2*OME(I)*XJA*F1/(2*DE*GE)
      GW(2,1)=GW(1,2)
      GW(2,2)=GW(2,2)+((3-4*XNU)*ALOG(1./RA)+RD2**2)*OME(I)*XJA*F1/
     1(2*DE*GE)
      HW(1,1)=HW(1,1)-RDN*((1-2*XNU)+2*RD1**2)/(RA*DE)*OME(I)*XJA*F1
      HW(1,2)=HW(1,2)-(RDN*2*RD1*RD2+(1-2*XNU)*(ETA1*RD2-ETA2*RD1))*
     1OME(I)*XJA*F1/(RA*DE)
      HW(2,1)=HW(2,1)-(RDN*2*RD1*RD2+(1-2*XNU)*(ETA2*RD1-ETA1*RD2))*
     1OME(I)*XJA*F1/(RA*DE)
      HW(2,2)=HW(2,2)-RDN*((1-2*XNU)+2*RD2**2)*OME(I)*XJA*F1/(RA*DE)
      GW(1,3)=GW(1,3)+((3-4*XNU)*ALOG(1./RA)+RD1**2)*OME(I)*XJA*F2/
     1(2*DE*GE)
      GW(1,4)=GW(1,4)+RD1*RD2*OME(I)*XJA*F2/(2*DE*GE)
      GW(2,3)=GW(1,4)
      GW(2,4)=GW(2,4)+((3-4*XNU)*ALOG(1./RA)+RD2**2)*OME(I)*XJA*F2/
     1(2*DE*GE)
      HW(1,3)=HW(1,3)-RDN*((1-2*XNU)+2*RD1**2)/(RA*DE)*OME(I)*XJA*F2
      HW(1,4)=HW(1,4)-(RDN*2*RD1*RD2+(1-2*XNU)*(ETA1*RD2-ETA2*RD1))*
     1OME(I)*XJA*F2/(RA*DE)
      HW(2,3)=HW(2,3)-(RDN*2*RD1*RD2+(1-2*XNU)*(ETA2*RD1-ETA1*RD2))*
     1OME(I)*XJA*F2/(RA*DE)
      HW(2,4)=HW(2,4)-RDN*((1-2*XNU)+2*RD2**2)*OME(I)*XJA*F2/(RA*DE)
      GW(1,5)=GW(1,5)+((3-4*XNU)*ALOG(1./RA)+RD1**2)*OME(I)*XJA*F3/
     1(2*DE*GE)
      GW(1,6)=GW(1,6)+RD1*RD2*OME(I)*XJA*F3/(2*DE*GE)
      GW(2,5)=GW(1,6)
      GW(2,6)=GW(2,6)+((3-4*XNU)*ALOG(1./RA)+RD2**2)*OME(I)*XJA*F3/
     1(2*DE*GE)
      HW(1,5)=HW(1,5)-RDN*((1-2*XNU)+2*RD1**2)/(RA*DE)*OME(I)*XJA*F3
      HW(1,6)=HW(1,6)-(RDN*2*RD1*RD2+(1-2*XNU)*(ETA1*RD2-ETA2*RD1))*
     1OME(I)*XJA*F3/(RA*DE)
      HW(2,5)=HW(2,5)-(RDN*2*RD1*RD2+(1-2*XNU)*(ETA2*RD1-ETA1*RD2))*
```

```
     1OME(I)*XJA*F3/(RA*DE)
   40 HW(2,6)=HW(2,6)-RDN*((1-2*XNU)+2*RD2**2)*OME(I)*XJA*F3/(RA*DE)
      RETURN
      END
```

5. Routine LOCINEQ

This subroutine computes using standard Gauss quadrature and a special quadrature formula, the (2×6) submatrix GW when the collocation point is one of the nodes within the element under consideration.

The integration process is completely analogous to that shown for potential problems in section 2.8. The integrals are split into two parts: one without singularity which is integrated by means of the standard formula; and the other, the loarithmic part, that is integrated using a special quadrature formula. More details of the integration process are given in section 2.8. The only difference is now that kernels to be integrated have some more terms that in the case of potential problems. Also GW is now (2×6) instead of (3).

The listing for this subroutine is as follows:

```
C-----------------------------------------------------------------------
      SUBROUTINE LOCINEQ(XG1,YG1,XG2,YG2,XG3,YG3,GW,NODO)
C
C  PROGRAM 40
C
C  THIS SUBROUTINE COMPUTES THE GW MATRIX WHEN THE COLLOCATION
C  POINT IS ONE OF THE NODES OF THE INTEGRATION ELEMENT.
C  THE COEFFICIENTS ARE COMPUTED BY NUMERICAL INTEGRATION:
C  THE NON SINGULAR PART IS COMPUTED USING STANDARD GAUSS QUADRATURE,
C  THE LOGARITHMIC PART IS COMPUTED USING A SPECIAL QUADRATURE FORMULA.
C
      COMMON N,L,GE,XNU,INP,IPR
      DIMENSION GI(10),OME(10),GIL(10),OMEL(10),GW(2,6),R(2)
C
C  DATA FOR THE GAUSS QUADRATURE
C
      DATA GI/0.9739065285,-0.9739065285,0.8650633666,-0.8650633666
     @,0.6794095682,-0.6794095682,0.4333953941,-0.4333953941,
     @0.1488743389,-0.1488743389/
      DATA OME/0.0666713443,0.0666713443,0.1494513491,0.1494513491
     @,0.2190863625,0.2190863625,0.2692667193,0.2692667193,
     @0.2955242247,0.2955242247/
C
C  DATA FOR THE SPECIAL QUADRATURE
C
      DATA GIL/0.0090426309,0.0539712662,0.1353118246,0.2470524162
     @,0.3802125396,0.5237923179,0.6657752055,0.7941904160,
     @0.8981610912,0.9688479887/
      DATA OMEL/0.1209551319,0.1863635425,0.1956608732,0.1735771421
     @,0.1356956729,0.0936467585,0.0557877273,0.0271598109,
     @0.0095151826,0.0016381576/
C
C  SET LOCAL COORDINATES SYSTEM
C
      GOTO(1,2,3),NODO
    1 X3=XG3-XG1
      Y3=YG3-YG1
      X2=XG2-XG1
      Y2=YG2-YG1
      A1=(X3-2*X2)*0.5
      B1=X2
      A2=(Y3-2*Y2)*0.5
      B2=Y2
      GO TO 4
```

```
    2 X3=XG3-XG2
      Y3=YG3-YG2
      X1=XG1-XG2
      Y1=YG1-YG2
      A1=X1+X3
      B1=X3-X1
      A2=Y1+Y3
      B2=Y3-Y1
      GO TO 4
    3 X2=XG2-XG3
      Y2=YG2-YG3
      X1=XG1-XG3
      Y1=YG1-YG3
      A1=(X1-2*X2)*0.5
      B1=-X2
      A2=(Y1-2*Y2)*0.5
      B2=-Y2
    4 CONTINUE
C
      DO 10 I=1,2
      DO 10 J=1,6
   10 GW(I,J)=0.
      A=A1**2 + A2**2
      B=2*(A1*B1 + A2*B2)
      C=B1**2 + B2**2
      CONT1=(3-4*XNU)/(8.*3.1415926*GE*(1-XNU))
      CONT2=CONT1/(3-4*XNU)
      DO 250 I=1,10
      T1=((A1*GI(I)+B1)**2)/((A2*GI(I)+B2)**2 + (A1*GI(I)+B1)**2)
      T2=((A2*GI(I)+B2)**2)/((A2*GI(I)+B2)**2 + (A1*GI(I)+B1)**2)
      T3=(A1*GI(I)+B1)*(A2*GI(I)+B2)/((A1*GI(I)+B1)**2+(A2*GI(I)+B2)**2)
C
C  COMPUTE SHAPE FUNCTIONS FOR NUMERICAL INTEGRATION
C
      F3=0.5*GI(I)*(GI(I)+1.)
      F2=1.-GI(I)**2
      F1=0.5*GI(I)*(GI(I)-1.)
      FL3=GIL(I)*(2.*GIL(I)-1.)
      FL2=4.*GIL(I)*(1.-GIL(I))
      FL1=(GIL(I)-1.)*(2.*GIL(I)-1.)
      FLN3=0.5*GIL(I)*(GIL(I)+1.)
      FLN2=1.-GIL(I)**2
      FLN1=0.5*GIL(I)*(GIL(I)-1.)
C
C  COMPUTE GW COEFFICIENTS
C
      GO TO(50,60,70) NODO
   50 XJA1=SQRT((4*A1*GIL(I)-2*A1+0.5*X3)**2+(4*A2*GIL(I)-2*A2+0.5*Y3)**
     *2)*2
      XJA2=SQRT((A1*GI(I)*2+0.5*X3)**2+(A2*GI(I)*2+0.5*Y3)**2)
      XLO=-ALOG(2*SQRT((GI(I)*A1+B1)**2+(GI(I)*A2+B2)**2))
      S3=CONT1*(FL3*XJA1*OMEL(I)+F3*XJA2*XLO*OME(I))
      S2=CONT1*(FL2*XJA1*OMEL(I)+F2*XJA2*XLO*OME(I))
      S1=CONT1*(FL1*XJA1*OMEL(I)+F1*XJA2*XLO*OME(I))
      GO TO 200
   60 XJA1=SQRT((0.5*B1-A1*GIL(I))**2+(0.5*B2-A2*GIL(I))**2)
      XJA11=SQRT((0.5*B1+A1*GIL(I))**2+(0.5*B2+A2*GIL(I))**2)
      XJA2=SQRT((0.5*B1+A1*GI(I))**2+(0.5*B2+A2*GI(I))**2)
      XLO=-0.5*ALOG((GI(I)*A1*0.5+B1*0.5)**2+(GI(I)*A2*0.5+B2*0.5)**2)
      S3=CONT1*((FLN1*XJA1+FLN3*XJA11)*OMEL(I)+F3*XJA2*XLO*OME(I))
      S2=CONT1*(FLN2*(XJA1+XJA11)*OMEL(I)+F2*XJA2*XLO*OME(I))
      S1=CONT1*((FLN3*XJA1+FLN1*XJA11)*OMEL(I)+F1*XJA2*XLO*OME(I))
      GO TO 200
   70 XJA1=SQRT((2*A1-4*A1*GIL(I)-0.5*X1)**2+(2*A2-4*A2*GIL(I)-0.5*Y1)**
     *2)*2
      XJA2=SQRT((2*A1*GI(I)-0.5*X1)**2+(2*A2*GI(I)-0.5*Y1)**2)
      XLO=-ALOG(2*SQRT((A1*GI(I)+B1)**2+(A2*GI(I)+B2)**2))
      S3=CONT1*(FL1*XJA1*OMEL(I)+F3*XJA2*XLO*OME(I))
      S2=CONT1*(FL2*XJA1*OMEL(I)+F2*XJA2*XLO*OME(I))
      S1=CONT1*(FL3*XJA1*OMEL(I)+F1*XJA2*XLO*OME(I))
  200 GW(1,5)= GW(1,5)+S3+CONT2*F3*T1*XJA2*OME(I)
      GW(1,3)= GW(1,3)+S2+CONT2*F2*T1*XJA2*OME(I)
      GW(1,1)= GW(1,1)+S1+CONT2*F1*T1*XJA2*OME(I)
```

