1970

User's manual for axcst finite element program, February 1970

Erhard G. Schultchen

Celal N. Kostem

USER'S MANUAL FOR AXCST
FINITE ELEMENT PROGRAM

by

Erhard G. Schultchen

Celal N. Kostem

Department of Civil Engineering
Fritz Engineering Laboratory
Lehigh University
Bethlehem, Pennsylvania

February 1970

Fritz Engineering Laboratory Report No. 237.66

TABLE OF CONTENTS

1. INTRODUCTION

AXCST is a program for the stress and displacement analysis of axi-symmetric elasticity problems using the Finite Element method. In the program triangular elements with a linear displacement field are used.

The major blocks of this program are similar to those of the CSTES program (Ref. 1). The AXCST program can be used for the linear elastic analysis of axi-symmetric solids with axi-symmetric boundary conditions of any type. In addition, the program can solve thermal stress problems with arbitrary axi-symmetric temperature distribution.

2. ANALYTIC FORMULATION

2.1 Displacement Field

Stresses and displacements of a solid with axi-symmetric geometry, boundary conditions, and loading are also axi-symmetric. If the problem is formulated in cylindrical coordinates r, θ, and z, this implies that stresses and displacements are independent of θ. Thus, the three-dimensional elasticity problem will be reduced to a two-dimensional case, where the coordinates r and z completely define the stresses and displacements of the body in all directions.

The Finite Element approach for solving elasticity problems requires the definition of a displacement field within the element. The simplest form of this function is based on a complete first order polynomial in the following form:

$$u(r,z) = e_1 + e_2 r + e_3 z$$

$$v(r,z) = e_4 + e_5 r + e_6 z \qquad (1)$$

This assumption immediately defines the strain variation within each element, which is found to be constant except for ε_θ. Since the tangential strain varies as an inverse function of r, the normal stresses throughout the element will vary in radial direction as well.

The linear displacement function defined in Eq. (1) satisfies the compatibility conditions along the edges of the elements as long as nodal point continuity is maintained.

Expressing the unknown coefficients $e_1,\ldots,e_6$ in terms of nodal point displacements, the displacement function can be rewritten:

$$\begin{bmatrix} u\ (r,z) \\ v\ (r,z) \end{bmatrix} = \frac{1}{2\Delta} \begin{bmatrix} 1 \\ r \\ z \end{bmatrix}^T \begin{bmatrix} a_i & a_j & a_k \\ b_i & b_j & b_k \\ c_i & c_j & c_k \end{bmatrix} \begin{bmatrix} u_i & v_i \\ u_j & v_j \\ u_k & v_k \end{bmatrix} \qquad (2)$$

where

$$\begin{aligned} a_i &= r_j z_k - r_k z_j & b_i &= z_j - z_k & c_i &= r_k - r_j \\ a_j &= r_k z_i - r_i z_k & b_j &= z_k - z_i & c_j &= r_i - r_k \\ a_k &= r_i z_j - r_j z_i & b_k &= z_i - z_j & c_k &= r_j - r_i \end{aligned} \qquad (3)$$

$2\Delta = b_j c_k - b_k c_j$ $\quad |\Delta|$ = area of triangle.

It should be noted that the axi-symmetric CST-elements are actually complete rings with triangular cross-section, while the nodal "points" at which they are connected are circular lines in plane view (Fig. 1).

2.2 Fundamental Matrices

In the Finite Element formulation of elasticity problems four fundamental matrices are used:

(1) B - matrix,

relates the elemental strains to the nodal point displacements.

(2) D - matrix,

relates the elemental stresses to the elemental strains, known as the elasticity matrix.

(3) k^e - matrix,

relates the nodal point reactions to the nodal point displacements of each element, known as the element stiffness matrix.

(4) K - matrix,

relates the nodal point reactions to the nodal point displacements of the whole structure, known as the global stiffness matrix.

2.3 B - Matrix

From the theory of elasticity the following relationships can be found for the axi-symmetric state of strain (Ref. 2):

$$\epsilon_r = \frac{\partial u}{\partial r} \qquad \gamma_{rz} = \frac{\partial u}{\partial z} + \frac{\partial v}{\partial r}$$

$$\epsilon_z = \frac{\partial v}{\partial z} \qquad \gamma_{\theta r} = 0 \qquad (4)$$

$$\epsilon_\theta = \frac{u}{r} \qquad \gamma_{\theta z} = 0$$

Substituting the displacement function (Eq. (2)) into the strain-displacement relations (Eq. (4)), the elemental strains can be expressed in terms of nodal point displacements:

$$\underline{\varepsilon} = \underline{B}\ \underline{\delta}^e$$

$$\begin{bmatrix} \varepsilon_r \\ \varepsilon_z \\ \varepsilon_\theta \\ \gamma_{rz} \end{bmatrix} = \frac{1}{2\Delta} \begin{bmatrix} b_i & - & b_j & - & b_k & - \\ - & c_i & - & c_j & - & c_k \\ p_i & - & p_j & - & p_k & - \\ c_i & b_i & c_j & b_j & c_k & b_k \end{bmatrix} \begin{bmatrix} u_i \\ v_i \\ u_j \\ v_j \\ u_k \\ v_k \end{bmatrix} \tag{5}$$

where

$$p_i = \frac{a_i}{r} + b_i + \frac{c_i z}{r}$$

$$p_j = \frac{a_j}{r} + b_j + \frac{c_j z}{r}$$

$$p_k = \frac{a_k}{r} + b_k + \frac{c_k z}{r}$$

It is convenient to split the $\underline{B}$ - matrix into a constant and a variable part:

$$\underline{B} = \underline{B}_1 + \underline{B}_2$$

$\underline{B}_1$ defines here the element strains at the center of the triangle, while $\underline{B}_2$ takes into account the variation of ε_θ throughout the element.

$$\underline{B}_2 = \frac{1}{2\Delta} \begin{bmatrix} b_i & 0 & b_j & 0 & b_k & 0 \\ 0 & c_i & 0 & c_j & 0 & c_k \\ p_{ci} & 0 & p_{cj} & 0 & p_{ck} & 0 \\ c_i & b_i & c_j & b_j & c_k & b_k \end{bmatrix} \tag{6}$$

$$\underline{B}_2 = \frac{1}{2\Delta}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ q_i & 0 & q_j & 0 & q_k & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

where

$$p_{c_m} = \frac{a_m}{r_c} + b_m + \frac{c_m z_c}{r_c} \qquad m = i,j,k$$

$$q_m = \frac{a_m}{r} + \frac{c_m z}{r} - \frac{a_m}{r_c} - \frac{c_m z_c}{r_c}$$

$$r_c = \frac{1}{3}(r_i + r_j + r_k)$$

$$z_c = \frac{1}{3}(z_i + z_j + z_k)$$

coordinates of center of triangle

The expressions for p_{ci}, p_{cj}, and p_{ck} can be rewritten in the following form:

$$p_{ci} = p_{cj} = p_{ck} = \frac{2\Delta}{3r_c} = p_c$$

2.4 Elasticity Matrix

The elasticity matrix $\underline{D}$ defines the stress-strain relationship within each element. For an isotropic material the $\underline{D}$ matrix is of the following form:

$$\underline{\sigma} = \underline{D}\ \underline{\varepsilon}$$

$$\begin{bmatrix} \sigma_r \\ \sigma_z \\ \sigma_\theta \\ \tau_{rz} \end{bmatrix} = E \frac{(1-\nu)}{(1+\nu)\ (1-2\nu)} \begin{bmatrix} 1 & \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & - \\ \frac{\nu}{1-\nu} & 1 & \frac{\nu}{1-\nu} & - \\ \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 1 & - \\ - & - & - & \frac{1-2\nu}{2(1-\nu)} \end{bmatrix} \begin{bmatrix} \varepsilon_r \\ \varepsilon_z \\ \varepsilon_\theta \\ \gamma_{rz} \end{bmatrix} \quad (7)$$

2.5 Element Stiffness Matrix

A general expression for the element stiffness matrix, which relates nodal point reactions and nodal point displacements of each element, can be derived by applying the principle of virtual work (Ref. 3).

$$\underline{k}^e = \int_V \underline{B}^T\ \underline{D}\ \underline{B}\ dV \quad (8)$$

This integration has to be carried out over the total volume of the ring-element, however, due to axi-symmetry this volume integral can be simplified to an integral over the triangular cross section of the element.

$$\underline{k}^e = 2\pi \int_\Delta \underline{B}^T\ \underline{D}\ \underline{B}\ r\ dr\ dz \quad (9)$$

Separating the constant and variable parts of the $\underline{B}$ - matrix results in the following summation:

$$\underline{k}^e = 2\pi \int_\Delta \underline{B}_1^T \underline{D}\, \underline{B}_1\, r\, dr\, dz + 2\pi \int_\Delta \underline{B}_1^T \underline{D}\, \underline{B}_2\, r\, dr\, dz$$
$$+ 2\pi \int_\Delta \underline{B}_2^T \underline{D}\, \underline{B}_1\, r\, dr\, dz + 2\pi \int_\Delta \underline{B}_2^T \underline{D}\, \underline{B}_2\, r\, dr\, dz \qquad (10)$$

Noting that $\int_\Delta \underline{B}_2\, r\, dr\, dz = \int_\Delta \underline{B}_2^T\, r\, dr\, dz = 0$ and $\int_\Delta r\, dr\, dz = r_c \left|\Delta\right|$ the element stiffness matrix can be expressed as

$$\underline{k}^e = \underline{k}_1 + \underline{k}_2$$

or (11)

$$\underline{k}^e = 2\pi\, r_c \left|\Delta\right| \underline{B}_1^T \underline{D}\, \underline{B}_1 + 2\pi \int_\Delta \underline{B}_2^T \underline{D}\, \underline{B}_2\, r\, dr\, dz$$

The numerical evaluation of $\underline{k}_1$ presents no difficulties, however, problems arise in the integration of certain terms of $\underline{k}_2$. From Eqs. (6) and (7)

$$\underline{B}_2^T \underline{D}\, \underline{B}_2 = \frac{E}{4\Delta^2} \frac{(1-\nu)}{(1+\nu)\,(1-2\nu)} \begin{bmatrix} q_i q_i & - & q_j q_i & - & q_k q_i & - \\ - & - & - & - & - & - \\ q_i q_j & - & q_j q_j & - & q_k q_j & - \\ - & - & - & - & - & - \\ q_i q_k & - & q_j q_k & - & q_k q_k & - \\ - & - & - & - & - & - \end{bmatrix} \qquad (12)$$

where

$$q_m q_n = q_n q_m = \left(\frac{a_m}{r} + \frac{c_m z}{r} - \frac{a_m}{r_c} - \frac{c_m z_c}{r_c}\right)\left(\frac{a_n}{r} + \frac{c_n z}{r} - \frac{a_n}{r_c} - \frac{c_n z_c}{r_c}\right)$$

$$m = i,j,k$$

$$n = i,j,k$$

Integration of these terms leads to

$$\int_\Delta q_m q_n \, r \, dr \, dz = |\Delta|\left[a_m a_n\left(\frac{F_1}{\Delta} - \frac{1}{r_c}\right) + \left(a_m c_n + a_n c_m\right)\left(\frac{F_2}{\Delta} - \frac{z_c}{r_c}\right) + c_m c_n\left(\frac{F_3}{\Delta} - \frac{z_c^2}{r_c}\right)\right] \tag{13}$$

In the derivation of this expression the following integrals have to be evaluated:

$$\int_\Delta dr \, dz = |\Delta| \qquad \int_\Delta \frac{1}{r} dr \, dz = F_1$$

$$\int_\Delta r \, dr \, dz = |\Delta| \, r_c \qquad \int_\Delta \frac{z}{r} dr \, dz = F_2 \tag{14}$$

$$\int_\Delta z \, dr \, dz = |\Delta| \, z_c \qquad \int_\Delta \frac{z^2}{r} dr \, dz = F_3$$

