differ significantly from small scale laboratory results. In addition, it is not possible to model or even map all the faults, dykes and joints present in the rock mass.

Very often, therefore, it is necessary to select a range for each of the above parameters and to try to assess their influence on the final results. In addition, analyses are often carried out to compare the relative stability of a number of different geometries or to assess the effect of different primitive stress conditions on a particular geometry rather than to predict categorically the stability of a given excavation. Confidence in the method is obtained once predictions have been observed in the field.

The above uncertainties are imposed by the particular problem at hand. These have to be considered together with the numerical uncertainties imposed by the particular numerical technique being used. When boundary element or finite element analyses are carried out, one of the first questions concerns the size of the grid or mesh. One rule of thumb which has proved particularly useful is described in some detail below.

It is not advisable to attempt calculating stresses within one element width of each element, where the width is defined as the square root of the element area. This restriction arises primarily from the assumption that displacements and tractions are constant over any one element. As a result a displacement jump or discontinuity arises at each element boundary resulting in localised stresses which are infinite. The tractions, however, usually vary much more slowly than displacements except of course at edges or corners, where the true tractions are discontinuous. If it is necessary to calculate stresses within one element width, then these should be calculated over the element centroid rather than above the element boundaries (Diering, 1980b).

This simple rule leads to numerous other rules of thumb which assist the user in drawing up a boundary element mesh.

(i) The aspect ratio of these areas of adjacent elements in the same plane should not exceed 4 or 5 to 1.

(ii) When a tabular excavation is being modelled, elements on opposite surfaces of the excavation should be separated by at least one element width.

(iii) Elements on opposite sides of solid material should be separated by at least three element widths. This enables a reasonable stress distribution to be obtained within the intervening solid material.

(iv) The angles between elements at corners or edges should exceed sixty degrees, if possible.

(v) At least two and preferably three elements should be used to span the side or edge of an excavation.

The following factors contributed towards drawing up and explaining the above rules.

(a) The use of numerical integration of limited accuracy can be expected to break down if elements are too close to one another.
(b) Considerable experience has been gained from the use of the two-dimensional programme MINAP (Crouch, 1976) and three-dimensional displacement discontinuity formulations for tabular excavations (Starfield and Crouch, 1973).
(c) The use of the programmes to model problems for which analytic or other solution were available.

In addition, it was found that a badly specified boundary element mesh usually produced a set of equations which was divergent or converged very slowly.

The above rules were used in drawing up the meshes for the practical examples which are described below. It should be borne in mind at all times that the computer programmes described here are merely tools which provide the engineer with stress and displacement information about a real problem. Interpretation of the results to predict failure or select pillar sizes etc must still rely on engineering judgement and experience.

## 9.1 Programme BEM

### 9.1.1 Underground excavations close to surface

A nickel mine in Southern Africa is at present mining two major ore bodies in close proximity. The larger and smaller are here termed orebodies A and B respectively. Collapse of the 18 m thick crown pillar above the workings in orebody A prompted a stress analysis of the situation to determine whether further failures could be expected. The complexity of the situation required a three-dimensional analysis. Even so, the geometry was considerably simplified as shown in Fig 9.1 for the analysis. The situation was modelled as a homogeneous elastic problem using 300 boundary elements and stresses and displacements were determined at numerous points in and around the crown pillars of the two excavations.

From the displacements it was apparent that overall surface uplift occurred as a result of mining but that the immediate hangingwall of excavation A subsided under its own weight. An extensive zone of horizontal and vertical tension was revealed by the calculated stresses and this can be attributed to the surface deformation pattern. A representative example of the tensile zone is shown in Fig 9.2 as a NE-SW section through the centre of the excavation. Because rock is weak in tension and because of extensive weathering of the near surface rock, it was possible to delineate a potential zone of failure around excavation A. In this zone, failure would typically result from the combination of tensile failure close to surface and shear failure extending down to the excavation. Calculated shear stresses in this region exceeded measured values of cohesion. The modelled and observed failure zones are shown in Fig 9.3. The actual failure zone extended approximately 40 m to the north of the modelled failure zone, this difference being due to the presence of a weak geological contact (between the talc carbonate host rock and neighbouring greestone). Calculations show that it is also possible that

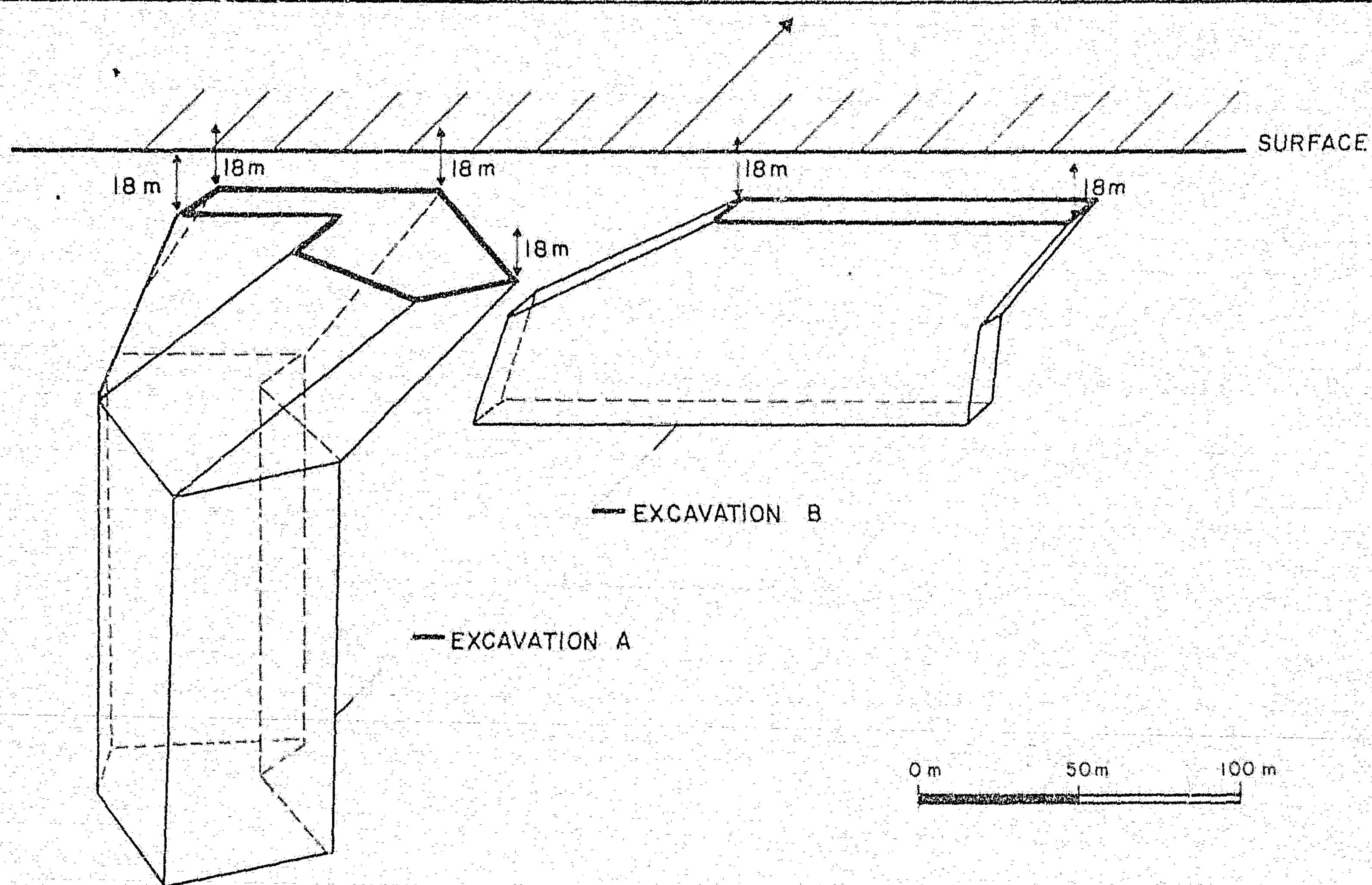


FIG. 9.1 GEOMETRY USED TO MODEL FAILURE OF CROWN PILLAR ABOVE EXCAVATION A

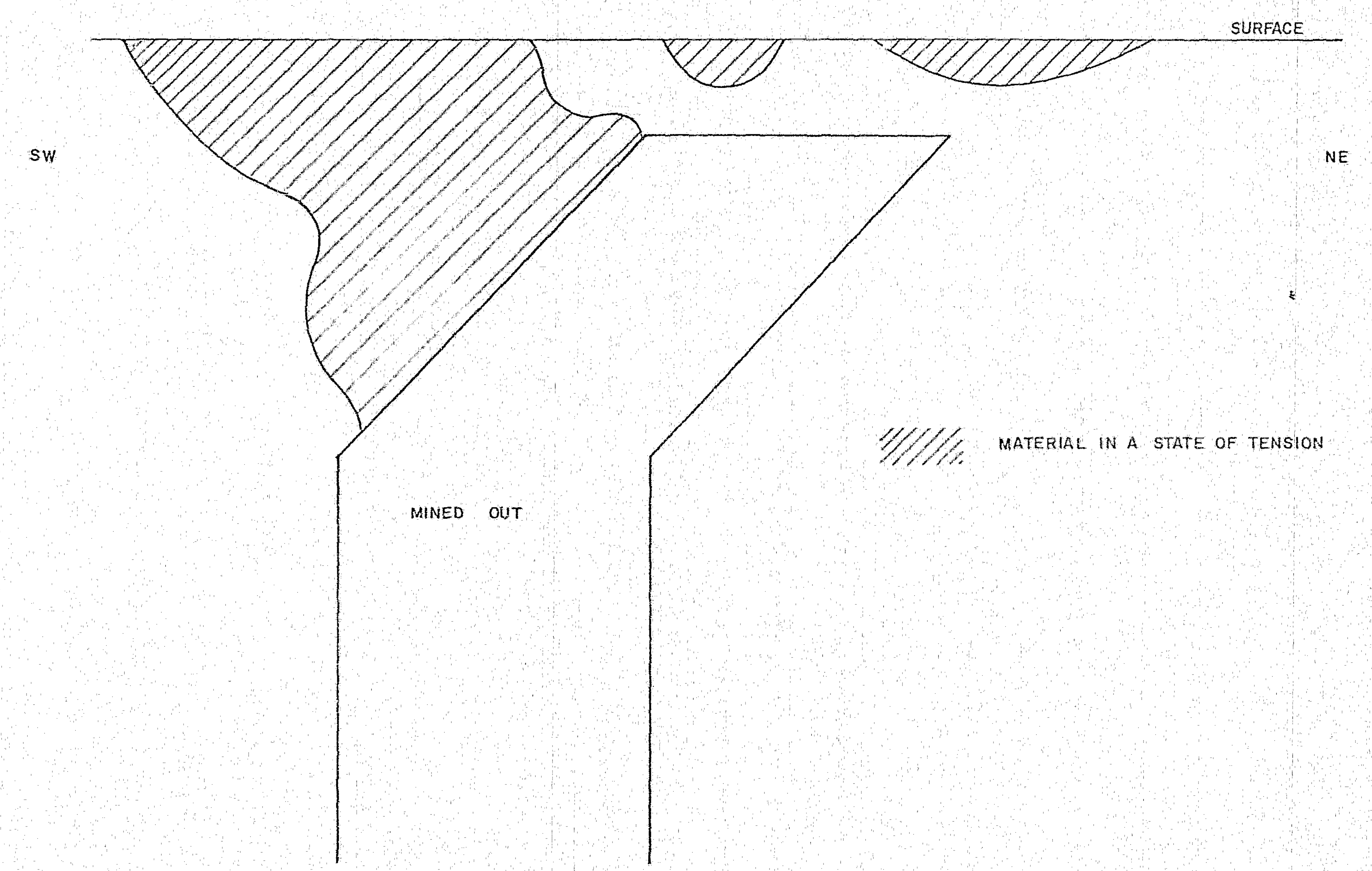


FIG. 9.2 SECTION THROUGH BOUNDARY ELEMENT MODEL SHOWING ZONES OF TENSION ABOVE THE EXCAVATION

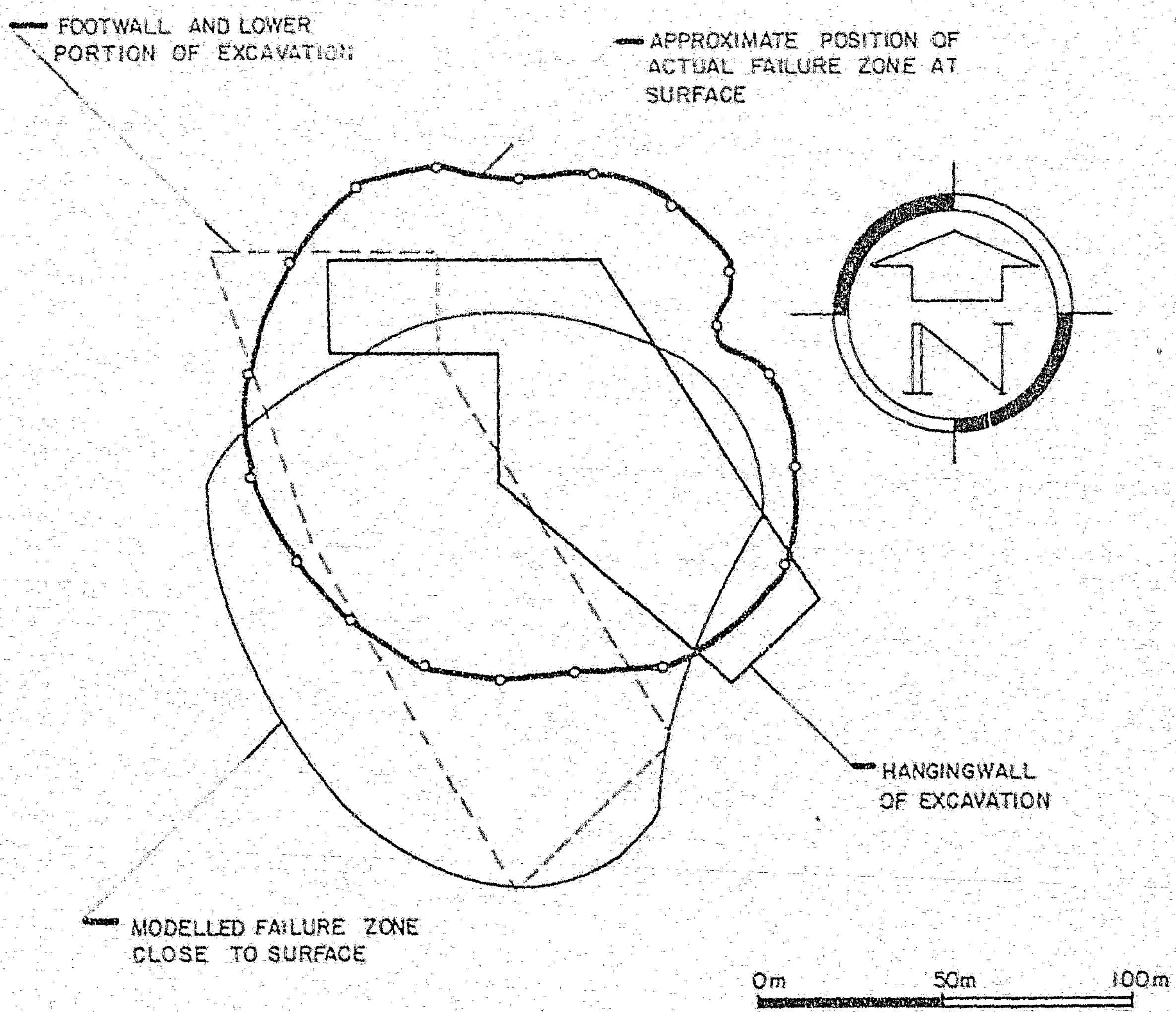


FIG. 9.3 PLAN SHOWING OBSERVED AND MODELLED ZONES OF FAILURE RELATIVE TO THE HANGINGWALL AND FOOTWALL OF EXCAVATION A

futher collapse of the crown pillar will take place south of the present zone. This corresponds with future mining at depth which will be further to the south as shown by the plan ouline of the lower portion of excavation A in Fig 9.3.

It is important to emphasize that this analysis was carried out using a bare minimum of boundary elements and after making several simplifying assumptions about the excavation geometry. Thus failure of excavation A was able to be modelled by a simple elastic boundary element analysis while excavation B, which has a smaller span, was shown to be stable due to the absence of any extensive tensile zone.

### 9.1.2 Interaction of underground and open pit excavations

Fig 9.4 shows a schematic representation of the geometry for this problem. A number of items were of interest, the most important being:

(i) the stability of the crown pillar between the underground and open pit excavations,

(ii) the stability of the hangingwall above the larger underground excavation,

(iii) the optimum size and shape for the pillar between the stope and larger underground excavation (Fig 9.4), and

(iv) the stresses below the stope, since this is where orepasses and service excavations are located.

Use was made of one plane of symmetry to reduce the total number of elements, but more than 350 elements were required for the problem. Unlike the previous example, stresses and displacements were required at a number of widely separated points, so that a fairly uniform and fine mesh was required. As the primitive stresses were not known with any certainty, crown pillar will take place south of the present zone. This corresponds with future mining at depth which will be further to the south as shown by the plan ouline of the lower portion of excavation A in Fig 9.3.

It is important to emphasize that this analysis was carried out using a bare minimum of boundary elements and after making several simplifying assumptions about the excavation geometry. Thus failure of excavation A was able to be modelled by a simple elastic boundary element analysis while excavation B, which has a smaller span, was shown to be stable due to the absence of any extensive tensile zone.

### 9.1.2 Interaction of underground and open pit excavations

Fig 9.4 shows a schematic representation of the geometry for this problem. A number of items were of interest, the most important being:

(i) the stability of the crown pillar between the underground and open pit excavations,

(ii) the stability of the hangingwall above the larger underground excavation,

(iii) the optimum size and shape for the pillar between the stope and larger underground excavation (Fig 9.4), and

(iv) the stresses below the stope, since this is where ore-passes and service excavations are located.

Use was made of one plane of symmetry to reduce the total number of elements, but more than 350 elements were required for the problem. Unlike the previous example, stresses and displacements were required at a number of widely separated points, so that a fairly uniform and fine mesh was required. As the primitive stresses were not known with any certainty,

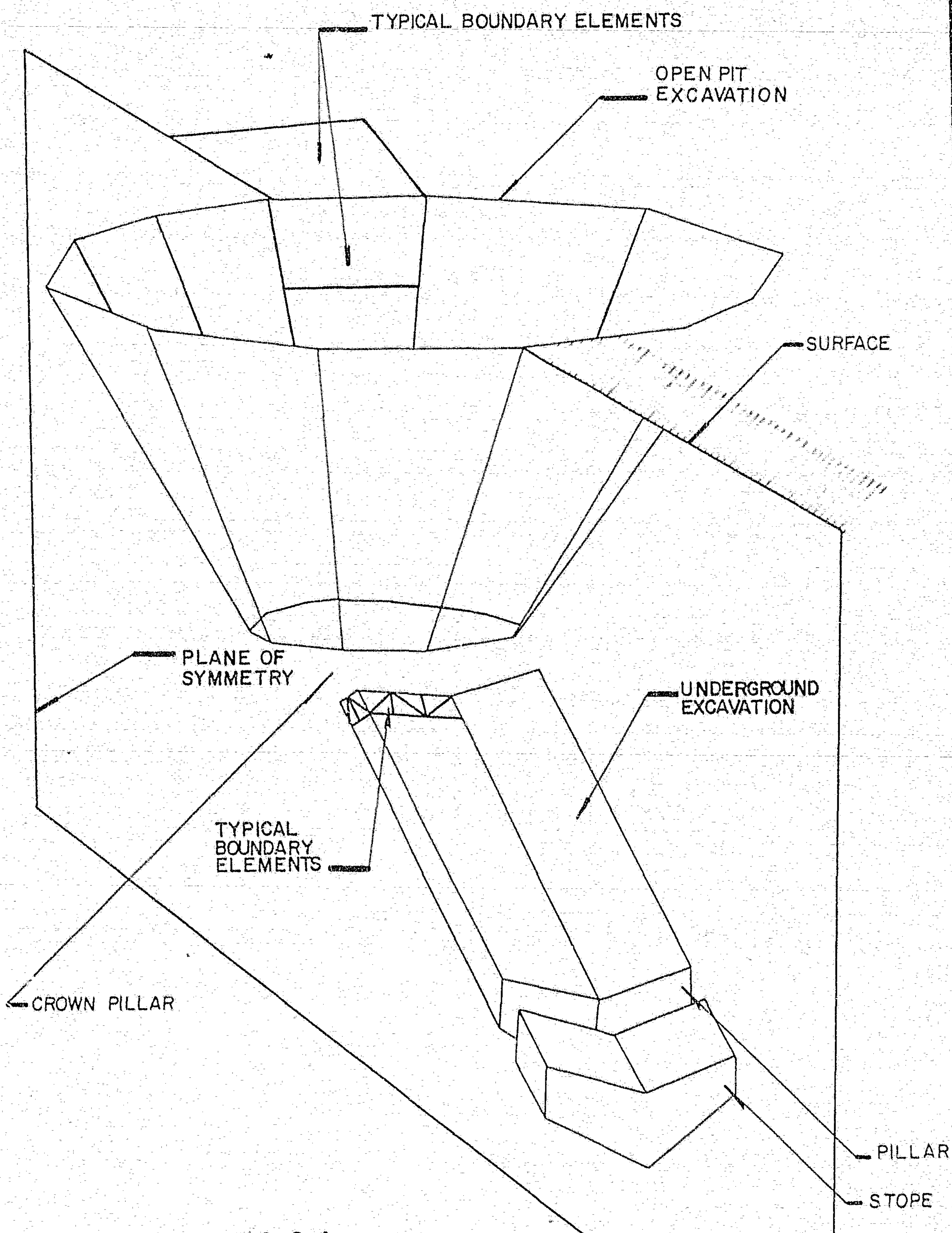


FIG. 9.4

GEOMETRY USED TO MODEL INTERACTION OF OPEN PIT AND UNDERGROUND EXCAVATIONS it was necessary to do a number of different runs to check the effects of different primitive stress states.