```
      GW(1,6)= GW(1,6)+ CONT2*T3*XJA2*F3*OME(I)
      GW(1,4)= GW(1,4)+ CONT2*T3*XJA2*F2*OME(I)
      GW(1,2)= GW(1,2)+ CONT2*T3*XJA2*F1*OME(I)
      GW(2,5)= GW(1,6)
      GW(2,3)= GW(1,4)
      GW(2,1)=GW(1,2)
      GW(2,6)= GW(2,6)+ S3+CONT2*T2*XJA2*F3*OME(I)
      GW(2,4)=GW(2,4)+S2+CONT2*T2*XJA2*F2*OME(I)
      GW(2,2)= GW(2,2)+ S1+CONT2*T2*XJA2*F1*OME(I)
C
  250 CONTINUE
C
      RETURN
      END
```

6. Routine INTEREQ

This subroutine first reorders the vectors DFI and FI in such a way that all the boundary displacements are stored in FI and all the tractions in DFI. It then computes the displacements and stresses at internal points.

The displacement at any interior point is given by

$$\mathbf{u}^i = \sum_{j=1}^{NE} \left\{ \int_{\Gamma_j} \mathbf{u}^* \phi \, d\Gamma \right\} \mathbf{p}^j - \sum_{j=1}^{NE} \left\{ \int_{\Gamma_j} \mathbf{p}^* \phi \, d\Gamma \right\} \mathbf{u}^j \tag{4.67}$$

where the integrals along the boundary elements are computed numerically by calling again the subroutine EXTINEQ.

Analogously, the stresses are given by

$$\sigma_{kl}^i = \sum_{j=1}^{NE} \left\{ \int_{\Gamma_j} \mathbf{D}_{kl} \phi \, d\Gamma \right\} \mathbf{p}^j - \sum_{j=1}^{NE} \left\{ \int \mathbf{S}_{kl} \phi \, d\Gamma \right\} \mathbf{u}^j \tag{4.68}$$

where

$$\mathbf{D}_{kl} = [D_{1kl}, D_{2kl}]; \qquad \mathbf{S}_{kl} = [S_{1kl}, S_{2kl}]$$

The values of D_{mkl} and S_{mkl} have been given in equations (3.110) and (3.111). The integrals along the elements in equation (4.68) are computed by calling the subroutine SIGMAEQ.

The listing of INTEREQ is as follows,

```
C-----------------------------------------------------------------------
      SUBROUTINE INTEREQ(FI,DFI,KODE,CX,CY,X,Y,SSOL,DSOL)
C
C  PROGRAM 41
C
C  THIS SUBROUTINE COMPUTES THE VALUES OF STRESSES AND DISPLACEMENTS AT
C  INTERNAL POINTS.
C
      COMMON N,L,GE,XNU,INP,IPR
      DIMENSION FI(1),DFI(1),KODE(1),CX(1),CY(1)
      DIMENSION X(1),Y(1),SSOL(1),DSOL(1)
      DIMENSION HW(2,6),GW(2,6)
      DIMENSION D11(6),D12(6),D22(6),S11(6),S12(6),S22(6)
C
C  REARRANGE FI AND DFI ARRAYS TO STORE ALL THE VALUES OF THE
C  DISPLACEMENT IN FI AND ALL THE VALUES TRACTION IN DFI
C
```

```
      NE=N/2
      DO 180 I=1,NE
      DO 170 J=1,6
      IF(KODE(6*I-6+J)) 110,110,170
  110 IF((I-NE).NE.0 .OR. J.LT.5) GO TO 125
      IF(KODE(J-4)) 114,114,113
  113 CH=FI(J-4)*GE
      FI(J-4)=DFI(6*I-6+J)
      DFI(6*I-6+J)=CH
      GO TO 170
  114 DFI(6*I-6+J)=DFI(J-4)
      GO TO 170
  125 IF(I.EQ.1 .OR. J.GT.2 .OR. KODE(6*I-8+J).EQ.1) GO TO 130
      DFI(6*I-6+J)=DFI(6*I-8+J)
      GO TO 170
  130 CH=FI(4*I-4+J)*GE
      FI(4*I-4+J)=DFI(6*I-6+J)
      DFI(6*I-6+J)=CH
  170 CONTINUE
  180 CONTINUE
C
C  COMPUTE THE VALUES OF STRESSES AND DISPLACEMENTS AT INTERNAL POINTS
C
      IF(L.EQ.0) GO TO 50
      DO 240 K=1,L
      SSOL(3*K-2)=0.
      SSOL(3*K-1)=0.
      SSOL(3*K)=0.
      DSOL(2*K-1)=0.
      DSOL(2*K)=0.
      DO 230 I=1,NE
      CALL EXTINEQ(CX(K),CY(K),X(2*I-1),Y(2*I-1),X(2*I),Y(2*I),
     1X(2*I+1),Y(2*I+1),HW,GW)
      CALL SIGMAEQ(CX(K),CY(K),X(2*I-1),Y(2*I-1),X(2*I),Y(2*I),
     1X(2*I+1),Y(2*I+1),D11,D12,D22,S11,S12,S22)
      DO 220 J=1,6
      IJ4=4*I-4+J
      IF(IJ4.GT.(4*NE)) IJ4=J-4
      SSOL(3*K-2)=SSOL(3*K-2)+D11(J)*DFI(6*I-6+J)-S11(J)*FI(IJ4)
      SSOL(3*K-1)=SSOL(3*K-1)+D12(J)*DFI(6*I-6+J)-S12(J)*FI(IJ4)
      SSOL(3*K)=SSOL(3*K)+D22(J)*DFI(6*I-6+J)-S22(J)*FI(IJ4)
      DSOL(2*K-1)=DSOL(2*K-1)+GW(1,J)*DFI(6*I-6+J)-HW(1,J)*FI(IJ4)
  220 DSOL(2*K)=DSOL(2*K)+GW(2,J)*DFI(6*I-6+J)-HW(2,J)*FI(IJ4)
  230 CONTINUE
  240 CONTINUE
   50 RETURN
      END
```

7. Routine SIGMAEQ

The integrals of the S and D coefficients multiplied by the shape functions in equation (4.63) are evaluated using Gaussian quadrature. The procedure is similar to that used in routine EXTINEQ, the only difference being that now the expressions of D and S given by equation (3.110) and (3.111) are computed at the integration points instead of the values of u^* and p^* as in EXTINEQ.

The FORTRAN listing of SIGMAEQ is as follows,

```
C-----------------------------------------------------------------------
      SUBROUTINE SIGMAEQ(XP,YP,X1,Y1,X2,Y2,X3,Y3,D11,D12,D22,S11,S12
     *,S22)
C
C  PROGRAM 42
C
C  THIS SUBROUTINE COMPUTES THE VALUES OF THE S AND D MATRICES
C  USING GAUSS QUADRATURE IN ORDER TO COMPUTE THE STRESS
C  AT ANY INTERNAL POINT.
```

```
C  RA            = RADIUS
C  RD1,RD2,RDN   = RADIUS DERIVATIVES
C  ETA1,ETA2     = COMPONENTS OF THE UNIT NORMAL TO THE ELEMENT
C  XCO,YCO       = INTEGRATION POINT ALONG THE ELEMENT
C  XJA           = JACOBIAN
C
      COMMON N,L,GE,XNU,INP,IPR
      DIMENSION D11(6),D12(6),D22(6),S11(6),S12(6),S22(6)
      DIMENSION GI(10),OME(10)
      DATA GI/0.9739065285,-0.9739065285,0.8650633666,-0.8650633666
     @,0.6794095683,-0.6794095682,0.4333953941,-0.4333953941,
     @0.1488743389,-0.1488743389/
      DATA OME/0.0666713443,0.0666713443,0.1494513491,0.1494513491
     @,0.2190863625,0.2190863625,0.2692667193,0.2692667193,
     @0.2955242247,0.2955242247/
      DO 20 J=1,6
      D11(J)=0.
      D12(J)=0.
      D22(J)=0.
      S11(J)=0.
      S12(J)=0.
   20 S22(J)=0.
C
      FA=1-4*XNU
      AL=1-2*XNU
      A=X3-2*X2+X1
      B=(X3-X1)/2
      C=Y3-2*Y2+Y1
      D=(Y3-Y1)/2
      DE=4*3.141592*(1-XNU)
      DO 40 I=1,10
C
C  COMPUTE VALUES OF THE SHAPE FUNCTIONS AT THE INTEGRATION POINTS
C
      F1=GI(I)*(GI(I)-1)*0.5
      F2=1.-GI(I)**2
      F3=GI(I)*(GI(I)+1)*0.5
C
C  COMPUTE GEOMETRICAL PARAMETERS
C
      XCO=X1*F1+X2*F2+X3*F3
      YCO=Y1*F1+Y2*F2+Y3*F3
      XJA=SQRT((GI(I)*A+B)**2+(GI(I)*C+D)**2)
      ETA1=(GI(I)*C+D)/XJA
      ETA2=-(GI(I)*A+B)/XJA
      RA=SQRT((XP-XCO)**2+(YP-YCO)**2)
      RD1=(XCO-XP)/RA
      RD2=(YCO-YP)/RA
      RDN=RD1*ETA1+RD2*ETA2
C
C  COMPUTE D AND S COEFFICIENTS
C
      D11(1)=D11(1)+(AL*RD1+2*RD1**3)*OME(I)*XJA*F1/(DE*RA)
      D11(2)=D11(2)+(2*RD1**2*RD2-AL*RD2)*OME(I)*XJA*F1/(DE*RA)
      D11(3)=D11(3)+(AL*RD1+2*RD1**3)*OME(I)*XJA*F2/(DE*RA)
      D11(4)=D11(4)+(2*RD1**2*RD2-AL*RD2)*OME(I)*XJA*F2/(DE*RA)
      D11(5)=D11(5)+(AL*RD1+2*RD1**3)*OME(I)*XJA*F3/(DE*RA)
      D11(6)=D11(6)+(2*RD1**2*RD2-AL*RD2)*OME(I)*XJA*F3/(DE*RA)
      D12(1)=D12(1)+(AL*RD2+2*RD1**2*RD2)*F1/(DE*RA)*OME(I)*XJA
      D12(2)=D12(2)+(AL*RD1+2*RD1*RD2**2)*F1/(DE*RA)*OME(I)*XJA
      D12(3)=D12(3)+(AL*RD2+2*RD1**2*RD2)*F2/(DE*RA)*OME(I)*XJA
      D12(4)=D12(4)+(AL*RD1+2*RD1*RD2**2)*F2/(DE*RA)*OME(I)*XJA
      D12(5)=D12(5)+(AL*RD2+2*RD1**2*RD2)*F3/(DE*RA)*OME(I)*XJA
      D12(6)=D12(6)+(AL*RD1+2*RD1*RD2**2)*F3/(DE*RA)*OME(I)*XJA
      D22(1)=D22(1)+(2*RD1*RD2**2-AL*RD1)*F1/(DE*RA)*OME(I)*XJA
      D22(2)=D22(2)+(AL*RD2+2*RD2**3)*F1/(DE*RA)*OME(I)*XJA
      D22(3)=D22(3)+(2*RD1*RD2**2-AL*RD1)*F2/(DE*RA)*OME(I)*XJA
      D22(4)=D22(4)+(AL*RD2+2*RD2**3)*F2/(DE*RA)*OME(I)*XJA
      D22(5)=D22(5)+(2*RD1*RD2**2-AL*RD1)*F3/(DE*RA)*OME(I)*XJA
      D22(6)=D22(6)+(AL*RD2+2*RD2**3)*F3/(DE*RA)*OME(I)*XJA
      S11(1)=S11(1)+(2*RDN*(AL*RD1+XNU*2*RD1-4*RD1**3)+4*XNU*ETA1
     1*RD1**2+AL*(2*ETA1*RD1**2+2*ETA1)-FA*ETA1)*2*GE*F1/(DE*RA**2)*
     2OME(I)*XJA
```