For the general case $r_i \neq r_j \neq r_k$ and $r_i \neq 0$, $r_j \neq 0$, $r_k \neq 0$ (Fig. 2), the integrals F_1, F_2, and F_3 can be written as

$$F_1 = \int_\Delta \frac{1}{r} dr \, dz = I_{11} + I_{12} + I_{13}$$

$$I_{11} = \frac{a_i}{c_i} (Lnr_k - Lnr_j) + b_i \qquad \text{along k-j}$$

$$I_{12} = \frac{a_j}{c_j} (Lnr_i - Lnr_k) + b_j \qquad \text{along i-k}$$

$$I_{13} = \frac{a_k}{c_k} (Lnr_j - Lnr_i) + b_k \qquad \text{along j-i}$$

$$F_2 = \int_\Delta \frac{z}{r} \, dz \, dr = I_{21} + I_{22} + I_{23}$$

$$I_{21} = -\frac{1}{2} \frac{a_i^2}{c_i^2} (Lnr_k - Lnr_j) - \frac{a_i}{c_i} b_i - \frac{1}{4} \frac{b_i}{c_i} b_i (r_k + r_j)$$

$$I_{22} = -\frac{1}{2} \frac{a_j^2}{c_j^2} (Lnr_i - Lnr_k) - \frac{a_j}{c_j} b_j - \frac{1}{4} \frac{b_j}{c_j} b_j (r_i + r_k)$$

$$I_{23} = -\frac{1}{2} \frac{a_k^2}{c_k^2} (Lnr_j - Lnr_i) - \frac{a_k}{c_k} b_k - \frac{1}{4} \frac{b_k}{c_k} b_k (r_j + r_i)$$

$$F_3 = \int_\Delta \frac{z^2}{r} \, dz \, dr = I_{31} + I_{32} + I_{33}$$

$$I_{31} = +\frac{1}{3} \frac{a_i^3}{c_i^3} (Lnr_k - Lnr_j) + \frac{a_i^2}{c_i^2} b_i + \frac{1}{2} \frac{a_i}{c_i} \frac{b_i}{c_i} b_i (r_k + r_j)$$

$$+ \frac{1}{9} \frac{b_i^3}{c_i^3} (r_k^3 - r_j^3)$$

$$I_{32} = +\frac{1}{3} \frac{a_j^3}{c_j^3} (Lnr_i - Lnr_k) + \frac{a_j^2}{c_j^2} b_j + \frac{1}{2} \frac{a_j}{c_j} \frac{b_j}{c_j} b_j (r_i + r_k)$$

$$+ \frac{1}{9} \frac{b_j^3}{c_j^3} (r_i^3 - r_k^3)$$

$$I_{33} = +\frac{1}{3}\frac{a_k^3}{c_j^3}(\mathrm{Lnr}_j - \mathrm{Lnr}_i) + \frac{a_k^2}{c_k^2} b_k + \frac{1}{2}\frac{a_k}{c_k}\frac{b_k}{c_k} b_k (r_j + r_i)$$

$$+\frac{1}{9}\frac{b_k^3}{c_k^3}(r_j^3 - r_i^3)$$

It can be shown that certain simplifications of these expressions are possible; for instance, in the evaluation of F_1 the sum of $b_i + b_j + b_k$ is indentically zero. Whereas this is true for the general case of $r_m \neq r_n$ and $r_m \neq 0$, algebraic operations of this type are not permissible in certain special cases where terms of I_{pq} become improper. These special cases can be classified into two groups:

Type 1: $r_m = r_n$

Type 2: $r_m = 0$

A Type 1 singularity arises when two nodal points, m and n, have the same value of r (for instance, $r_k = r_j$). This means that the corresponding value of c_i (Eq. (3)) becomes zero. Since c_i appears in the denominator of several terms of I_{pq}, these expressions will become infinite or at least improper. However, from Fig. 2 it can be seen that in this case the sub-integral over the area k j j′ k′ does not exist, which means that $I_{11} = I_{21} = I_{31} = 0$.

A singularity of Type 2 is encountered in the case of $r_m = 0$. Those terms of I_{pq} which become improper for $r_m = r_i = 0$ can be reduced to the following form:

$$T_{pi} = \left(\frac{a_j^p}{c_j^p} - \frac{a_k^p}{c_j^p}\right)\mathrm{Lnr}_i \qquad p = 1,2,3 \qquad (16)$$

The evaluation of T_{pi} for $r_i = 0$ leads to an indefinite expression of the form $0 \cdot \infty$; however, by applying L'Hospital's Rule it can be shown that the limit of all T_{pm} terms is zero.

Although it is possible that both types of singularities occur simultaneously ($r_m = r_n = 0$), this condition requires no special attention since the conclusions drawn previously are still valid in this case.

The foregoing paragraphs show that the evaluation of the element stiffness matrix for axi-symmetric triangular elements is relatively complicated compared to the CST (constant strain triangle) element in plane stress problems. The reason for this lies in the difficulties encountered in generating the $\underline{k}_2$ - matrix.

Therefore, a number of developers of Finite Element programs tried to circumvent these difficulties by using certain approximations (Refs. 4,5,6,7).

The simplest way of doing this is by assuming $\underline{k}_2 = 0$ (Ref. 4). Recalling that $\underline{k}_2$ takes into account the variation of ε_θ inside the element, this approximation is certainly justified for large values of r_m. On the other hand, it can be shown that the elements of $\underline{k}_2$ are of the same magnitude as those of $\underline{k}_1$ for elements in the vicinity of the axis of rotation. Therefore, the use of this approximation may cause large inaccuracies.

It was shown that the existence of certain special conditions, for which a number of sub-integrals become singular or do not exist,

eliminates the possibility of simplifying the expressions for the integrals $F_1,\ldots,F_3$.

To circumvent this difficulty it was suggested (Ref. 5) to assign a small quantity to those values that cause the integrals to become singular. Test runs show that the approximation $r_m = \varepsilon$ instead of $r_m = 0$ gives results which seem to converge to the true value. On the contrary, the approximation $r_m - r_n = \varepsilon$ results in an excessive error build-up even for computers with a relatively long word length. For this reason, it seems to be questionable whether an approximation of this type should be used.

Another very common method of evaluating the element stiffness matrix for axi-symmetric triangular elements is the use of numerical integration techniques (Refs. 6,7). Obviously, this procedure gives only approximate results, regardless of whether those special conditions for the nodal point coordinates are met or not. In general, it appears to be doubtful whether numerical integration has a distinct advantage as far as execution time and storage space is concerned, when compared with the exact evaluation procedure.

2.6 Global Stiffness Matrix

One of the major problems in a Finite Element program is the assemblage of the global stiffness matrix $\underline{K}$ of the structure. A detailed description of this procedure is given in Ref. 1.

2.7 Auxiliary Matrices

In order to make the analytic formulation of the Finite Element method complete, two additional relationships have to be established.

Since the AXCST program includes the analysis of thermal strain problems, expressions have to be derived to transform the thermal strain loading condition $\underline{\varepsilon}^t$ to equivalent nodal point loads $\underline{F}^t$.

$$\underline{\varepsilon}^t = \begin{bmatrix} \varepsilon_r^t \\ \varepsilon_z^t \\ \varepsilon_\theta^t \\ \gamma_{r\theta}^t \end{bmatrix} = \begin{bmatrix} \alpha\, t \\ \alpha\, t \\ \alpha\, t \\ 0 \end{bmatrix} \tag{17}$$

By applying the principle of virtual work to this problem the following relationship can be obtained:

$$\begin{aligned} \underline{F}^t &= \int_V \underline{B}^T \underline{D}\, \underline{\varepsilon}^t \, dV \\ &= 2\pi \int_\Delta \left(\underline{B}_1^T + \underline{B}_2^T \right) \underline{D}\, \underline{\varepsilon}^t \, r \, dr \, dz \end{aligned} \tag{18}$$

Recalling that $\int_\Delta \underline{B}_2^T \, r \, dr \, dz = 0$, this expression can be simplified:

$$\begin{aligned} \underline{F}^t &= 2\pi \, \underline{B}_1^T \underline{D}\, \underline{\varepsilon}^t \int r \, dr \, dz \\ &= 2\pi \, \underline{B}_1^T \underline{D}\, \underline{\varepsilon}^t \, r_c \, |\Delta| \end{aligned} \tag{19}$$

After evaluating $\underline{B}_1^T \underline{D} \underline{\varepsilon}^t$ explicitly the nodal point loads $\underline{F}^t$ finally can be written as

$$\underline{F}_t = \frac{E}{(1-2\nu)} \cdot \alpha \cdot t \pi r_c \cdot \frac{|\Delta|}{\Delta} \begin{bmatrix} b_i + p_c \\ c_i \\ b_j + p_c \\ c_j \\ b_k + p_c \\ c_k \end{bmatrix} \tag{20}$$

As a last step in a Finite Element analysis, the elemental stresses have to be expressed in terms of nodal point displacements. For this purpose the following relationship has to be used:

$$\begin{aligned} \underline{\sigma} &= \underline{D} \underline{B} \underline{\delta}^e - \underline{D} \underline{\varepsilon}^t \\ &= \underline{D} (\underline{B} \underline{\delta}^e - \underline{\varepsilon}^t) \end{aligned} \tag{21}$$

3. GENERAL PROGRAM DESCRIPTION

3.1 Application to Boundary Value Problems

The AXCST program can be used for solving the following type of elasticity problems:

(a) Surface Forces, (S)

(b) Body Surfaces, (F)

(c) Displacement Boundary Conditions

(d) Thermal Strain

It is understood, of course, that these loading conditions are axi-symmetrical.

As a characteristic of the Finite Element Method, all force and displacement boundary conditions have to be prescribed at the nodal points, as nodal point loads or as nodal point displacements. This makes it possible to prescribe displacements not only at the surface, but also in the interior of the body.

To keep the program as general as possible no attempt was made to incorporate the evaluation of nodal point loads due to distributed surface or body forces. The procedure for obtaining the nodal point loads due to arbitrary distributed forces will be outlined in Section 3.2.

Only in the case of thermal strain problems the program will generate the nodal point loads using data for the coefficient of thermal expansion and the temperature change, which have to be prescribed for every element. Again, more specific features like temperature

interpolation or temperature variation within the element was not included, but can be handled by assigning nodal point loads directly.

3.2 Nodal Point Loads due to Distributed Forces

The evaluation of nodal point loads due to distributed surface forces is commonly done by applying the principle of virtual work (Refs. 5,8).

Using the following notation

S_n = nodal point load due to distributed surface forces

$\bar{\Delta}_n$ = virtual displacement at nodal point n in the direction of S_n

$\underline{f}^T = \begin{bmatrix} f_r\ (r,z) \\ f_z\ (r,z) \end{bmatrix}$ = distributed surface forces

$\underline{\bar{\Delta s}} = \begin{bmatrix} \bar{u}_s\ (r,z) \\ \bar{v}_s\ (r,z) \end{bmatrix}$ = virtual displacement of the surface s

$\int_s ds$ = integral over the element surface

the nodal point load can be written as

$$S_n \cdot \bar{\Delta}_n = \int_s \underline{f}^T\, \underline{\bar{\Delta s}}\, ds \tag{22}$$

By assuming $\bar{\Delta}_n = 1$ it follows

$$S_n = \int_s \left[f_r \cdot \bar{u}_s + f_z \cdot \bar{v}_s \right] ds \tag{23}$$

Here $\bar{u}_s$ and $\bar{v}_s$ represent a compatible displacement field, having a unit displacement at nodal point n in the direction of S_n.