It proved worthwhile to write a small programme which generated a line or plane of points or benchmarks at which stresses were to be calculated. For example, three planes of benchmarks were inserted into the pillar between the stope and underground excavation and three in the stope footwall.

Typically, from two to four hundred points were selected as benchmarks and again it proved worthwhile to write a small programme to analyse the points. This programme high-lighted areas of high tension, or high shear on specified planes. It also described the position of each point, ie five metres in the footwall of the stope or fifteen metres in the hangingwall etc. It is felt that this type of "post processing" facility should form an integral part of the analysis cycle.

Thus a fairly simple analysis was able to provide adequate information to the design engineer enabling him to answer the above questions with increased confidence.

### 9.1.3 Circular pile in sandstone

Although this is not a mining problem, it is included here since it demonstrates the interaction between subregions with different elastic properties as well as the effects of slip across the vertical interface between pile and sandstone. 186 elements were used together with 2 planes of symmetry. The pile diameter was 1,8 m and the vertical pile load equivalent to 5,28 MPa (Fig 9.5). Table 9.1 gives a comparison of vertical stresses and displacements at various points and it is seen that the load shedding characteristics of the pile depend significantly upon the choice of cohesion and angle of friction. The pile and sandstone moduli were:

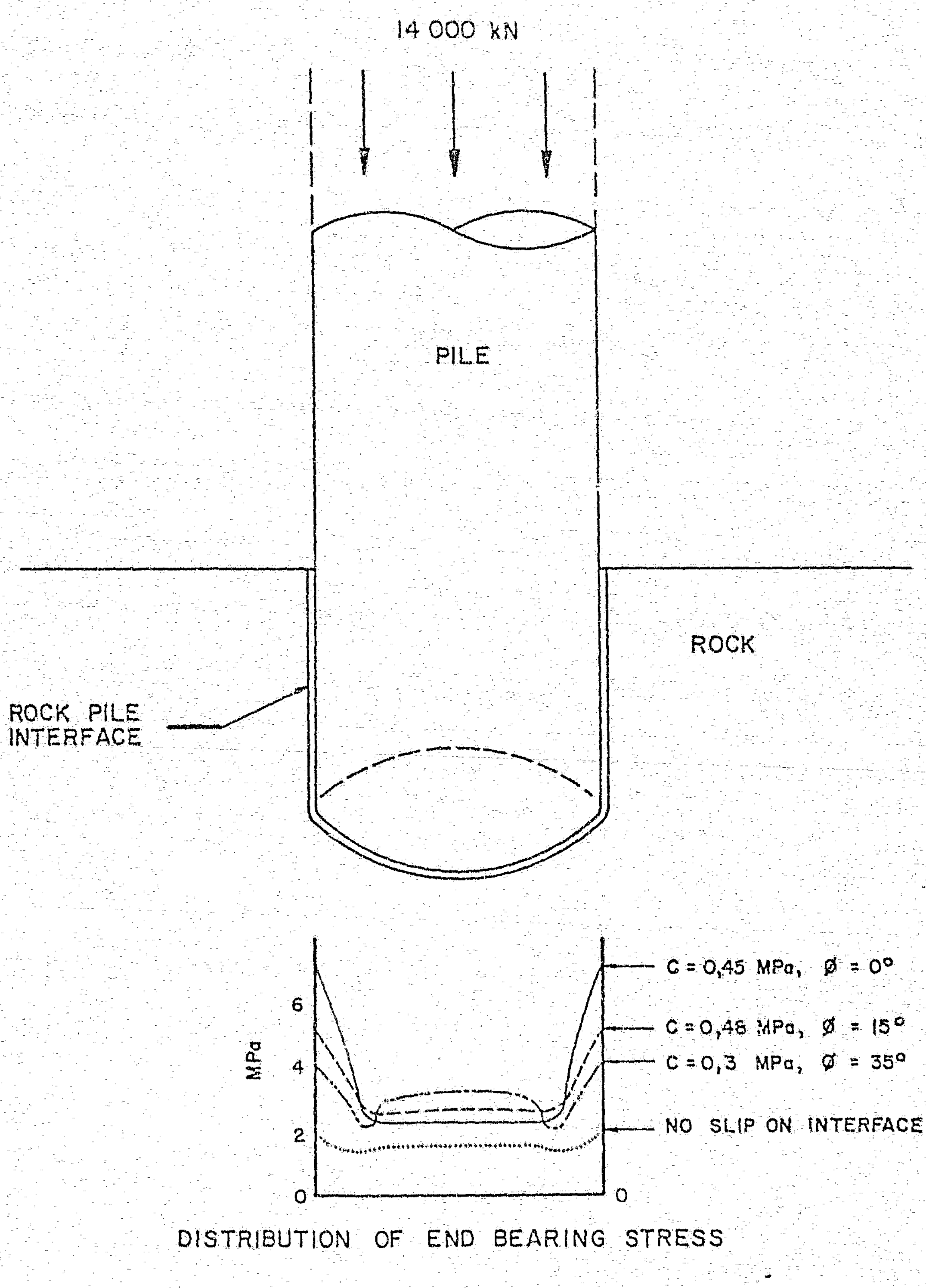


FIG. 9.5 PILE SOCKETED IN SANDSTONE

$E_1 = 26\ 000$ MPa

$\nu_1 = 0{,}22$

$E_2 = 20\ 000$ MPa

$\nu_2 = 0{,}25$

respectively.

TABLE 9.1: LOAD SHEDDING OF CIRCULAR PILE AND SETTLEMENTS FOR VARIOUS COHESIONS AND ANGLES OF FRICTION

| Item | Run 1 | Run 2 | Run 3 |
|---|---|---|---|
| Cohesion (MPa) | 9999 | 0,48 | 0,45 |
| Angle of friction | 30° | 15° | 0° |
| Settlement (mm) | 0,95 | 1,07 | 1,10 |
| Vertical stress below centre (1) | 1,40 | 2,40 | 2,60 |
| Resisting stress at surface (2) | 3,00 | 0,90 | 0,45 |
| Pressure at base edge of pile(3) | 1,40 | 3,80 | 4,0 |
| Vertical stress at radial distance of 1,5 m 0,25 m below pile (4) | 0,50 | 0,31 | 0,28 |

In particular, Table 9.1 shows how the load is transferred directly below the pile as cohesion and angle of friction of the pile/sandstone interface are reduced. The load distribution at the end of the pile is also shown in Fig 9.5. It should also be noted that no other boundary element formulation has been found in the literature capable of modelling this particular situation.

## 9.2 Programme MBEM

### 9.2.1 Tabular underground excavation close to an open pit excavation.

A copper mine in Central Africa is currently mining two tabular ore bodies, the upper by open pit and the lower by underground block caving. As the two excavations approach one another, the stability of the open pit slopes is gradually being reduced. The problem was well suited to an analysis using the mixed boundary element programme. Some 220 boundary elements were used to model the open pit and surrounding ground surface, while the underground excavation was initially modelled with a 12x12 and subsequently with a 20x24 grid of displacement discontinuity elements. A schematic representation of the situation is shown in Fig 9.6.

This example differs from the previous examples in that displacements, not stresses, were of interest. Although the surrounding rockmass is grossly non-homogenous, this simplified homogeneous elastic analysis was carried out with the aim of correlating surface elastic deformations with observed cave cracks. Since the boundary element method gives displacements more reliably than stresses, it was felt that the rather coarse mesh of 220 elements was adequate.

A major advantage of the displacement discontinuity element formulation is that it is easy to alter the underground mine layout without having to re-draw the entire mesh. Contours of surface displacements for various underground mine layouts were used to assess relative influences of open pit and underground mining in each case. Details of the grid used to model the underground mining are shown in Fig 9.7. Different shading patterns show the extent of mining for various alternatives. A typical contour map of vertical displacements for the "present" mining layout is shown in Fig 9.8.

### 9.2.2 Tabular excavation close to a vertical fault

The presence of major faults in the gold mines of Southern Africa gives rise to numerous rock mechanics problems. In particular, large amounts of energy may be released as the mining approaches a fault. As the energy released sometimes manifests itself in the form of rockbursts and often results in adverse mining conditions, the manner in which the fault is approached is of considerable importance. Two-dimensional modelling of this type of situation has been well described by Crouch (1976). Usually, however, the mining and fault make up a fully three-dimensional problem.

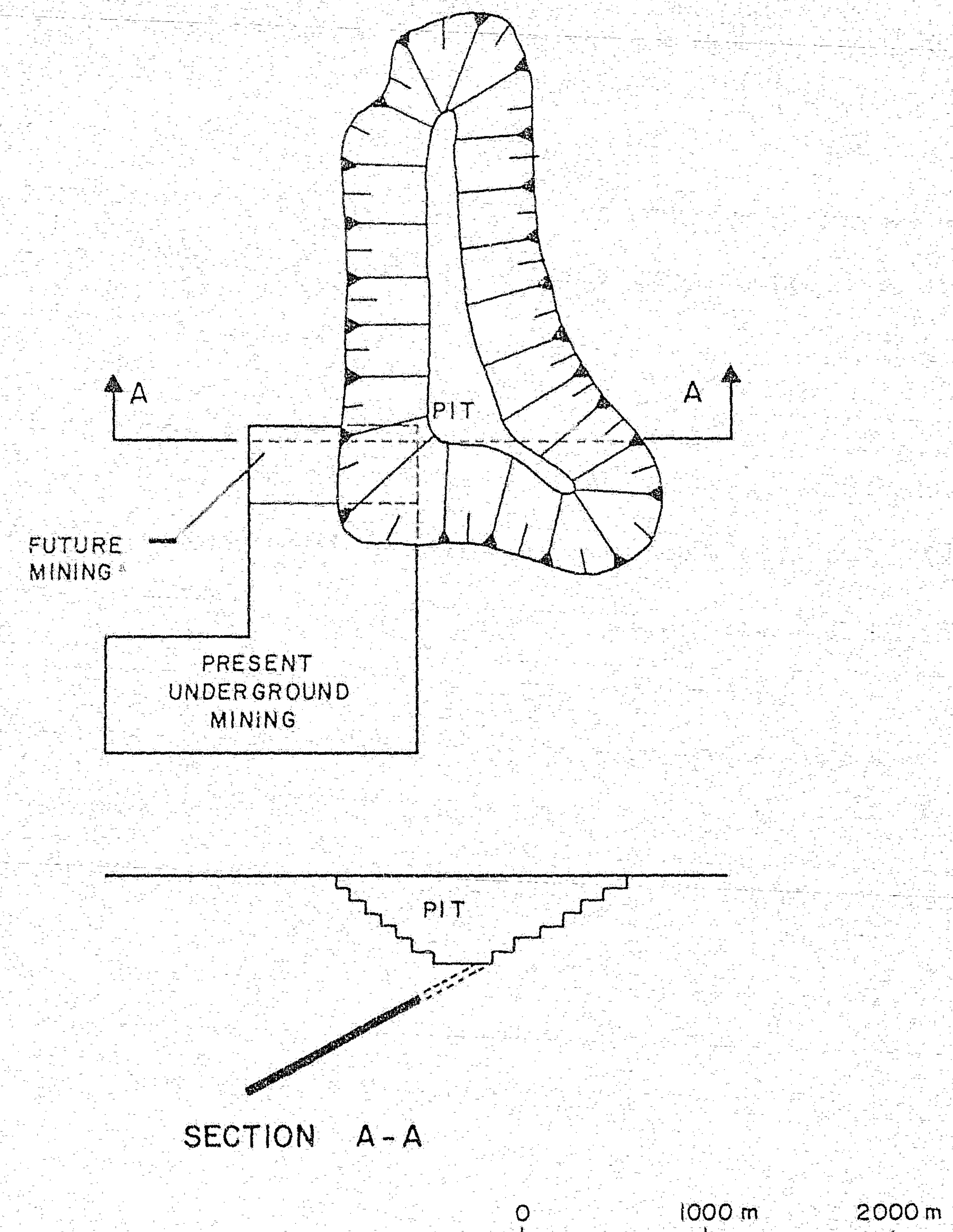


FIG. 9.6 SCHEMATIC REPRESENTATION OF OPEN PIT AND TABULAR UNDERGROUND EXCAVATIONS

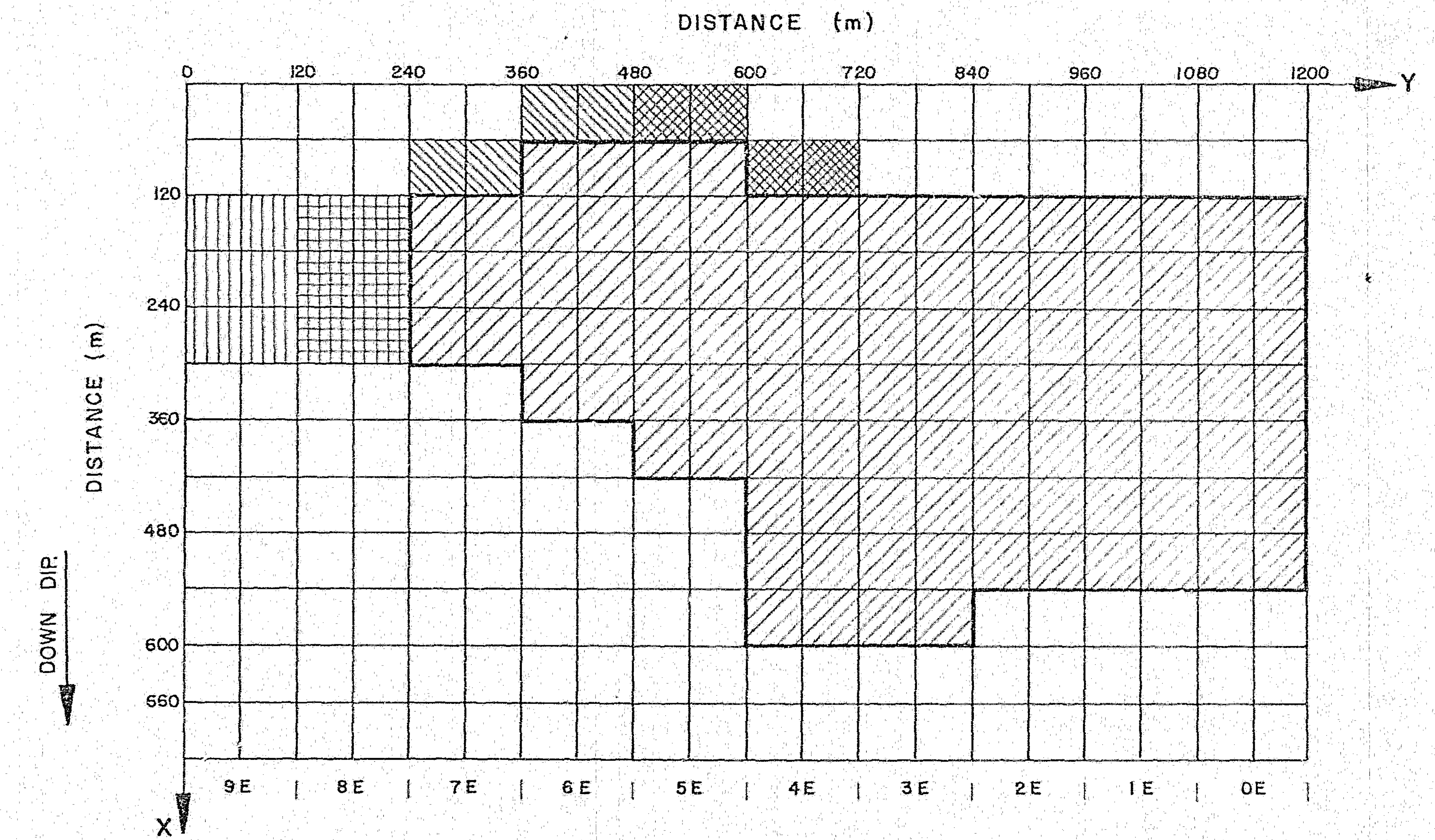


FIG. 9.7 **DETAILS OF UNDERGROUND GRID OF SEAM ELEMENTS.**

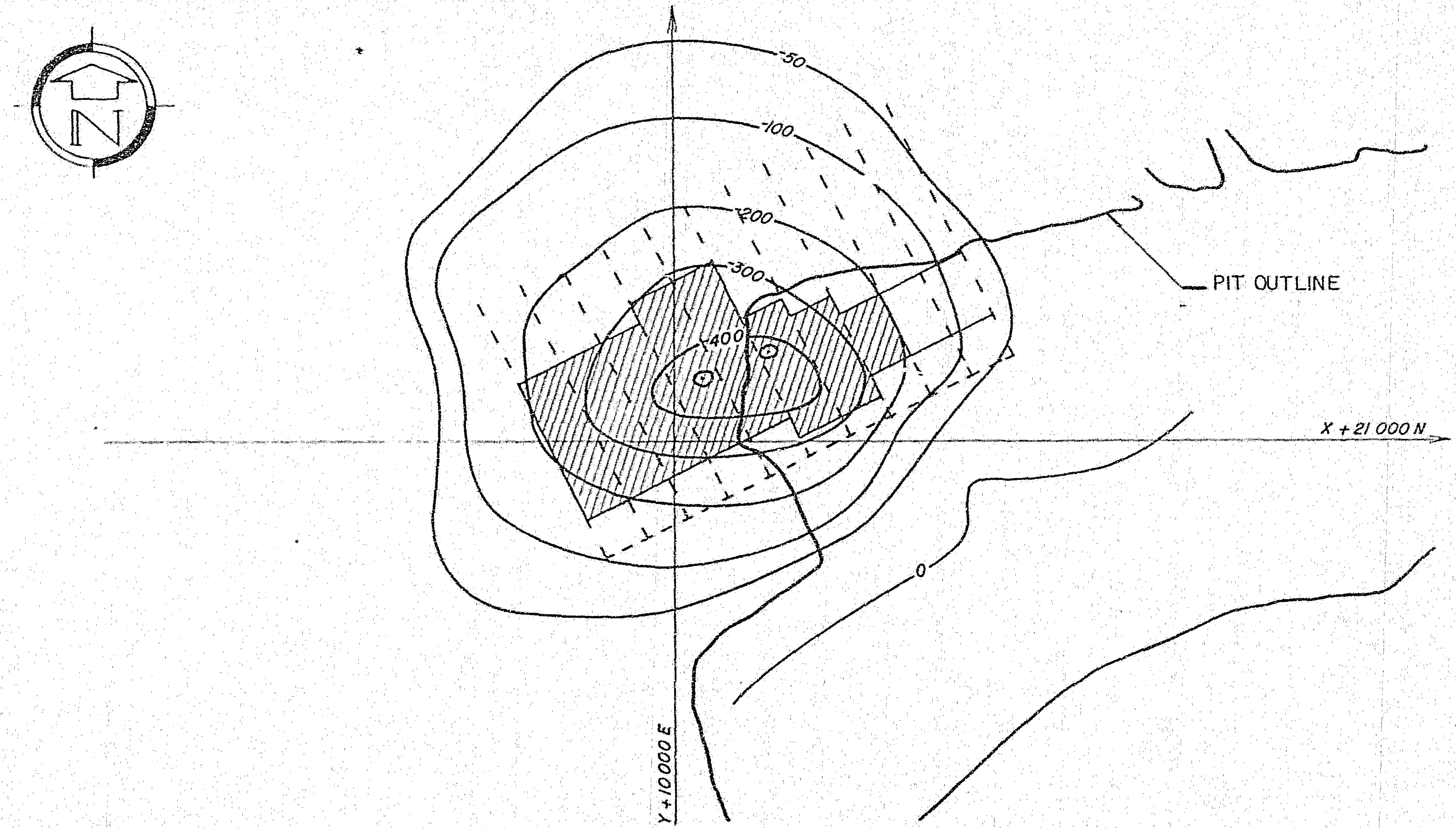


FIG. 9.8 : CONTOURS OF VERTICAL DISPLACEMENT (Z DIRECTION) — PRESENT MINING SITUATION UP TO 7E

A simple test problem was devised to test the MBEM programme. A six by six array of displacement discontinuity elements was placed perpendicular to a ten by ten array of double or interface boundary elements (Fig 9.9).

Two mining steps were carried out and in the second step, the amount and extent of slip occurring along the fault was much greater than for the first step. Material properties and loading for the analysis were:

| | | |
|---|---|---|
| Young's Modulus | = | 60 000 MPa |
| Poisson's Ratio | = | 0,2 |
| Fault Cohesion | = | 0 |
| Fault Angle of Friction | = | 10 degrees |
| Vertical Primitive Stress | = | 60 MPa |
| Horizontal Primitive Stress | = | 30 MPa |

Table 9.2 shows variation of closure vs distance from the fault along a section normal to the fault through the centre of the excavation. The three-dimensional programme RIDE was used to obtain check values for the case in which no fault was present. (ie no fault slip allowed). Maximum fault shear displacements are also shown in each case.