```
      S11(2)=S11(2)+(2*RDN*(AL*RD2-4*RD1**2*RD2)+4*XNU*ETA1*RD1*RD2+
     1AL*2*ETA2*RD1**2-FA*ETA2)*2*GE*F1/(DE*RA**2)*OME(I)*XJA
      S11(3)=S11(3)+(2*RDN*(AL*RD1+XNU*2*RD1-4*RD1**3)+4*XNU*ETA1
     1*RD1**2+AL*(2*ETA1*RD1**2+2*ETA1)-FA*ETA1)*2*GE*F2/(DE*RA**2)*
     2OME(I)*XJA
      S11(4)=S11(4)+(2*RDN*(AL*RD2-4*RD1**2*RD2)+4*XNU*ETA1*RD1*RD2+
     1AL*2*ETA2*RD1**2-FA*ETA2)*2*GE*F2/(DE*RA**2)*OME(I)*XJA
      S11(5)=S11(5)+(2*RDN*(AL*RD1+XNU*2*RD1-4*RD1**3)+4*XNU*ETA1
     1*RD1**2+AL*(2*ETA1*RD1**2+2*ETA1)-FA*ETA1)*2*GE*F3/(DE*RA**2)*
     2OME(I)*XJA
      S11(6)=S11(6)+(2*RDN*(AL*RD2-4*RD1**2*RD2)+4*XNU*ETA1*RD1*RD2+
     1AL*2*ETA2*RD1**2-FA*ETA2)*2*GE*F3/(DE*RA**2)*OME(I)*XJA
      S12(1)=S12(1)+(2*RDN*(XNU*RD2-4*RD1**2*RD2)+2*XNU*(ETA1*RD2*
     1RD1+ETA2*RD1**2)+AL*(2*ETA1*RD1*RD2+ETA2))*2*GE*F1/
     2(DE*RA**2)*OME(I)*XJA
      S12(2)=S12(2)+(2*RDN*(XNU*RD1-4*RD1*RD2**2)+2*XNU*(ETA1*RD2**2
     1+ETA2*RD1*RD2)+AL*(2*ETA2*RD1*RD2+ETA1))*2*GE*F1/
     2(DE*RA**2)*OME(I)*XJA
      S12(3)=S12(3)+(2*RDN*(XNU*RD2-4*RD1**2*RD2)+2*XNU*(ETA1*RD2*
     1RD1+ETA2*RD1**2)+AL*(2*ETA1*RD1*RD2+ETA2))*2*GE*F2/
     2(DE*RA**2)*OME(I)*XJA
      S12(4)=S12(4)+(2*RDN*(XNU*RD1-4*RD1*RD2**2)+2*XNU*(ETA1*RD2**2
     1+ETA2*RD1*RD2)+AL*(2*ETA2*RD1*RD2+ETA1))*2*GE*F2/
     2(DE*RA**2)*OME(I)*XJA
      S12(5)=S12(5)+(2*RDN*(XNU*RD2-4*RD1**2*RD2)+2*XNU*(ETA1*RD2*
     1RD1+ETA2*RD1**2)+AL*(2*ETA1*RD1*RD2+ETA2))*2*GE*F3/
     2(DE*RA**2)*OME(I)*XJA
      S12(6)=S12(6)+(2*RDN*(XNU*RD1-4*RD1*RD2**2)+2*XNU*(ETA1*RD2**2
     1+ETA2*RD1*RD2)+AL*(2*ETA2*RD1*RD2+ETA1))*2*GE*F3/
     2(DE*RA**2)*OME(I)*XJA
      S22(1)=S22(1)+(2*RDN*(AL*RD1-4*RD1*RD2**2)+4*XNU*ETA2*RD1*RD2+
     1AL*2*ETA1*RD2**2-FA*ETA1)*2*GE*F1/(DE*RA**2)*OME(I)*XJA
      S22(2)=S22(2)+(2*RDN*(AL*RD2+2*XNU*RD2-4*RD2**3)+4*XNU*ETA2
     1*RD2**2+AL*(2*ETA2*RD2**2+2*ETA2)-FA*ETA2)*2*GE*F1/
     2(DE*RA**2)*OME(I)*XJA
      S22(3)=S22(3)+(2*RDN*(AL*RD1-4*RD1*RD2**2)+4*XNU*ETA2*RD1*RD2+
     1AL*2*ETA1*RD2**2-FA*ETA1)*2*GE*F2/(DE*RA**2)*OME(I)*XJA
      S22(4)=S22(4)+(2*RDN*(AL*RD2+2*XNU*RD2-4*RD2**3)+4*XNU*ETA2
     1*RD2**2+AL*(2*ETA2*RD2**2+2*ETA2)-FA*ETA2)*2*GE*F2/
     2(DE*RA**2)*OME(I)*XJA
      S22(5)=S22(5)+(2*RDN*(AL*RD1-4*RD1*RD2**2)+4*XNU*ETA2*RD1*RD2+
     1AL*2*ETA1*RD2**2-FA*ETA1)*2*GE*F3/(DE*RA**2)*OME(I)*XJA
   40 S22(6)=S22(6)+(2*RDN*(AL*RD2+2*XNU*RD2-4*RD2**3)+4*XNU*ETA2
     1*RD2**2+AL*(2*ETA2*RD2**2+2*ETA2)-FA*ETA2)*2*GE*F3/
     2(DE*RA**2)*OME(I)*XJA
      RETURN
      END
```

8. Routine OUTPTEQ

This subroutine prints the results in the following order.

(i) Displacements at boundary nodes.
(ii) Tractions at boundary nodes (tractions 'before' and 'after' each node are printed).
(iii) Internal point displacements.
(iv) Internal point stresses.

The listing is as follows:

```
C-----------------------------------------------------------------------
      SUBROUTINE OUTPTEQ(X,Y,FI,DFI,CX,CY,SSOL,DSOL)
C
C  PROGRAM 43
C
C  THIS SUBROUTINE PRINTS THE VALUES OF THE DISPLACEMENTS
```

```
C  AND TRACTIONS AT BOUNDARY NODES. IT ALSO PRINTS THE VALUES
C  OF DISPLACEMENTS AND STRESSES AT INTERNAL POINTS
C
      DIMENSION X(1),Y(1),FI(1),DFI(1)
      DIMENSION CX(1),CY(1),SSOL(1),DSOL(1)
      COMMON N,L,GE,XNU,INP,IPR
      NE=N/2
C
      WRITE(IPR,100)
  100 FORMAT(' ',79('*')//1X,'RESULTS'//2X,'BOUNDARY NODES'///10X
     1,'X',17X,'Y',11X,'DISPLACEMENT X',4X,'DISPLACEMENT Y'/)
      DO 10 I=1,N
   10 WRITE(IPR,200) X(I),Y(I),FI(2*I-1),FI(2*I)
  200 FORMAT(4(4X,E14.7))
      WRITE(IPR,150)
  150 FORMAT(///28X,'TRACTION X',3X,'TRACTION Y',3X,'TRACTION X'
     1,3X,'TRACTION Y'/7X,'X',12X,'Y',7X,'BEFORE NODE',2X,'BEFORE NODE'
     2,2X,'AFTER NODE',3X,'AFTER NODE'/)
      WRITE(IPR,250) X(1),Y(1),DFI(6*NE-1),DFI(6*NE),DFI(1),DFI(2)
      WRITE(IPR,250) X(2),Y(2),DFI(3),DFI(4),DFI(3),DFI(4)
      DO 15 I=2,NE
      WRITE(IPR,250) X(2*I-1),Y(2*I-1),DFI(6*I-7),DFI(6*I-6),
     1DFI(6*I-5),DFI(6*I-4)
   15 WRITE(IPR,250) X(2*I),Y(2*I),DFI(6*I-3),DFI(6*I-2),
     1DFI(6*I-3),DFI(6*I-2)
  250 FORMAT(6(1X,E12.5))
C
      IF(L.EQ.0) GO TO 30
      WRITE(IPR,300)
  300 FORMAT(///2X,'INTERNAL POINTS DISPLACEMENTS'//8X,'X',15X,'Y',9X
     1,'DISPLACEMENT X',2X,'DISPLACEMENT Y')
      DO 20 K=1,L
   20 WRITE(IPR,400)CX(K),CY(K),DSOL(2*K-1),DSOL(2*K)
      WRITE(IPR,350)
  350 FORMAT(//2X,'INTERNAL POINTS STRESSES'//8X,'X',15X,'Y',12X,
     1'SIGMA X',10X,'TAU XY',9X,'SIGMA Y')
      DO 25 K=1,L
   25 WRITE(IPR,450) CX(K),CY(K),SSOL(3*K-2),SSOL(3*K-1),SSOL(3*K)
  400 FORMAT(4(2X,E14.7))
  450 FORMAT(5(2X,E14.7))
   30 WRITE(IPR,500)
  500 FORMAT(' ',79('*'))
      RETURN
      END
```

Example 4.2

The circular cavity under internal pressure described in Example 4.1 is now studied using 12 quadratic elements. This discretization gives the same number of nodes as the one which used 24 constant elements and results can then be easily compared.

The input for the ELQUABE code is as follows

CIRCULAR CAVITY (DATA)