To illustrate this procedure the nodal point loads due to uniform pressure acting on a plane z_s = constant will be evaluated (Fig. 3).

From Eq. (23) follows

$$S_{i_z} = \int_s \left[f_r \cdot \bar{u}_s + f_z \bar{v}_s \right] ds \tag{24}$$

Since $f_r\ (r,z) = 0$ and $f_z\ (r,z) = -p_z$ on surface s, Eq. (25) can be rewritten as

$$S_{iz} = -p_z \int_s \bar{v}_s\ ds \tag{25}$$

A compatible displacement function $\bar{v}_s$ with $\bar{v}_i = 1$ can be obtained from Eq. (2)

$$\bar{v}_s\ (r,z_s) = \frac{1}{2\Delta} \begin{bmatrix} 1 \\ r \\ z_s \end{bmatrix}^T \begin{bmatrix} a_i & a_j & a_k \\ b_i & b_j & b_k \\ c_i & c_j & c_k \end{bmatrix} \begin{bmatrix} \bar{v}_i = 1 \\ 0 \\ 0 \end{bmatrix}$$

$$= \frac{1}{2\Delta} \left(a_i + b_i r + c_i z_s \right) \tag{26}$$

Since the surface integral $\int_s ds$ is independent of z in this particular case, the following simplification can be made:

$$\int_s ds = 2\pi \int r\, dr \tag{27}$$

Using Eqs. (25), (26), and (27) the nodal point load S_{iz} can be written as

$$S_{iz} = -\frac{2\pi p_z}{2\Delta} \int_{r_i}^{r_k} \left[(a_i + c_i z_s)\, r + b_i\, r^2 \right] dr \tag{28}$$

After executing this integration and replacing a_i, b_i, c_i, and 2Δ, according to Eq. (3), the following expression for S_{iz} will be obtained:

$$S_{iz} = -\frac{\pi p_z}{3} (r_k - r_i)\, (r_k + 2r_i) \tag{29a}$$

Using the same procedure to evaluate S_{jz} and S_{kz} yields

$$S_{kz} = -\frac{\pi p_z}{3} (r_k - r_i)\, (r_i + 2r_k) \tag{29b}$$

$$S_{jz} = 0 \tag{29c}$$

Another very common type of loading, uniform pressure on a surface with r_s = constant (Fig. 4), will result in the following expressions for the nodal point loads:

$$S_{ir} = S_{jr} = -\pi p_r r_s\, (z_j - z_i) \tag{30a,b}$$

$$S_{kr} = 0 \tag{30c}$$

The generality of this procedure makes it possible to evaluate nodal point loads for arbitrary surface forces, although the expressions become rather involved for more complicated cases. This procedure is also applicable to body forces if the integration is executed over the volume of the element instead of its surface.

3.3 Solution of Simultaneous Equations

The equilibrium equation for the total system of elements can be written as

$$\underline{F} = \underline{K}\,\underline{\delta} \tag{31}$$

where $\underline{K}$ = global stiffness matrix

$\underline{F}$ = nodal force vector

$\underline{\delta}$ = nodal point displacement vector

N = total number of nodal points

The vectors $\underline{F}$ and $\underline{\delta}$ consist of N sub-vectors $\underline{F}_n$ and $\underline{\delta}_n$, which contain the force and displacement components for each nodal point.

$$\underline{F}_n = \begin{bmatrix} F_{r_n} \\ F_{z_n} \end{bmatrix} \qquad \underline{\delta}_n = \begin{bmatrix} u_n \\ v_n \end{bmatrix} \tag{32}$$

In the AXCST program an over-relaxation type of approach is used for solving the large system of simultaneous equations (Refs. 1,9). The basic formula for this iterative procedure can be written as:

$$\underline{\delta}_n^{(j+1)} = \underline{\delta}_n^{(j)} + \beta\, \underline{k}_{nn}^{-1}\left[\underline{F}_n - \sum_{i=1}^{n-1} \underline{k}_{ni}\, \underline{\delta}_i^{(j+1)} - \sum_{i=n+1}^{N} \underline{k}_{ni}\, \delta_i^{(j)}\right] \tag{33}$$

where β = over-relaxation factor (1.85)

j = number of iteration cycle

$\underline{k}_{ni}$ = (2x2) sub-matrix of global stiffness matrix

$\underline{k}_{nn}^{-1}$ = nodal point flexibility matrix

The use of this over-relaxation method offers a number of distinct advantages:

(a) The large but sparsely populated global stiffness matrix can be stored very effectively, that is, for a given storage space a maximum number of nodal points can be handled.

(b) The error build-up and propagation, a very serious problem for computers with short word length, is reduced to a minimum.

(c) No restrictions concerning the "band-width" have to be imposed on the enumeration of the nodal points.

On the other hand, iterative methods of this type tend to require more execution time as compared to direct methods of solving simultaneous equations. This is particularly true for certain types of problems, where the convergence of the solution presents difficulties. However, the same reasons that cause a slow convergence of the iterative solution

might cause an increased error build-up for direct methods. This error remains undetectable unless elaborate error analysis is performed.

3.4 Displacement Boundary Conditions

From Eq. (33) the displacement increment per iteration cycle can be written as

$$\Delta\delta_n^{(j)} = \underline{k}_{nn}^{-1} \left[F_n - \sum_{i=1}^{n-1} \underline{k}_{ni}\, \delta_i^{(j+1)} - \sum_{j=n+1}^{N} \underline{k}_{ni}\, \delta_i^{(j)} \right] \tag{34}$$

or

$$\Delta\delta_n^{(j)} = \underline{k}_{nn}^{-1} \cdot \underline{P}_n^{(j)} \tag{35a}$$

$$\begin{bmatrix} \Delta u_n \\ \Delta v_n \end{bmatrix}^{(j)} = \begin{bmatrix} f_{rr} & f_{rz} \\ f_{zr} & f_{zz} \end{bmatrix} \begin{bmatrix} P_r \\ P_z \end{bmatrix}^{(j)} \tag{35b}$$

$\underline{P}_n$ denotes here the difference between the nodal point load $\underline{F}_n$ and those forces caused by the displacement of adjacent nodal points. $\underline{P}_n$ is commonly referred to as "unbalanced force".

Whereas Eq. (35) is valid for all nodal points without restraints, certain modifications of the nodal point flexibility matrix $\underline{k}_{nn}^{-1}$ are necessary for those points where displacement boundary conditions are prescribed (Ref. 9).

In general, two types of restraints have to be distinguished:

(1) the nodal point is fixed in all directions

(2) the nodal point is free to move along a line with an angle φ

Denoting the support reactions with $\underline{S}_n$, Eq. (35) can be rewritten for a restrained nodal point as follows:

$$\begin{bmatrix} \Delta u_n \\ \Delta v_n \end{bmatrix} = \begin{bmatrix} f_{rr}^* & f_{rz}^* \\ f_{zr}^* & f_{zz}^* \end{bmatrix} \begin{bmatrix} P_r + S_r \\ P_z + S_z \end{bmatrix} \tag{36}$$

For a Type 1 restraint the displacement increments $\underline{\Delta\delta}_n$ are identical zero. It is apparent that this condition can only be met by setting the modified stiffness matrix $\underline{k}_{nn}^{*-1}$ equal to zero.

$$f_{rr}^* = f_{rz}^* = f_{zr}^* = f_{zz}^* = 0 \tag{37}$$

For a Type 2 restraint (Fig. 5) Eq. (36) reads:

$$\begin{bmatrix} \Delta u_n \\ \Delta u_n \cdot \tan\varphi \end{bmatrix} = \begin{bmatrix} f_{rr}^* & f_{rz}^* \\ f_{zr}^* & f_{zz}^* \end{bmatrix} \begin{bmatrix} P_r - S \cdot \sin\varphi \\ P_z + S \cdot \cos\varphi \end{bmatrix} \tag{38}$$

Eliminating the unknown support reaction S results in the following expressions for $\underline{k}_{nn}^{*-1}$

$$\begin{aligned} f_{rr}^* &= \frac{f_{rr} - \alpha f_{rz}}{1 - \alpha \tan\varphi} & f_{rz}^* &= \frac{f_{rz} - \alpha f_{yy}}{1 - \alpha \tan\varphi} \\ f_{zr}^* &= f_{rr}^* \tan\varphi & f_{zz}^* &= f_{rz}^* \cdot \tan\varphi \end{aligned} \tag{39}$$

where

$$\alpha = \frac{f_{rr} \tan\varphi - f_{rz}}{f_{rz} \tan\varphi - f_{zz}}$$

Special attention has to be paid to the case where φ equals $\pi/2$. This corresponds to the case where nodal point is free to move in z - direction. Since $\tan\varphi$ is infinite in this case, Eq. (38) has to be rewritten

$$\begin{bmatrix} 0 \\ \Delta v_n \end{bmatrix} = \begin{bmatrix} f_{rr}^* & f_{rz}^* \\ f_{zr}^* & f_{zz}^* \end{bmatrix} \begin{bmatrix} P_r + S_r \\ P_z \end{bmatrix} \tag{40}$$

Expressing S_r in terms of P_r and P_z yields

$$f_{rr}^* = f_{rz}^* = f_{zr}^* = 0$$

$$f_{zz}^* = f_{zz} - \frac{f_{rz}^2}{f_{rr}}$$

3.5 General Description of Input

The minimum amount of information that has to be provided for a Finite Element analysis includes the following items:

1) Control Variables

   Number of nodal points, number of elements, number of restrained nodal points, control variables for output. In case an iterative method for solving the system of simultaneous equations is used, a convergence limit and/or a maximum number of iteration cycles has to be provided.

2) Element Data

   Nodal point numbers i,j,k
   Material properties (Young's modulus, Poisson's ratio)

3) Nodal Point Data

   Coordinates of nodal point.
   Nodal point loads.

4) Restrained Nodal Point Data

Number of restrained nodal point.
Type of restraint and direction in which the nodal point is free to move.

In the AXCST program two features were incorporated which require additional input information. These features are:

1) The program will generate nodal point loads due to thermal strain. For this purpose the coefficient of thermal expansion and the temperature change has to be provided for each element.

2) The program will read in initial values for the nodal point displacements. These values have the character of initial guesses for unrestrained nodal points, whereas they are treated as specified displacements for restrained nodal points.

It is possible to incorporate numerous other features in this program. Of particular interest would be program parts for generating nodal point loads due to various types of surface and body forces. However, to keep the input as general and at the same time as simple as possible, no provisions of this type were made. For the same reason all input data have to be punched on separate cards for each element or nodal point respectively. It is apparent that for specific types of problems the amount of input data could be greatly reduced. In those cases it is left to the users to modify the input of this program according to the specific purpose.

3.6 General Description of Output

The Finite Element analysis of elasticity problems provides the following numerical results:

1) Nodal point displacements

2) Element strains and stresses

These nodal point displacements define the displacements throughout the body uniquely since the linear displacement function used in the AXCST program satisfies compatibility along the edges of the elements (Section 2.1). However, the element stresses and strains will generally not match along the interface of two adjacent elements.

From comparisons with elasticity problems, of which an exact solution is known, it was found that the best numerical results are obtained by assigning the element stresses and strains to the center of the element. To obtain stress and strain values also for the nodal points a number of authors applied certain weighted average procedures. Unfortunately, the criteria used by different authors to determine the weighting coefficients contradict each other in a certain way.

A second problem arises in connection with nodal point stresses. In many elasticity problems the maximum stresses occur at the boundary of the body rather than in its interior. In this case the stresses at the boundary points will be biased noticeably, since their value is based on the average of the interior element stresses, not on their extrapolated values. Therefore, it seems to be doubtful whether the nodal point stresses should be used at all as a result of the Finite Element analysis.

