A Further Development of the Boundary Integral Technique for Elastostatics

A thesis submitted for the degree of Doctor of Philosophy in the Faculty of Engineering and Applied Science, University of Southampton.

J.C. Lachat

February 1975

"Comme on l'a souvent dit,
il n'est pas de mathématiques
sans larmes à l'usage des
physiciens et des ingénieurs."

Laurent Schwartz

to Annie
and
Virginie

ACKNOWLEDGMENTS

The author thanks Dr. C.A. Brebbia for his guidance and valuable criticism at all stages of the work. Thanks are due to M. Adamowicz, Directeur Général of the Centre Technique des Industries Mécaniques (CETIM), M. Simon, Directeur Général des Recherches, and M. Chabert, Adjoint au Directeur Général des Recherches, for allowing time to complete the work, to Professor T.A. Cruse for his valuable advice ; to my colleagues J.O. Watson for assistance in programming and M. Dubois for useful discussions, and to Mme Scheuer for typing the thesis.

ABSTRACT

Faculty of Engineering and Applied Science

Jean-Claude Lachat

Doctor of Philosophy

The field equations of plane and three-dimensional elastostatics are transformed, by a general method applicable to any elliptic equation, into boundary integral equations. For the discretisation of these equations, the boundary is represented by three-node segments with quadratic variation of geometry in the two-dimensional case, and by eight-node quadrilaterals and six-node triangles also with quadratic variation of geometry in three dimensions. Over each boundary element, displacement and traction are considered to vary linearly, quadratically or cubically with respect to the intrinsic coordinates. In three dimensions, the elastic body is divided into subregions, for each of which the integral equation is discretised. By specifying continuity and equilibrium across interfaces, a global system of equations is obtained ; this system is of banded form. The subregions may have different elastic properties. Gaussian quadrature formulae are used to evaluate all integrals but those of the product of the strongly singular kernel and shape functions corresponding to the functional singularity ; these integrals and the coefficient of the free term of the integral equation are calculated indirectly, by considering rigid body translations. The program for three-dimensional analysis chooses the order of integration formula for each surface element according to the rapidity of variation of the integrand. Equation coefficients are scaled so that numerical stability is such that the system may be solved by elimination without iteration on the residues. The system is reduced by block solution, all load cases being treated simultaneously. Examples of two and three-dimensional analyses are presented, and comparisons are made with results obtained experimentally and by the finite element method. It is shown that the boundary integral equation method may be used to analyse a wide range of practical problems, and that in most cases it is a relatively economical method of calculation.

CONTENTS

I - INTRODUCTION

During the past fifteen years, various numerical techniques have been developed into efficient tools for the solution of problems of continuum mechanics. In particular, the finite element method is now well established as a means of solving practical problems in many fields, including elasticity, plasticity and fluid mechanics. The method has proved most versatile, but its use does present certain problems. Even where the most sophisticated data generation and checking facilities are available, the cost of preparation of data is very high. The finite element method involves the solution of a large number of simultaneous equations ; even though the matrix is of banded form and symmetric,,the computer time spent in reducing the system is high. In addition most of the information given by a finite element program is never used ; the volume of output is such that the engineer finds difficulty in selecting the results of interest to him. Finally, the displacement method, which is the most commonly used, gives good results for displacements but less accurate results for stresses, which are in most cases of greater interest to the engineer. In particular, the calculated stress field is discontinuous across element interfaces. It is possible that these difficulties associated with the finite element method may be overcome by using the boundary integral equation method.

...

The integral method consists of the transformation of the field equations, which describe the behaviour of the unknown function inside and on the boundary of the domain, to an integral equation, relating the unknown and possibly certain of its derivatives to the given value on the boundary.

Whether it is preferable to solve a partial differential equation subject to boundary conditions, or to solve an integral equation, clearly depends upon the technique available for the solution of the two types of equation. If an analytic solution is envisaged, the transformation of the field equations to an integral equation is of little interest. However, if numerical techniques are to be used, the transformation is worth considering for the following reasons.

For $x \in S$, the boundary of the region under consideration, the integral relationships become the relationships between u and certain of its derivatives, and if one is able to solve these equations, one obtains $u(x), x \in S$. From this information one may calculate directly $u(x), x \in V$ from the integral relationship between displacement at an interior point and the values of displacement and its derivatives on the boundary.