```
  CIRCULAR CAVITY UNDER INTERNAL PRESSURE  (12 QUADRATIC ELEMENTS)
12 5 94500. .1
-0.76985 -2.87295
-1.4872  -2.57575
-2.1032 -2.1032
-2.57585 -1.4872
-2.87295 -0.76985
-2.9745 0.
-2.87295 0.76985
-2.57585  1.4872
-2.1032  2.1032
-1.4872  2.57585
-0.76985  2.87295
0.       2.9743
```

```
0.76985  2.87295
1.4872   2.57575
2.1032   2.1032
2.57585  1.4872
2.87295  0.76985
2.9745 0.
2.87295 -0.76985
2.57585  -1.4872
2.1032   -2.1032
1.4872   -2.57585
0.76985  -2.87295
0.       -2.9743
1  -25.88 1 -96.59  1  -50.  1 -86.6 1  -70.71 1 -70.71
1  -70.71 1 -70.71  1  -86.6 1 -50.  1  -96.59 1 -25.88
1  -96.59 1 -25.88  1  -100. 1  0.   1  -96.59 1  25.88
1  -96.59 1  25.88  1  -86.6 1  50.  1  -70.71 1  70.71
1  -70.71 1  70.71  1  -50.  1  86.6 1  -25.88 1  96.59
1  -25.88 1  96.59  0   0.  1  100. 1 25.88 1  96.59
1 25.88 1  96.59  1 50.  1  86.6 1 70.71 1  70.71
1 70.71 1  70.71  1 86.6 1  50.  1 96.59 1  25.88
1 96.59 1  25.88  1 100. 0  0.   1 96.59 1 -25.88
1 96.59 1 -25.88  1 86.6 1 -50.  1 70.71 1 -70.71
1 70.71 1 -70.71  1 50.  1 -86.6 1 25.88 1 -96.59
1 25.88 1 -96.59  0   0.  1 -100. 1  -25.88 1 -96.59
4. 0. 2.82843 2.82843 -4. 0. 6. 0. 10. 0.
```

The output given below also presents symmetry of results, same absolute values (with different sign) for hoop and radial stresses at internal points and exponential decay of both displacements and stresses with distance from the cavity.

The comparison is as follows.

Table 4.2 12 Quadratic Elements Discretization

Distance from centre of cavity	Elasticity theory	Constant elements		Quadratic elements	
		Stress	% Error	Stress	% Error
4	−56.25	−57.234	1.75%	−55.275	1.73%
6	−25.00	−25.295	1.18%	−24.565	1.74%
10	−9.00	−9.106	1.18%	−8.844	1.73%

(Notice that only internal points with comparatively large stresses are taken into consideration as the stresses are too low for points further away from the cavity to give a meaningful error estimate.)

It is interesting to see that in this case the use of quadratic elements does not appreciably alter the error in the results on the contrary, they are overall less accurate for the same number of degrees of freedom. This is due to the problem being rather of a special character. As we will see in what follows, although constant elements behave well in problems such as this they give poor results when used to model problems with bending.

CIRCULAR CAVITY (OUTPUT)

```
*******************************************************************************
 CIRCULAR CAVITY UNDER INTERNAL PRESSURE  (12 QUADRATIC ELEMENTS)

 DATA
 NUMBER OF BOUNDARY ELEMENTS= 12
 NUMBER OF INTERNAL POINTS=  5
 SHEAR MODULUS= 0.9450000E+05
 POISSON RATIO= 0.1000000E+00
```

BOUNDARY NODES COORDINATES

NODE	X	Y
1	-0.7698500E+00	-0.2872950E+01
2	-0.1487200E+01	-0.2575750E+01
3	-0.2103200E+01	-0.2103200E+01
4	-0.2575850E+01	-0.1487200E+01
5	-0.2872950E+01	-0.7698500E+00
6	-0.2974500E+01	0.0000000E+00
7	-0.2872950E+01	0.7698500E+00
8	-0.2575850E+01	0.1487200E+01
9	-0.2103200E+01	0.2103200E+01
10	-0.1487200E+01	0.2575850E+01
11	-0.7698500E+00	0.2872950E+01
12	0.0000000E+00	0.2974300E+01
13	0.7698500E+00	0.2872950E+01
14	0.1487200E+01	0.2575750E+01
15	0.2103200E+01	0.2103200E+01
16	0.2575850E+01	0.1487200E+01
17	0.2872950E+01	0.7698500E+00
18	0.2974500E+01	0.0000000E+00
19	0.2872950E+01	-0.7698500E+00
20	0.2575850E+01	-0.1487200E+01
21	0.2103200E+01	-0.2103200E+01
22	0.1487200E+01	-0.2575850E+01
23	0.7698500E+00	-0.2872950E+01
24	0.0000000E+00	-0.2974300E+01

BOUNDARY CONDITIONS

	PRESCRIBED VALUES											
	-----FIRST NODE------				-----SECOND NODE-----				-----THIRD NODE------			
ELE	X DIR.	C	Y DIR.	C	X DIR.	C	Y DIR.	C	X DIR.	C	Y DIR.	C
1	-25.880	1	-96.590	1	-50.000	1	-86.600	1	-70.710	1	-70.710	1
2	-70.710	1	-70.710	1	-86.600	1	-50.000	1	-96.590	1	-25.880	1
3	-96.590	1	-25.880	1	-100.000	1	0.000	1	-96.590	1	25.880	1
4	-96.590	1	25.880	1	-86.600	1	50.000	1	-70.710	1	70.710	1
5	-70.710	1	70.710	1	-50.000	1	86.600	1	-25.880	1	96.590	1
6	-25.880	1	96.590	1	0.000	0	100.000	1	25.880	1	96.590	1
7	25.880	1	96.590	1	50.000	1	86.600	1	70.710	1	70.710	1
8	70.710	1	70.710	1	86.600	1	50.000	1	96.590	1	25.880	1
9	96.590	1	25.880	1	100.000	1	0.000	0	96.590	1	-25.880	1
10	96.590	1	-25.880	1	86.600	1	-50.000	1	70.710	1	-70.710	1
11	70.710	1	-70.710	1	50.000	1	-86.600	1	25.880	1	-96.590	1
12	25.880	1	-96.590	1	0.000	0	-100.000	1	-25.880	1	-96.590	1

****

RESULTS

BOUNDARY NODES

X	Y	DISPLACEMENT X	DISPLACEMENT Y
-0.7698500E+00	-0.2872950E+01	-0.4072179E-03	-0.1519695E-02
-0.1487200E+01	-0.2575750E+01	-0.7866786E-03	-0.1362574E-02
-0.2103200E+01	-0.2103200E+01	-0.1112431E-02	-0.1112502E-02
-0.2575850E+01	-0.1487200E+01	-0.1362535E-02	-0.7866730E-03
-0.2872950E+01	-0.7698500E+00	-0.1519704E-02	-0.4072844E-03
-0.2974500E+01	0.0000000E+00	-0.1573262E-02	0.1928129E-08
-0.2872950E+01	0.7698500E+00	-0.1519712E-02	0.4072882E-03
-0.2575850E+01	0.1487200E+01	-0.1362557E-02	0.7866755E-03
-0.2103200E+01	0.2103200E+01	-0.1112482E-02	0.1112481E-02
-0.1487200E+01	0.2575850E+01	-0.7866530E-03	0.1362540E-02
-0.7698500E+00	0.2872950E+01	-0.4071935E-03	0.1519707E-02
0.0000000E+00	0.2974300E+01	0.0000000E+00	0.1573358E-02
0.7698500E+00	0.2872950E+01	0.4072177E-03	0.1519692E-02
0.1487200E+01	0.2575750E+01	0.7866795E-03	0.1362572E-02
0.2103200E+01	0.2103200E+01	0.1112433E-02	0.1112499E-02
0.2575850E+01	0.1487200E+01	0.1362536E-02	0.7866717E-03
0.2872950E+01	0.7698500E+00	0.1519705E-02	0.4072846E-03