TABLE 9.2: FLAT REEF INTERSECTING A VERTICAL FAULT

| Programme | Step | Fault Slip | Fault Shear (mm) | Closure (mm) | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | 15 m | 45 m | 75 m | 105 m | 135 m | 165 m |
| RIDE | 1 | NO | 0 | 63 | 90 | 103 | 103 | 91 | 64 |
| MBEM | 1 | NO | 0 | 61 | 88 | 102 | 102 | 89 | 63 |
| MBEM | 1 | YES | 30 | 81 | 96 | 105 | 104 | 90 | 63 |
| MBEM | 2 | YES | 53 | 125 | 127 | 130 | 125 | 110 | 83 |
| MINAP | - | NO | 0 | 120 | 163 | 181 | 181 | 163 | 120 |
| MINAP | - | YES | 125 | 268 | 258 | 245 | 224 | 192 | 136 |

Although this is a very simple analysis in which a bare minimum of fault interface elements and displacement discontinuity elements were used, it demonstrates the marked difference between equivalent two and three-dimensional analyses.

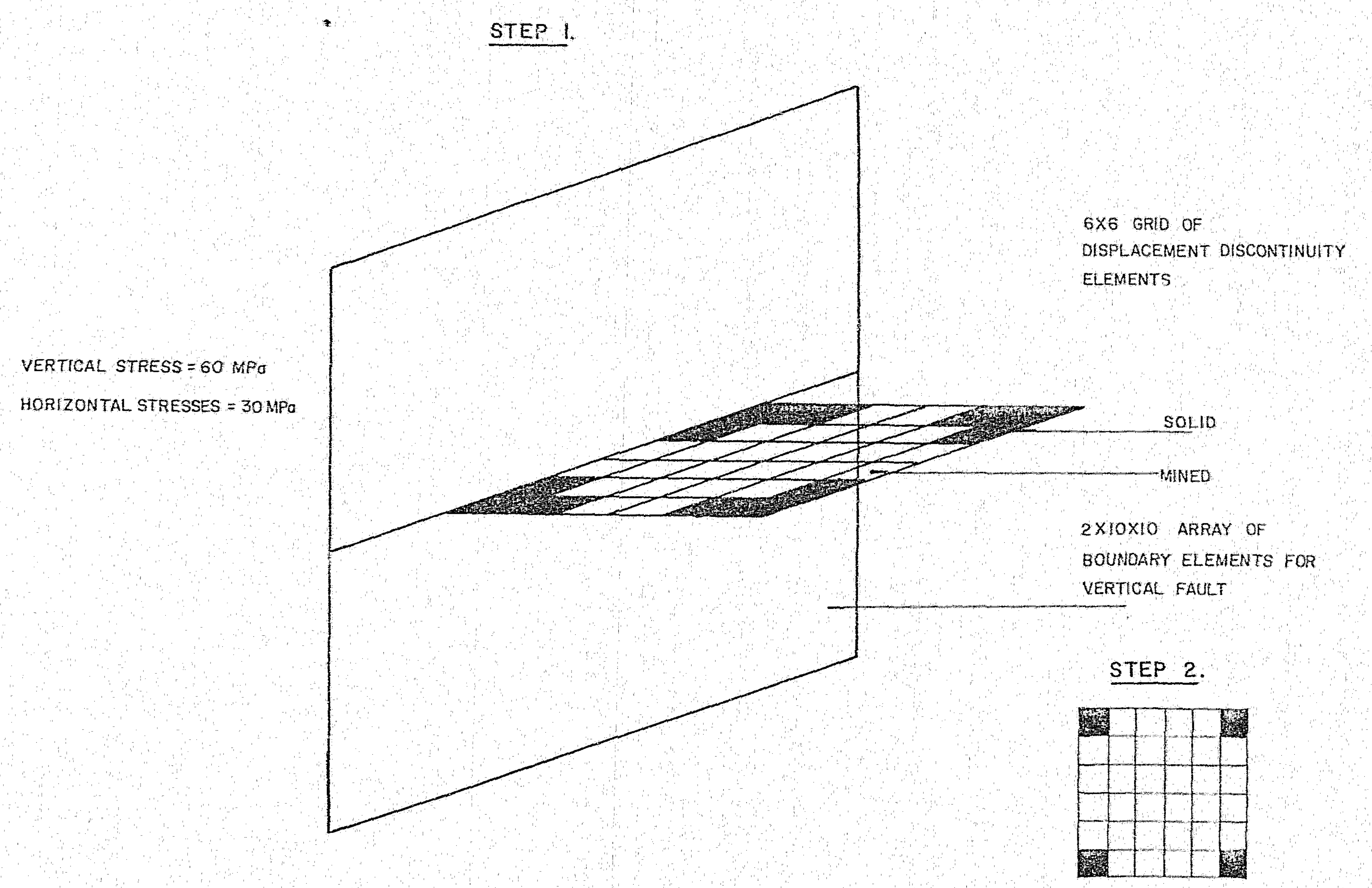


FIG. 9.9 TABULAR EXCAVATION MINING UP TO A VERTICAL FAULT

It also shows that thousands, rather than hundreds, of elements are required to model this type of problem realistically in practice. The simple type of analyses is nevertheless useful for studying representative problems and for modelling general effects.

## 9.3 Programme MINAPH

### 9.3.1 Mining of a coal seam below a dolerite sill

Many South African coal mines have coal seams which are overlain by fairly thick dolerite sills. The stresses in these sills and their effect on stresses in and around the coal seams are of considerable interest. Fig 9.10 shows an example selected to test the MINAPH programme for this type of problem. The stresses and displacements for the 12 reference points in Fig 9.10 are given in Table 9.3. These values are compared with an equivalent homogeneous run.

TABLE 9.3: COMPARISON OF STRESSES AND DISPLACEMENTS FOR NON-HOMOGENEOUS AND HOMOGENEOUS ANALYSES

| Reference points | Non-homogeneous model | | | Homogeneous model | | |
|---|---|---|---|---|---|---|
| | $u_y$ (m) | $\sigma_x$ (MPa) | $\sigma_y$ (MPa) | $u_y$ (m) | $\sigma_x$ (MPa) | $\sigma_y$ (MPa) |
| 1 | −0,043 | 0,6 | 1,1 | −0,039 | 1,0 | 0,4 |
| 2 | −0,044 | 0,5 | 1,4 | −0,040 | 0,8 | 0,1 |
| 3 | −0,044 | −6,4 | 0,3 | −0,042 | 0,3 | −0,1 |
| 4 | −0,044 | −2,1 | 0,0 | −0,044 | −1,0 | −0,1 |
| 5 | − | − | 0,0 | | − | 0,0 |
| 6 | 0,011 | −1,4 | 0,3 | 0,013 | −0,2 | 0,2 |
| 7 | −0,035 | 0,5 | 1,3 | −0,032 | 0,4 | 1,6 |
| 8 | −0,034 | −5,9 | 2,8 | −0,030 | 0,0 | 2,9 |
| 9 | −0,034 | −5,0 | 6,4 | −0,028 | −0,4 | 4,6 |
| 10 | −0,029 | 0,3 | 10,7 | −0,023 | 0,8 | 9,4 |
| 11 | −0,011 | − | 12,9 | 0,010 | − | 12,0 |
| 12 | −0,006 | −0,5 | 0,2 | −0,003 | 0,7 | 8,6 |

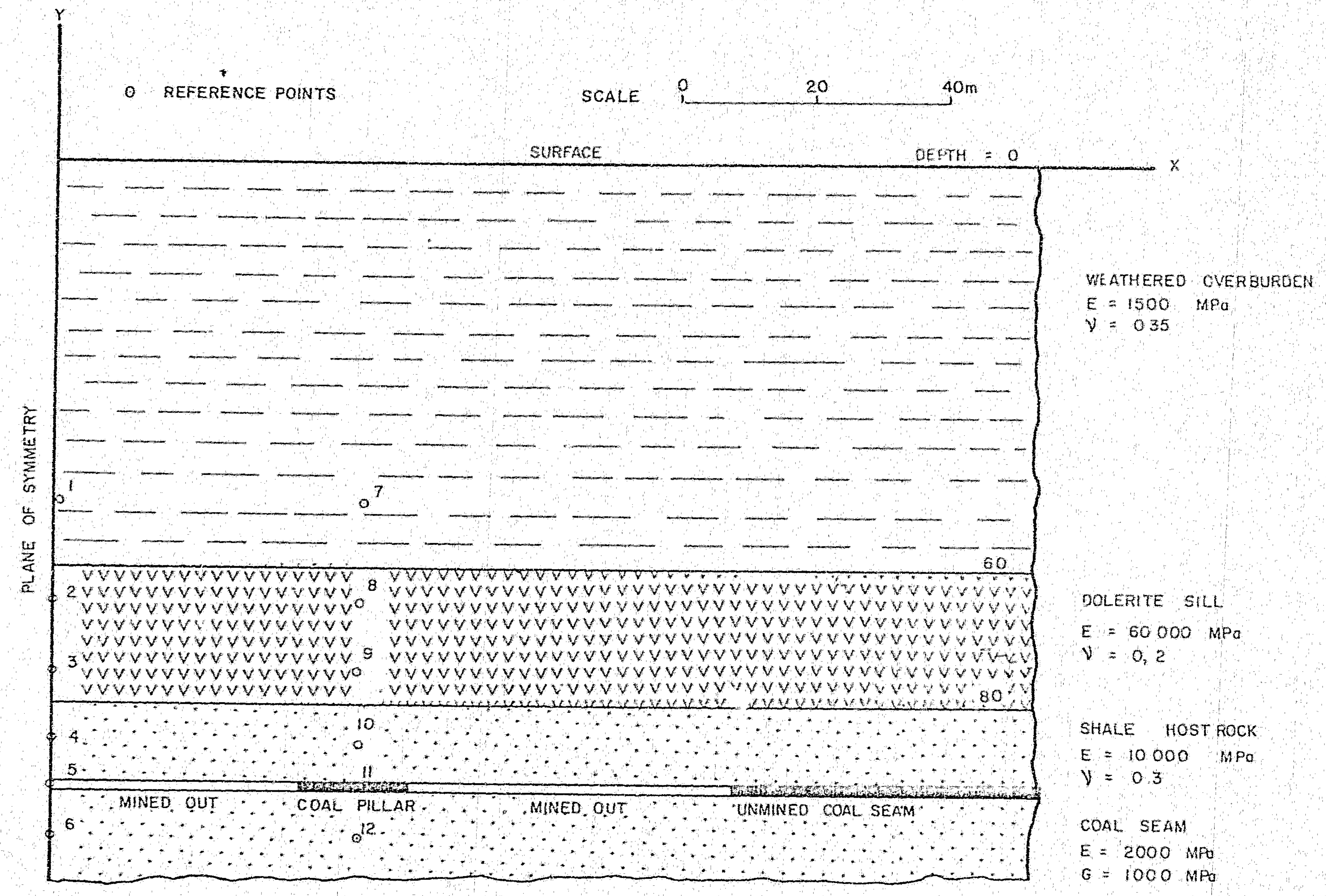


FIG 1.0 COAL SEAM EXTRACTION BELOW A DOLERITE SILL

Although the displacements are similar in each case, the homogeneous model fails to detect t   ry high horizontal tension in the sill while over-estimating the extent of induced tension above the excavation. As rock (especially jointed rock) is weak in tension, it is important to know where zones of tension occur since this demonstrates if and where failure is likely to initiate. The non-homogenous linear analysis cannot model the complete failure path, but is at least an improvement over the equivalent linear homogenous analysis.

### 9.3.2 Tabular excavation with different footwall and hangingwall properties

This problem was selected to see how different hangingwall and footwall materials might affect the behaviour of a fault in close proximity to a tabular excavation (Fig 9.11). Constant field stresses of 30 MPa and 60 MPa were applied in the horizontal and vertical directions respectively. Elastic parameters for the hangingwall and footwall were as follows.

| | | |
|---|---|---|
| Hangingwall: | $E_1$ | $= 50\ 000$ MPa |
| | $\nu_1$ | $= 0{,}35$ |
| Footwall: | $E_2$ | $= 80\ 000$ MPa |
| | $\nu_2$ | $= 0{,}20$ |

The analyses showed that the normal stresses and hence the shear strengths along the fault were the same, within a few percent, in both the hangingwall and footwall rocks. The shear displacements were about 1,5 times greater in the hangingwall than in the footwall. Consequently, to a first degree of approximation, some 60% of the energy dissipated along the fault would be in the hangingwall.

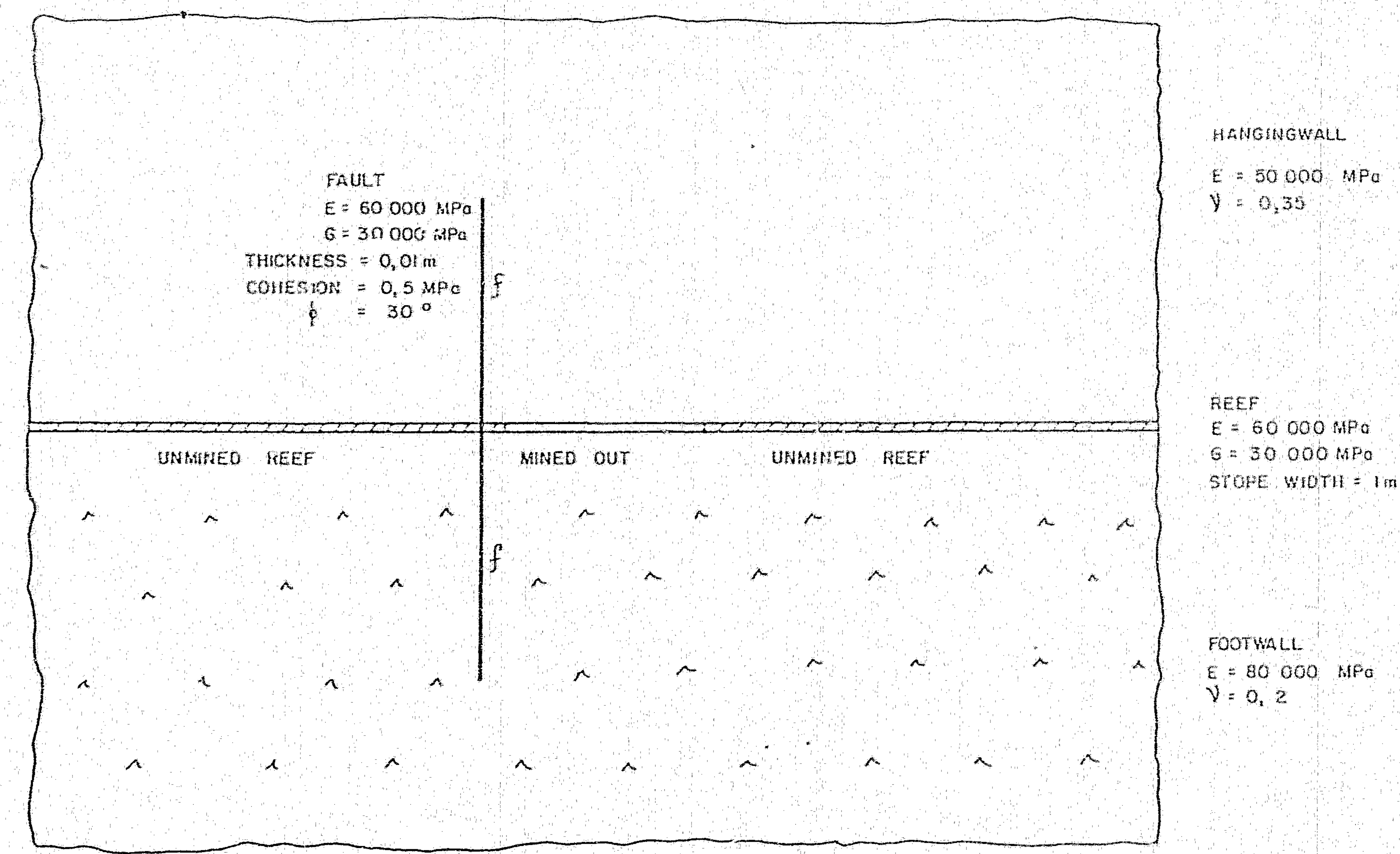


FIG. 9.11 TABULAR EXCAVATION APPROACHING A FAULT

## CHAPTER 10 GENERAL DISCUSSION AND CONCLUSIONS

It is difficult to evaluate the utility of the formulations described in this dissertation when the range of boundary, displacement discontinuity and finite element formulations already in existence is so large. When considering three-dimensional elastostatic formulations directly applicable to mining engineering problems, then the finite element method becomes very inefficient in most cases as large numbers of elements are usually required.

The displacement discontinuity method has found wide application for the modelling of tabular excavations in homogeneous ground and in two dimensions has been extended (Crouch, [illegible]6) to enable realistic modelling of mining in homogeneous faulted ground. The extension of the displacement discontinuity method described here which enables mining in non-homogeneous ground to be modelled is therefore a logical and useful extension.

In three dimensions, the main applications of the displacement discontinuity method in South Africa have been the modelling of tabular excavations. In the coal mining industry, multiple coal seams at relatively shallow depths are often encountered. The "multi-reef" MINSIM or SHAMIP programmes developed by the Chamber of Mines of South Africa are well suited to this type of problem in most cases and it is only when outcropping coal seams or very irregular ground surfaces are encountered, that the mixed boundary element method of this dissertation would prove useful. The multilayered stratigraphies in many instances are not well suited to any type of three-dimensioanl analysis since a very large number of finite or boundary elements is normally required. Such problems are probably best modelled by a three-dimensional finite element analysis, especially if use can be made of numerous planes of symmetry or repetition.

A wide variety of displacement discontinuity formulations exist for modelling the tabular excavations encountered in the gold and platinum mines of South Africa including the SHAMIP, MINSIM and "Multi Reef MINSIM" programmes of the Chamber of Mines and the FREEF, DREEF and MREEF programmes of Crouch. These programmes do not allow for different hangingwall and footwall material or for the presence of faults or dykes. The formulations of this dissertation provide means of modelling these situations. Practical problems of this type would require a large number of elements, probably in excess of two thousand. A computer larger than the Data General Nova 3 on which these programmes were developed would be required. This should not detract from the utility of the programmes, however, as the Data General Nova 3 is a very small mini-computer by modern standards.

Perhaps the best applications for the programmes of this dissertation arise from the interaction of open pit and underground excavations or for underground caverns such as are encountered in underground hydro electric power stations and massive underground mining operations for example. The three-dimensional boundary element programmes of the Chamber of Mines and of Lachat and Watson are also well suited to this type of problem. These formulations are not designed to run on small mini-computers and as such are not as readily available to the rock mechanics practitioner and other users.

Finally, a brief summary of the good points and shortcomings of the formulations of this dissertation is given. In addition, scope for future development is high-lighted.

The main advantages of the formulations of this dissertation may be summarised as follows:

(1) The combination of boundary and displacement discontinuity elements in a single programme is demonstrated to be both possible and practical.

(2) The lumping scheme described here forms an essential part of the formulations for the following main reasons:
   (i) Reduction of disk storage and execution time.
   (ii) Reduction of core storage requirements.
   (iii) Numerous checks for user input errors.

(3) The facility for allowing non linear behaviour on interface boundaries should prove extremely useful in the future. This has

already been demonstrated by the two-dimensional programme MINAP of Crouch (1976).

(4) The use of an iterative solution technique as opposed to a direct technique has the following main advantages:

  (i) Reduction in execution time by at least an order of magnitude for most problems.
  (ii) It enables the lumping mechanism to be implemented.
  (iii) It enables non-linear behaviour of displacement discontinuity or interface boundary elements to be modelled fairly efficiently.

(5) The formulations are designed for and execute on a small mini-computer making them both more accessible and cheaper to run for those users who have in house mini-computers and previously had to rely on a computer beureux for this type of analysis.

(6) Both the boundary and displacement discontinuity elements are the constant displacement/traction type so that their accuracies are very similar. This, the simplest element type, has the following advantages:

  (i) It is well suited to the development of formulations such as have been described here.
  (ii) Failure of boundary interface elements poses no special problems.
  (iii) In mining problems, negligible errors are introduced by the assumption of constant tractions over each element, since the primitive or virgin stress field which gives rise to the tractions on each element usually varies slowly with depth.

The main disadvantages of the BEM and MBEM programmes are as follows:

(1) The use of constant displacement/traction elements often necessitates the use of a large number of elements. Most of the basic ideas

described in the dissertation, however, should be applicable to more sophisticated element types.

(2) Some problem geometries, especially those involving more than one subregion, produce sets of equations which cannot readily be solved using the method of Successive over relaxation.

(3) The present MBEM formulation caters only for one regular array of displacement discontinuity elements, so that mining up to a fault could be modelled, but not mining through a fault.

Very considerable scope for further development of and improvements to the BEM and MBEM formulations exists at present.

(1) A linear or higher order variation of displacements and tractions over each element is desirable.

(2) The numerical integration and lumping schemes, although efficient at present, can still be improved and optimised, especially for very large problems.

(3) It is suspected that alternative iterative techniques might perform better than the Successive over relaxation method being used at present. Two methods look fairly promising, the first being the conjugate gradient method (Reid, 1970) and the second, a partial elimination scheme (Tuff, 1981).

(4) For large problems, it is not only generation of the boundary element mesh that is time consuming but also interpretation of the output data. Current development is aimed at improving the "pre-" and "post-processing" facilities available to the BEM and MBEM programmes. An interface with computer graphics would be particularly useful.

(5) Problems involving the earth's surface are frequently encountered. A semi infinite formulation based on Mindlin's solution for a point load in a half space seems both feasible and desirable.

(6) The present MBEM programme has demonstrated the utility of a combined boundary and displacement discontinuity element programme. A computer package allowing for complete mixing of displacement discontinuity and boundary elements with a number of subregions seems to be a logical extension to the current "state of the art" programmes available to the rock mechanics practitioner.

The boundary element method of stress analysis has already been used widely in various rock mechanics aspects of mine design. The formulations described above were written specially for execution on a small mini-computer. This renders them more accessible to the small group user. Limited storage requirements necesitate judicious use of elements and careful choice of simplifying assumptions about excavation geomtry and material properties. However, since the type of problem described here is usually so complex as to defy very accurate modelling even using very large computers, a more rewarding approach is to simplify the problem with the aim of evaluating general effects and calculating stresses and displacements which can reasonably be related to the actual problem and used for the purposes of comparison.