From the point of view of numerical analysis, the advantage is therefore that, since the problem is transformed from one over the domain V to one over the boundary S, it is sufficient for the solution of the problem to discretise only the boundary, and solve the resulting equations. The dimension of the problem is reduced by one (e.g. in elasticity the solution of equations over the body is replaced by the study of behaviour on the surface), which greatly simplifies the use of the computer program, especially the specification of data and interpretation of results, and also reduces the order of the system of simultaneous equations to be solved.

Historically, the integral methods are a development of the method of Fredholm which consists of the application of potential theory in conjunction with the theory of linear integral equations.

Fredholm himself was the first to apply the method to elasticity (35). After this development, numerous publications on the application of Fredholm's equations appeared, but today most of these are considered to be of no interest, because only special problems are analysed, and the mathematical treatment lacks rigor. Until 1950, the elastostatic boundary value problem had not been thoroughly studied except for certain special cases : the first fundamental boundary value problem (displa-

...

cement given on the boundary), the plane problem, axisymmetric problems, and some others. This was due to the fact that the theory of regular integral equations of Fredholm, taken in its classical context, appeared to be inadequate for the treatment of the second fundamental boundary value problem(stress resultant given on the boundary), and the third fundamental boundary value problem(linear or quasi-linear combination of displacement and stress resultant given on the boundary). The system of partial differential equations of elasticity in terms of displacement is an elliptic system, and the problem of integration where the boundary values of displacement are given (first fundamental problem) is a Dirichlet problem, similar to the classical Dirichlet problem for harmonic functions, and so the method of Fredholm may be applied to the first fundamental problem of elastostatics as it may be applied to the Dirichlet problem for harmonic functions. This has been shown by Fredholm.

More recently, Kupradze (20), Kinoshita and Mura (68), have obtained by the method of potentials,integral equations for the first and second fundamental problems. Although these integral equations are singular, the authors apply the theorem of Fredholm;the conclusions are correct but rigorous proofs are lacking.

In 1963, Kupradze (20) reconsidered the boundary value problems, for the more general case of periodic oscillations of

an elastic body, and established proofs of existence and uniqueness for homogeneous and piecewise homogeneous bodies. This work is apparently the first in which rigorous and general proofs, based on multidimensional singular integral equation theory are given.

Kupradze has presented two methods of numerical approximation of the integral equations (20). Rizzo (29) analysed by computer some simple problems of elastostatics in 1967. Rizzo and Shippy (30) investigated in 1968 plane inclusion problems and Cruse and Rizzo (7, 8) generalised their method to elastodynamic analysis, by integral transformation (subject to the condition that a Laplace transform with respect to time is possible). Cruse and Vanburen (43) applied the integral equation method to the three dimensional analysis of a fracture specimen with an edge crack. In 1971, Swedlow and Cruse (46) presented an elastoplastic analysis for anisotropic compressible materials subject to strain-hardening, but gave no numerical examples.

Since then Mendelson (57) has presented an analysis of the elastoplastic problem, accompanied by numerical results. It has therefore been demonstrated that the integral method is applicable to a wide range of problems (dynamics, plasticity), and especially the work of Cruse has shown that complicated geometries and boundary conditions can be modelled.

...

The starting point for the work presented here is the study by Cruse of the two and three-dimensional problems of elastostatics. Cruse considers the boundary integral equation :

$$\frac{1}{2} u_i(x) + \int_S T_{ij}(x,y)\, u_j(y)\, dS = \int_S U_{ij}(x,y)\, t_j(y)\, dS$$

in which S is the surface of the body, x and y are points on the surface, and u and t are the boundary values of displacement and traction respectively. The kernels T and U are derived from the solution of Kelvin's problem of the point load in the infinite elastic space. T is of order $\frac{1}{r^2}$, where r is the distance between x and y and therefore it is necessary to take the Cauchy principal value of the integral on the left side of the equation. U is of order $\frac{1}{r}$ so no such problem arises for the evaluation of the integral on the right side. In each direction, at every point on S, either u or t is known, and the equation becomes an integral equation of the first or second kind for the unknown function. Once the integral equation has been solved, both u and t are known at every point on the surface, and interior values may be calculated by simple integrations.

For the problem of plane elasticity, Cruse represents the boundary by straight line segments and takes as the unknowns u and t at the centre of each segment (6). u and t are supposed to be constant over each segment. ...

In three dimensions, the surface is represented by plane triangles, and again u and t are taken to be constant over each triangle. More recently, Cruse has presented formulations in which u and t are taken to vary linearly over each boundary segment or triangle (53), the unknowns being associated with the extremities or vertices of the surface elements rather than their centroids. In this latter formulation, the coefficient of the free term of the integral equation is different from $\frac{1}{2}$ in the case in which the point x is on an edge or corner. Cruse finds that linear variation is more efficient than the assumption that u and t are constant over each boundary element (53), and suggests that the accuracy of the method could further be improved by considering higher order variation of functions and curved boundary elements.