```
 0.2974500E+01    0.0000000E+00    0.1573262E-02    0.0000000E+00
 0.2872950E+01   -0.7698500E+00    0.1519712E-02   -0.4072880E-03
 0.2575850E+01   -0.1487200E+01    0.1362557E-02   -0.7866766E-03
 0.2103200E+01   -0.2103200E+01    0.1112481E-02   -0.1112482E-02
 0.1487200E+01   -0.2575850E+01    0.7866526E-03   -0.1362543E-02
 0.7698500E+00   -0.2872950E+01    0.4071933E-03   -0.1519709E-02
 0.0000000E+00   -0.2974300E+01    0.0000000E+00   -0.1573361E-02
```

X	Y	TRACTION X BEFORE NODE	TRACTION Y BEFORE NODE	TRACTION X AFTER NODE	TRACTION Y AFTER NODE
-0.76985E+00	-0.28730E+01	-0.25880E+02	-0.96590E+02	-0.25880E+02	-0.96590E+02
-0.14872E+01	-0.25758E+01	-0.50000E+02	-0.86600E+02	-0.50000E+02	-0.86600E+02
-0.21032E+01	-0.21032E+01	-0.70710E+02	-0.70710E+02	-0.70710E+02	-0.70710E+02
-0.25759E+01	-0.14872E+01	-0.86600E+02	-0.50000E+02	-0.86600E+02	-0.50000E+02
-0.28730E+01	-0.76985E+00	-0.96590E+02	-0.25880E+02	-0.96590E+02	-0.25880E+02
-0.29745E+01	0.00000E+00	-0.10000E+03	0.00000E+00	-0.10000E+03	0.00000E+00
-0.28730E+01	0.76985E+00	-0.96590E+02	0.25880E+02	-0.96590E+02	0.25880E+02
-0.25759E+01	0.14872E+01	-0.86600E+02	0.50000E+02	-0.86600E+02	0.50000E+02
-0.21032E+01	0.21032E+01	-0.70710E+02	0.70710E+02	-0.70710E+02	0.70710E+02
-0.14872E+01	0.25759E+01	-0.50000E+02	0.86600E+02	-0.50000E+02	0.86600E+02
-0.76985E+00	0.28730E+01	-0.25880E+02	0.96590E+02	-0.25880E+02	0.96590E+02
0.00000E+00	0.29743E+01	-0.32454E-03	0.10000E+03	-0.32454E-03	0.10000E+03
0.76985E+00	0.28730E+01	0.25880E+02	0.96590E+02	0.25880E+02	0.96590E+02
0.14872E+01	0.25758E+01	0.50000E+02	0.86600E+02	0.50000E+02	0.86600E+02
0.21032E+01	0.21032E+01	0.70710E+02	0.70710E+02	0.70710E+02	0.70710E+02
0.25759E+01	0.14872E+01	0.86600E+02	0.50000E+02	0.86600E+02	0.50000E+02
0.28730E+01	0.76985E+00	0.96590E+02	0.25880E+02	0.96590E+02	0.25880E+02
0.29745E+01	0.00000E+00	0.10000E+03	0.61401E-03	0.10000E+03	0.61401E-03
0.28730E+01	-0.76985E+00	0.96590E+02	-0.25880E+02	0.96590E+02	-0.25880E+02
0.25759E+01	-0.14872E+01	0.86600E+02	-0.50000E+02	0.86600E+02	-0.50000E+02
0.21032E+01	-0.21032E+01	0.70710E+02	-0.70710E+02	0.70710E+02	-0.70710E+02
0.14872E+01	-0.25759E+01	0.50000E+02	-0.86600E+02	0.50000E+02	-0.86600E+02
0.76985E+00	-0.28730E+01	0.25880E+02	-0.96590E+02	0.25880E+02	-0.96590E+02
0.00000E+00	-0.29743E+01	0.24495E-03	-0.10000E+03	0.24495E-03	-0.10000E+03

INTERNAL POINTS DISPLACEMENTS

X	Y	DISPLACEMENT X	DISPLACEMENT Y
0.4000000E+01	0.0000000E+00	0.1169808E-02	-0.4138201E-10
0.2828430E+01	0.2828430E+01	0.8271520E-03	0.8271745E-03
-0.4000000E+01	0.0000000E+00	-0.1169808E-02	-0.1176431E-08
0.6000000E+01	0.0000000E+00	0.7798681E-03	-0.2637535E-10
0.1000000E+02	0.0000000E+00	0.4679200E-03	-0.9115411E-09

INTERNAL POINTS STRESSES

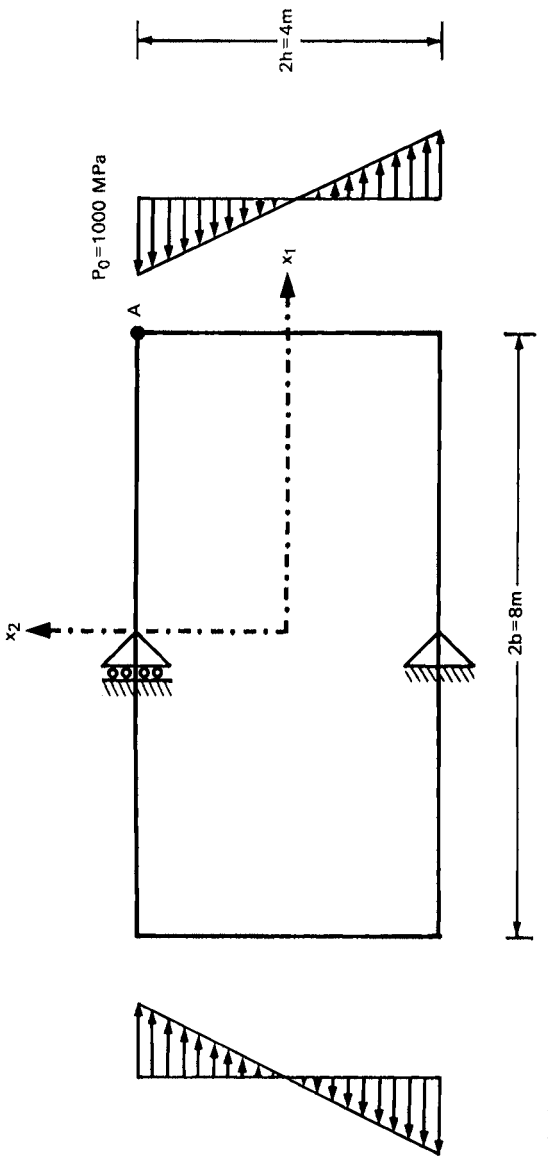
X	Y	SIGMA X	TAU XY	SIGMA Y
0.4000000E+01	0.0000000E+00	-0.5527559E+02	-0.2949312E-03	0.5527643E+
0.2828430E+01	0.2828430E+01	0.2465338E-02	-0.5527016E+02	0.3696308E-
-0.4000000E+01	0.0000000E+00	-0.5527559E+02	-0.2649724E-03	0.5527644E+
0.6000000E+01	0.0000000E+00	-0.2456595E+02	-0.1923647E-03	0.2456613E+
0.1000000E+02	0.0000000E+00	-0.8843707E+01	-0.5847029E-04	0.8843761E+

****

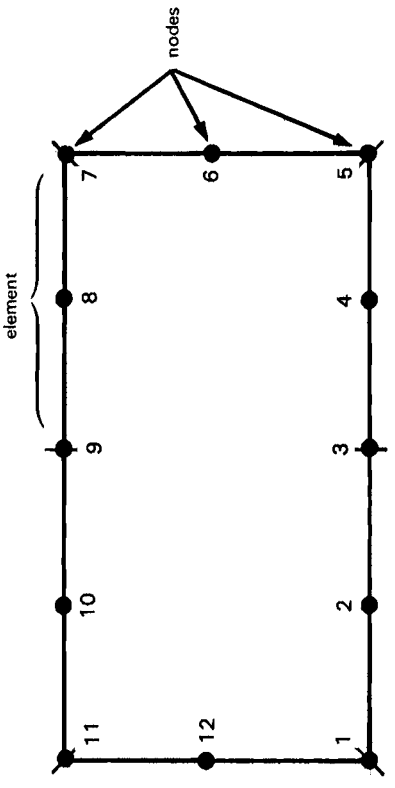
Example 4.3

The following example demonstrates the use of quadratic elements for a simple problem which because of flexure, can not be solved accurately using constant elements.

Figure 4.9 shows a rectangular plate under a linear distribution of tractions in the horizontal direction, which represents two applied moments. The plate is considered to be in plane stress with shear modulus $\mu = G = 80{,}000$ MPa and Poisson's ratio $\nu = 0.25$. The boundary has been discretized into only 6 quadratic



(a) Plate under linearly varying end tractions



(b) Discretization of the plate into six quadratic elements

Figure 4.9 Plate stretching elements under flexion

elements with tractions prescribed along all elements (note tractions in horizontal sides are prescribed as zero in the two directions), with the exception of the nodes located on the vertical axes where the displacements can be prescribed as indicated in the figure. These displacements are the minimum needed to avoid rigid body motions and make the problem well posed for solution.

The input required by the program ELQUABE is as follows:

RECTANGULAR PLATE (DATA)

```
  RECTANGULAR PLATE UNDER FLEXURAL MOMENT (6 QUADRATIC ELEMENTS)
6 1 80000. 0.2
-4. -2. -2. -2. 0. -2. 2. -2. 4. -2. 4. 0. 4. 2. 2. 2. 0. 2.
-2. 2. -4. 2. -4. 0.
1 0. 1 0. 1 0. 1 0. 0 0. 1 0.
0 0. 1 0. 1 0. 1 0. 1 0. 1 0.
1 1000. 1 0. 1  0. 1 0.   1  -1000.    1 0.
1 0. 1 0. 1 0. 1 0. 0 0. 0 0.
0 0. 0 0. 1 0. 1 0. 1 0. 1 0.
1 1000.    1 0. 1 0. 1 0.   1 -1000. 1 0.
0. 0.
```

This produces the following results:

RECTANGULAR PLATE (OUTPUT)

```
**************************************************************************
 RECTANGULAR PLATE UNDER FLEXURAL MOMENT (6 QUADRATIC ELEMENTS)


 DATA
 NUMBER OF BOUNDARY ELEMENTS=  6
 NUMBER OF INTERNAL POINTS=  1
 SHEAR MODULUS= 0.8000000E+05
 POISSON RATIO= 0.2000000E+00


 BOUNDARY NODES COORDINATES


   NODE          X                Y

      1     -0.4000000E+01    -0.2000000E+01
      2     -0.2000000E+01    -0.2000000E+01
      3      0.0000000E+00    -0.2000000E+01
      4      0.2000000E+01    -0.2000000E+01
      5      0.4000000E+01    -0.2000000E+01
      6      0.4000000E+01     0.0000000E+00
      7      0.4000000E+01     0.2000000E+01
      8      0.2000000E+01     0.2000000E+01
      9      0.0000000E+00     0.2000000E+01
     10     -0.2000000E+01     0.2000000E+01
     11     -0.4000000E+01     0.2000000E+01
     12     -0.4000000E+01     0.0000000E+00


 BOUNDARY CONDITIONS

                                    PRESCRIBED VALUES
      -----FIRST NODE------   -----SECOND NODE-----   -----THIRD NODE------
ELE   X DIR.  C   Y DIR.  C   X DIR.  C   Y DIR.  C   X DIR.  C   Y DIR.  C