The AXCST program provides the following output:

1) Input Data

Control variables
Element data (including center coordinates)
Nodal point data
Restrained nodal point data

2) Force Unbalance (printed in specified cycle intervals)

3) Displacement and Stresses (printed in specified cycle intervals)

Number of cycle, corresponding force unbalance
Nodal point displacements
Element stresses (σ_r, σ_z, σ_θ, τ_{rz}, σ_1, σ_2, direction of σ_1)

In case there are distinct reasons to use nodal point stresses as a result of the analysis, their values can be calculated and printed out by specifying the proper control variables (see Section 4.1).

4. USER'S GUIDE

4.1 Input Information

A. Header Card - FORMAT (1X,52H.....)

B. Control Card - FORMAT (6I4,6X,E10.3)

Cols.	Input
1 - 4	Number of elements
5 - 8	Number of nodal points
9 - 12	Number of restrained boundary points
13 - 16	Print interval of force unbalance
17 - 20	Print interval of displacements and stresses
21 - 24	Maximum number of cycles
31 - 40	Convergence limit for unbalanced force
50	0 - output of nodal point stresses suppressed 1 - output of nodal point stresses executed

C. Element Data - 1 Card per element FORMAT (3I4,8X,4F10.0)

Cols.	Input
1 - 4	Nodal point number i
5 - 8	Nodal point number j
9 - 12	Nodal point number k
21 - 30	Modulus of elasticity
31 - 40	Poisson's ratio
41 - 50	Coefficient of thermal expansion
51 - 60	Temperature change

D. Nodal Point Data - 1 Card per nodal point FORMAT (6F10.0)

Cols.	Input
1 - 10	R - Ordinate
11 - 20	Z - Ordinate
21 - 30	R - Load
31 - 40	Z - Load
41 - 50	R - Displacement (*)
51 - 60	Z - Displacement (*)

Note: (*) On free nodal points these are initial guesses; on restrained nodal points these are specified displacements.

E. Boundary Point Data - 1 Card per boundary point
FORMAT (2I4,2X,F10.0)

Cols.	
1 - 4	Nodal point number
8	0 - nodal point is fixed in both directions 1 - nodal point is fixed in R direction 2 - nodal point is free to move along a line of slope S

Cols.	Input
11 - 20	Slope S (type 2 boundary point only)

Note: Cards of type C and type D have to be placed in their natural sequence. To avoid errors the element or nodal point number should be punched in columns 71 - 80.

4.2 Assumptions, Limitations, Remarks

A. Limitations

1) Maximum number of elements: 600

2) Maximum number of nodal points: 310

3) Maximum number of nodal points adjacent to a certain point: 8

4) Elements and nodal points have to be numbered in natural sequence (data cards have to be placed in the same order).

B. Assumption

Over-relaxation factor: 1.85

C. Initial guesses for displacements

For boundary condition type 2 (nodal point is forced to move along a line of slope S): Initial guesses of displacements perpendicular to the prescribed slope will be treated as imposed displacements.

D. Direction of principal stresses

The calculated value is the angle between the positive R axis and the direction of the maximum positive (or minimum negative) principal stress. In case that both values $(\sigma_r - \sigma_z)$ and τ_{rz} are very small this angle becomes meaningless.

E. Required field length

CM - $100{,}000_8$

By using CM - $140{,}000_8$ the program can handle up to 530 nodal points and 1000 elements.

5. REFERENCES

1. Schultchen, E. G. and Kostem, C. N.
USER'S MANUAL FOR CSTES FINITE ELEMENT PROGRAM, Lehigh University, Fritz Engineering Laboratory Report No. 237.58, 1969.

2. Timoshenko, S. and Goodier, J. N.
THEORY OF ELASTICITY, McGraw-Hill Book Company, New York, 1951.

3. Zienkiewicz, O. C.
THE FINITE ELEMENT METHOD IN STRUCTURAL AND CONTINUUM MECHANICS", McGraw-Hill Book Company, New York, 1967.

4. Tanaka, H. and Kawamata, S. et al.
RECENT RESEARCHES OF STRUCTURAL MECHANICS, UNO SHOTEN, Tokyo, 1968.

5. Gallagher, R. M., Editor
FINITE ELEMENT ANALYSIS, Course Notes, Cornell University, 1969.

6. Jones, R. M. and Crose, J. G.
SAAS II - FINITE ELEMENT STRESS ANALYSIS OF AXISYMMETRIC SOLIDS, Aerospace Corporation, Aerospace Report No. TR-0200 (S4980) -1, 1968.

7. Meek, J. L. and Carey, G.
ANALYSIS BY COMPUTER - AXISYMMETRIC SOLUTION OF ELASTO-PLASTIC PROBLEMS BY FINITE ELEMENT METHODS, University of Queensland, Bulletin No. 11, 1969.

8. Clough, R. W. and Rashid, Y.
FINITE ELEMENT ANALYSIS OF AXI-SYMMETRIC SOLIDS, Journal of the Engineering Mechanics Division, Vol. 91, No. EM1, February 1965.

9. Wilson, E. L.
FINITE ELEMENT ANALYSIS OF TWO-DIMENSIONAL STRUCTURES, Ph.D. Dissertation, University of California, Berkeley, 1963.