The formulations of this dissertation have already been tested and used successfully on a number of practical mining problems and as such it is felt that they constitute useful additions to the tool box of the rock mechanics engineer.

REFERENCES

BANNERJEE, P.K. and BUTTERFIELD, R. (1977). Boundary element method in Geomechanics in Finite Elements in Geomechanics, ed. G. Gudekus, Chichester, Wiley.

CROUCH, S.L. (1976). Analysis of stresses and displacements around underground excavations. An application of the displacement discontinuity method, University of Minnesota Geomechanics Report.

CRUSE, T.A. (1969). Numerical solutions in three-dimensional elastostatics. Int. Jnl. Solids and Structures, Vol 5 pp 1259-1274.

CRUSE, T.A. (1973). An improved boundary integral equation method for three-dimensional stress analysis. Computers and Structures, Vol 3 pp 509-527.

DEIST, F.H. and GEORGIADIS, E (1976). A Computer System for three-dimensional elastic analysis using a boundary element approach. Chamber of Mines of South Africa, Research Report No 43/76, Project No GS 1510.

DIERING, J.A.C. (1977). Analysis of Single Seam Extraction. An application of the displacement discontinuity method B.Sc. Hons. Dissertation Submitted to the University of the Witwatersrand.

DIERING, J.A.C. (1978). Report on implementation of 2-D stress analysis programme MINAP and development of software for 3-D stress analysis programme SPUD on the mini-computer. Report on RP 36, Rock Burst Research Project, Western Deep Levels Gold Mining Company, South Africa.

REFERENCES

DIERING, J.A.C. (1980a). Simulation of mining in non-homogeneous ground using the displacement discontinuity method, Jnl. S. Afr. Inst. Min. Metall. Vol 80, No 7 pp 169-173.

DIERING, J.A.C. (1980b). An improved method for the determination, by a MINSIM type of analysis, of stresses and displacements around tabular excavations Jnl. S. Afr. Inst. Min. Metall. Vol 80, No 12 pp 225-228.

FRÖBERG, C. (1970). An Introduction to Numerical Analysis. Addison-Wesley.

GAY, N.C. (1975). In situ stress measurements in Southern Africa. Technophysics, 29 pp 447-459.

HAMMER, P.C., MARLOWE, O.P. and STROUD, A.H. (1956). Numerical Integration over Simplexes and Cones, Math Tables Aids Comp. Vol 10, pp 130-137.

HOCKING, G. (1977). Development and application of boundary integral and rigid block methods for Geotechnics. Ph.D. Thesis. University of London, U.K.

LACHAT, J.C. (1975). A further development of the Boundary Integral technique for elastostatics. Ph.D. Thesis, University of Southhampton, U.K.

LACHAT, J.C. and WATSON, J.O. (1976). Effective numerical treatment of boundary integral equations: A formulation for three-dimensional elastostatics. Int. Jnl. for Num. Meth. in Eng. Volume 10, pp 991-1005.

MORRIS, J.P.E. (1975). SHAMIP: A Computer programme for Elastic Analysis of large size, tabular, mono-planar shallow, hard ore-bodies, Johannesburg, Chamber of Mines of South Africa, Research Report No 51/75, 221 pp.

REFERENCES

REID, J.K. (1970). On the Method of conjugate gradients for the solution of large spare systems of linear equations in Solution of Large Sparse Sets of Linear Equations.

SALAMON, M.D.G. (1963). Elastic analysis of displacements and stresses induced by the mining of seam or reef deposits, Part I Jnl, of S. Afr. Inst. Min and Metall, Vol 64 (4) pp 129-149.

SALAMON, M.D.G. (1964a). Elastic analysis of displacements and stresses induced by the mining of seam or reef deposits, Part II, Jnl. S. Afr. Inst. Min. and Metall, Vol 64 (6) pp 197-218.

SALAMON, M.D.G. (1964b). Elastic analysis of displacements and stresses induced by the mining of seam or reef deposits, Part III, Jnl. S. Afr. Inst. Min and Metall, Vol 64 (10) pp 468-500.

SALAMON, M.D.G. (1964c). Analysis of displacements and stresses induced by the mining of seam of reef deposits, Part IV, Jnl S. Afr. Inst. Min. Metall. Vol 65 (5) pp 319-338.

STARFIELD, A.M. and CROUCH S.L. (1973). Elastic analysis of single seam extraction in New Horizons in Rock Mechanics. (Ed H.R. Hardy, JR. and R. Stefhanko) ASCE, N.Y., pp 421-439.

TUFF, D.A. (1981). Partial elimination technique for structural equations. Proceedings of the symposium on design applications of the finite element method, Vol 3, University of Cape Town, S. Africa.

REFERENCES

ZIENKIEWICZ, O.C. (1971). The Finite Element Method in Engineering Science, N.Y. McGraw-Hill.

ZIENKIEWICZ, O.C. (1977). The Finite Element Method. London, McGraw-Hill.

# A P P E N D I X 1

## EQUIVALENCE OF DISPLACEMENT DISCONTINUITY AND BOUNDARY ELEMENT STRESS AND DISPLACEMENT FUNCTIONS

## APPENDIX 1: EQUIVALENCE OF DISPLACEMENT DISCONTINUITY AND BOUNDARY ELEMENT STRESS AND DISPLACEMENT FUNCTIONS

The stresses and displacements induced at interior points by the displacement discontinuity elements are given (Starfield and Crouch, 1973) in terms of three harmonic functions $\phi_1$, $\phi_2$, and $\phi_3$. It is shown here that these functions are equivalent to the Kelvin solution used for the boundary elements. (This equivalence made combining the two element types much easier from a programming point of view since only Kelvin type solutions were used. The solution in terms of $\phi_1$, $\phi_2$, and $\phi_3$ is not as easily programmed as the Kelvin type solution).

The displacment component (in the a -direction of the co-ordinate system chosen for the displacement discontinuity elements) is (Starfield and Crouch, 1973).

$$u_3(p) = 2(1-\nu)\,\phi_{3,3} + (1-2\nu)(\phi_{1,1} + \phi_{2,2}) - z\,\phi_{,3} \qquad \text{(A1)}$$

where: $\phi = \phi_{i,i}$

$\phi_i = C_i/r$

$$C_i = \frac{a\, d_i}{8\pi(1-\nu)}$$

$a$ = Area of displacement discontinuity element

= $4h^2$ (h is the element half width)

$\phi_{,3} = \phi_{i,i3}$

$r$ = Distance from element to interior point p

$d_i$ = Displacement discontinuity component in the i direction

From (A1)

$$u_3(p) = 2(1-\nu)\frac{-c_3 z}{r^3} - (1-2\nu)\left(\frac{-c_1 x}{r^3} - \frac{c_2 y}{r^3}\right) \qquad \text{(A2)}$$

$$- z\left(\frac{-c_3}{r^3} - \frac{3xzc_1}{r^5} - \frac{3yzc_2}{r^5} - \frac{3z^2c_3}{r^5}\right)$$

Or in tensor notation

$$u_3(p) = \frac{\ell^2}{2\pi(1-\nu)}\Big[-2(1-\nu)r_{,3}\,d_3 + (1-2\nu)(r_{,1}\,d_1 + r_{,2}\,d_2)$$

$$+ r_{,3}\left(d_3 - 3r_{,1}\,r_{,3}\,d_1 - 3r_{,2}\,r_{,3}\,d_2 - 3r_{,3}\,r_{,3}\,d_3\right)\Big]$$

$$= \frac{\ell^2}{2(1-\nu)\pi r^2}\left[-(2(1-\nu)-1)r_{,3}\,d_3 - (1-2\nu)(r_{,1}\,d_1 + r_{,2}\,d_2) - 3r_{,3}^2\,r_{,i}\,d_i\right]$$

$$= \frac{-\ell^2}{2(1-\nu)\pi r^2}\left[(1-2\nu)r_{,i}\,d_i - 3r_{,3}^2\,r_{,i}\,d_i\right] \qquad \text{(A3)}$$

The equation for $u_3(p)$ is found (for a single element) from (3) as

$$u_3(p) = -\Delta T_{3j}(p,n)\,u_j(n) + \Delta u_{3j}(p,n)\,t_j(n) \qquad \text{(A4)}$$

In order to compare (A4) and (A3) the displacements $u_j$ in (A4) were replaced by $d_j$ in accordance with the definition of a displacement discontinuity (11). The second term in (A4) is not present in a displacement discontinuity element since the $t_j$ components have opposite signs on the two displacement discontinuity surfaces and cancel one another. Also in comparing (A4) and (A3) the integration implicit in the T term is evaluated using a one point integral formula so that (A4) may be rewritten as:

$$u_3(p) = -a\, T_{3j}(p,n)\, d_j(n)$$

$$= \frac{-a(1-2\nu)}{8\pi r^2(1-\nu)}\left[\frac{\partial r}{\partial n}\left(\delta_{3j} + \frac{3}{1-2\nu}\, r_{,3}\, r_{,j}\right) - n_j\, r_{,3} + n_3\, r_{,j}\right] d_j(n) \qquad \text{(A5)}$$

Now the choice of co-ordinate system is such that

$n_i = (n_1, n_2, n_3) = (0, 0, 1)$

A5 therefore reduces to

$$u_3(p) = \frac{-a(1-2\nu)}{8\pi r^2(1-\nu)}\left[r_{,3}\left(\delta_{3j} + \frac{3}{1-2\nu}\, r_{,3}\, r_{,j}\right) - \delta_{3j}\, r_{,3} + r_{,j}\right] d_j(n)$$

$$= \frac{-a(1-2\nu)}{8\pi r^2(1-\nu)}\left[r_{,j}\, d_j(n) + \frac{3}{1-2\nu}\, r_{,3}^{\,2}\, r_{,j}\, d_j(n)\right] \qquad \text{(A6)}$$

This is the same as (A3)

The equivalence of the displacements $u_1$ and $u_2$ and stresses may be derived in similar fashion to that above.