Where the structure is symmetric with respect to one or more of the coordinate axes, Cruse discretises only part of the surface, so reducing the order of the system of equations to be solved.

Cruse performs the integrations necessary to calculate the matrix coefficients and second member analytically. Whilst this approach may be more efficient than numerical integration in simple cases, it would be very difficult to integrate analytically the products of kernels and higher order variations of u and t over curved boundary elements. In general, the coefficients of u and t appearing in the system of equations are not of the same order, and the matrix is ill-conditioned. Cruse multiplies the ...

coefficients of t by the shear modulus, so obtaining a system which can be solved by Gauss reduction.

To calculate stresses at surface points, one may use directly the integral formula, or calculate the stress tensor from the traction and surface strain. Cruse uses the latter procedure, which is very much more efficient than integration.

The present study is limited to the elastostatic problem in two and three dimensions. A general method, based on the theory of distributions is presented of proceeding from an elliptic operator to the corresponding fundamental solution and boundary integral equation. Improved techniques, based upon parametric representation of geometry and the functions u and t, are used in the numerical formulation. The surface elements are considered in general to be curved, with quadratic variation of geometry with respect to the intrinsic coordinates, as for the finite elements of Zienkiewicz and Ergatoudis (65), (67). Displacement and traction are considered to vary linearly, quadratically or cubically with respect to the intrinsic coordinates. Gaussian integration formulae are used, and the coefficient of the free term of the integral equation is calculated numerically. For construction of the system of equations, the unit of distance is taken to be the greatest dimension of the elastic body, and that of stress is taken to be the modulus of elasticity. The system is expressed in nondimen-

...

sional form, and may be expected to be numerically stable. The matrix usually obtained by the integral equation method is fully populated, a characteristic which renders the method less efficient in certain cases. This disadvantage is overcome by dividing the elastic body into subregions and writing the integral equation for each subregion. In this way, a matrix of banded form is obtained, and, incidentally, several different materials may be considered. To limit the amount of computer central memory required, a block solver is used to reduce the system of equations. The parametric representation of geometry and functions allows an elegant formulation of the direct calculation of stresses on the surface.

II - BASIC FORMULATION OF THE BOUNDARY INTEGRAL EQUATIONS

1.- Transformation of the Navier equation into an integral relationship

The integral relationship known as Somigliana's identity is normally calculated using Betti's theorem and a limiting process (6).

In the present study, the relationship is established in terms of generalised functions, by the use of distribution theory. This technique is an elegant and powerful tool for the transformation of an elliptic operator, for which a fundamental solution is known, into an integral relationship. The theory of this technique is given by Schwartz (32).

For convenience, operator symbols are used in the following derivation : ∇ is the gradient operator, $\nabla.$ is the divergence operator. Symbols without indices represent tensors (S); with indices they are the components S_{ij} of the tensor S

The equations of elastostatics in terms of displacement are those of Navier :

$$\Delta^* u = \lambda \nabla(\nabla.u) + \mu \nabla.(\nabla u) + \mu \nabla.(\nabla u)^T = -f \quad (2.1.1)$$

where f is the body force.

The unknown displacements $u = (u_1, u_2, u_3)$ must satisfy (2.1.1) at every point $x = (x_1, x_2, x_3)$ of the domain V being considered, and must also satisfy the boundary conditions : u, t given on the boundary S of V, t being the stress resultant :

$$t = \sigma . n = T^{(n)} u$$

where n is the outward normal and $T^{(n)}$ is the differential operator :

$$T^{(n)} = \lambda n \nabla . + \mu n . \nabla + \mu n . \nabla^T \qquad (2.1.2)$$

To derive a formulation of the above boundary value problem in terms of distributions, one applies in the sense of distribution theory the operator Δ^* to $H(S)u$ where $H(S)$ is the characteristic function of the domain V with boundary S :

$$H(S) = \begin{cases} 1 & x \in V \\ 0 & x \notin V \end{cases}$$

The relationships given in reference (14) are used to express the relation 2.1.1 in terms of distributions :

$$\begin{aligned}
\nabla(\nabla.(H(S)u)) &= H(S)\{\nabla(\nabla.u)\} - \nabla(n.u\,\delta(S)) - n(\nabla.u)\delta(S) \\
\nabla.(\nabla(H(S)u)) &= H(S)\{\nabla.(\nabla u)\} - \nabla.((nu)\delta(S)) - (n.\nabla u)\delta(S) \\
\nabla.(\nabla(H(S)u))^T &= H(S)\{\nabla.(\nabla u)^T\} - \nabla.((nu)^T\delta(S)) - (n.\nabla u)^T)\delta(S)
\end{aligned} \qquad (2.1.3)$$

where $\delta(S)$ is the Dirac distribution on the surface S and $(\ \ )$ is a distribution and $\{\ \ \}$ represents a regular function.