6. PROGRAM LISTING

```
      PROGRAM AXCST (INPUT,TAPE1=INPUT,OUTPUT,TAPE2=CUTPUT)                AXCST  1
C*                                                                         AXCST  2
C*    FINITE ELEMENT ANALYSIS OF AXI-SYMMETRIC SOLIDS                      AXCST  3
C*                                                                         AXCST  4
      DIMENSION NPI(600),NPJ(600),NPK(600),E(600),P(600),AL(600),DT(600)   AXCST  5
      DIMENSION TH(600),SR(600),SZ(600),ST(600),TA(600)                    AXCST  6
      DIMENSION R(310),Z(310),RLOAD(310),ZLOAD(310),DR(310),DZ(310)        AXCST  7
      DIMENSION SRR(310,9),SRZ(310,9),SZR(310,9),SZZ(310,9),NN(310,9)      AXCST  8
      DIMENSION FRR(310),FRZ(310),FZR(310),FZZ(310)                        AXCST  9
      DIMENSION B(6,6),D(6,6),S(6,6)                                       AXCST 10
      DIMENSION AA(3),BB(3),CC(3),AO(3),RA(3),RB(3),LL(3)                  AXCST 11
      EQUIVALENCE (AL(600),TH(600)),(DT(600),SR(600))                      AXCST 12
      IN=1                                                                 AXCST 13
      IO=2                                                                 AXCST 14
      DATA RELA/1.85/                                                      AXCST 15
C*                                                                         AXCST 16
C*    INPUT                                                                AXCST 17
C*                                                                         AXCST 18
      WRITE(IO,300)                                                        AXCST 19
      READ(IN,400)                                                         AXCST 20
      WRITE(IO,400)                                                        AXCST 21
      WRITE(IO,301)                                                        AXCST 22
      READ(IN,401) NELE,NNOP,NBOP,INUN,INST,MAXC,TOLE,NPST                 AXCST 23
      WRITE(IO,302) NELE                                                   AXCST 24
      WRITE(IO,303) NNOP                                                   AXCST 25
      WRITE(IO,304) NBOP                                                   AXCST 26
      WRITE(IO,305) INUN                                                   AXCST 27
      WRITE(IO,306) INST                                                   AXCST 28
      WRITE(IO,307) MAXC                                                   AXCST 29
      WRITE(IO,308) TOLE                                                   AXCST 30
      WRITE(IO,309) RELA                                                   AXCST 31
      READ(IN,402) (NPI(M),NPJ(M),NPK(M),E(M),P(M),AL(M),DT(M),M=1,NELE)   AXCST 32
      READ(IN,403) (R(N),Z(N),RLOAD(N),ZLOAD(N),DR(N),DZ(N),N=1,NNOP)      AXCST 33
      WRITE(IO,310)                                                        AXCST 34
      WRITE(IO,311) (N,R(N),Z(N),RLOAD(N),ZLOAD(N),DR(N),DZ(N),N=1,NNOP)   AXCST 35
      WRITE(IO,312)                                                        AXCST 36
      DO 101 N=1,NNOP                                                      AXCST 37
      DO 100 L=1,9                                                         AXCST 38
      SRR(N,L)=SRZ(N,L)=SZR(N,L)=SZZ(N,L)=0.0                              AXCST 39
  100 NN(N,L)=0                                                            AXCST 40
  101 NN(N,1)=N                                                            AXCST 41
C*                                                                         AXCST 42
C*    FORMATION OF ELEMENT STIFFNESS MATRIX                                AXCST 43
C*                                                                         AXCST 44
      PI=4.0*ATAN(1.0)                                                     AXCST 45
      CD=90.0/PI                                                           AXCST 46
      DO 112 M=1,NELE                                                      AXCST 47
      DO 102 II=1,6                                                        AXCST 48
      DO 102 JJ=1,6                                                        AXCST 49
  102 B(II,JJ)=D(II,JJ)=S(II,JJ)=0.0                                       AXCST 50
      I=LL(1)=NPI(M)                                                       AXCST 51
      J=LL(2)=NPJ(M)                                                       AXCST 52
      K=LL(3)=NPK(M)                                                       AXCST 53
      AA(1)=R(J)*Z(K)-R(K)*Z(J)                                            AXCST 54
      AA(2)=R(K)*Z(I)-R(I)*Z(K)                                            AXCST 55
      AA(3)=R(I)*Z(J)-R(J)*Z(I)                                            AXCST 56
      BB(1)=Z(J)-Z(K)                                                      AXCST 57
```

```
      BB(2)=Z(K)-Z(I)                                                  AXCST 58
      BB(3)=Z(I)-Z(J)                                                  AXCST 59
      CC(1)=R(K)-R(J)                                                  AXCST 60
      CC(2)=R(I)-R(K)                                                  AXCST 61
      CC(3)=R(J)-R(I)                                                  AXCST 62
      RC=(R(I)+R(J)+R(K))/3.0                                          AXCST 63
      ZC=(Z(I)+Z(J)+Z(K))/3.0                                          AXCST 64
      WRITE(IO,313) M,I,J,K,E(M),P(M),AL(M),DT(M),RC,ZC                AXCST 65
      ZR=ZC/RC                                                         AXCST 66
      AREA=(BB(2)*CC(3)-BB(3)*CC(2))/2.0                               AXCST 67
      IF(AREA.EQ.0.0) GO TO 135                                        AXCST 68
      ARE=ABS(AREA)                                                    AXCST 69
      TT=2.0*AREA/(3.0*RC)                                             AXCST 70
      PP=1.0-2.0*P(M)                                                  AXCST 71
      TH(M)=E(M)*AL(M)*DT(M)/PP                                        AXCST 72
      THE=PI*RC*TH(M)*SIGN(1.0,AREA)                                   AXCST 73
      DO 103 L=1,3                                                     AXCST 74
      LB=2*L                                                           AXCST 75
      LA=LB-1                                                          AXCST 76
      LC=LL(L)                                                         AXCST 77
      RLOAD(LC)=RLOAD(LC)+THE*(BB(L)+TT)                               AXCST 78
      ZLOAD(LC)=ZLOAD(LC)+THE*CC(L)                                    AXCST 79
      B(1,LA)=B(4,LB)=BB(L)                                            AXCST 80
      B(2,LB)=B(4,LA)=CC(L)                                            AXCST 81
  103 B(3,LA)=TT                                                       AXCST 82
      DC=E(M)*PI*RC/(4.0*ARE*(1.0+P(M)))                               AXCST 83
      DB=DC*2.0/PP                                                     AXCST 84
      DA=DB*(1.0-P(M))                                                 AXCST 85
      DB=DB*P(M)                                                       AXCST 86
      DD=DA/RC                                                         AXCST 87
      D(1,1)=D(2,2)=D(3,3)=DA                                          AXCST 88
      D(1,2)=D(1,3)=D(2,1)=D(2,3)=D(3,1)=D(3,2)=DB                     AXCST 89
      D(4,4)=DC                                                        AXCST 90
      DO 104 II=1,4                                                    AXCST 91
      DO 104 JJ=1,6                                                    AXCST 92
      S(II,JJ)=0.0                                                     AXCST 93
      DO 104 KK=1,4                                                    AXCST 94
  104 S(II,JJ)=S(II,JJ)+D(II,KK)*B(KK,JJ)                              AXCST 95
      DO 105 II=1,4                                                    AXCST 96
      DO 105 JJ=1,6                                                    AXCST 97
  105 D(II,JJ)=S(II,JJ)                                                AXCST 98
      DO 106 II=1,6                                                    AXCST 99
      DO 106 JJ=1,6                                                    AXCST100
      S(II,JJ)=0.0                                                     AXCST101
      DO 106 KK=1,4                                                    AXCST102
  106 S(II,JJ)=S(II,JJ)+B(KK,II)*D(KK,JJ)                              AXCST103
      BLOI=BLOJ=BLOK=0.0                                               AXCST104
      IF(R(I).GT.0.0) BLOI=ALOG(R(I))                                  AXCST105
      IF(R(J).GT.0.0) BLOJ=ALOG(R(J))                                  AXCST106
      IF(R(K).GT.0.0) BLOK=ALOG(R(K))                                  AXCST107
      AO(1)=BLOK-BLOJ                                                  AXCST108
      AO(2)=BLOI-BLOK                                                  AXCST109
      AO(3)=BLOJ-BLOI                                                  AXCST110
      RA(1)=R(K)+R(J)                                                  AXCST111
      RA(2)=R(I)+R(K)                                                  AXCST112
      RA(3)=R(J)+R(I)                                                  AXCST113
      RB(1)=RA(1)*RA(1)-R(K)*R(J)                                      AXCST114
      RB(2)=RA(2)*RA(2)-R(I)*R(K)                                      AXCST115
```

```
      RB(3)=RA(3)*RA(3)-R(J)*R(I)                                       AXCST116
      FA=FB=FC=0.0                                                      AXCST117
      DO 107 L=1,3                                                      AXCST118
      IF(CC(L).EQ.0.0) GO TO 107                                        AXCST119
      AC=AA(L)/CC(L)                                                    AXCST120
      BC=BB(L)/CC(L)                                                    AXCST121
      CO=AC*AC(L)                                                       AXCST122
      BO=BC*BB(L)                                                       AXCST123
      FA=FA+CO+BB(L)                                                    AXCST124
      FB=FB-AC*(CO/2.0+BB(L))-BO*RA(L)/4.0                              AXCST125
      FC=FC+AC*AC*(CO/3.0+BB(L))+BO*(AC*RA(L)/2.0+BC*RB(L)/9.0)         AXCST126
  107 CONTINUE                                                          AXCST127
      FA=FA/AREA-1.0/RC                                                 AXCST128
      FB=FB/AREA-ZR                                                     AXCST129
      FC=FC/AREA-ZR*ZC                                                  AXCST130
      DO 108 II=1,3                                                     AXCST131
      DO 108 JJ=1,3                                                     AXCST132
      IS=2*II-1                                                         AXCST133
      JS=2*JJ-1                                                         AXCST134
  108 S(IS,JS)=S(IS,JS)+DD*(AA(II)*AA(JJ)*FA+(AA(II)*CC(JJ)+AA(JJ)*CC(IIAXCST135
     1))*FB+CC(II)*CC(JJ)*FC)                                           AXCST136
*                                                                       AXCST137
*     FORMATION OF GLOBAL STIFFNESS MATRIX                              AXCST138
*                                                                       AXCST139
      DO 111 II=1,3                                                     AXCST140
      DO 111 JJ=1,3                                                     AXCST141
      LM=LL(II)                                                         AXCST142
      MM=0                                                              AXCST143
  109 MM=MM+1                                                           AXCST144
      IF(MM.GE.10) GO TO 136                                            AXCST145
      IF(NN(LM,MM).EQ.LL(JJ)) GO TO 110                                 AXCST146
      IF(NN(LM,MM).NE.0) GO TO 109                                      AXCST147
  110 NN(LM,MM)=LL(JJ)                                                  AXCST148
      IB=2*II                                                           AXCST149
      IA=IB-1                                                           AXCST150
      JB=2*JJ                                                           AXCST151
      JA=JB-1                                                           AXCST152
      SRR(LM,MM)=SRR(LM,MM)+S(IA,JA)                                    AXCST153
      SRZ(LM,MM)=SRZ(LM,MM)+S(IA,JB)                                    AXCST154
      SZR(LM,MM)=SZR(LM,MM)+S(IB,JA)                                    AXCST155
  111 SZZ(LM,MM)=SZZ(LM,MM)+S(IB,JB)                                    AXCST156
  112 CONTINUE                                                          AXCST157
      DO 114 N=1,NNOP                                                   AXCST158
      NO=10                                                             AXCST159
  113 NO=NO-1                                                           AXCST160
      IF(NN(N,NO).EQ.0) GO TO 113                                       AXCST161
  114 NN(N,1)=NO                                                        AXCST162
      DO 115 N=1,NNOP                                                   AXCST163
      DET=SRR(N,1)*SZZ(N,1)-SRZ(N,1)*SZR(N,1)                           AXCST164
      FRR(N)=SZZ(N,1)/DET                                               AXCST165
      FRZ(N)=-SRZ(N,1)/DET                                              AXCST166
      FZR(N)=-SZR(N,1)/DET                                              AXCST167
  115 FZZ(N)=SRR(N,1)/DET                                               AXCST168
*                                                                       AXCST169
*     DISPLACEMENT BOUNDARY CONDITIONS                                  AXCST170
*                                                                       AXCST171
      IF(NBOP.EQ.0) GO TO 121                                           AXCST172
      WRITE(IO,314)                                                     AXCST173
```

```
      WRITE(IO,315)                                                    AXCST174
      DO 120 L=1,NBOP                                                  AXCST175
      READ(IN,404) NP,NT,SLO                                           AXCST176
      WRITE(IO,316) NP,NT,SLO                                          AXCST177
      IF(NT-1) 118,117,116                                             AXCST178
  116 DET=(FRR(NP)*SLO-FRZ(NP))/(FZR(NP)*SLO-FZZ(NP))                  AXCST179
      COF=1.0-DET*SLO                                                  AXCST180
      FRR(NP)=(FRR(NP)-DET*FZR(NP))/COF                                AXCST181