# APPENDIX 2

## PROGRAMME LISTING FOR PROGRAMME BEM

```
*******************************************************************************************************************
C
C     --------------------------------------------------------------------------------
C     SOURCE LISTING FOR PROGRAM BEM
C
C     PROGRAM WRITTEN BY J. A. C. DIERING      1980/81
C
C     COPYRIGHT RESIDES WITH STEFFEN ROBERTSON AND KIRSTEN INC.
C
C     SUBPROGRAMS AND SUBROUTINES ARE GIVEN IN ALPHABETIC ORDER
C     EACH IS SEPARATED BY A LINE OF STARS (*****)
C
C     --------------------------------------------------------------------------------
*******************************************************************************************************************

HEADR
BEMPRINT IND
ARCAP
BELL
BEM
BEM1
BEM4
BEM6
BEMDOT
BEMINF
BEMLIN
BEMSHAPE
COMMON
DCOSF
DCOSL
DCOSP
DCOSV
DIAGCO
DISTZ
DUMPO
ERROR
GENGEN
IBLOAD
INITT
INPUT1
INTEGB
LDBLK
LESET
LGEOM
LOAD
LOAD1
LODNOD
LODVEC
LSET
LUMSUM
MODGEN
MODSET
NODGEN
NULL
NULL2
NULL4
NULLI
```

```
FRINK
SETFILE
SOLIN
SUMELT
SUMSOL
SUMUP
SYMFX
TOTOL
TTOU
TUL
TVAL
UDT
UVAL
XLEN
XLEND
XLENT

****************************************************************************************************

      FUNCTION AREAF(I,J,K,X,N)
C
C     FUNCTION TO CALCULATE THE AREA OF AN ELEMENT.
C
      DIMENSION X(3,N)
      A=SQRT(DIST2(X(1,I),X(1,J)))
      B=SQRT(DIST2(X(1,J),X(1,K)))
      C=SQRT(DIST2(X(1,K),X(1,I)))
      S=0.5*(A+B+C)
      AREAF=SQRT(S*(S-A)*(S-B)*(S-C))
      RETURN
      END

****************************************************************************************************

C
C     SUBROUTINE TO RING-A-DING.
C
      SUBROUTINE BELL
      TYPE ""
      RETURN
      END

****************************************************************************************************

C     JACD                                                              *
      PARAMETER JQ=13,JU=89,JY=700
      COMMON X(3,JY),ALUMP(JU),XLUMP(3,JU),DLUMC(3,JU),LKOD(JU),LMOD(JU),
     *  IFACE(JU),TLUMP(3,JU),ULUMP(3,JU),
     2  NDIR(3,8),NMULT(3,8),XSYM,YSYM,ZSYM,ISYM,NSYM(5),PRIMO(3,3),PRIM1(3,3),
     3  EMOD(5),PR(5),E,G,V,V1,V2,V3,V4,V5,V6,V7,V8,V9,CK,CD,CONS,IBC(3),
     4  TRANS,TRAN7,TRAN21,TRAN42,TRANL,TRANLL,WGHT(11),COORD(3,11),
     5  NODE,NITER,NLUMP,ND1MEL,ND1ML,ND,ISTAR,IBUF,LIM,NSLIP,NUMOR,NMOD,
     6  ONE,NIL,TWO,THIRD,HALF,ZERO,I1,I2,I3,PI,ALT(3,3,5),DELTA(3,3),OMEGA,
     7  TT(9),UU(9),IREAD,IRITE,A,B,C,R,RSQ,ICHAN,ISSW(20,,XCON(3),
     8  DEL,ARMIN,ARMAX,CURV,RATT,RATB,ARRAT,ISWAP,R1,R2,R3,R4,K1,K2,K3
C
      COMMON /EM/ INDEX(4,JQ),DCEL(3,JQ),XCEL(3,JQ),T(3,JQ),U(3,JQ),AREA(JQ)
     E  ,IELT(JQ),KOD(JQ),LIND(JU),NUMEL,COH,PHI,THIK
C
```

```
      COMMON /RE/ INDEXR(4,JQ),DCELR(3,JQ),XCELR(3,JQ),TR(3,JQ),UR(3,JQ)
     R  ,AREAR(JQ),IELTR(JQ),KODR(JQ),LINDR(JU),NUMELR,CONR,PHIR,THIKR
C
C      PROGRAM BEM (MAIN CONTROL PROGRAM)
C
C
      CALL JOBTIME("BEM   ",6)
      CALL KRONO(2,IRITE,TON,TINC)
      CALL INITI
      CALL SWAP("BEM1.PR",IER)
      IF(IER.NE.1)WRITE(IRITE) " BEM1 SWAP ER",IER
      IF(ISWAP.NE.1)STOP AFTER BEM1
      IF(ISSW(1).NE.1)CALL SWAP("BEMINF.PR",IER)
      IF(ISWAP.NE.1)STOP AFTER BEMINF
      IF(IER.NE.1)WRITE(IRITE) "BEMINF ER",IER
      IF(ISSW(2).NE.1)CALL UD1
      IF(ISSW(2).EQ.NIL.OR.ISSW(2).EQ.2)CALL SWAP("SOLIN.PR",IER)
      IF(ISWAP.NE.1)STOP AFTER SOLIN
      IF(IER.NE.1)WRITE(IRITE) "SOLIN ER",IER
      IF(ISSW(6).EQ.NIL)CALL SWAP("BEM4.PR",IER)
      IF(IER.NE.1)WRITE(IRITE) " BEM4 SWAP ER",IER
      IF(ISWAP.NE.1)STOP AFTER BEM4
      IF(NUMOR.LT.1)GO TO 1
      CALL SWAP("BEMLIN.PR",IER)
      IF(IER.NE.1)WRITE(IRITE) "BEMLIN SWAP ER",IER
      IF(ISWAP.NE.1)STOP AFTER BEMLIN
      CALL SWAP("BEM6.PR",IER)
      IF(IER.NE.1)WRITE(IRITE) "BEM6 ER",IER
      IF(ISWAP.NE.1)STOP AFTER BEM6
    1 CONTINUE
C     CALL JOBTIME("BEM   ",6)
      CALL KRONO(5,IRITE,TON,TINC)
      STOP
      END

*****************************************************************************************************************

C     JACD
      PARAMETER JQ=13,JU=89,JY=700
      COMMON X(3,JY),ALUMP(JU),XLUMP(3,JU),DLUMC(3,JU),LKOD(JU),LMOD(JU),
     *  IFACE(JU),TLUMP(3,JU),ULUMP(3,JU),
     2  NDIR(3,8),NMULT(3,3),XSYM,YSYM,ZSYM,ISYM,NSYM(5),PRIM0(3,3),PRIM1(3,3),
     3  EMOD(5),PR(5),E,G,V,V1,V2,V3,V4,V5,V6,V7,V8,V9,CK,CD,CONS,IBC(3),
     4  TRANS,TRAN7,TRAN21,TRAN42,TRANL,TRANLL,WGHT(11),COORD(3,11),
     5  NODE,NITER,NLUMP,ND1MEL,ND1ML,ND,ISTAR,IBUF,LIM,NSLIP,NUMOR,NMOD,
     6  ONE,NIL,TWO,THIRD,HALF,ZERO,I1,I2,I3,P1,ALT(3,3,3),DELTA(3,3),OMEGA,
     7  TT(9),UU(9),IREAD,IRITE,A,B,C,R,RSQ,ICHAN,ISSW(20),XCOM(3),
     8  DEL,ARMIN,ARMAX,CURV,RATT,RATB,ARRAT,ISWAP,R1,R2,R3,R4,K1,K2,K3
C
      COMMON /EM/ INDEX(4,JQ),DCEL(3,JQ),XCEL(3,JQ),T(3,JQ),U(3,JQ),AREA(JQ)
     E  ,IELT(JQ),KOD(JQ),LIND(JU),NUMEL,CON,PHI,THIK
C
      COMMON /RE/ INDEXR(4,JQ),DCELR(3,JQ),XCELR(3,JQ),TR(3,JQ),UR(3,JQ)
     R  ,AREAR(JQ),IELTR(JQ),KODR(JQ),LINDR(JU),NUMELR,CONR,PHIR,THIKR
C
C      BEM1 (INPUT SECTION)
C
C
      ISWAP=33
      CALL INPUT
      CALL LGEOM
```

```
      ISWAP=1
      CALL BACK
      STOP
      END
*************************************************************************************************************************
C     JACD                                                                              *
      PARAMETER JQ=13,JU=89,JY=700
      COMMON X(3,JY),ALUMP(JU),XLUMP(3,JU),DLUMC(3,JU),LKOD(JU),LMOD(JU),
     * IFACE(JU),TLUMP(3,JU),ULUMP(3,JU),
     2 NDIR(8,8),NMULT(3,8),XSYM,YSYM,ZSYM,ISYM,NSYM(5),PRIMO(3,3),PRIM1(3,3),
     3 EMOD(5),PR(5),E,G,V,V1,V2,V3,V4,V5,V6,V7,V8,V9,CK,CD,CONS,IBC(3),
     4 TRAN5,TRAN7,TRAN21,TRAN42,TRANL,TRANLL,WGHT(11),COORD(3,11),
     5 NODE,NITER,NLUMP,NDIMEL,NDIML,ND,ISTAR,IBUF,LIM,NSLIP,NUMOR,NMOD,
     6 ONE,NIL,TWO,THIRD,HALF,ZERO,I1,I2,I3,PI,ALT(3,3,3),DELTA(3,3),OMEGA,
     7 TT(9),UU(9),IREAD,IRITE,A,B,C,R,RSQ,ICHAN,ISSW(20),XCOM(3),
     8 DEL,ARMIN,ARMAX,CURV,RATI,RATB,ARRAT,ISWAP,R1,R2,R3,R4,K1,K2,K3
C
      COMMON /EN/ INDEX(4,JQ),DCEL(3,JQ),XCEL(3,JQ),T(3,JQ),U(3,JQ),AREA(JQ)
     E ,IELT(JQ),KOD(JQ),LIND(JU),NUMEL,COH,PHI,THIK
C
      COMMON /RE/ INDEXR(4,JQ),DCELR(3,JQ),XCELR(3,JQ),TR(3,JQ),UR(3,JQ)
     R ,AREAR(JQ),IELTR(JQ),KODR(JQ),LINDR(JU),NUMELR,COHR,PHIR,THIKR
C
C     BEM4  --- SUBPROGRAM TO OUTPUT THE RESULTS.
C
      ISWAP=33
      CALL OPEN(2,"LUMSTOR.DE",1,IER)
      IF(IER.NE.1)TYPE "LUMSTOR OP ER",IER
      WRITE(IRITE,101)
      DO 1 NL=1,NLUMP
      CALL LDBLK(NL,1,1)
      WRITE(IRITE,102)NL
      DO 1 N=1,NUMEL
      KODE=KOD(N)
      CALL IBLOAD(KODE,IBC)
    1 WRITE(IRITE,100)N,IELT(N),(XCEL(I,N),I=1,3),(U(I,N),I=1,3),(T(I,N),I=1,3),
     1KOD(N),AREA(N),IFACE(N),(IBC(K),K=1,3)
  100 FORMAT(I3,I5,F7.2,2(F9.2,1X),3(F9.5,1X),3(F9.4,1X),I6,4X,F9.2,I6,
     *3X,3I2,1X)
  101 FORMAT(" ELEMENT X-COORD"T20"Y-COORD"T30"Z-COORD"T43"U-X"T53"U-Y"T63
     1"U-Z"T73"T-X"T83"T-Y"T93"T-Z"T99"KODE"T107"AREA"T114"IFACE   IBC",
     *//)
  102 FORMAT(/" LUMP"I3,/)
      ISWAP=1
      CALL BACK
      STOP
      END
*************************************************************************************************************************
C     JACD                                                                              *
      PARAMETER JQ=13,JU=89,JY=700
      COMMON X(3,JY),ALUMP(JU),XLUMP(3,JU),DLUMC(3,JU),LKOD(JU),LMOD(JU),
     * IFACE(JU),TLUMP(3,JU),ULUMP(3,JU),
     2 NDIR(8,8),NMULT(3,8),XSYM,YSYM,ZSYM,ISYM,NSYM(5),PRIMO(3,3),PRIM1(3,3),
     3 EMOD(5),PR(5),E,G,V,V1,V2,V3,V4,V5,V6,V7,V8,V9,CK,CD,CONS,IBC(3),
     4 TRAN5,TRAN7,TRAN21,TRAN42,TRANL,TRANLL,WGHT(11),COORD(3,11),
     5 NODE,NITER,NLUMP,NDIMEL,NDIML,ND,ISTAR,IBUF,LIM,NSLIP,NUMOR,NMOD,
```

```
     6  ONE,NIL,TWO,THIRD,HALF,ZERO,I1,I2,I3,PI,ALT(3,3,3),DELTA(3,3),OMEGA,
     7  TT(9),UU(9),IREAD,IRITE,A,B,C,R,RSQ,ICHAN,ISSW(20),XCOM(3),
     8  DEL,ARMIN,ARMAX,CURV,RATT,RATB,ARRAT,ISWAP,R1,R2,R3,R4,K1,K2,K3
C
      COMMON /EM/ INDEX(4,JQ),DCEL(3,JQ),XCEL(3,JQ),T(3,JQ),U(3,JQ),AREA(JQ)
     E  ,IELT(JQ),KOD(JQ),LIND(JQ),NUMEL,CON,PHI,THIK
C
      COMMON /RE/ INDEXR(4,JQ),DCELR(3,JQ),XCELR(3,JQ),TR(3,JQ),UR(3,JQ)
     R  ,AREAR(JQ),IELTR(JQ),KODR(JQ),LINDR(JQ),NUMELR,CONR,PHIR,THIKR
C
      COMMON /CO/ AO,BO,W
C
C     BEM6 --- BENCHMARKS (INTERIOR POINTS FOR STRESSES AND DISPL CALCS)
C
      DIMENSION XX(3),XEL(3),XC(3),DIS(3),TRAC(3),DCO(3),DISP(3)
      DIMENSION STRES(3,3),UN(3,JY)
      EQUIVALENCE (STRES,TT)
      WRITE(IRITE,100)
  100 FORMAT(/"   OUTPUT FOR FIELD POINTS",/"    **********************",/
     1/" -VE STRESSES ARE COMPRESSIVE",/
     1" SIG-1 IS THE MAX PRINC STRESS",/
     1" SIG-2 IS THE INT PRINC STRESS",/
     1" SIG-3 IS THE MIN PRINC STRESS",/)

C
C     CONSTANTS.
C
      ISWAP=30
C
      CALL OPEN(4,"CONSTOR.BE",1,IER)
      IF(IER.NE.1)TYPE "CONSTOR OP ER",IER
      CALL OPEN(3,"BENSTORE",3,IER)
      IF(IER.NE.1)TYPE "BENSTORE OP ER",IER
      CALL OPEN(6,"BENCH ",1,IER)
      IF(IER.NE.1)TYPE "OPEN ER",IER
      CALL OPEN(5,"BEMLIN.BE",1,IER)
      IF(IER.NE.1)TYPE "BEMLIN.BE OP ER",IER
      CALL OPEN(2,"LUMSTOR.BE",1,IER)
      IF(IER.NE.1)TYPE "LUMSTOR OP ER",IER
      READ BINARY (5) UN
      READ(4)TRANL,TRAN42,TRAN21,TRAN7,TRAN5,AO,BO,W
      WRITE(IRITE,303)TRAN42,TRAN21,TRAN7,TRAN5,TRANL,W,AO,BO
  303 FORMAT(/" TRANSITION VALUE FOR 42 POINT INTEGRATION =",F10.2,/
     1       "                          21               =",F10.2,/
     1       "                           7               =",F10.2,/
     1       "                           3               =",F10.2,/
     1      ," TRANSITION VALUE FOR    LUMPING           =",F10.2,/
     1       " W ( 0 FOR CONSTANT 1 FOR LINEAR           =",F10.2,/
     1       " (A,B) FOR SHAPE FUNCTION                  =",2F10.2,/)
      DO 1 KK=1,NUMOR
      READ BINARY (6) NBEN,JBEN,XX,MODUL
      TYPE XX,MODUL
      CALL MODBEN(MODUL)
      NS=NSYM(MODUL)
      CALL NULL(DISP,3)
      CALL NULL(STRES,9)
      DO 66 NL=1,NLUMP
      KODL=LKOD(NL)
      IF(KODL.EQ.NIL)GO TO 66
C
```

```
C     DECIDE ABOUT LUMPING
C
      DO 66 ISYM=1,NS
C
C     LOAD SYMMETRY CONSTANTS.
C
      IF(LMOD(NL).NE.MODUL)GO TO 66
      IDIR=NDIR(ISYM,NS)
      IF(IDIR.EQ.NIL)GO TO 66
      CALL LESET(1,XC,XLUMP(1,NL),0,DCO,DLUMC(1,NL))
      AR=ALUMP(NL)
      RSQ=DIST2(XX,XC)
      RATIO=RSQ/AR
      IF(RATIO.LT.TRANL)GO TO 60
      CALL LESET(0,TRAC,TLUMP(1,NL),0,DIS,ULUMP(1,NL))
      CALL INTEGB(I1,I1,I2,I3,TRAC,DIS,AR,I1,STRES,DISP,XX,XC,DCO,UN)
      GO TO 66
   60 CONTINUE
      CALL LDBLK(NL,1,.)
      DO 8 N=1,NUNEL
      CALL LESET(0,DCO,DCEL(1,N),1,XC,XCEL(1,N))
      AR=AREA(N)
      RSQ=DIST2(XC,XX)
      RATIO=RSQ/AR
      CALL LESET(0,TRAC,T(1,N),0,DIS,U(1,N))
      IF(RATIO.LT.TRANS)GO TO 9
      CALL INTEGB(I1,I1,I2,I3,TRAC,DIS,AR,I1,STRES,DISP,XX,XC,DCO,UN)
      GO TO 8
    9 CALL LODNOD(NOD1,NOD2,NOD3,NOD4,INDEX(1,N))
      IN=NOD1
      JN=NOD2
      KN=NODE+1
      CALL LSET(X(1,KN),XCEL(1,N))
      CALL LSET(UN(1,KN),U(1,N))
      AR=AREA(N)
      NQ=3
      IF(NOD4.NE.NIL)NQ=4
      DO 6 NSID=1,NQ
      GO TO (361,362,363,364),NSID
  361 IN=NOD1
      JN=NOD2
      GO TO 365
  362 IN=NOD2
      JN=NOD3
      GO TO 365
  363 IN=NOD3
      JN=NOD4
      IF(NQ.EQ.3)JN=NOD1
      GO TO 365
  364 IN=NOD4
      JN=NOD1
  365 CALL INTEGB(N,IN,JN,KN,TRAC,DIS,AR,NIL,STRES,DISP,XX,XC,DCO,UN)
    6 CONTINUE
    8 CONTINUE
   66 CONTINUE
      WRITE(IRITE,105)
  105 FORMAT(/" IDENTITY",T14"X"T22"Y"T30"Z"T40"U"T48"V"T56"W"T70"OXX"
     1T80"OXY"T88"OXZ"T96"OYY"T104"OYZ"T112"OZZ")
      WRITE(IRITE,101)NBEN,JBEN,XX,DISP,((STRES(I,J),J=I,3),I=1,3)
      WRITE BINARY (3) NBEN,JBEN,XX,DISP,((STRES(I,J),J=I,3),I=1,3)
```

```
  101 FORMAT(1X,2I4,2X,3(F7.3,1X),2X,3(F7.4,1X),T69,6(F7.3,1X))
C
C     ADD PRIM STRESSES.
C
      FAC=XX(3)
      DO 30 I=1,3
      DO 30 J=1,3
   30 STRES(I,J)=STRES(I,J)+FAC*PRIM1(I,J)+PRIMO(I,J)
      WRITE(IRITE,103)((STRES(I,J),J=1,3),I=1,3)
      WRITE BINARY (2)((STRES(I,J),J=1,3),I=1,3)
  103 FORMAT(30X"TOTAL STRESSES ARE:"T69,6(F7.3,1X))
      IF(ISSW(7).EQ.NIL)CALL PRINK(STRES,IRITE)
    1 CONTINUE
      ISWAP=1
      CALL BACK
      STOP
      END
*****************************************************************************************************************************************
C
C     DOT PRODUCT
C
      FUNCTION DOT(X,Y,N)
      DIMENSION X(1),Y(1)
      DOT=0
      DO 1 I=1,N
    1 DOT=DOT+X(I)*Y(I)
      RETURN
      END
*****************************************************************************************************************************************
C     JACD                                                                         *
      PARAMETER JQ=13,JU=89,JY=700
      COMMON X(3,JY),ALUMP(JU),XLUMP(3,JU),DLUMC(3,JU),LKOD(JU),LMOD(JU),
     *  IFACE(JU),TLUMP(3,JU),ULUMP(3,JU),
     2  NDIR(8,8),NMULT(3,8),XSYM,YSYM,ZSYM,ISYM,NSYM(5),PRIMO(3,3),PRIM1(3,3),
     3  EMOD(5),PR(5),E,G,V,V1,V2,V3,V4,V5,V6,V7,V8,V9,CK,CD,CONS,IBC(3),
     4  TRANS,TRAN7,TRAN21,TRAN42,TRANL,TRANLL,WGHT(11),COORD(3,11),
     5  NODE,NITER,NLUMP,ND1MEL,ND1ML,ND,ISTAR,IBUF,LIM,NSLIP,NUMOR,NMOD,
     6  ONE,NIL,TWO,THIRD,HALF,ZERO,I1,I2,I3,PI,ALT(3,3,3),DELTA(3,3),OMEGA,
     7  TT(9),UU(9),IREAD,IRITE,A,B,C,R,RSQ,ICHAN,ISSW(2 ),XCOM(3),
     8  DEL,ARMIN,ARMAX,CURV,RATT,RATB,ARRAT,ISWAP,R1,R2,R3,R4,K1,K2,K3
C
      COMMON /EM/ INDEX(4,JQ),DCEL(3,JQ),XCEL(3,JQ),T(3,JQ),U(3,JQ),AREA(JQ)
     E  ,IELT(JQ),KOD(JQ),LIND(JU),NUMEL,COH,PHI,THIK
C
      COMMON /RE/ INDEXR(4,JQ),DCELR(3,JQ),XCELR(3,JQ),TR(3,JQ),UR(3,JQ)
     R  ,AREAR(JQ),IELTR(JQ),KODR(JQ),LINDR(JU),NUMELR,COHR,PHIR,THIKR
C
      COMMON /BU/ BUF(1152)
C
C     DEMINF --- SUBPROGRAM TO CALCULATE INFL COEFS ON BOUNDARY ELEMENTS
C
      DIMENSION XC(3),XEL(3),XRL(3),XE(3),XR(3),TEMP(3),DLC(3),
     1  TEMPLL(3),DCOSR(3)
      ISWAP=33
      ISTAR=ICHAN
      IBUF=NIL
```

```
      CALL KRONO(2,IRITE,TON,TINC)
      WRITE(IRITE,550)
      CALL SETFILE
C
C     START LUMP-LUMP LOOPS
C
      DO 511 MOD1=1,NMOD
      CALL MODSET(MOD1)
      NS=NSYM(MOD1)
      DO 511 ML=1,NLUMP
      LKODR=LKOD(ML)
      IF(LKODR.LE.NIL)GO TO 511
      IF(LMOD(ML).NE.MOD1)GO TO 511
      CALL LDBLK(ML,1,2)
C     TYPE "MLUMP=",ML,NUMELR
      CALL LSET(XRL,XLUMP(1,ML))
      CALL NULL(TEMPLL,3)
      DO 19 NL=1,NLUMP
      LINDR(NL)=NIL
      LKODE=LKOD(NL)
      IF(LKODE.EQ.NIL)GO TO 19
      MOD2=LMOD(NL)
      IF(MOD2.NE.MOD1)GO TO 19
      ARL=ALUMP(NL)
      RSQ=DIST2(XRL,XLUMP(1,NL))
      RATIO=RSQ/ARL
      IF(RATIO.LT.TRANLL)GO TO 14
C     TYPE "L-L",ML,NL
      LINDR(NL)=1
      CALL NULL2(TT,UU,9)
      DO 18 ISYM=1,NS