  1     0.000 1     0.000 1     0.000 1     0.000 1     0.000 0     0.000 1
  2     0.000 0     0.000 1     0.000 1     0.000 1     0.000 1     0.000 1
  3 1000.000 1     0.000 1     0.000 1     0.000 1-1000.000 1     0.000 1
  4     0.000 1     0.000 1     0.000 1     0.000 1     0.000 0     0.000 0
  5     0.000 0     0.000 0     0.000 1     0.000 1     0.000 1     0.000 1
  6 1000.000 1     0.000 1     0.000 1     0.000 1-1000.000 1     0.000 1
**************************************************************************
```

```
RESULTS
 BOUNDARY NODES
        X                Y            DISPLACEMENT X    DISPLACEMENT Y
   -0.4000000E+01   -0.2000000E+01    -0.1999997E-01     0.1999998E-01
   -0.2000000E+01   -0.2000000E+01    -0.9999990E-02     0.5000000E-02
    0.0000000E+00   -0.2000000E+01     0.0000000E+00    -0.2648449E-08
    0.2000000E+01   -0.2000000E+01     0.1000000E-01     0.4999989E-02
    0.4000000E+01   -0.2000000E+01     0.2000001E-01     0.1999998E-01
    0.4000000E+01    0.0000000E+00     0.1396984E-07     0.1874997E-01
    0.4000000E+01    0.2000000E+01    -0.1999997E-01     0.1999997E-01
    0.2000000E+01    0.2000000E+01    -0.9999976E-02     0.4999987E-02
    0.0000000E+00    0.2000000E+01     0.0000000E+00     0.0000000E+00
   -0.2000000E+01    0.2000000E+01     0.9999998E-02     0.4999997E-02
   -0.4000000E+01    0.2000000E+01     0.2000002E-01     0.2000000E-01
   -0.4000000E+01    0.0000000E+00    -0.2517481E-08     0.1875000E-01

      X              Y          TRACTION X   TRACTION Y   TRACTION X   TRACTION Y
                                BEFORE NODE  BEFORE NODE  AFTER NODE   AFTER NODE
-0.40000E+01 -0.20000E+01 -0.10000E+04  0.00000E+00  0.00000E+00  0.00000E+00
-0.20000E+01 -0.20000E+01  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
 0.00000E+00 -0.20000E+01 -0.10431E-02  0.00000E+00 -0.10431E-02  0.00000E+00
 0.20000E+01 -0.20000E+01  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
 0.40000E+01 -0.20000E+01  0.00000E+00  0.00000E+00  0.10000E+04  0.00000E+00
 0.40000E+01  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
 0.40000E+01  0.20000E+01 -0.10000E+04  0.00000E+00  0.00000E+00  0.00000E+00
 0.20000E+01  0.20000E+01  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
 0.00000E+00  0.20000E+01 -0.18626E-03 -0.18626E-03 -0.18626E-03 -0.18626E-03
-0.20000E+01  0.20000E+01  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
-0.40000E+01  0.20000E+01  0.00000E+00  0.00000E+00  0.10000E+04  0.00000E+00
-0.40000E+01  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00

 INTERNAL POINTS DISPLACEMENTS

      X              Y          DISPLACEMENT X  DISPLACEMENT Y
 0.0000000E+00  0.0000000E+00   0.1111766E-07  -0.1250002E-02

 INTERNAL POINTS STRESSES

      X              Y            SIGMA X          TAU XY          SIGMA Y
 0.0000000E+00  0.0000000E+00  0.1640320E-03  0.6055832E-04  0.6484985E-
**********************************************************************
```

Results for u_1 and u_2 displacements at corner A (figure 4.9(a)) are summarized in Table 4.3, where they are compared against the analytical solution and the values obtained using the program ELCONBE with 50 constant elements. It can be seen that while the accuracy of the quadratic elements is excellent, poor results are obtained using constant elements. This demonstrates that constant elements are not to be recommended for problems with flexural strains, although they give good results in other applications.

Table 4.3 Displacements at Corner A (figure 4.9)

Displacement	Analytical solution	6 Quadratic element solution		50 Constant elements	
		Values	Error (%)	Values	Error (%)
u_1	−0.02	−0.02	0%	−0.0157	21.5%
u_2	0.02	0.02	0%	+0.0165	17.5%

(All results in m.)

Example 4.4

Another flexural problem which can be solved accurately using quadratic elements is the cantilever beam shown in figure 4.10. The beam has been discretized using 12 quadratic elements and is under a transverse parabolic load distribution at its two ends. On the end on the left hand side one also considers horizontal displacement constraints as shown in the figure, with the lower corner node displacements

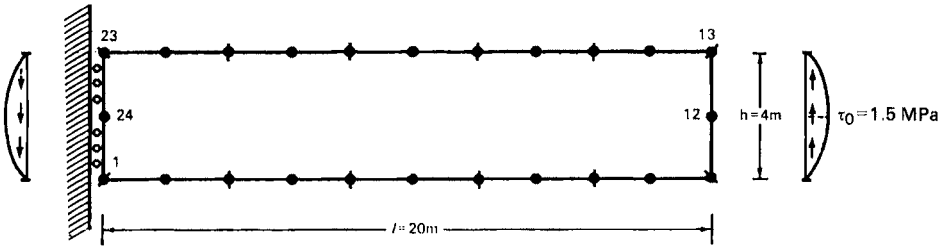


Figure 4.10 Cantilever beam under end loading

set both equal to zero. The problem is in plane stress and the material constants are assumed to be the same as those in Example 4.3.

The beam theory gives for this problem a free end deflection of,

$$u_2 = \frac{Pl^3}{3EI} = 10 \text{ mm} \tag{a}$$

where P is the total load applied at the end, l is the length of the beam, E its modulus of elasticity and I the moments of inertia of the beam.

The stress distributions along the fixed part of the boundary is linear in accordance with the beam theory and equal to

$$\sigma = \frac{Mh}{2I} = 30 \text{ MPa} \tag{b}$$

Boundary Element Values of vertical displacements at nodes 11–12–13 at one end and normal tractions at nodes 23–24–1 at the other end are given in Table 4.4. The results agree well with those of formulae (a) and (b).

Table 4.4 Vertical Displacements and Tractions for the Cantilever Beam of figure 4.10

Node	Vertical displacement u_2	Normal tractions σ_n
11	10.166	—
12	10.169	—
13	10.166	—
23	—	29.85
24	—	−0.00
1	—	−29.85

The input for this problem to run program ELQUABE is as follows:

CANTILEVER BEAM (DATA)

```
  CANTILEVER BEAM UNDER TRANSVERSAL END LOAD (12 QUADRATIC ELEMENTS)
 12 1  80000. 0.2
 0. 0. 2. 0. 4. 0. 6. 0. 8. 0. 10. 0. 12. 0. 14. 0. 16. 0. 18. 0. 20. 0.
 20. 2. 20. 4. 18. 4. 16. 4. 14. 4. 12. 4. 10. 4. 8. 4. 6. 4. 4. 4.
 2. 4. 0. 4. 0. 2.
 1 0. 0 0. 1 0. 1 0. 1 0. 1 0. 1 0.
 1 0. 1 0. 1 0. 1 0. 1 0. 1 0. 1 0.
 1 0. 1 0. 1 0. 1 0. 1 0. 1 0. 1 0.
 1 0. 1 0. 1 0. 1 0. 1 0. 1 0. 1 0.
 1 0. 1 0. 1 0. 1 0. 1 0. 1 0. 1 0.
 1 0. 1 0.    1 0. 1 1500. 1 0. 1 0.
 1 0. 1 0. 1 0. 1 0. 1 0. 1 0. 1 0.
 1 0. 1 0. 1 0. 1 0. 1 0. 1 0. 1 0.
 1 0. 1 0. 1 0. 1 0. 1 0. 1 0. 1 0.
 1 0. 1 0. 1 0. 1 0. 1 0. 1 0. 1 0.
 0 0. 1 0.    0 0. 1 -1500. 0 0. 1 0.
10 2.
```

The corresponding output is listed below,

CANTILEVER BEAM (OUTPUT)

```
********************************************************************************
 CANTILEVER BEAM UNDER TRANSVERSAL END LOAD (12 QUADRATIC ELEMENTS)

DATA
NUMBER OF BOUNDARY ELEMENTS= 12
NUMBER OF INTERNAL POINTS=  1
SHEAR MODULUS= 0.8000000E+05
POISSON RATIO= 0.2000000E+00