      FRZ(NP)=(FRZ(NP)-DET*FZZ(NP))/COF                                AXCST182
      FZR(NP)=FRR(NP)*SLO                                              AXCST183
      FZZ(NP)=FRZ(NP)*SLO                                              AXCST184
      GO TO 120                                                        AXCST185
  117 FZZ(NP)=FZZ(NP)-FZR(NP)*FRZ(NP)/FRR(NP)                          AXCST186
      GO TO 119                                                        AXCST187
  118 FZZ(NP)=0.0                                                      AXCST188
  119 FRR(NP)=FRZ(NP)=FZR(NP)=0.0                                      AXCST189
*                                                                      AXCST190
*     ITERATIVE SOLUTION OF SIMULTANEOUS EQUATIONS                     AXCST191
*                                                                      AXCST192
  120 CONTINUE                                                         AXCST193
  121 NCY=0                                                            AXCST194
      IUN=INUN                                                         AXCST195
      IST=INST                                                         AXCST196
      IND=1                                                            AXCST197
  122 WRITE(IO,317)                                                    AXCST198
  123 SUM=0.0                                                          AXCST199
      DO 125 N=1,NNOP                                                  AXCST200
      NUM=NN(N,1)                                                      AXCST201
      FR=RLOAD(N)                                                      AXCST202
      FZ=ZLOAD(N)                                                      AXCST203
      DO 124 L=2,NUM                                                   AXCST204
      M=NN(N,L)                                                        AXCST205
      FR=FR-SRR(N,L)*DR(M)-SRZ(N,L)*DZ(M)                              AXCST206
  124 FZ=FZ-SZR(N,L)*DR(M)-SZZ(N,L)*DZ(M)                              AXCST207
      DDR=FRR(N)*FR+FRZ(N)*FZ-DR(N)                                    AXCST208
      DDZ=FZR(N)*FR+FZZ(N)*FZ-DZ(N)                                    AXCST209
      DR(N)=DR(N)+RELA*DDR                                             AXCST210
      DZ(N)=DZ(N)+RELA*DDZ                                             AXCST211
  125 SUM=SUM+ABS(SRR(N,1)*DDR+SRZ(N,1)*DDZ)+ABS(SZR(N,1)*DDR+SZZ(N,1)*DAXCST212
     1DZ)                                                              AXCST213
      NCY=NCY+1                                                        AXCST214
      IF(NCY.LT.IUN) GO TO 126                                         AXCST215
      IUN=IUN+INUN                                                     AXCST216
      WRITE(IO,318) NCY,SUM                                            AXCST217
  126 IF(SUM.LE.TOLE.OR.NCY.GE.MAXC) IND=2                             AXCST218
      IF(NCY.GE.IST) GO TO 127                                         AXCST219
      GO TO (123,127) IND                                              AXCST220
*                                                                      AXCST221
*     NODAL POINT DISPLACEMENTS                                        AXCST222
*                                                                      AXCST223
  127 WRITE(IO,319) NCY,SUM                                            AXCST224
      WRITE(IO,320)                                                    AXCST225
      DO 128 N=1,NNOP,2                                                AXCST226
      L=N+1                                                            AXCST227
      IF(L.GT.NNOP) L=NNOP                                             AXCST228
  128 WRITE(IO,321) N,DR(N),DZ(N),L,DR(L),DZ(L)                        AXCST229
*                                                                      AXCST230
*     ELEMENT STRESSES                                                 AXCST231
```

```
*                                                                       AXCST232
      WRITE(IO,322)                                                     AXCST233
      WRITE(IO,323)                                                     AXCST234
      IST=IST+INST                                                      AXCST235
      DO 129 M=1,NELE                                                   AXCST236
      I=NPI(M)                                                          AXCST237
      J=NPJ(M)                                                          AXCST238
      K=NPK(M)                                                          AXCST239
      BI=Z(J)-Z(K)                                                      AXCST240
      BJ=Z(K)-Z(I)                                                      AXCST241
      BK=Z(I)-Z(J)                                                      AXCST242
      CI=R(K)-R(J)                                                      AXCST243
      CJ=R(I)-R(K)                                                      AXCST244
      CK=R(J)-R(I)                                                      AXCST245
      AREA=(BJ*CK-BK*CJ)/2.0                                            AXCST246
      TT=2.0*AREA/(R(I)+R(J)+R(K))                                      AXCST247
      ER=BI*DR(I)+BJ*DR(J)+BK*DR(K)                                     AXCST248
      EZ=CI*DZ(I)+CJ*DZ(J)+CK*DZ(K)                                     AXCST249
      ET=TT*(DR(I)+DR(J)+DR(K))                                         AXCST250
      GA=CI*DR(I)+CJ*DR(J)+CK*DR(K)+BI*DZ(I)+BJ*DZ(J)+BK*DZ(K)          AXCST251
      DC=E(M)/(4.0*AREA*(1.0+P(M)))                                     AXCST252
      DB=DC*2.0/(1.0-2.0*P(M))                                          AXCST253
      DA=DB*(1.0-P(M))                                                  AXCST254
      DB=DB*P(M)                                                        AXCST255
      SR(M)=DA*ER+DB*(EZ+ET)-TH(M)                                      AXCST256
      SZ(M)=DA*EZ+DB*(ER+ET)-TH(M)                                      AXCST257
      ST(M)=DA*ET+DB*(ER+EZ)-TH(M)                                      AXCST258
      TA(M)=DC*GA                                                       AXCST259
  129 CALL STRESS (SR(M),SZ(M),ST(M),TA(M),M,CD)                        AXCST260
      GO TO (122,130) INO                                               AXCST261
*                                                                       AXCST262
*     NODAL POINT STRESSES                                              AXCST263
*                                                                       AXCST264
  130 IF(NPST.NE.1) GO TO 500                                           AXCST265
      WRITE(IO,324)                                                     AXCST266
      WRITE(IO,325)                                                     AXCST267
      DO 134 N=1,NNOP                                                   AXCST268
      SSR=SSZ=SST=TTA=FR=FZ=FT=0.0                                      AXCST269
      DO 133 M=1,NELE                                                   AXCST270
      I=NPI(M)                                                          AXCST271
      J=NPJ(M)                                                          AXCST272
      K=NPK(M)                                                          AXCST273
      IF(N.EQ.I) GO TO 132                                              AXCST274
      IF(N.EQ.J) GO TO 131                                              AXCST275
      IF(N.NE.K) GO TO 133                                              AXCST276
      I=NPK(M)                                                          AXCST277
      K=NPI(M)                                                          AXCST278
      GO TO 132                                                         AXCST279
  131 I=NPJ(M)                                                          AXCST280
      J=NPI(M)                                                          AXCST281
  132 QR=ABS(R(J)+R(K)-2.0*R(I))                                        AXCST282
      QZ=ABS(Z(J)-Z(I))+ABS(Z(K)-Z(I))                                  AXCST283
      Q=QR+QZ                                                           AXCST284
      FFR=QR/Q                                                          AXCST285
      FFZ=QZ/Q                                                          AXCST286
      FR=FR+FFR                                                         AXCST287
      FZ=FZ+FFZ                                                         AXCST288
      FT=FT+1.0                                                         AXCST289
```

```
      SSR=SSR+SR(M)*FFR                                                 AXCST290
      SSZ=SSZ+SZ(M)*FFZ                                                 AXCST291
      SST=SST+ST(M)                                                     AXCST292
      TTA=TTA+TA(M)                                                     AXCST293
  133 CONTINUE                                                          AXCST294
      PSR=SSR/FR                                                        AXCST295
      PSZ=SSZ/FZ                                                        AXCST296
      PST=SST/FT                                                        AXCST297
      PTA=TTA/FT                                                        AXCST298
  134 CALL STRESS (PSR,PSZ,PST,PTA,N,CO)                                AXCST299
      GO TO 500                                                         AXCST300
  135 WRITE(IO,326) M                                                   AXCST301
      GO TO 500                                                         AXCST302
  136 WRITE(IO,327) LM                                                  AXCST303
  300 FORMAT(1H1,//)                                                    AXCST304
  301 FORMAT(///,* INPUT  DATA*,///)                                    AXCST305
  302 FORMAT(* NUMBER OF ELEMENTS              =*,I5,/)                 AXCST306
  303 FORMAT(* NUMBER OF NODAL POINTS          =*,I5,/)                 AXCST307
  304 FORMAT(* NUMBER OF BOUNDARY POINTS       =*,I5,/)                 AXCST308
  305 FORMAT(* PRINT INTERVAL UNBALANCE        =*,I5,/)                 AXCST309
  306 FORMAT(* PRINT INTERVAL STRESSES         =*,I5,/)                 AXCST310
  307 FORMAT(* CYCLE LIMIT                     =*,I5,/)                 AXCST311
  308 FORMAT(* TOLERANCE LIMIT                 =*,E13.3,/)              AXCST312
  309 FORMAT(* OVER-RELAXATION FACTOR          =*,F13.3,/)              AXCST313
  310 FORMAT(1H1,//,* POINT*,8X,*R-ORD*,10X,*Z-ORD*,9X,*R-LOAD*,9X,*Z-LOAXCST314
     1AD*,10X,*R-DISP*,11X,*Z-DISP*,//)                                 AXCST315
  311 FORMAT(I4,1X,4F15.5,2F17.8)                                       AXCST316
  312 FORMAT(1H1,//,* ELEMENT*,4X,*I*,7X,*J*,7X,*K*,6X,*E-MODULUS*,6X,*PAXCST317
     1OISSON*,7X,*ALPHA*,8X,*DELTA T*,11X,*R-CENT*,7X,*Z-CENT*,//)      AXCST318
  313 FORMAT(1X,4(I4,4X),E11.3,F12.4,F15.8,F13.3,5X,2F13.5)              AXCST319
  314 FORMAT(1H1,//,* DISPLACEMENT BOUNDARY CONDITIONS*,//)             AXCST320
  315 FORMAT(* POINT*,9X,*TYPE*,13X,*SLOPE*,//)                        AXCST321
  316 FORMAT(I4,13X,I1,F20.6)                                           AXCST322
  317 FORMAT(1H1,//,* CYCLE*,6X,*FORCE UNBALANCE*,/)                    AXCST323
  318 FORMAT(I5,8X,E12.5)                                               AXCST324
  319 FORMAT(1H1,*CYCLE :*,I6,15X,*FORCE UNBALANCE :*,E15.5,///)        AXCST325
  320 FORMAT(//,2(* POINT*,9X,*R-DISPLACEMENT*,6X,*Z-DISPLACEMENT*,18X),AXCST326
     1//)                                                               AXCST327
  321 FORMAT(2(I4,5X,2E20.7,18X))                                       AXCST328
  322 FORMAT(1H1,//,* ELEMENT STRESSES*)                                AXCST329
  323 FORMAT(////,* ELEMENT*,11X,*SIGMA R*,11X,*SIGMA Z*,11X,*SIGMA T*,1AXCST330
     11X,*TAU R-Z*,11X,*SIG-MAX*,11X,*SIG-MIN*,11X,*DIRECTION*,//)      AXCST331
  324 FORMAT(1H1,//,* NODAL POINT STRESSES      (WEIGHTED AVERAGE OF ELEAXCST332
     1MENT STRESSES)*)                                                  AXCST333
  325 FORMAT(////,*  POINT *,11X,*SIGMA R*,11X,*SIGMA Z*,11X,*SIGMA T*,1AXCST334
     11X,*TAU R-Z*,11X,*SIG-MAX*,11X,*SIG-MIN*,11X,*DIRECTION*,//)      AXCST335
  326 FORMAT(1H1,10X,*ZERO AREA                    ELEMENT NUMBER :*,I5)AXCST336
  327 FORMAT(1H1,10X,*MORE THAN 8 NODAL POINTS ADJACENT TO POINT :*,I5) AXCST337
  400 FORMAT(1X,52H                                                    )AXCST338
  401 FORMAT(6I4,E16.3,I10)                                             AXCST339
  402 FORMAT(3I4,8X,4F10.0)                                             AXCST340
  403 FORMAT(6F10.0)                                                    AXCST341
  404 FORMAT(2I4,F12.0)                                                 AXCST342
  500 CALL EXIT                                                         AXCST343
      END                                                               AXCST344
```

```
      SUBROUTINE STRESS (R,Z,T,G,N,CD)                                  AXCST345
C                                                                       AXCST346
C     DIRECTION OF PRINCIPAL STRESSES                                   AXCST347
C                                                                       AXCST348
      IO=2                                                              AXCST349
      SS=(R+Z)/2.0                                                      AXCST350
      SD=R-Z                                                            AXCST351
      SA=SQRT(SD*SD/4.0+G*G)                                            AXCST352
      SMAX=SS+SA                                                        AXCST353
      SMIN=SS-SA                                                        AXCST354
      TT=SIGN(1.0,G)                                                    AXCST355
      IF(SD.NE.0.0) GO TO 100                                           AXCST356
      FI=TT*45.0                                                        AXCST357
      GO TO 101                                                         AXCST358
  100 FI=CD*ATAN(2.0*G/SD)                                              AXCST359
      IF(SD.LT.0.0) FI=TT*90.0+FI                                       AXCST360
  101 WRITE(IO,300) N,R,Z,T,G,SMAX,SMIN,FI                              AXCST361
  300 FORMAT(I5,3X,6F18.4,F18.2)                                        AXCST362
      RETURN                                                            AXCST363
      END                                                               AXCST364
```