      IDIR=NDIR(ISYM,NS)
      IF(IDIR.EQ.NIL)GO TO 18
      CALL LESET(1,XEL,XLUMP(1,NL),0,DLC,DLUMC(1,NL))
      RSQ=DIST2(XRL,XEL)
C
C     LUMP-LUMP COEFFICIENTS.
C
      R=SQRT(RSQ)
      WCK=CK*ARL/RSQ
      WCR=ARL*CONS/R
      CALL TTUU(XEL,XRL,WCK,WCR,DLC)
   18 CONTINUE
      CALL LOAD1(TEMPLL,LKODE,ULUMP(1,NL),TLUMP(1,NL))
      GO TO 19
   14 LINDR(NL)=3
      IF(RATIO.LT.TRANL.OR.LKODE.EQ.6)GO TO 19
      LINDR(NL)=2
C     TYPE "E-L",ML,NL
   19 CONTINUE
C
C     LUMP-ELT COEFFICIENTS.
C
      DO 1 M=1,NUMELR
      MELR=IELTR(M)
C     TYPE "ELT",MELR,M
      LKODR=KODR(M)
      CALL LSET(TEMP,TEMPLL)
      CALL LSET(XR,XCELR(1,M))
      CALL LODNOD(NOD1,NOD2,NOD3,NOD4,INDEXR(1,M))
```

```
      CALL DCOSV(NOD1,NOD2,NOD3,DCOSR,X,ND)
      DO 3 NL=1,NLUMP
      LND=LINDR(NL)
      IF(LND.NE.2)GO TO 7
      LKODE=LKOD(NL)
      ARL=ALUMP(NL)
      CALL NULL2(TT,UU,9)
      DO 9 ISYM=1,NS
      IDIR=NDIR(ISYM,NS)
      IF(IDIR.EQ.NIL)GO TO 9
      CALL LESET(1,XEL,XLUMP(1,NL),0,DLC,DLUMC(1,NL))
      RSQ=DIST2(XR,XEL)
      R=SQRT(RSQ)
      WCK=CK*ARL/RSQ
      WCR=ARL*CONS/R
      CALL TTUU(XEL,XR,WCK,WCR,DLC)
    9 CONTINUE
      CALL LOAD1(TEMP,LKODE,ULUMP(1,NL),TLUMP(1,NL))
      GO TO 3
    7 IF(LND.NE.3)GO TO 3
C
C     ELT-ELT COEFFICIENTS
C
      CALL LDBLK(NL,1,1)
C     TYPE "DET LUMP",NL
      DO 6 N=1,NUMEL
      MELE=IELT(N)
      KODE=KOD(N)
C
C     WORK OUT A BETTER SCHEME IN INPUT
C
      IF(LKODE.EQ.(-1))KODE=-1
      CALL NULL2(TT,UU,9)
      DO 111 ISYM=1,NS
      IDIR=NDIR(ISYM,NS)
      IF(IDIR.EQ.NIL)GO TO 111
      IF(M.NE.N)GO TO 30
      IF(ML.NE.NL)GO TO 30
      IF(ISYM.GT.1)GO TO 30
      CALL DIAGCO(M)
      GO TO 111
   30 CONTINUE
      CALL LODNOD(NOD1,NOD2,NOD3,NOD4,INDEX(1,N))
      AR=AREA(N)
      NQ=1
      CALL LESET(1,XC,XCEL(1,N),0,DLC,DCEL(1,N))
      RSQ=DIST2(XR,XC)
      RATIO=RSQ/AR
      NUMTRI=1
      NINTB=11
      NINTE=11
      MINT=1
      W1=AR
      IF(RATIO.GE.TRANS)GO TO 50
      IF(NOD4.EQ.NIL)GO TO 50
      NQ=2
      AR=AREAF(NOD1,NOD2,NOD3,X,ND)
   50 CONTINUE
      DO 11 NTQ=11,NQ
      IF(NTQ.EQ.1)GO TO 53
```

```
      AR=AREAF(NOD1,NOD3,NOD4,X,ND)
      NOD2=NOD4
      CALL XCENT(NOD1,NOD2,NOD3,XC,X,ND)
      CALL SYMFX(XSYM,YSYM,ZSYM,NMULT(1,ISYM),XC)
      NINTB=11
      NINTE=11
      NUMTRI=1
      MINT=1
   53 CONTINUE
      IN=NOD1
      JN=NOD2
      KN=NOD3
      W1=AR
C
      FUDGE=1
      RSQ=DIST2(XR,XC)
      RATIO=RSQ/AR
      RCL=RSQ/20.
C     IF(RATIO.LT.0.1)WRITE(IRITE,100)RATIO,MELR,MELE,NIL
  100 FORMAT(" CHECK ELTS TOO CLOSE RATIO,M,N,ER TYPE="F6.3,3I5)
      IF(RATIO.GE.TRANS)GO TO 20
      MINT=3
      NINTB=8
      NINTE=10
      IF(RATIO.GE.TRAN7)GO TO 20
      MINT=7
      NINTB=1
      NINTE=7
C
C     IF INTERIOR PROBLEM (EG CUBE) GO TO 25
C
C     IF(DOT(BLC,BCOSR,3).GT.(-HALF))GO TO 25
C     IF(RATIO.GT.(1.5))GO TO 25
C     FUDGE=1.5/RATIO
C  25 CONTINUE
      IF(RATIO.GT.TRAN21)GO TO 20
      MINT=21
      KN=NODE+1
      AR=AR/3.
      NUMTRI=3
      CALL XCENT(NOD1,NOD2,NOD3,XEL,X,ND)
      CALL LSET(X(1,KN),XEL)
      IF(RATIO.GT.TRAN42)GO TO 20
      MINT=42
      NUMTRI=6
      AR=AR/2.
      JN=NODE+2
   20 CONTINUE
C
C     SUBSECTION FOR NUMERICAL QUADRATURE
C
C     INFLUENCE OF ELEMENT N ON ELEMENT M (N.NE.M)
C
C     IF(MINT.GT.5)TYPE "ELT",IELT(N),MINT
      DO 11 NTRI=1,NUMTRI
      IF(NUMTRI.NE.6)GO TO 48
      IF(NTRI.EQ.1.OR.NTRI.EQ.6)IN=NOD1
      IF(NTRI.EQ.2.OR.NTRI.EQ.3)IN=NOD2
      IF(NTRI.EQ.4.OR.NTRI.EQ.5)IN=NOD3
      IF(NTRI.GT.3)GO TO 41
```

```
      DO 441 L=11,13
  441 X(L,JN)=(X(L,NOD1)+X(L,NOD2))*HALF
      GO TO 21
   41 IF(NTRI.GT.4)GO TO 42
      DO 442 L=11,13
  442 X(L,JN)=(X(L,NOD2)+X(L,NOD3))*HALF
      GO TO 21
   42 DO 443 L=11,13
  443 X(L,JN)=(X(L,NOD3)+X(L,NOD1))*HALF
      GO TO 21
   43 CONTINUE
      IF(NTRI.EQ.1)GO TO 21
      IF(NTRI.EQ.2)IN=NOD3
      IF(NTRI.LT.3)GO TO 21
      IN=NOD1
      JN=NOD3
   21 CONTINUE
      DO 11 NUM=NINTB,NINTE
      W1=WGHT(NUM)*AR*FUDGE
      IF(NINTB.EQ.11)GO TO 64
      C1=COORD(1,NUM)
      C2=COORD(2,NUM)
      C3=COORD(3,NUM)
      DO 24 L=11,13
   24 XC(L)=C1*X(L,IN)+C2*X(L,JN)+C3*X(L,KN)
      CALL SYMFX(XSYM,YSYM,ZSYM,NMULT(1,ISYM),XC)
      RSQ=DIST2(XR,XC)
      IF(RSQ.LT.RCL)WRITE(IRITE,100)RATIO,MELR,MELE,IONE
   64 R=SQRT(RSQ)
      WCK=W1*CK/RSQ
      WCR=W1*CONS/R
      CALL FTUU(XC,XR,WCK,WCR,DLC)
   11 CONTINUE
  111 CONTINUE
      CALL LOAD1(TEMP,KODE,U(1,N),T(1,N))
    6 CONTINUE
    3 CONTINUE
      WRITE BINARY (3) TEMP
    1 CONTINUE
      CALL LIBLK(NL,2,2)
      IBM=IBUF-2
      WRITE(IRITE,551)ISTAR,(BUF(I),I=IBM,IBUF),(LINDR(J),J=1,NLUMP)
  551 FORMAT(1X,I5,2X,3F8.5,2X,95I1)
  550 FORMAT(//)
  511 CONTINUE
      CALL DUMPO(IBUF,BUF,ISTAR,1)
      WRITE(IRITE,550)
      CALL KRONO(5,IRITE,TON,TINC)
      CALL RESET
      ISWAP=1
      CALL BACK
      STOP
      END
**********************************************************************************************************************************
C
C     BEMLIN --- AVERAGE NODAL DISPLACEMENTS FROM ELT VALUES (OPTIONAL)
C
C     JACD                                                   *
```

```
      PARAMETER JQ=13,JU=89,JY=700
      COMMON X(3,JY),ALUMP(JU),XLSMP(3,JU),DLUMC(3,JU),LKOD(JU),LMOD(JU),
     *  IFACE(JU),TLUMP(3,JU),ULUMP(3,JU),
     2  NDIR(8,8),NMULT(3,8),XSYM,YSYM,ZSYM,ISYM,NSYM(5),PRIMO(3,3),PRIM1(3,3),
     3  EMOD(5),PR(5),E,G,V,V1,V2,V3,V4,V5,V6,V7,V8,V9,CK,CD,CONS,IDC(3),
     4  TRAN5,TRAN7,TRAN21,TRAN42,TRANL,TRANLL,WGHT(11),COORD(3,11),
     5  NODE,NITER,NLUMP,NDIMEL,NDIML,ND,ISTAR,IBUF,LIM,NSLIP,NJMOR,NMOD,
     6  ONE,NIL,TWO,THIRD,HALF,ZERO,I1,I2,I3,PI,ALT(3,3,3),DELTA(3,3),OMEGA,
     7  TT(9),UU(9),IREAD,IRITE,A,B,C,R,RSQ,ICHAN,ISSW(20),XCOM(3),
     8  DEL,ARMIN,ARMAX,CURV,RATT,RATB,ARRAT,ISWAP,R1,R2,R3,R4,K1,K2,K3
C
      COMMON /EM/ INDEX(4,JQ),DCEL(3,JQ),XCEL(3,JQ),T(3,JQ),U(3,JQ),AREA(JQ)
     E  ,IELT(JQ),KOD(JQ),LIND(JU),NUMEL,COH,PHI,THIK
C
      COMMON /RE/ INDEXR(4,JQ),DCELR(3,JQ),XCELR(3,JQ),TR(3,JQ),UR(3,JQ)
     R  ,AREAR(JQ),IELTR(JQ),KODR(JQ),LINDR(JU),NUMELR,COHR,PHIR,THIKR
C
      DIMENSION UN(3,JY),SUM(JY)
      ISWAP=33
      TYPE "INTO BEMLIN"
      CALL OPEN(3,"BEMLIN.BE",3,IER)
      IF(IER.NE.1)TYPE "BEMLIN OP ER",IER
      CALL OPEN(2,"LUMSTOR.BE",1,IER)
      IF(IER.NE.1)TYPE "LUMSTOR OP ER",IER
      ND3=ND*3
      CALL NULL(UN,ND3)
      CALL NULL(SUM,ND)
      DO 1 NL=1,NLUMP
      LKODE=LKOD(NL)
      IF(LKODE.EQ.NIL)GO TO 1
      CALL LDBLK(NL,1,1)
      DO 2 N=1,NUMEL
      DO 2 J=1,4
      NJ=INDEX(J,N)
      IF(NJ.EQ.NIL)GO TO 2
      WGT=AREA(N)/DIST2(X(1,NJ),XCEL(1,N))
      DO 3 I=1,3
    3 UN(I,NJ)=UN(I,NJ)+U(I,N)*WGT
      SUM(NJ)=SUM(NJ)+WGT
    2 CONTINUE
    1 CONTINUE
      DO 4 NJ=1,ND
      VAL=SUM(NJ)
      IF(VAL.EQ.ZERO)GO TO 4
      DO 5 I=1,3
    5 UN(I,NJ)=UN(I,NJ)/VAL
    4 CONTINUE
C     CALL OPEN(15,"NODFIX.BE",1,IER)
C     IF(IER.NE.1)TYPE "NODFIX OP ER",IER
C     DO 20 N=1,NODE
C     READ(15,END=21)NOD,KODE
C     GO TO (20,22,23,24,25),KODE
C  22 UN(1,NOD)=ZERO
C     UN(2,NOD)=ZERO
C     UN(3,NOD)=ZERO
C     GO TO 20
C  23 UN(1,NOD)=ZERO
C     GO TO 20
C  24 UN(2,NOD)=ZERO
C     GO TO 20
```

```
C  25 UN(3,NOD)=ZERO
C  20 CONTINUE
C  21 CONTINUE
      WRITE BINARY (5) UN
      TYPE "EXIT BEMLIN"
      ISWAP=1
      CALL BACK
      STOP
      END
*****************************************************************************************************************
C
C     A=0 FOR FULL NODAL VALUES
C       1 FOR NO NODAL VALUES
C     B=0 FOR NO ELT VALUES
C       1 FOR FULL ELT VALUES
C     W=0 FOR ELT VALUES ONLY
C       1 FOR COMBINED VALUES
C
C
C     SHAPE FUNCTION FOR 'LAGRANGE TRIANGLE' (OPTIONAL)
C
      SUBROUTINE BEMSHAPE(X1,X2,X3,KQUAD,UN,U,N1,N2,N3,DIS,ND)
      COMMON /CO/ A,B,W
      DIMENSION UN(3,ND),U(1),DIS(3)
C     GO TO (1,2,3),KQUAD
C   1 RHO=X1
C     GO TO 4
C   2 RHO=4.*X1*X2
C     GO TO 4
C   3 RHO=27.*X1*X2*X3
      RHO=27.*X1*X2*X3
    4 RHOD=A+RHO*(B-A)
      RM1=(1.-RHOD)*W
      RHOW=RHOD*W+(1.-W)
      DO 5 L=1,3
    5 DIS(L)=RHOW*U(L)+RM1*(X1*UN(L,N1)+X2*UN(L,N2)+X3*UN(L,N3))
      RETURN
      END
*****************************************************************************************************************
C     JACD                                                        *
      PARAMETER JQ=13,JU=82,JY=700,JY3=2100
      COMMON X(3,JY),ALUMP(JU),XLUMP(3,JU),DLUMC(3,JU),LKOD(JU),LMOD(JU),
     *  IFACE(JU),TLUMP(3,JU),ULUMP(3,JU),
     2  NDIR(8,8),NMULT(3,8),XSYM,YSYM,ZSYM,ISYM,NSYM(5),PRIM0(3,3),PRIM1(3,3),
     3  EMOD(5),PR(5),E,G,V,V1,V2,V3,V4,V5,V6,V7,V8,V9,CK,CD,CONS,IBC(3),
     4  TRANS,TRAN7,TRAN21,TRAN42,TRANL,TRANLL,WGHT(11),COORD(3,11),
     5  NODE,NITER,NLUMP,NDIMEL,NDIML,ND,ISTAR,IBUF,LIM,NSLIP,NUMOR,NMOD,
     6  ONE,NIL,TWO,THIRD,HALF,ZERO,I1,I2,I3,PI,ALT(3,3,3),DELTA(3,3),OMEGA,
     7  TT(9),UU(9),IREAD,IRITE,A,B,C,R,RSQ,ICHAN,ISSW(20),XCOM(3),
     8  DEL,ARMIN,ARMAX,CURV,RATT,RATB,ARRAT,ISWAP,R1,R2,R3,R4,K1,K2,K3
C
      COMMON /EM/ INDEX(4,JQ),DCEL(3,JQ),XCEL(3,JQ),T(3,JQ),U(3,JQ),AREA(JQ)
     E  ,IELT(JQ),KOD(JQ),LIND(JU),NUMEL,CON,PHI,THIK
C
      COMMON /RE/ INDEXR(4,JQ),DCELR(3,JQ),XCELR(3,JQ),TR(3,JQ),UR(3,JQ)
     R  ,AREAR(JQ),IELTR(JQ),KODR(JQ),LINDR(JU),NUMELR,CONR,PHIR,THIKR
```

```
C
**************************************************************************************************************************
C
C     SUBROUTINE TO CALCULATE LOCAL COORD SYSTEM ABOUT DC(A,B,C).
C
      SUBROUTINE DCOSF(A,B,C,DC)
      DIMENSION DC(3,3)
      TOL=0.00001
      DC(3,1)=A
      DC(3,2)=B
      DC(3,3)=C
      AMO=ABS(A)
      IF(AMO.LT.TOL)AMO=TOL
      IF(A.LT.0.)AMO=-AMO
      DENOM=SQRT(AMO*AMO+B*B)
      TEMP=AMO/DENOM
      DC(2,1)=-B/DENOM
      DC(2,3)=0
      DC(2,2)=TEMP
C
C     DIP DC
C
      DC(1,1)=TEMP*C
      DC(1,2)=-DC(2,1)*C
      DC(1,3)=DC(2,1)*B-TEMP*A
      RETURN
      END
**************************************************************************************************************************
      SUBROUTINE DCOSL(I,J,A,B,C,X,ND)
C
C     SUBROUTINE TO FIND DIR COSINES OF A LINE.
C
      DIMENSION X(3,ND)
      R=SQRT(DIST2(X(1,I),X(1,J)))
      A=(X(1,J)-X(1,I))/R
      B=(X(2,J)-X(2,I))/R
      C=(X(3,J)-X(3,I))/R
      RETURN
      END
**************************************************************************************************************************
      SUBROUTINE DCOSP(I,J,K,A,B,C,X,ND)
      DIMENSION X(3,ND),DUM(3,3)
C
C    SUBROUTINE TO FIND DIRECTION COSINES OF PLANE GIVEN THE NODES
C
      DO 1 N=1,3
      DUM(2,N)=X(N,J)-X(N,I)
    1 DUM(3,N)=X(N,K)-X(N,I)
      A1=DUM(2,2)*DUM(3,3)-DUM(2,3)*DUM(3,2)
      A2=DUM(2,3)*DUM(3,1)-DUM(2,1)*DUM(3,3)
      A3=DUM(2,1)*DUM(3,2)-DUM(2,2)*DUM(3,1)
      R=SQRT(A1*A1+A2*A2+A3*A3)
      A=A1/R
      B=A2/R
```

```
      C=A3/R
      RETURN
      END
*************************************************************************************************************
      SUBROUTINE DCOSV(I,J,K,B,X,ND)
      DIMENSION X(3,ND),B(3),DUM(3,3)
C
C     SUBROUTINE TO FIND DIRECTION COSINE VECTOR OF PLANE GIVEN THE NODES
C
      DO 1 N=1,3
      DUM(2,N)=X(N,J)-X(N,I)
    1 DUM(3,N)=X(N,K)-X(N,I)
      A1=DUM(2,2)*DUM(3,3)-DUM(2,3)*DUM(3,2)
      A2=DUM(2,3)*DUM(3,1)-DUM(2,1)*DUM(3,3)
      A3=DUM(2,1)*DUM(3,2)-DUM(2,2)*DUM(3,1)
      R=SQRT(A1*A1+A2*A2+A3*A3)
      B(1)=A1/R
      B(2)=A2/R
      B(3)=A3/R
      RETURN
      END
*************************************************************************************************************
      PARAMETER JQ=13,JU=89,JY=700
      SUBROUTINE DIAGLO(M)
C     $ACD                                                                    *
      COMMON X(3,JY),ALUMP(JU),XLUMP(3,JU),DLUMC(3,JU),LROD(JU),LMOD(JU),
     *  IFACE(JU),TLUMP(3,JU),ULUMP(9,JU),
     2  NDIR(8,8),NMULT(3,3),XSYM,YSYM,ZSYM,ISYM,NSYM(5),PRIMO(3,3),PRIMI(3,3),
     3  LMOD(5),PR(5),E,G,V,V1,V2,V3,V4,V5,V6,V7,V8,V9,CK,CD,CONS,IRC(3),
     4  TRANS,TRAN7,TRAN21,TRAN42,TRANL,TRANLL,WGHT(11),COORD(3,11),
     5  NODE,NITER,NLUMP,NDIMEL,NDIML,ND,ISTAR,IHUF,LIM,NSLIP,NUMUR,NMOD,
     6  ONE,NIL,TWO,THIRD,HALF,ZERO,I1,I2,I3,PI,ALT(3,3,3),DELTA(3,3),OMEGA,
     7  TT(9),UU(9),IREAD,IRITE,A,B,C,R,RSQ,ICHAN,ISSW(20),XCOM(3),
     8  DEL,ARMIN,ARMAX,CURV,RATT,RATB,ARRAT,ISWAP,R1,R2,R3,R4,K1,K2,K3
C
      COMMON /EM/ INDEX(4,JQ),DCEL(3,JQ),XCEL(3,JQ),T(3,JQ),U(3,JQ),AREA(JQ)
     E  ,IELT(JQ),KOD(JQ),LIND(JU),NUMEL,COH,PHI,THIK
C
      COMMON /E/ E1(3),E2(3),I,J
C
C     SUBROUTINE TO CALCULATE DIAGONAL MATRIX VALUES (IE SELF EFFECTS)
C
      IVAL=0
      DO 3 I=1,3
      DO 3 J=1,3
      IVAL=IVAL+1
      TT(IVAL)=ZERO
      IF(I.EQ.J)TT(IVAL)=-HALF
    3 UU(IVAL)=ZERO
      NR=3
      CALL LODNOD(N1,N2,N3,N4,INDEX(1,M))
      IF(N4.NE.NIL)NR=4
      CALL LSET(XCOM,XCEL(1,M))
      A=DCEL(1,M)
      B=DCEL(2,M)
      C=DCEL(3,M)
```

```
      DO 1 NROT=1,NR
      L1=N1
      L2=N2
      IF(NROT.EQ.1)GO TO 30
      IF(NR.EQ.4)GO TO 20
      IF(NROT.EQ.2)L1=N3
      IF(NROT.EQ.2)L2=N1
      IF(NROT.EQ.3)L1=N2
      IF(NROT.EQ.3)L2=N3
      GO TO 30
   20 CONTINUE
      IF(NROT.NE.2)GO TO 31
      L1=N4
      L2=N1
      GO TO 30
   31 IF(NROT.NE.3)GO TO 32
      L1=N3
      L2=N4
      GO TO 30
   32 L1=N2
      L2=N3
   30 CONTINUE
      CALL DCOSL(L2,L1,A1,B1,C1,X,ND)
      E1(1)=A1
      E1(2)=B1
      E1(3)=C1
      E2(1)=B*E1(3)-C*E1(2)
      E2(2)=C*E1(1)-A*E1(3)
      E2(3)=A*E1(2)-B*E1(1)
C
C     CALCULATE DISTANCES
C
      D=ZERO
      DI=ZERO
      DJ=ZERO
      DO 2 L=1,3
      D =D +E2(L)*(X(L,L1)-XCOM(L))
      DI=DI+E1(L)*(X(L,L1)-XCOM(L))
      DJ=DJ+E1(L)*(X(L,L2)-XCOM(L))
    2 CONTINUE
C
      RI=SQRT(DIST2(XCOM,X(1,L1)))
      RJ=SQRT(DIST2(XCOM,X(1,L2)))
      IVAL=0
      DO 1 I=1,3
      DO 1 J=1,3
      IVAL=IVAL+1
      UU(IVAL)=UU(IVAL)-UVAL(DI,RI,D)+UVAL(DJ,RJ,D)
      IF(I.EQ.J)GO TO 1
      TT(IVAL)=TT(IVAL)-TVAL(DI,RI)+TVAL(DJ,RJ)
    1 CONTINUE
      RETURN
      END

**************************************************************************************************************

C
C     DISTANCE BETWEEN TWO POINTS.
C
      FUNCTION DIST2(X,Y)
```

```
      DIMENSION X(1),Y(1)
      DISTZ=(X(1)-Y(1))**2+(X(2)-Y(2))**2+(X(3)-Y(3))**2
      RETURN
      END
*********************************************************************************************************************
C
C     SUBROUTINE TO HANDLE DISK TRANSFERS.
C
      SUBROUTINE DUMPO(IBUF,BUF,ISTAR,IRW)
      DIMENSION BUF(1)
      NBLOK=9
      IER=555
      IF(IRW.EQ.1)CALL WRBLK(2,ISTAR,BUF,NBLOK,IER)
      IF(IRW.EQ.2)CALL RDBLK(2,ISTAR,BUF,NBLOK,IER)
      IBUF=0