BOUNDARY NODES COORDINATES

  NODE          X                   Y

     1      0.0000000E+00       0.0000000E+00
     2      0.2000000E+01       0.0000000E+00
     3      0.4000000E+01       0.0000000E+00
     4      0.6000000E+01       0.0000000E+00
     5      0.8000000E+01       0.0000000E+00
     6      0.1000000E+02       0.0000000E+00
     7      0.1200000E+02       0.0000000E+00
     8      0.1400000E+02       0.0000000E+00
     9      0.1600000E+02       0.0000000E+00
    10      0.1800000E+02       0.0000000E+00
    11      0.2000000E+02       0.0000000E+00
    12      0.2000000E+02       0.2000000E+01
    13      0.2000000E+02       0.4000000E+01
    14      0.1800000E+02       0.4000000E+01
    15      0.1600000E+02       0.4000000E+01
    16      0.1400000E+02       0.4000000E+01
    17      0.1200000E+02       0.4000000E+01
    18      0.1000000E+02       0.4000000E+01
    19      0.8000000E+01       0.4000000E+01
    20      0.6000000E+01       0.4000000E+01
    21      0.4000000E+01       0.4000000E+01
    22      0.2000000E+01       0.4000000E+01
    23      0.0000000E+00       0.4000000E+01
    24      0.0000000E+00       0.2000000E+01
```

BOUNDARY CONDITIONS

	PRESCRIBED VALUES											
	-----FIRST NODE------				-----SECOND NODE-----				-----THIRD NODE------			
ELE	X DIR.	C	Y DIR.	C	X DIR.	C	Y DIR.	C	X DIR.	C	Y DIR.	C
1	0.000	1	0.000	0	0.000	1	0.000	1	0.000	1	0.000	1
2	0.000	1	0.000	1	0.000	1	0.000	1	0.000	1	0.000	1
3	0.000	1	0.000	1	0.000	1	0.000	1	0.000	1	0.000	1
4	0.000	1	0.000	1	0.000	1	0.000	1	0.000	1	0.000	1
5	0.000	1	0.000	1	0.000	1	0.000	1	0.000	1	0.000	1
6	0.000	1	0.000	1	0.000	1	1500.000	1	0.000	1	0.000	1
7	0.000	1	0.000	1	0.000	1	0.000	1	0.000	1	0.000	1
8	0.000	1	0.000	1	0.000	1	0.000	1	0.000	1	0.000	1
9	0.000	1	0.000	1	0.000	1	0.000	1	0.000	1	0.000	1
10	0.000	1	0.000	1	0.000	1	0.000	1	0.000	1	0.000	1
11	0.000	1	0.000	1	0.000	1	0.000	1	0.000	1	0.000	1
12	0.000	0	0.000	1	0.000	0	-1500.000	1	0.000	0	0.000	1

****

RESULTS

BOUNDARY NODES

X	Y	DISPLACEMENT X	DISPLACEMENT Y
0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00
0.2000000E+01	0.0000000E+00	0.2831624E+00	0.1641325E+00
0.4000000E+01	0.0000000E+00	0.5382484E+00	0.5995840E+00
0.6000000E+01	0.0000000E+00	0.7601910E+00	0.1273359E+01
0.8000000E+01	0.0000000E+00	0.9555768E+00	0.2155845E+01
0.1000000E+02	0.0000000E+00	0.1118159E+01	0.3217330E+01
0.1200000E+02	0.0000000E+00	0.1254107E+01	0.4428082E+01
0.1400000E+02	0.0000000E+00	0.1357250E+01	0.5758359E+01
0.1600000E+02	0.0000000E+00	0.1433837E+01	0.7178458E+01
0.1800000E+02	0.0000000E+00	0.1477282E+01	0.8658765E+01
0.2000000E+02	0.0000000E+00	0.1492648E+01	0.1016628E+02
0.2000000E+02	0.2000000E+01	0.1201406E-05	0.1016907E+02
0.2000000E+02	0.4000000E+01	-0.1492655E+01	0.1016628E+02
0.1800000E+02	0.4000000E+01	-0.1477283E+01	0.8658760E+01
0.1600000E+02	0.4000000E+01	-0.1433836E+01	0.7178457E+01
0.1400000E+02	0.4000000E+01	-0.1357249E+01	0.5758362E+01
0.1200000E+02	0.4000000E+01	-0.1254107E+01	0.4428082E+01
0.1000000E+02	0.4000000E+01	-0.1118160E+01	0.3217330E+01
0.8000000E+01	0.4000000E+01	-0.9555784E+00	0.2155848E+01
0.6000000E+01	0.4000000E+01	-0.7601921E+00	0.1273359E+01
0.4000000E+01	0.4000000E+01	-0.5382496E+00	0.5995839E+00
0.2000000E+01	0.4000000E+01	-0.2831629E+00	0.1641333E+00
0.0000000E+00	0.4000000E+01	0.0000000E+00	-0.2929009E-06
0.0000000E+00	0.2000000E+01	0.0000000E+00	-0.4010320E-01

X	Y	TRACTION X BEFORE NODE	TRACTION Y BEFORE NODE	TRACTION X AFTER NODE	TRACTION Y AFTER NODE
0.00000E+00	0.00000E+00	-0.29853E+05	0.00000E+00	0.00000E+00	0.39985E-01
0.20000E+01	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.40000E+01	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.60000E+01	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.80000E+01	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.10000E+02	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.12000E+02	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.14000E+02	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.16000E+02	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.18000E+02	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.20000E+02	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.20000E+02	0.20000E+01	0.00000E+00	0.15000E+04	0.00000E+00	0.15000E+04
0.20000E+02	0.40000E+01	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.18000E+02	0.40000E+01	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.16000E+02	0.40000E+01	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.14000E+02	0.40000E+01	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.12000E+02	0.40000E+01	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.10000E+02	0.40000E+01	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00

```
0.80000E+01  0.40000E+01  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
0.60000E+01  0.40000E+01  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
0.40000E+01  0.40000E+01  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
0.20000E+01  0.40000E+01  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
0.00000E+00  0.40000E+01  0.00000E+00  0.00000E+00  0.29853E+05  0.00000E+00
0.00000E+00  0.20000E+01  0.44191E-01 -0.15000E+04  0.44191E-01 -0.15000E+04