7. ACKNOWLEDGEMENTS

Thanks and appreciation are due to Mrs. K. Michele Kostem for her help in the preparation of the report and to Mrs. Jane Lenner for typing and proofreading the manuscript.

8. FIGURES

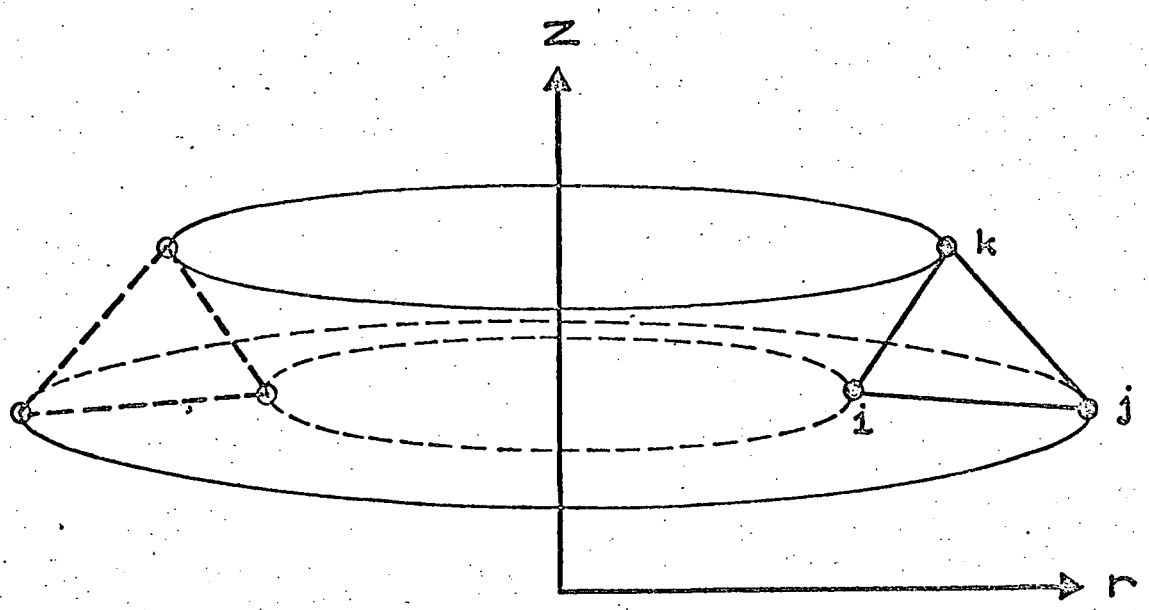


Fig. 1: Triangular Ring Element

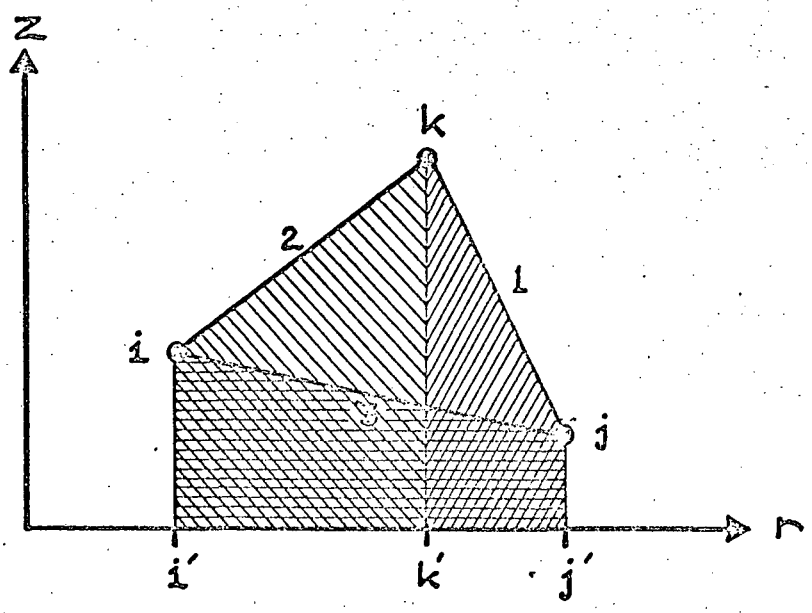


Fig. 2: Integration Scheme

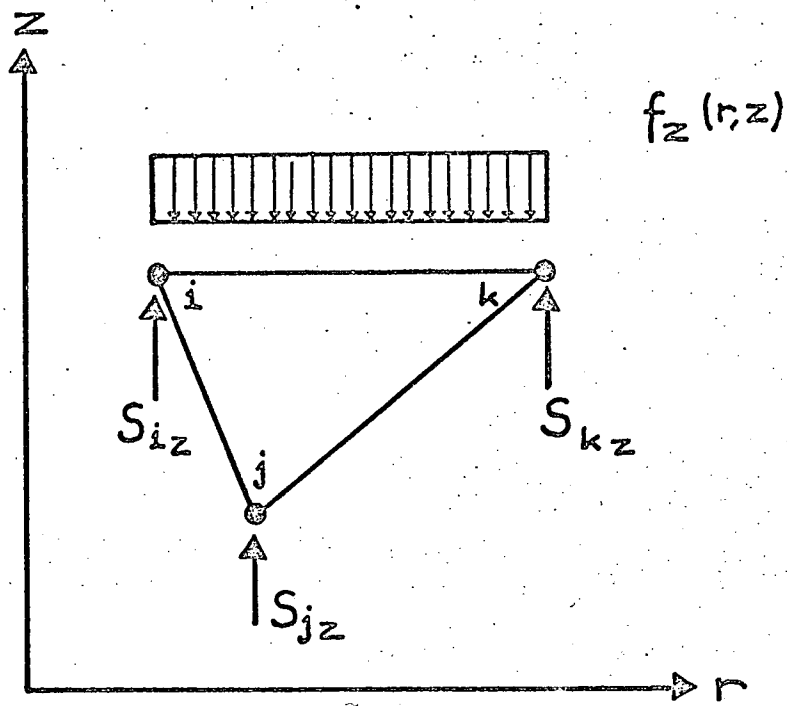


Fig. 3a: Element Geometry and Loading

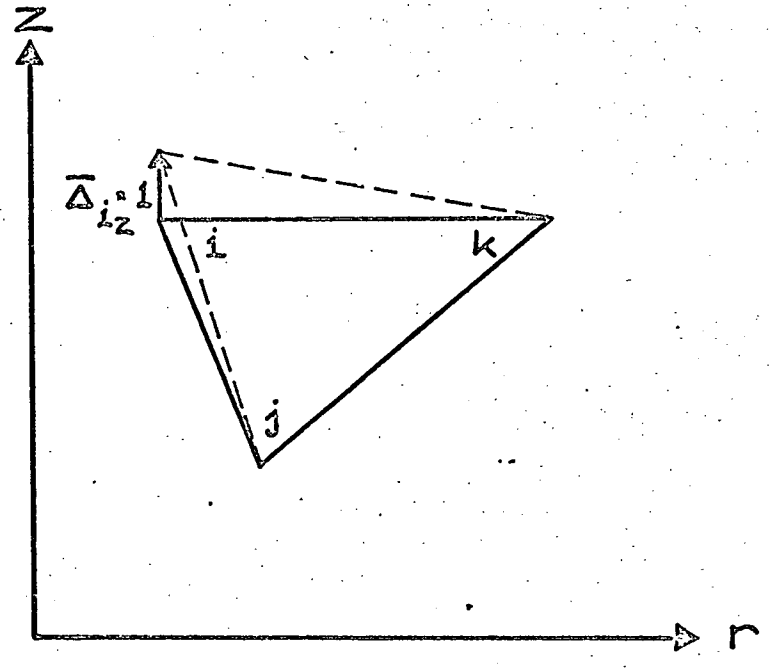


Fig. 3b: Virtual Displacement $\bar{\Delta}_{i_z}$

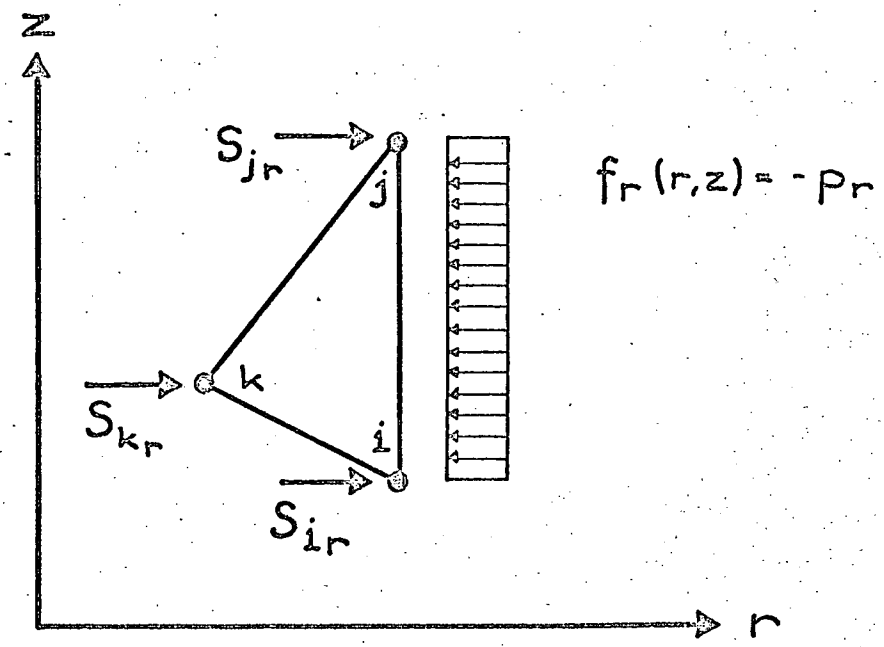


Fig. 4: Element Geometry and Loading

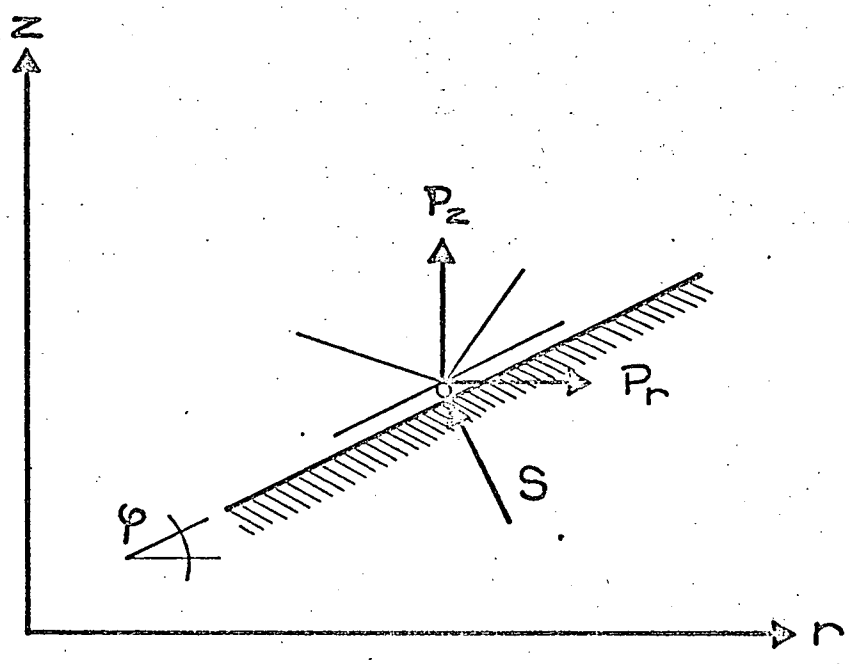
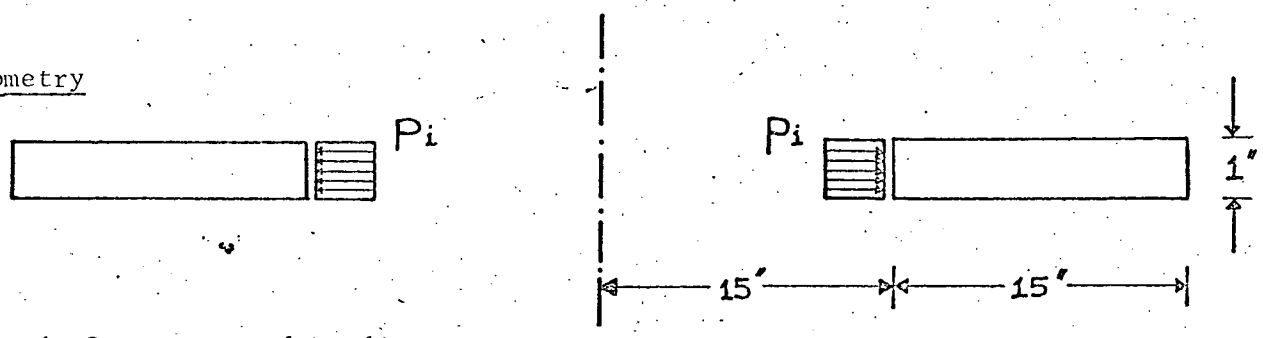


Fig. 5: Restrained Boundary Point

9. APPENDIX

EXAMPLE: Thick-walled cylinder under internal pressure (plane stress solution)

Geometry

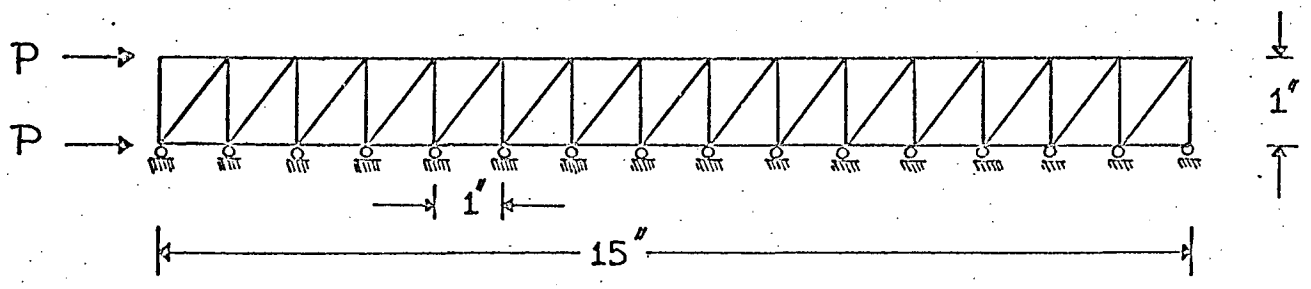


Elastic Constants and Loading

E = 30,000 ksi p_i = 100 ksi

ν = 0.30 P = 4712.39 kip

Element Layout



Number of Elements: 30

Number of Nodal Points: 32

Units

Stresses: ksi

Length, displacements: in.