      ISTAR=ISTAR+NBLOK
      IF(IER.EQ.1)RETURN
      TYPE "<207>IER,ISTAR,IRW",IER,ISTAR,IRW
      PAUSE FOR A WHILE
      RETURN
      END
*********************************************************************************************************************
C
C     OUTPUT OF ERROR MESSAGES
C
      SUBROUTINE ERROR(IEL,N,M,I,F)
      IF(I.EQ.3)GO TO 10
      IF(IEL.GT.1000)GO TO 10
      IEL=IEL*10.+I
      IF(I.EQ.14)IEL=(IEL-14)*10+I
   10 CONTINUE
      GO TO (1,2,3,4,5,6,7,8,9,1,1,1,1,14),I
    2 TYPE "ELT",N," SPECIFIED MORE THAN ONCE"
      GO TO 1
    3 TYPE "NODE",N," NOT IN POSITION",M," AS REQUIRED"
      GO TO 1
    4 TYPE "ELT",N," IS UNDEFINED"
      GO TO 1
    5 TYPE "ELT",N," HAS ILLEGAL NODE NUMBER IN POSITION",M
      GO TO 1
    6 TYPE "NODE",M," IN ELT",N," IS NEGATIVE ?"
      GO TO 1
    7 TYPE "QUADRILATERAL ELT",N," IS HAS ONLY 3 NODES ?"
      GO TO 1
    8 TYPE "ELT",N," IS CURVED IN COMPONENT",M," CURVATURE=",F
      GO TO 1
    9 TYPE "ELT",N," HAS VERY BIG OR SMALL AREA"
      GO TO 1
   14 TYPE "ELT",N," NOT AT AN INTERFACE (ILLEGAL FOR CODE 7 ELEMENTS)"
    1 CONTINUE
      RETURN
      END
*********************************************************************************************************************
      PARAMETER JQ=13,JU=89,JY=700
```

```
C
C      MODE 1 = POINT MODE
C           2 = LINE  MODE
C           3 = PLANE MODE
C           4 = ELT   MODE
C           5 = PLANE MODE WITH VECTOR OFFSET (USING ELEMENT NODES)
C
C
C     GENERATE BENCHMARKS ALONG LINES , PLANES, OR AT POINTS
C
      SUBROUTINE GENBEN
C     JACD
      COMMON X(3,JY),ALUMP(JU),XLUMP(3,JU),DLUMC(3,JU),LKOD(JU),LMOD(JU),
     *  IFACE(JU),TLUMP(3,JU),ULUMP(3,JU),
     2  NDIR(8,8),NMULT(3,8),XSYM,YSYM,ZSYM,ISYM,NSYM(5),PRIMO(3,3),PRIM1(3,3),
     3  EMOD(5),PR(5),E,G,V,V1,V2,V3,V4,V5,V6,V7,V8,V9,CK,CD,CONS,IBC(3),
     4  TRANS,TRAN7,TRAN21,TRAN42,TRANL,TRANLL,WGHT(11),COORD(3,11),
     5  NODE,NITER,NLUMP,NDIMEL,NDIML,ND,ISTAR,IBUF,LIM,NSLIP,NUMOR,NMOD,
     6  ONE,NIL,TWO,THIRD,HALF,ZERO,I1,I2,I3,PI,ALT(3,3,3),DELTA(3,3),OMEGA,
     7  TT(9),UU(9),IREAD,IRITE,A,B,C,R,RSQ,ICHAN,ISSW(20),XCOM(3),
     8  DEL,ARMIN,ARMAX,CURV,RAT1,RATB,ARRAT,ISWAP,R1,R2,R3,R4,K1,K2,K3
C
      COMMON /EM/ INDEX(4,JQ),DCEL(3,JQ),XCEL(3,JQ),T(3,JQ),U(3,JQ),AREA(JQ)
     E  ,IEL1(JQ),KOD(JQ),LIND(JU),NUMEL,COH,PHI,THIK
C
      COMMON/LA/ JPNT(11),JLINE(11),JPLN(11),JELT(11)
      DIMENSION XBEN(3),XE(3),XB(3)
      DATA JPNT/"INDIVIDUAL FIELD POINT"/
      DATA JLINE/"FIELD POINT STRING NO "/
      DATA JPLN/"PLANAR SET NUMBER     "/
      DATA JELT/"SET NORMAL TO ELT NO  "/
C
      NLINE=NUMOR
      NUMOR=NIL
      TYPE "IN GENBEN",NLINE
      DO 3 KB=1,NLINE
      READ (1) NO,MODE,ISUB
      IF(ISUB.GT.NMOD)PAUSE SUBREGION ERROR
      IF(NO.LE.NIL)GO TO 3
      IF(MODE.LE.NIL)PAUSE MODE ERROR
      GO TO (10,20,30,40,50),MODE
   10 TYPE "POINT",NO
      DO 11 N=1,NO
      NUMOR=NUMOR+1
      READ (1) IVAL,XBEN
   11 WRITE BINARY (6) N,IVAL,XBEN,ISUB
      GO TO 3
   20 TYPE "LINE",NO
      DO 21 N=1,NO
      READ (1) XB,XE,NPT
      IF(NPT.LE.NIL)GO TO 21
      F=FLOAT(NPT-1)
      DO 22 NP=1,NPT
      FP=FLOAT(NP-1)/F
      DO 23 I=1,3
   23 XBEN(I)=XB(I)+FP*(XE(I)-XB(I))
      NUMOR=NUMOR+1
   22 WRITE BINARY (6) N,NP,XBEN,ISUB
   21 CONTINUE
      GO TO 3
```

```
30 TYPE "PLANE"
   DO 31 N=1,NO
   N1=NODE+1
   N2=NODE+2
   N3=NODE+3
   N4=NODE+4
   READ(1)((X(J,N),J=1,3),N=N1,N4)
   READ(1)N12,N14
C
C  N1=(-1,-1)  N2=(1,-1)   N3=(1,1)   N4=(-1,1)
C   N12=NX  ,  N14=NY
C
   DXL=1./FLOAT(N12)
   DYL=1./FLOAT(N14)
   XBL=-1.-DXL
   YBL=-1.-DYL
   DO 32 NX=1,N12
   DO 32 NY=1,N14
   XL=XBL+NX*DXL*2.
   YL=YBL+NY*DYL*2.
   OMX=1.-XL
   OPX=1.+XL
   OMY=1.-YL
   OPY=1.+YL
   DO 33 I=1,3
33 XBEN(I)=0.25*(OMX*OMY*X(I,N1)+OPX*OMY*X(I,N2)
  1              +OPX*OPY*X(I,N3)+OMX*OPY*X(I,N4))
   NUMOR=NUMOR+1
32 WRITE BINARY (6) NX,NY,XBEN,ISUB
31 CONTINUE
   GO TO 3
C
C  ELT CENTRED BENCHMARKS
C  LUMP NO,EWB,EWE,NO OF POINTS
C  IF ELT NO=N1L , ALL ELEMENTS IN THE LUMP ARE DONE
C
40 TYPE "ELT"
   DO 41 N=1,NO
   READ(1) NL,NEL,DB,DE,NPT
   CALL LIBLK(NL,1,1)
   NUM=NUMEL
   IF(NEL.NE.N1L)NUM=1
   DO 44 NE=1,NUM
   NN=NE
   IF(NEL.NE.N1L)NN=NEL
   CALL LSET(XCOM,XCEL(1,NN))
   DO 42 NP=1,NPT,
   FP=FLOAT(NP-1)/FLOAT(NPT-1)
   DO 43 I=1,3
43 XBEN(I)=XCOM(I)+DCEL(I,NN)*(DB+FP*(DE-DB))
   NUMOR=NUMOR+1
42 WRITE BINARY (6) NP,IELT(NN),XBEN,ISUB
44 CONTINUE
41 CONTINUE
   GO TO 3
C
C  N1,N2,N3,N4 DEFINES "SHAPE" QUADRILATERAL
C  XCOM  DEFINES OFFSET
C  N12,N14 = NO OF COLS AND ROWS
C
```

```
   50 TYPE "VECTOR OFFSET"
      DO 51 N=1,NO
      READ(1) N1,N2,N3,N4
      READ(1)XCON
      READ(1)N12,N14
C
C     N1=(-1,-1)  N2=(1,-1)   N3=(1,1)   N4=(-1,1)
C      N12-NX  ,  N14=NY
C
      DXL=1./FLOAT(N12)
      DYL=1./FLOAT(N14)
      XBL=-1.-DXL
      YBL=-1.-DYL
      DO 52 NX=1,N12
      DO 52 NY=1,N14
      XL=XBL+NX*DXL*2.
      YL=YBL+NY*DYL*2.
      OMX=1.-XL
      OPX=1.+XL
      OMY=1.-YL
      OPY=1.+YL
      DO 53 I=1,3
   53 XBEN(I)=0.25*(OMX*OMY*X(I,N1)+OPX*OMY*X(I,N2)
     1                +OPX*OPY*X(I,N3)+OMX*OPY*X(I,N4))+XCOM(I)
      NUMOR=NUMOR+1
   52 WRITE BINARY (6) NX,NY,XBEN,ISUB
   51 CONTINUE
    3 CONTINUE
      TYPE "EXIT GENBEN"
      RETURN
      END
****************************************************************************************************************
C
C     SET UP IBC WITH BOUNDARY CONDITION CODES
C
      SUBROUTINE IBLOAD(KOD,IBC)
      DIMENSION IBC(3)
      IT=1
      ID=2
      I1=IT
      I2=IT
      I3=IT
      IBC(1)=0
      IBC(2)=0
      IBC(3)=0
      IF(KOD.LE.0.OR.KOD.GT.11)RETURN
      GO TO (20,2,3,4,5,6,7,8,9,7,9),KOD
    2 I1=ID
      I2=ID
      I3=ID
      GO TO 20
    3 I1=ID
      I2=IT
      I3=IT
      GO TO 20
    4 I1=IT
      I2=ID
      I3=IT
```

```
      GO TO 20
    5 I1=IF
      I2=IF
      I3=ID
      GO TO 20
    6 PAUSE ILLEGAL KODE 6
    7 I1=3
      I2=3
      I3=3
      GO TO 20
    8 I1=4
      I2=4
      I3=4
      GO TO 20
    9 I1=3
      I2=3
      I3=4
      GO TO 20
   20 CONTINUE
      IBC(1)=I1
      IBC(2)=I2
      IBC(3)=I3
      RETURN
      END

*******************************************************************************************************************

      PARAMETER JQ=19,JU=69,JY=700
      SUBROUTINE INITT
C     JACD
      COMMON X(3,JY),ALUMP(JU),XLUMP(3,JU),DLUMC(3,JU),LKOD(JU),LMOD(JU),
     * IFACE(JU),TLUMP(3,JU),ULUMP(3,JU),
     2 NDIR(8,8),NMULT(3,8),XSYM,YSYM,ZSYM,ISYM,NSYM(5),PRIMO(3,3),PRIM1(3,3),
     3 EMOD(5),PR(5),E,G,V,V1,V2,V3,V4,V5,V6,V7,V8,V9,CK,CD,CONS,IBC(3),
     4 TRANS,TRAN7,TRAN21,TRAN42,TRANL,TRANLL,WGHT(11),COORD(3,11),
     5 NODE,NITER,NLUMP,NDIMEL,NDIML,ND,ISTAR,IBUF,LIM,NSLIP,NUMOR,NMOD,
     6 ONE,NIL,TWO,THIRD,HALF,ZERO,I1,I2,I3,PI,ALT(3,3,3),DELTA(3,3),OMEGA,
     7 TT(9),UU(9),IREAD,IRITE,A,B,C,R,RSQ,ICHAN,ISSW(20),XCOM(3),
     8 DEL,ARMIN,ARMAX,CURV,RATT,RATB,ARRAT,ISWAP,R1,R2,R3,R4,K1,K2,K3
C
C     SUBROUTINE TO INITIALIZE THE MATRICES.
C
      TYPE "INTO INITT"
      ND=JY-5
      NDIML=JU
      NDIMEL=JQ
      IBFSZ=1152
      LIM=IBFSZ-18
      ZERO=0.
      NIL=0
      ONE=1.
      IONE=1
      TWO=2.
      HALF=0.5
      THIRD=1./3.
      I1=1
      I3=2
      I3=3
      PI=3.14159
C
```

```
      DO 1 I=1,3
      DO 1 J=1,3
      DELTA(I,J)=ZERO
      DELTA(I,I)=ONE
      DO 1 K=1,3
    1 ALT(I,J,K)=ZERO
      ALT(1,2,3)=ONE
      ALT(2,3,1)=ONE
      ALT(3,1,2)=ONE
      ALT(1,3,2)=-ONE
      ALT(3,2,1)=-ONE
      ALT(2,1,3)=-ONE
      THIRD=ONE/3.
      A1=0.05961587
      A2=0.79742699
      B1=0.47014206
      B2=0.10128651
      W1=0.225
      W2=0.13239415
      W3=0.12593918
C
      COORD(1,1)=THIRD
      COORD(2,1)=THIRD
      COORD(3,1)=THIRD
C
      COORD(1,2)=A1
      COORD(2,2)=B1
      COORD(3,2)=B1
C
      COORD(1,3)=B1
      COORD(2,3)=A1
      COORD(3,3)=B1
C
      COORD(1,4)=B1
      COORD(2,4)=B1
      COORD(3,4)=A1
C
      COORD(1,5)=A2
      COORD(2,5)=B2
      COORD(3,5)=B2
C
      COORD(1,6)=B2
      COORD(2,6)=A2
      COORD(3,6)=B2
C
      COORD(1,7)=B2
      COORD(2,7)=B2
      COORD(3,7)=A2
C
      COORD(1,8)=HALF
      COORD(2,8)=HALF
      COORD(3,8)=ZERO
C
      COORD(1,9)=HALF
      COORD(2,9)=ZERO
      COORD(3,9)=HALF
C
      COORD(1,10)=ZERO
      COORD(2,10)=HALF
      COORD(3,10)=HALF
```

```
C
      COORD(1,11)=THIRD
      COORD(2,11)=THIRD
      COORD(3,11)=THIRD
C
      WGHT(1)=W1
      WGHT(2)=W2
      WGHT(3)=W2
      WGHT(4)=W2
      WGHT(5)=W3
      WGHT(6)=W3
      WGHT(7)=W3
      WGHT(8)=THIRD
      WGHT(9)=THIRD
      WGHT(10)=THIRD
      WGHT(11)=ONE
C
      ND3=ND*3
      CALL NULL(X,ND3)
      DO 3 J=1,8
      IF(J.LT.4)NMULT(J,1)=IONE
      NDIR(1,J)=1
      DO 3 I=2,8
      IF(J.LT.4)NMULT(J,1)=-1
      NDIR(I,J)=0
      NDIR(1,8)=1
    3 NDIR(J,J)=1
      NDIR(2,5)=IONE
      NDIR(2,6)=IONE
      NDIR(3,5)=IONE
      NDIR(3,7)=IONE
      NDIR(4,6)=IONE
      NDIR(4,7)=IONE
      NMULT(2,2)=IONE
      NMULT(3,2)=IONE
      NMULT(1,3)=IONE
      NMULT(3,3)=IONE
      NMULT(1,4)=IONE
      NMULT(2,4)=IONE
      NMULT(3,5)=IONE
      NMULT(2,6)=IONE
      NMULT(1,7)=IONE
      TYPE "EXIT INITT"
C
      RETURN
      END
*******************************************************************************************************************
      PARAMETER J0=13,JU=89,JY=700
      SUBROUTINE INPUT
C     JACD                                                    *
      COMMON X(3,JY),ALUMP(JU),XLUMP(3,JU),DLUMC(3,JU),LKOD(JU),LMOD(JU),
     * IFACE(JU),TLUMP(3,JU),ULUMP(3,JU),
     2 NDIR(8,8),NMULT(3,8),XSYM,YSYM,ZSYM,ISYM,NSYM(5),PRIMO(3,3),PRIMI(3,3),
     3 ENOD(5),PR(5),E,G,V,V1,V2,V3,V4,V5,V6,V7,V8,V9,CK,CD,CONS,IBC(3),
     4 TRANS,TRAN7,TRAN21,TRAN42,TRANL,TRANLL,WGHT(11),COORD(3,11),
     5 NODE,NITER,NLUMP,NDIMEL,NDIML,ND,ISTAR,IBUF,LIM,NSLIP,NUMOR,NMOD,
     6 ONE,NIL,TWO,THIRD,HALF,ZERO,I1,I2,I3,PI,ALT(3,3,3),DELTA(3,3),OMEGA,
     7 TT(9),UU(9),IREAD,IRITE,A,B,C,R,RSQ,ICHAN,ISSW(20),XCOM(3),
```

```
     &  DEL, ARMIN, ARMAX, CURV, RATT, RATD, ARRAT, ISKIP, R1, R2, R3, R4, K1, K2, K3
C
      COMMON /EM/ INDEX(4, JQ), DCEL(3, JQ), XCEL(3, JQ), T(3, JQ), U(3, JQ), AREA(JQ)
     E  , IELT(JQ), KOD(JQ), LIND(JQ), NUMEL, COH, PHI, THIK
C
      COMMON /RE/ INDEXR(4, JQ), DCELR(3, JQ), XCELR(3, JQ), TR(3, JQ), UR(3, JQ)
     R  , AREAR(JQ), IELTR(JQ), KODR(JQ), LINDR(JQ), NUMELR, COHR, PHIR, THIKR
C
      DIMENSION XE(3), IHED(32), DIS(3), TRAC(3), NIND(20)
C
C     SUBROUTINE TO READ INPUT VARIABLES AND SET CONSTANTS.
C
      TYPE "INTO INPUT"
      IREAD = 1
  100 FORMAT(S63)
      CALL SETFILE
      READ(IREAD, 202)ISSW
      IRITE=12
      IF(ISSW(5). EQ. 1)IRITE=10
      READ(IREAD, 100)IHED(1)
      WRITE(IRITE, 153)
  153 FORMAT(/1X, 128("*"), /)
      IF(IRITE. EQ. 12)WRITE(IRITE, 106)IHED
  106 FORMAT(/1X, "", 32A2, //)
      CALL KRONO(7, IRITE, TON, TINC)
      IF(IRITE. EQ. 12)WRITE(IRITE, 107)
  107 FORMAT(/1X, "BOUNDARY ELEMENT METHOD 3 D ANALYSIS (MARK 2)")
      WRITE(IRITE, 153)
C
C     READ AND WRITE INPUT DATA.
C
C
      WRITE(IRITE, 254)(ISSW(I), I=1, 7)
  254 FORMAT("     SWITCH CONTROLS ARE AS FOLLOWS", /
     *" ISSW 1 = 1 TO SKIP CALCULATION OF COEFFICIENTS. ", T60, I1, /
     *" ISSW 2 = 1 TO SKIP CALCULATION OF T AND U. ", T60, I1, /
     *"          = 2 TO CONTINUE CALCULATION OF T AND U", /
     *" ISSW 3 = 1 TO SKIP OUTPUT OF GEOMETRY. ", T60, I1, /
     *" ISSW 4 = 1 TO SKIP OUTPUT OF LUMP GEOMETRY. ", T60, I1, /
     *" ISSW 5 = 1 FOR OUTPUT TO SCREEN. ", T60, I1, /
     *" ISSW 6 = 1 TO SKIP OUTPUT OF TRACTIONS AND DISPLACEMENTS. ", T60, I1, /
     *" ISSW 7 = 1 TO SKIP PRINCIPAL STRESS ROUTINE", T60, I1, /
     *, /)
C
      READ(IREAD)TRANLL, TRANL, TRAN42, TRAN21, TRAN7, TRAN5
      READ(IREAD)ARMIN, ARMAX, CURV, RATT, RATD
      READ(IREAD)NITER, OMEGA, DEL, NSLIP
      READ(IREAD)NUMOR, NLUMP, NMOD
      READ(IREAD)XSYM, YSYM, ZSYM
      READ(IREAD)PRIMO, PRIM1
C
C
  202 FORMAT(25I1)
      WRITE(IRITE, 101)NITER, OMEGA, DEL, XSYM, YSYM, ZSYM
  101 FORMAT(/
     4          10X"MAX  ITERATIONS       = ", 6X, I4, /
     5          10X"RELAXION FACTOR       = ", F10. 2, /
     6          10X"CONVERGENCE RATIO     = ", F10. 5, /
     1          10X"X PLANE OF SYMMETRY = ", F10. 2, /
     2          10X"Y PLANE OF SYMMETRY = ", F10. 2, /
```

```
     3       10X"Z PLANE OF SYMMETRY = ",F10.2,/)
      WRITE(IRITE,102)NUMOR,NLUMP,TRAN42,TRAN21,TRAN7,TRAN5,TRANL,TRANLL
  102 FORMAT(10X"NO OF FIELD GROUPS  = ",6X,I4,/
     3       10X"NO OF LUMPS         = ",6X,I4,/
     5       10X"TRANS FOR 42 PT     = "F10.2,/
     7       10X"TRANS FOR 21 PT     = ",F10.2,/
     6       10X"TRANS FOR  7 PT     = ",F10.2,/
     7       10X"TRANS FOR  3 PT     = ",F10.2,/
     8       10X"ELEMENT LUMPING     = ",F10.2,/
     9       10X"LUMP    LUMPING     = ",F10.2,/
     1,/"  NOTE THAT KODE 33 IS A TRACTION FREE ELEMENT",/
     *,/)
C
      WRITE(IRITE,117)
  117 FORMAT(/" PRIMITIVE STRESSES ARE GIVEN BY ",/
     1" SIGMA = CONSTANT + DEPTH * OVERBURDEN VALUE",/
     1" THE VALUE OF SIGMA MUST BE - FOR COMPRESSIVE STRESSES",/)
      WRITE(IRITE,114)PRIMO
      WRITE(IRITE,115)PRIM1
  114 FORMAT(T10"  CONSTANT PRIMITIVE STRESS TENSOR IS:",/
     1T10" ----------------------------------------",//
     1T5,3(F10.3,1X),/T5,3(F10.3,1X),/T5,3(F10.3,1X),/)
  115 FORMAT(T10" OVERBURDEN PRIMITIVE STRESS TENSOR IS:",/
     1T10" ----------------------------------------",//
     1T5,3(F10.3,1X),/T5,3(F10.3,1X),/T5,3(F10.3,1X),/)
      IF(NMOD.GT.5)STOP 1
      WRITE(IRITE,113)NMOD
  113 FORMAT(" NO OF DIFFERENT SUBREGIONS =",I4,/)
      MAXSYM=NIL
      DO 104 I=1,NMOD
      READ(IREAD)J,EMOD(J),PR(J),NSYM(J)
      IF(NSYM(J).GT.9.OR.NSYM(J).LE.0)STOP 22
      IF(NSYM(J).GT.MAXSYM)MAXSYM=NSYM(J)
      IF(I.NE.J)STOP 2
      WRITE(IRITE,105)EMOD(I),PR(I),I,NSYM(I)
  104 CONTINUE
  105 FORMAT(" E ="F10.2"    POISSONS RATIO ="F6.2,"      SUBREGION"
     1," NO ="I4,"   SYMMETRY CODE ="I4)
C
C     READ IN NODAL COORDS
C
      NODE=NIL
      CALL NODGEN(X,NODE,ND,IREAD,IRITE)
      IF(NODE.GT.ND.OR.NLUMP.GT.NDIML)STOP 6
      XSYM=XSYM*TWO
      YSYM=YSYM*TWO
      ZSYM=ZSYM*TWO
      NLTEM=NLUMP
      IBIG=NDIMEL*3
C
C     READ IN LUMPS AND ELEMENTS
C
      WRITE(IRITE,180)
      DO 191 NL=1,NLUMP
      IF(ISSW(3).EQ.NIL.AND.NL.EQ.((NL/6)*6))WRITE(IRITE,180)
      IF(ISSW(3).EQ.NIL)WRITE(IRITE,185)
  185 FORMAT(" ")
      COH=ZERO
      PHI=ZERO
      THIK=ZERO
```

```
      TYPE NL,"=LUMP NO"
      READ(IREAD)LUMP,NUMEL,KODL,MOD1,MOD2
      IFREE=KODL
      IF(KODL.EQ.33)KODL=1
      NO=NUMEL
      NUMEL=IABS(NUMEL)
      IFACE(NL)=NIL