INTERNAL POINTS DISPLACEMENTS

      X              Y          DISPLACEMENT X  DISPLACEMENT Y
 0.1000000E+02  0.2000000E+01  -0.1154840E-06   0.3198673E+01

 INTERNAL POINTS STRESSES

      X              Y            SIGMA X          TAU XY          SIGMA Y
  0.1000000E+02  0.2000000E+01  0.1745605E-01  0.1438650E+04  -0.1143646E-
**********************************************************************************
```

Example 4.5

This problem represents the case of a plane-strain hollow cylinder under internal pressure as shown in figure 4.11. The pressure is assumed to be $p = 100$ N/mm^2, while the internal and external radii are $r_1 = 10$ mm and $r_2 = 25$ mm, respectively. The elastic constants of the material are; Elasticity Modulus $E = 200{,}000$ N/mm^2 and Poisson's ratio, $\nu = 0.25$.

Due to the symmetry of the problem only one quarter of the section needs to be discretized. The boundary conditions are shown in figure 4.10(b). Three different discretizations consisting of 4, 10 and 15 quadratic elements are used as shown in figures 4.11(c).

Table 4.5 shows the relevant displacements computed for the points A, B and C along the radius. The displacements are given in microns, i.e. 10^{-3} mm and are compared with the exact values given by the theory of elasticity. It is interesting to notice that even for the coarse discretization, results with 2% of the exact solution are obtained. When 15 elements are employed, the results are within a range of 0.1% of the exact solution.

Table 4.5 Radial Displacements for Hollow Cylinder under Internal Pressure (in 10^{-3} mm)

Node	Exact value	Discretization		
		4 elements	10 elements	15 elements
A	8.0325	7.8781	8.0246	8.0350
B	5.2912	5.1668	5.2845	5.2928
C	4.4526	4.3896	4.4520	4.4631

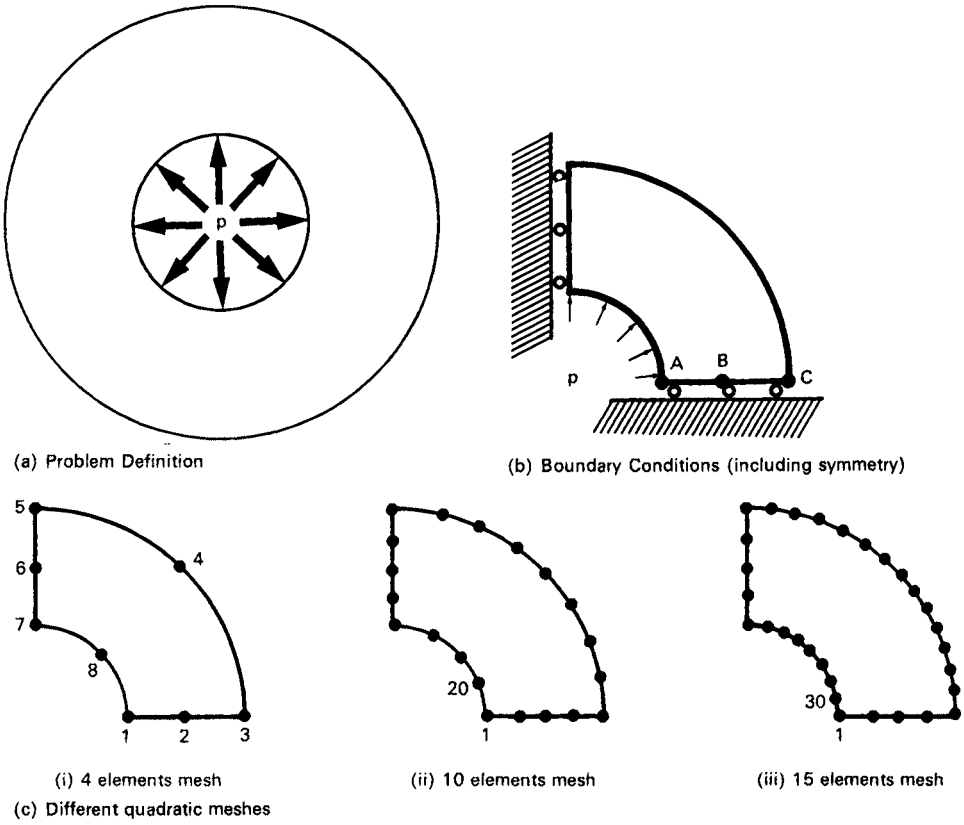


Figure 4.11 Hollow cylinder under internal pressure

Example 4.6

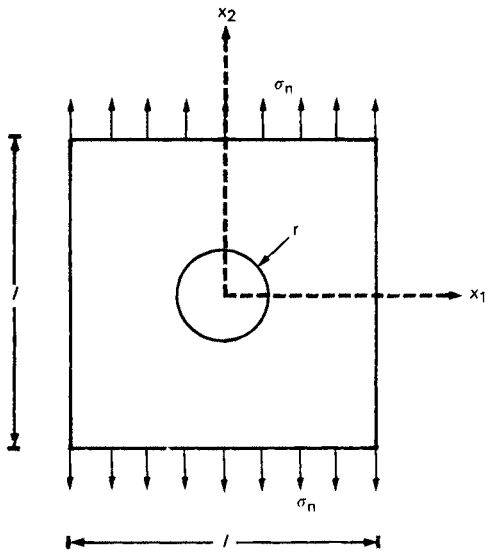
This example studies the stress concentration around a circular hole in a plate stretching specimen as shown in figure 4.12(a). The geometric and material constants are as follows:

$$\frac{l}{r} = 20$$

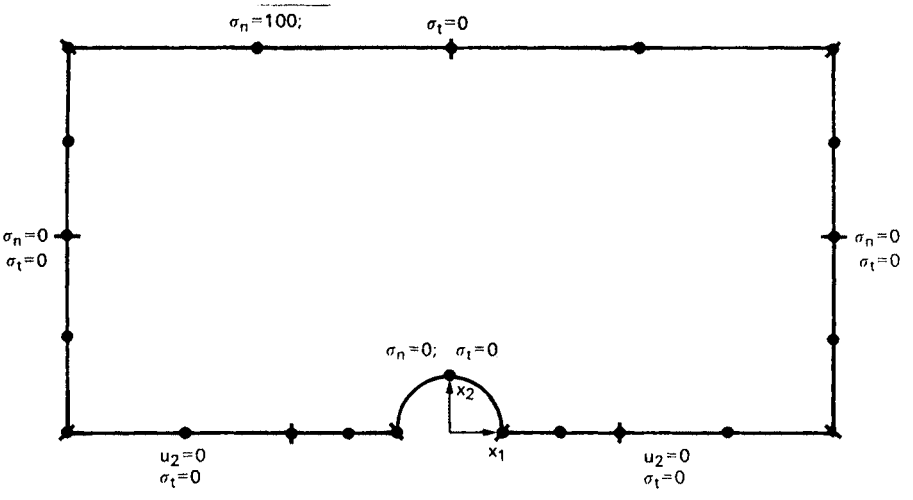
$$G = 80{,}000 \text{ MPa} \qquad \text{(a)}$$

$$\nu = 0.25$$

Figure 4.12(b) shows the boundary discretization for half the plate using only quadratic elements. Notice that only one element has been assumed over the circular section.



(a) Square plate with hole. Geometry and boundary conditions



(b) Boundary element discretization

Figure 4.12 Square plate with hole. Geometrical definition and boundary discretization

In order to check the accuracy of the Boundary Element solution the stress concentration factor around the hole is computed. This factor can be defined as,

$$K = \frac{(\sigma_{22})_{\max}}{\sigma_n} \tag{b}$$

where σ_n is the applied stress at the top and bottom end of the plate.

The boundary element solution gave a value of $\sigma_{22} = 303.66$ N/mm^2 at node A, which gives a stress concentration factor of $K = 3.0366$. An experimental solution for this problem gave $K = 3.03$, which shows a good agreement with the Boundary Element solution.

Exercises

4.1. Compute analytically the term G^{ii}_{11} that relates the first coordinate of a node with itself in a linear elements program.

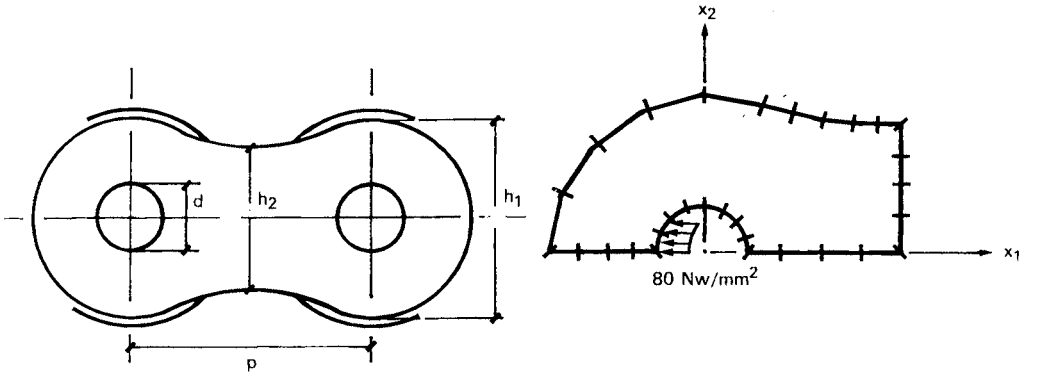
$$G^{ii}_{11} = \int_{\Gamma_{i-1}} \phi_2 u^*_{11}\, d\Gamma + \int_{\Gamma_i} \phi_1 u^*_{11}\, d\Gamma$$

4.2. Do the necessary transformations to the equation of exercise 4.1 to decompose it into integrals which may be computed either by standard Gaussian quadrature or by logarithmic Gaussian quadrature.

4.3. Discuss the effect of shifting the mid-node of a quadratic element from the centre on the numerical integrations (equation (4.56)).

4.4. Is there a limit of the distance from the mid-node to one end of a straight line quadratic element? Explain the reasons.

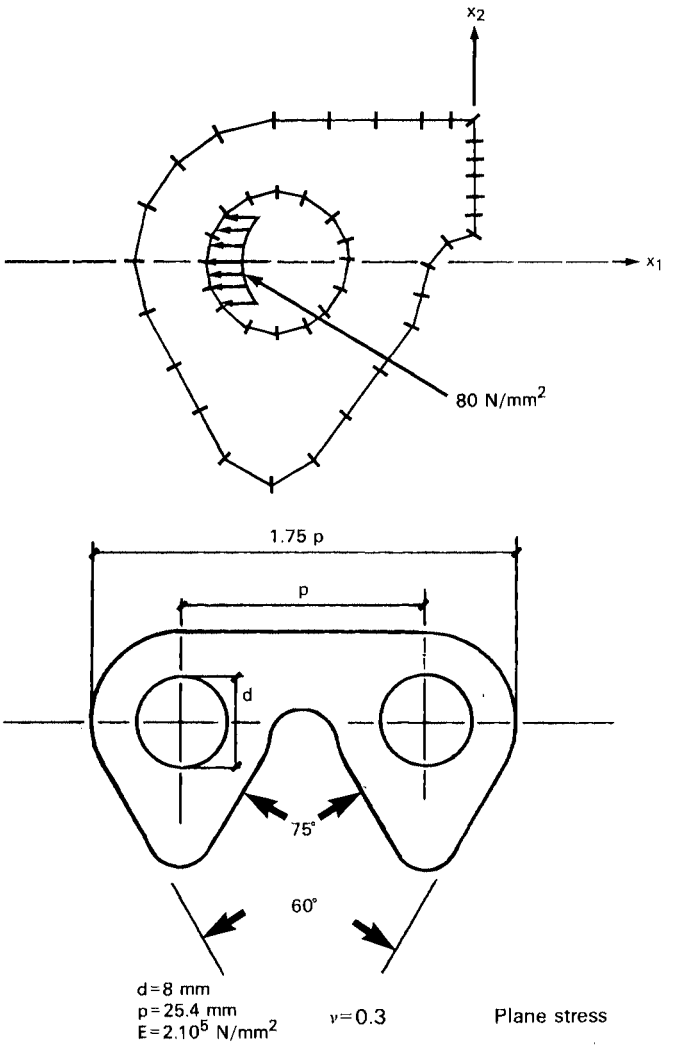
4.5. The figure shows a link plate of a roller chain. Assuming that both pins transmit their force as a uniform pressure over one quarter of the hole, determine the normal stress σ_{11} along the central section of the link using 32 constant elements to model one quarter of the problem (program ELCONBE).



p=25.4 mm
d=7.9 mm
h_1=20.8 mm
h_2=13.8 mm
$E = 2 \times 10^5$ Nw/mm^2
ν=0.3
Plane stress

4.6. Solve problem 4.5 using program ELQUABE and the same number of nodes with 16 quadratic elements. Solve the same problem using 32 quadratic elements. Compare the three solutions obtained for the problem. (The DIMENSIONS in the main program ELQUABE must be changed to run the 32 elements example.)

4.7. The link plate of a silent chain is shown in the figure. Using constant elements and the discretization of the figure compute the normal stresses along the central section.



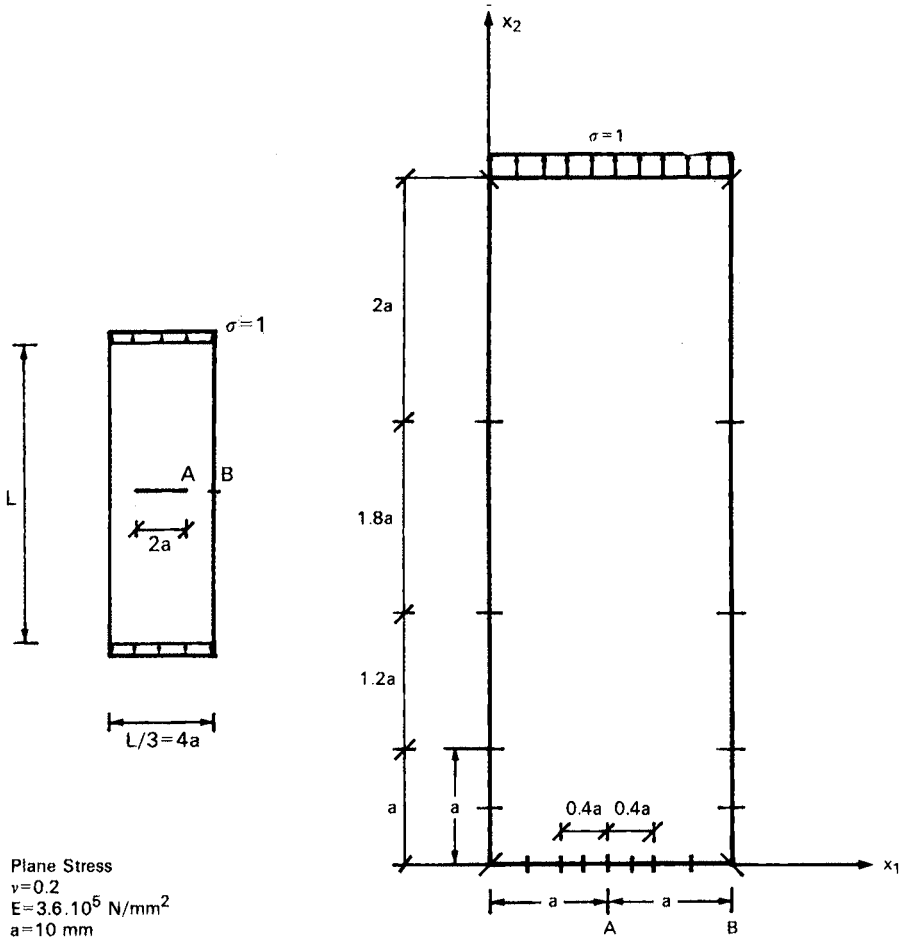
4.8. Introduce the necessary changes in program ELQUABE to solve multiboundary problems.

4.9. Solve problem 4.7 with the same number of nodes but using quadratic elements. Also solve the problem with quadratic elements and double the number of nodes. Compare the two solutions and also that of exercise 4.7.

4.10. The figure shows a centre cracked plate under traction. Because of the symmetry only one quarter of the plate is discretized. The σ_{22} stress near the crack tip for points along the line A–B is known to be of the form $\sigma_{22} = K_I/\sqrt{2\pi r}$, r being the distance to the tip and K_I the mode $-I$ stress intensity factor. Use the discretization (19

quadratic elements) shown in the figure to solve the problem and represent the value of $\sigma_{22}\sqrt{2\pi r}$ along the line A–B.

Comment on the results obtained near the tip. Compute the value of K_I by extrapolation to $r = 0$ of the last few nodes. Compare the value with the known solution $K_I = 6.67$, exact within 1%.



Centre cracked plate under traction and discretization of one quarter of the plate.