Note: Solid lines denote exact solution; dots and circles denote Finite Element solution

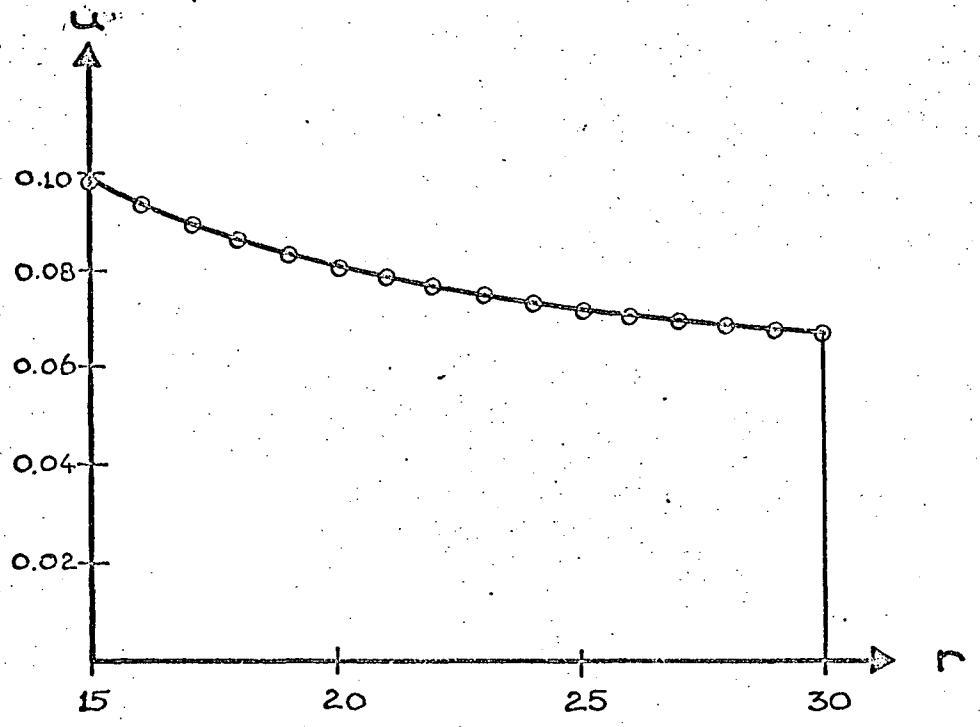


Fig. 1: Nodal Point Displacements

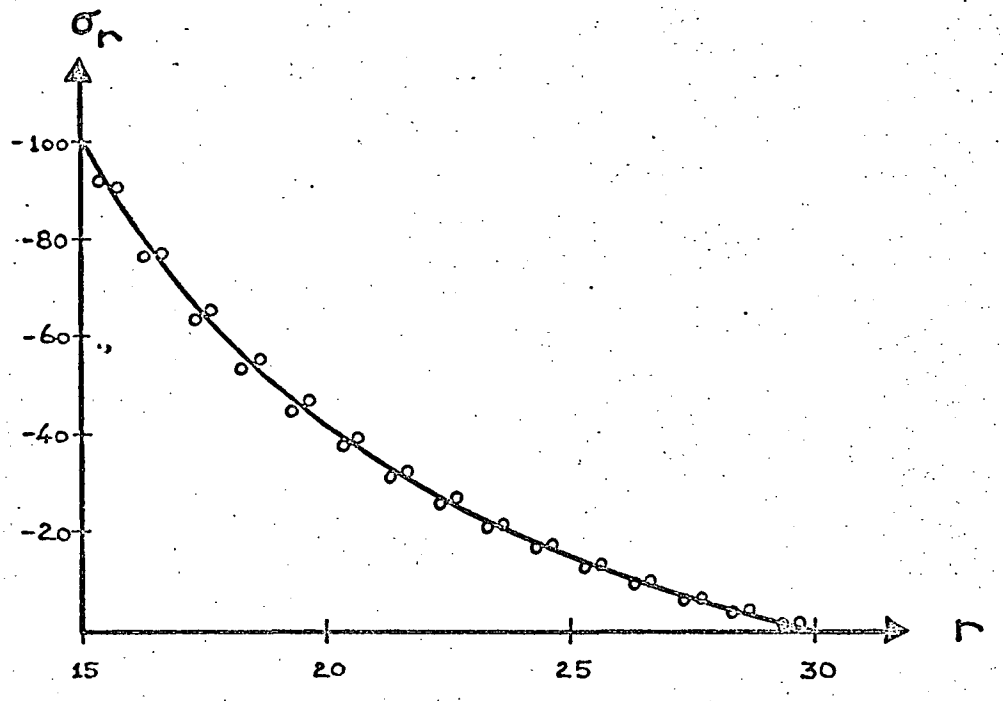


Fig. 2a: Element Stresses σ_r

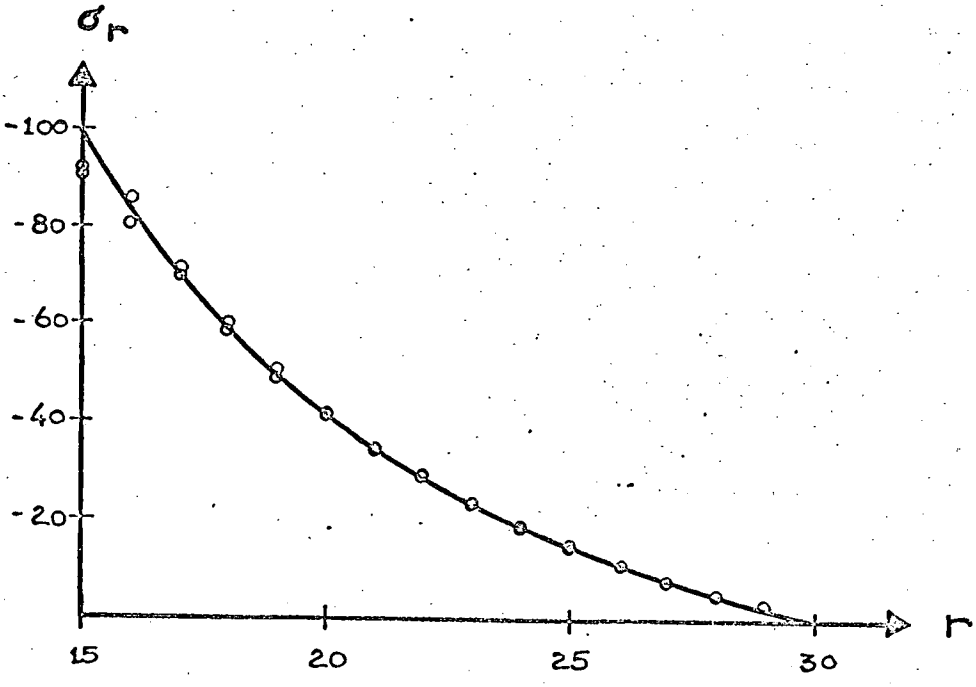


Fig. 2b: Nodal Point Stresses σ_r

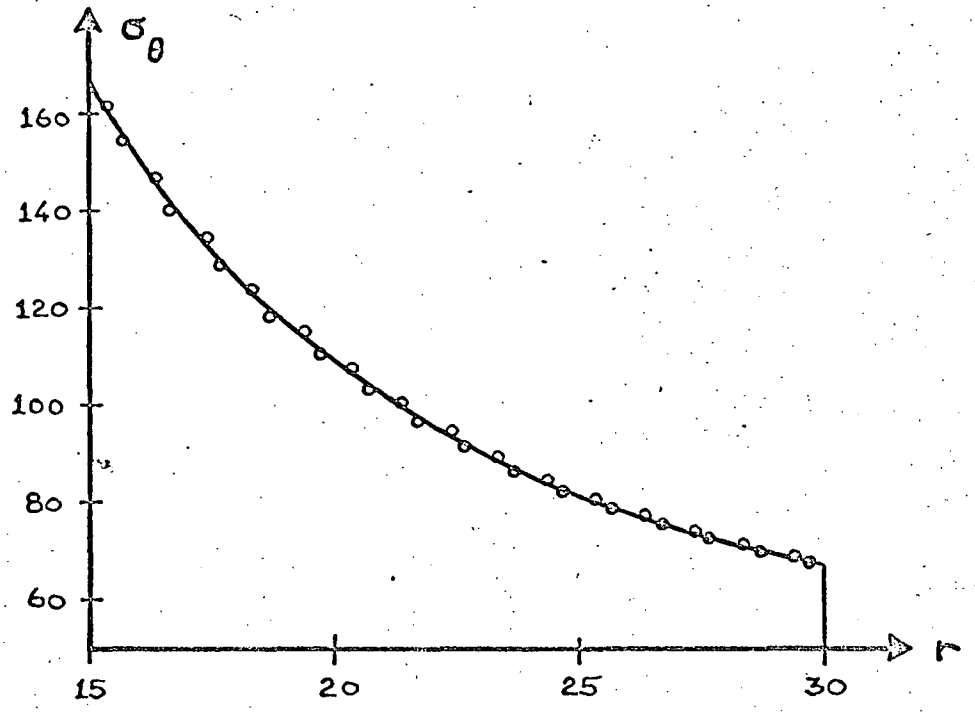


Fig. 3a: Element Stresses σ_θ

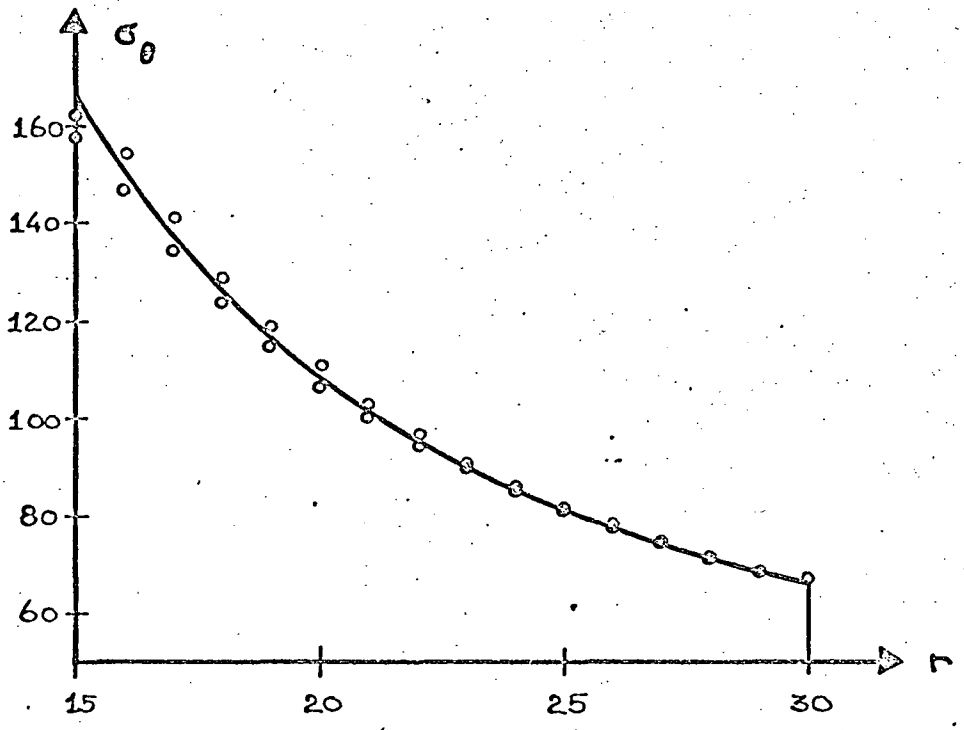


Fig. 3b: Nodal Point Stresses σ_θ