      IF(KODL.GT.6)READ (IREAD) CON,PHI,THIK
      PHI=TAN(PI*PHI/180.0)
      IF(KODL.LT.(-1))KODL=-1
      IF(KODL.GT.6.AND.MOD1.EQ.MOD2)STOP KODE 7 INTERFACE ERROR
      IF(MOD1.GT.5.OR.MOD2.GT.5)STOP 3
      IF(NUMEL.GT.NDIMEL)TYPE "ERROR 1",LUMP
      IF(LUMP.NE.NL)TYPE "POSSIBLE LUMP ERROR",NL,LUMP
      LMOD(NL)=MOD1
      NTEM=NDIMEL*3
      CALL NULLI(INDEX,NDIMEL,4)
      CALL NULLI(LIND ,NDIML,1)
      CALL NULLI(LINDR,NDIML,1)
      CALL NULL4(U,UR,T,TR,NTEM)
      CALL NULL4(XCEL,XCELR,DCEL,DCELR,NTEM)
      N=0
      DO 11 NE=1,NUMEL
      IF(NO.GT.NIL)GO TO 12
      READ(IREAD)NN,NUM,(NIND(J),J=1,NUM)
      IF(NIND(NUM).LT.NIL)NO=-NO
      NIND(NUM)=IABS(NIND(NUM))
      IF(NUM.GT.9)GO TO 14
      KEL=NUM-1
      DO 17 K=1,KEL
      N=N+1
      IF(N.GT.NDIMEL)STOP 13
      IELT(N)=NN
      NN=NN+1
      INDEX(1,N)=NIND(K)
      KP=K+1
      IF(KP.GT.KEL)KP=1
      INDEX(2,N)=NIND(KP)
      INDEX(3,N)=NIND(NUM)
   17 CONTINUE
      GO TO 11
   14 NROW=(NUM/4)-1
      KEL=NROW*3
      DO 15 K=1,KEL
      IROW=(K-1)/3
      N=N+1
      IF(N.GT.NDIMEL)STOP 13
      IELT(N)=NN
      NN=NN+1
      N1=IROW+K
      INDEX(1,N)=NIND(N1)
      INDEX(2,N)=NIND(N1+4)
      INDEX(3,N)=NIND(N1+5)
      INDEX(4,N)=NIND(N1+1)
   15 CONTINUE
      GO TO 11
   12 N=N+1
      IF(N.GT.NDIMEL)STOP 13
      READ(IREAD)NN,(INDEX(J,N),J=1,4)
      IELT(N)=NN
```

```
   11 CONTINUE
      NUMEL=N
      IF(KODL.NE.6)GO TO 18
      READ(IREAD)KODL
   19 READ(IREAD)IB,IE,DIS,TRAC
      DO 22 N=IB,IE
      CALL LSET(U(1,N),DIS)
   22 CALL LSET(T(1,N),TRAC)
      IF(IE.LT.NUMEL)GO TO 19
      LKOD(NL)=KODL
   18 CONTINUE
      LKOD(NL)=KODL
      IWILL=0
      IF(MOD2.EQ.0.OR.MOD2.EQ.MOD1)GO TO 60
      NLTEM=NLTEM+1
      IF(NLTEM.GT.NUIML)STOP 8
      IFACE(NL)=NLTEM
      IFACE(NLTEM)=NL
      IF(KODL.EQ.7)LKOD(NLTEM)=8
      IF(KODL.EQ.8)LKOD(NLTEM)=7
      LMOD(NLTEM)=MOD2
      IWILL=1
      NUMELR=NUMEL
      COHR=COH
      PHIR=PHI
      THIKR=THIK
   60 CONTINUE
      DO 1 N=1,NUMEL
      KODE=KODL
      IEL=NIL
      KOD(N)=KODE
      CALL LODNOD(N1,N2,N3,N4,INDEX(1,N))
C
C     GENERATE SECOND ELEMENT AT INTERFACE
C
      IF(IWILL.EQ.NIL)GO TO 62
      TYPE "GENERATE INTERFACE ELEMENT",N
      KODR(N)=KODE
      IF(KODE.EQ.7)KODR(N)=8
      IF(KODE.EQ.8)KODR(N)=7
      NNN=3
      IF(N4.NE.NIL)NNN=4
      NNP=NNN+1
      DO 61 J=1,NNN
   61 INDEXR(J,N)=INDEX(NNP-J,N)
   62 CONTINUE
C
C     CHECK GEOMETRY OF ELEMENTS.
C
      DO 10 J=1,4
      IF(INDEX(J,N).GT.NODE)CALL ERROR(IEL,N,J,5,DIFT)
      IF(J.LT.4.AND.INDEX(J,N).LT.1)CALL ERROR(IEL,N,J,6,DIFT)
   10 CONTINUE
      CALL DCOSP(N1,N2,N3,A,B,C,X,ND)
      A1=A
      B1=B
      C1=C
      AREA(N)=AREAF(N1,N2,N3,X,ND)
      IF(N4.EQ.NIL)GO TO 7
      CALL DCOSP(N1,N2,N3,A1,B1,C1,X,ND)
```

```
      DIFT=ABS(A-A1)
      IF(DIFT.GT.CURV.AND.(ABS(A)).GT.0.2)CALL ERROR(IEL,N,11,8,DIFT)
      DIFT=ABS(B-B1)
      IF(DIFT.GT.CURV.AND.(ABS(B)).GT.0.2)CALL ERROR(IEL,N,12,8,DIFT)
      DIFT=ABS(C-C1)
      IF(DIFT.GT.CURV.AND.(ABS(C)).GT.0.2)CALL ERROR(IEL,N,13,8,DIFT)
      AR=AREAF(N1,N3,N4,X,ND)
      RAT=AR/AREA(N)
      IF(RAT.LT.(0.2).OR.RAT.GT.(5.))
     1 WRITE(IRITE) "IRREGULAR QUAD ELEMENT",NL,N,RAT
      AREA(N)=AREA(N)+AR
    7 CONTINUE
      DCEL(1,N)=(A+A1)*HALF
      DCEL(2,N)=(B+B1)*HALF
      DCEL(3,N)=(C+C1)*HALF
      AREAR(N)=AREA(N)
      IF(AREA(N).LT.ARMIN.OR.AREA(N).GT.ARMAX)CALL ERROR(IEL,N,N,9,DIFT)
      CALL XCEN(INDEX(1,N),X,XCOM,ND)
C
C     CALCULATE B.C.'S
C
C     1  = SPECIFIED TRACTIONS.
C     2  = SPECIFIED DISPLACEMENTS.
C     3  =    "        Z-DISP
C     4  =    "        Y-DISP
C     5  =    "        X-DISP
C     6  =    "        B.C
C     7  = INTERFACE ELEMENTS
C     8  = FREE SURFACE (ZERO TRACTIONS)
C
C     IBC = 1 FOR TRACTIONS , 2 FOR DISPLACEMENTS.
C
      DEPTH=XCOM(3)
      IF(KODE.NE.9.AND.KODE.NE.1)GO TO 41
      DO 42 I=1,3
      DO 21 J=1,3
      SIG=-DEPTH*PRIM1(I,J)-PRIMO(I,J)
      IF(IFREE.EQ.33)SIG=0
   21 T(I,N)=SIG*DCEL(J,N)+T(I,N)
   42 CONTINUE
   41 CONTINUE
      IF(KODE.EQ.2)CALL NULL(U(1,N),3)
      DO 6 J=1,3
      DCELR(J,N)=DCEL(J,N)
      XCEL(J,N)=XCOM(J)
      XCELR(J,N)=XCOM(J)
      UR(J,N)=U(J,N)
    6 TR(J,N)=-T(J,N)
C
C
      KODE=KOD(N)
      CALL IBLOAD(KODE,IBC)
  180 FORMAT(/" LUMP   ELEMENT"T18"X-COORD"T28"Y-COORD"T38"Z-COORD"T44"   NODE "
     1," NUMBERS"T69"AREA"T77"DIR-X"T86"DIR-Y"T95"DIR-Z",6X,"B-C",
     1T115"KODE  ERROR CODE",/)
      IF(ISSW(3).EQ.1.AND.IEL.EQ.NIL)GO TO 3
      WRITE(IRITE,181)NL,N,IELT(N),(XCEL(I,N),I=1,3),(INDEX(I,N),I=1,4)
     1,AREA(N),(DCEL(I,N),I=1,3),(IBC(I),I=1,3),KOD(N),IEL
  181 FORMAT(I4,I5,I5,F10.2,F10.2,F10.2,T48,4I4,T61,F10.2,3(F8.2,1X)
     1  ,3I4,T113,I4,8X,I5)
```

```
    3 CONTINUE
    1 CONTINUE
      IF(ISSW(1).NE.0.OR.ISSW(2).NE.0)GO TO 191
      IF(KODL.GT.NIL)CALL LDBLK(NL,2,1)
      IF(IWILL.EQ.1)CALL LDBLK(NLTEM,2,2)
      IF(KODL.GE.NIL)GO TO 191
      CALL LDBLK(NL,1,1)
      LKODE=LKOD(NL)
      DO 196 N=1,NUMEL
      TYPE "N,KOD",N,LKODE
  196 KOD(N)=LKODE
      CALL LDBLK(NL,2,1)
  191 CONTINUE
C
C     READ IN FIELD POINT LOCATIONS
C
      IF(NUMOR.LE.NIL)GO TO 4
      CALL OPEN(6,"BENCH ",3,IER)
      IF(IER.NE.1)TYPE "BENCH OPEN ERROR",IER
      CALL GENBEN
    4 CONTINUE
      NLUMP=NLTEM
      ICHAN=2*NLUMP+2
      TYPE "EXIT INPUT",ICHAN
      RETURN
      END
*****************************************************************************************************************
      PARAMETER JQ=13,JU=89,JY=700
C
C     SUBROUTINE TO INTEGRATE OVER A TRIANGLE FOR FIELD POINTS
C
      SUBROUTINE INTEGR(N,IN,JN,KN,TRAC,DIS,AR,INTE,STRES,DISP,XX,XC,DCO,UN)
C    JACU                                                              x
      COMMON X(3,JY),ALUMP(JU),XLUMP(3,JU),DLUMC(3,JU),LKOD(JU),LMOD(JU),
     * IFACE(JU),TLUMP(3,JU),ULUMP(3,JU),
     2 NDIR(8,8),NMULT(3,3),XSYM,YSYM,ZSYM,ISYM,NSYM(5),PRIMO(3,3),PRIM1(3,3),
     3 EMOD(5),PR(5),E,G,V,V1,V2,V3,V4,V5,V6,V7,V8,V9,CK,CD,CONS,INC(3),
     4 TRANS,TRAN7,TRAN21,TRAN42,TRANL,TRANLL,WGHT(11),COORD(3,11),
     5 NODE,NITER,NLUMP,NDIMEL,NDIML,ND,ISTAR,IBUF,LIM,NSLIP,NUMOR,NMOD,
     6 ONE,NIL,TWO,THIRD,HALF,ZERO,I1,I2,I3,PI,ALT(3,3,3),DELTA(3,3),OMEGA,
     7 TT(9),UU(9),IREAD,IRITE,A,B,C,R,RSQ,ICHAN,ISSW(20),XCOM(3),
     8 DEL,ARMIN,ARMAX,CURV,RATT,RATB,ARRAT,ISWAP,R1,R2,R3,R4,K1,K2,K3
C
      COMMON /EM/ INDEX(4,JQ),DCEL(3,JQ),XCEL(3,JQ),T(3,JQ),U(3,JQ),AREA(JQ)
     E ,IELT(JQ),KOD(JQ),LIND(JU),NUMEL,COH,PHI,THIK
C
      COMMON /RE/ INDEXR(4,JQ),DCELR(3,JQ),XCELR(3,JQ),TR(3,JQ),UR(3,JQ)
     R ,AREAR(JQ),IELTR(JQ),KODR(JQ),LINDR(JU),NUMELR,COHR,PHIR,THIKR
C
      COMMON /CO/ AO,BO,W
      DIMENSION TRAC(1),DIS(1),STRES(3,3),DISP(3),XX(3),XC(3),DCO(3)
      DIMENSION UN(3,JY),RCOM(3)
      NINTB=11
      NINTE=11
      WI=AR
      MINT=1
      NUMTRI=1
      IF(INTE.EQ.1)GO TO 20
```

```
C
      CALL XCENT(IN,JN,KN,XC,X,ND)
      CALL SYMFX(XSYM,YSYM,ZSYM,NMULT(1,ISYM),XC)
      AR=AREAF(IN,JN,KN,X,ND)
      RSQ=DIST2(XX,XC)
      RATIO=RSQ/AR
      IF(RATIO.GE.TRAN5)GO TO 20
      MINT=3
      NINTB=8
      NINTE=10
      IF(RATIO.GE.TRAN7)GO TO 20
      MINT=7
      NINTB=1
      NINTE=7
      IF(RATIO.GT.TRAN21)GO TO 20
      LN=NODE+2
      DO 444 L=1,3
      UN(L,LN)=(UN(L,IN)+UN(L,JN))*HALF
  444 X(L,LN)=(X(L,IN)+X(L,JN))*HALF
      MINT=21
      AR=AR/2.
      NUMTRI=2
      IF(RATIO.LT.TRAN42)
     1 WRITE(IRITE)"ELT",IELT(N)," TOO CLOSE --- RATIO =",RATIO
   20 CONTINUE
      IF(MINT.GT.1)TYPE MINT,N,ISYM
      IF(MINT.GT.7)TYPE RATIO,N
      DO 6 NTRI=1,NUMTRI
      IF(NUMTRI.NE.2)GO TO 43
      IF(NTRI.EQ.2)IN=JN
      IF(NTRI.EQ.2)JN=JJ
      JJ=JN
      IF(NTRI.EQ.1)JN=LN
   43 CONTINUE
      DO 6 NUM=NINTB,NINTE
      IF(NINTB.EQ.11)GO TO 64
      C1=COORD(1,NUM)
      C2=COORD(2,NUM)
      C3=COORD(3,NUM)
      W1=WGHT(NUM)*AR
      DO 24 L=1,3
   24 XC(L)=C1*X(L,IN)+C2*X(L,JN)+C3*X(L,KN)
      CALL SYMFX(XSYM,YSYM,ZSYM,NMULT(1,ISYM),XC)
      CALL BEMSHAPE(C1,C2,C3,3,UN,U(1,N),IN,JN,KN,DIS,ND)
      CALL SYMFX(ZERO,ZERO,ZERO,NMULT(1,ISYM),DIS)
      RSQ=DIST2(XX,XC)
   64 R=SQRT(RSQ)
      R3=R*RSQ
      WCK=-W1*CK/RSQ
      WCR=W1*CONS/R
      WS=W1*CD/R3
      WD=-WCK
      DRDN=ZERO
      DO 5 L=1,3
      RCOM(L)=(XC(L)-XX(L))/R
    5 DRDN=DRDN+RCOM(L)*DCO(L)
      D3=DRDN*3.
      DO 6 I=1,3
      RCI=RCOM(I)
      RIV3=RCI*V3
```

```
      RIV2=RCI*V2
      DC1=DCO(1)
      R13=RC1*3.
      DO 6 J=1,3
      RCJ=RCOM(J)
      DCJ=DCO(J)
      D1J=DELTA(1,J)
      RIRJ=RCI*RCJ
      DV6=D1J*V6
      RNV7=V7*(DCI*RCJ+DCJ*RC1)
      TFUNC=WCR*(DRDN*(D1J+R1V3*RCJ)+DC1*RCJ-DCJ*RC1)*DIS(J)
      DISP(I)=DISP(1)+WCR*(D1J*V4+R1V2*RCJ)*TRAC(J)-TFUNC
      IF(J.LT. 1)GO TO 6
      SUM=ZERO
      DO 7 K=1,3
      RCK=RCOM(K)
      DCK=DCO(K)
      D1K=DELTA(1,K)
      DJK=DELTA(J,K)
      RIJK=RIRJ*RCK
      DUM=D1K*RCJ+DJK*RCI
      D1JRK=D1J*RCK
      DKIJ=WD*(DUM-D1JRK+V3*RIJK)*TRAC(K)
    7 SUM=SUM-WS*(D3*(D1JRK+V9*DUM-V8*R1JK)+RNV7*RCK+3.*RIRJ*DCK
     1+DCJ*D1K+DC1*DJK-V6*DCK*D1J)*DIS(K)+DKIJ
      STRES(1,J)=STRES(1,J)+SUM
    6 CONTINUE
      RETURN
      END

****************************************************************************************************
C
C     LOAD UP THE ELEMENT PARAMETERS FOR A LUMP
C
      SUBROUTINE LDBLK(NL,IRW,KOD)
      COMMON /EM/ BUF (256)
      COMMON /RE/ BUFR(256)
      NBLOK=2
      IBLK=NL*NBLOK
      IER=5555
      IF(IRW.EQ.1.AND.KOD.EQ.1)CALL RDBLK(2,IBLK,BUF ,NBLOK,IER)
      IF(IRW.EQ.1.AND.KOD.EQ.2)CALL RDBLK(2,IBLK,BUFR,NBLOK,IER)
      IF(IRW.EQ.2.AND.KOD.EQ.1)CALL WRBLK(2,IBLK,BUF ,NBLOK,IER)
      IF(IRW.EQ.2.AND.KOD.EQ.2)CALL WRBLK(2,IBLK,BUFR,NBLOK,IER)
      IF(IER.NE.1)TYPE "LDBLK ERROR ",IER,IRW,KOD
      RETURN
      END

****************************************************************************************************

      PARAMETER JQ=13,JU=89,JY=700
      SUBROUTINE LESET(IA1,A1,AR1,IA2,A2,AR2)
C
C     JACD
      COMMON X(3,JY),ALUMP(JU),XLUMP(3,JU),DLUMC(3,JU),LKOD(JU),LMOD(JU),
     * IFACE(JU),TLUMP(3,JU),ULUMP(3,JU),
     2 NDIR(8,8),NMULT(3,8),XSYM,YSYM,ZSYM,ISYM,NSYM(5),PRIMO(3,3),PRIM1(3,3),
     3 EMOD(5),PR(5),E,G,V,V1,V2,V3,V4,V5,V6,V7,V8,V9,CK,CO,CONS,IBC(3),
     4 TRANS,TRAN7,TRAN21,TRAN42,TRANL,TRANLL,WGHT(11),COORD(3,11),
```

```
     5 NODE, NITER, NLUMP, NDIMEL, NDIML, ND, ISTAR, IBUF, LIM, NSLIP, NUMOR, NMOD,
     6 ONE, NIL, TWO, THIRD, HALF, ZERO, I1, I2, I3, PI, ALT(3,3,3), DELTA(3,3), OMEGA,
     7 TT(9), UU(9), IREAD, IRITE, A, B, C, R, RSQ, ICHAN, ISSW(20), XCOM(3),
     8 DEL, ARMIN, ARMAX, CURV, RATT, RATB, ARRAT, ISWAP, R1, R2, R3, R4, K1, K2, K3
C
C     SUBROUTINE TO LOAD XEL AND DLC.
C
      DIMENSION A1(1), AR1(1), A2(1), AR2(1)
      DO 1 I=1,3
      A1(I)=AR1(I)
    1 A2(I)=AR2(I)
      IF(IA1.EQ.  1)CALL  SYMFX(XSYM, YSYM, ZSYM, NMULT(1, ISYM), A1)
      IF(IA1.EQ.NIL)CALL  SYMFX(ZERO, ZERO, ZERO, NMULT(1, ISYM), A1)
      IF(IA2.EQ.  1)CALL  SYMFX(XSYM, YSYM, ZSYM, NMULT(1, ISYM), A2)
      IF(IA2.EQ.NIL)CALL  SYMFX(ZERO, ZERO, ZERO, NMULT(1, ISYM), A2)
      RETURN
      END
********************************************************************************************************************
      PARAMETER JQ=13, JU=89, JY=700
      SUBROUTINE LGEOM
C     JACD                                                                    *
      COMMON X(3,JY), ALUMP(JU), XLUMP(3,JU), DLUMC(3,JU), LKOD(JU), LMOD(JU),
     * IFACE(JU), TLUMP(3,JU), ULUMP(3,JU),
     2 NDIR(8,8), NMULT(3,8), XSYM, YSYM, ZSYM, ISYM, NSYM(5), PRIMO(3,3), PRIM1(3,3),
     3 EMOD(5), PR(5), E, G, V, V1, V2, V3, V4, V5, V6, V7, V8, V9, CK, CD, CONS, IBC(3),
     4 TRANS, TRAN7, TRAN21, TRAN42, TRANL, TRANLL, WGHT(11), COORD(3,11),
     5 NODE, NITER, NLUMP, NDIMEL, NDIML, ND, ISTAR, IBUF, LIM, NSLIP, NUMOR, NMOD,
     6 ONE, NIL, TWO, THIRD, HALF, ZERO, I1, I2, I3, PI, ALT(3,3,3), DELTA(3,3), OMEGA,
     7 TT(9), UU(9), IREAD, IRITE, A, B, C, R, RSQ, ICHAN, ISSW(20), XCOM(3),
     8 DEL, ARMIN, ARMAX, CURV, RATT, RATB, ARRAT, ISWAP, R1, R2, R3, R4, K1, K2, K3
C
      COMMON /RE/ INDEXR(4,JQ), DCELR(3,JQ), XCELR(3,JQ), TR(3,JQ), UR(3,JQ)
     R , AREAR(JQ), IELTR(JQ), KODR(JQ), LINDR(JQ), NUMELR, COHR, PHIR, THIKR
C
C     CHECK LUMP GEOMETRY
C
      N=NDIML*3
      CALL NULL4(DLUMC, XLUMP, TLUMP, ULUMP, N)
      ATOT=ZERO
      WRITE(IRITE,100)
      DO 1 NL=1,NLUMP
      IER=NIL
      IF(LKOD(NL).EQ.NIL)GO TO 1
      ALUMP(NL)=ZERO
      CALL LDBLK(NL,1,2)
      DO 11 N=1,NUMELR
   11 ALUMP(NL)=ALUMP(NL)+AREAR(N)
      ATOT=ATOT+ALUMP(NL)
      DO 12 N=1,NUMELR
      IER=NIL
      RAT=AREAR(N)/ALUMP(NL)
      IF(RAT.LT.RATB)TYPE "ERROR 10",N,NL,RAT," < ",RATB
      IF(RAT.GT.RATT)TYPE "ERROR 11",N,NL,RAT," > ",RATT
      IF(RAT.LT.RATB)IER=10
      IF(RAT.GT.RATT)IER=11
      DO 2 I=1,3
      XLUMP(I,NL)=XLUMP(I,NL)+XCELR(I,N)*RAT
    2 DLUMC(I,NL)=DLUMC(I,NL)+DCELR(I,N)*RAT
```

```
     2 NDIR(8,8),NMULT(3,8),XSYM,YSYM,ZSYM,ISYM,NSYN(5),PRIMO(3,3),PRIM1(3,3),
     3 EMOD(5),PR(5),E,G,V,V1,V2,V3,V4,V5,V6,V7,V8,V9,CK,CD,CONS,IBC(3),
     4 TRANS,TRAN7,TRAN21,TRAN42,TRANL,TRANLL,WGHT(11),COORD(3,11),
     5 NODE,NITER,NLUMP,NDIMEL,NDIML,ND,ISTAR,IBUF,LIM,NSLIP,NUMOR,NMOD,
     6 ONE,NIL,TWO,THIRD,HALF,ZERO,I1,I2,I3,P1,ALT(3,3,3),DELTA(3,3),OMEGA,
     7 TT(9),UU(9),IREAD,IRITE,A,B,C,R,RSQ,ICHAN,ISSW(20),XCOM(3),
     8 DEL,ARMIN,ARMAX,CURV,RATT,RATB,ARRAT,ISWAP,R1,R2,R3,R4,P1,K2,K3
C
      COMMON /BU/ BUF(1152)
C
C     SUBPROGRAM TO LOAD INFL COEFFICIENTS ON BOUNDARY ELEMENTS.
C
      DIMENSION TEMP(3),T(1),U(1)
      IF(IBUF GT LIM)CALL DUMPO(IBUF,BUF,ISTAR,1)
      IVAL=0
      IF(NODE.LT.NIL)GO TO 20
      IF(NODE.LT.6)CALL IBLOAD(NODE,IBC)
      DO 4 I=I1,I3
      DO 4 J=I1,I3
      IVAL=IVAL+1
      IF(NODE.LT.6)GO TO 501
      IBUF=IBUF+1
      BUF(IBUF)=TT(IVAL)
      IBUF=IBUF+1
      BUF(IBUF)=UU(IVAL)
      GO TO 4
  501 IBB=IBC(J)
      IF(IBB NE 1)GO TO 502
      TEMP(I)=TEMP(I)+UU(IVAL)*T(J)
      IBUF=IBUF+1
      BUF(IBUF)=TT(IVAL)
      GO TO 4
  502 TEMP(I)=TEMP(I)+TT(IVAL)*U(J)
      IBUF=IBUF+1
      BUF(IBUF)=UU(IVAL)
    4 CONTINUE
      RETURN
   20 DO 5 I=1,3
      DO 5 J=1,3
      IVAL=IVAL+1
    5 TEMP(I)=TEMP(I)+TT(IVAL)*U(J)+UU(IVAL)*T(J)
      RETURN
      END
*************************************************************************************************************************
C
C     CONVERT NODE ARRAY INTO   I,J,K,L
C
      SUBROUTINE LODNOD(I,J,K,L,IN)
      DIMENSION IN(1)
      I=IN(1)
      J=IN(2)
      K=IN(3)
      L=IN(4)
      RETURN
      END
*************************************************************************************************************************
```