...

Substituting these expressions in 2.1.1 we have

$$\Delta^*\left(H(s)u\right) = H(s)\left\{\Delta^* u\right\} - \left(\lambda\,\nabla(n.u\,\delta(s)) + \mu\,\nabla.(nu\delta(s))\right.$$
$$+\mu\,\nabla.((nu)^T\delta(s)) - (\lambda\, n\nabla.u\,\delta(s) + \mu\, n.\nabla u\,\delta(s)$$
$$\left. + \mu\, n.(\nabla u)^T\delta(s)\right)$$

which may be rewritten as

$$\Delta^*\left(H(s)u\right) = H(s)\left\{\Delta^* u\right\} - {}^t T^{(n)}(u\,\delta(s)) - \left(T^{(n)}u\right)\delta(s) \qquad (2.1.4)$$

where $T^{(n)}$ is the operator defined by 2.1.2 and ${}^t T^{(n)}$ is the adjoint operator, in the sense of partial differential equation theory :

$${}^t T^{(n)} = \lambda\,\nabla n. + \mu\,\nabla.n + \mu\,\nabla.n^T \qquad (2.1.5)$$

Recalling that $\left\{\Delta^* u\right\} = -f$, 2.1.4 may be rewritten as the formulation of an interior Cauchy problem :

$$\Delta^*\left(H(s)u\right) = -H(s)f - {}^t T^{(n)}(u\,\delta(s)) - \left(T^{(n)}u\right)\delta(s) \qquad (2.1.6)$$

Now let E be an elementary solution corresponding to the operator Δ^* (31) :

$$\Delta^* E = \delta \qquad (2.1.7)$$

It is known, (31), (32), that a possible representation of the solution (if one exists) of (2.1.6) is given by the product of convolution of E and the right side of (2.1.6), that is :

$$H(s)u = -E * H(s)f - E * {}^t T^{(n)}(u\,\delta(s)) - E * \left(T^{(n)}u\right)\delta(s) \qquad (2.1.8)$$

...

This representation may be considered as the superposition of three potentials : the volume potential $-E*H(s)f$, the double layer potential $-E*^{t}T^{(n)}(u\,\delta(s))$, and the single layer potential $-E*(T^{(n)}u)\,\delta(s)$

One may transform the right side of (2.1.8) to show the part played by the physical quantities u and t. In fact, from $t = T^{(n)}u$, the last term of (2.1.8) is just $-E*t\,\delta(s)$. The function E is such that this term may be expressed as follows (62) :

$$-E*t\,\delta(s) = -\int_S E(x,y)\,t(y)\,ds_y$$

For the second last term of (2.1.8), one may carry out the differential operation ${}^{t}T^{(n)}$ upon E ; according to the properties of vector distributions, one obtains :

$$-E*{}^{t}T^{(n)}(u\,\delta(s)) = \int_S T_y^{n_y} E(x,y)\,u(y)\,dS_y$$

For the first term of the right side of (2.1.8)

$$-E*H(s)f = -\int_V E(x,y)\,f(y)\,dy$$

Therefore (2.1.8) may be written for $x \in V$ as follows :

$$u(x) = -\int_V E(x,y)\,f(y)\,dy + \int_S T_y^{(n_y)} E(x,y)\,u(y)\,dS_y - \int_S E(x,y)\,t(y)\,dS_y \qquad (2.1.9).$$

Remark : let us return to the relation (2.1.6). Developing the expression in terms of distributions, on a regular function v we have :

$$\langle \Delta^*(H(s)u), v\rangle = \langle H(s)u, \Delta^* v\rangle = \int_V u.\,\Delta^* v\,dx \quad \cdots$$

For the first term of the second part we have :

$$< H(s)\{\Delta^* u\}, v> = \int_V v . \Delta^* u \, dx$$

For the second term :

$$- < {}^t T^{(n)}(u\, \delta(s)), v> = \int_S u . T^{(n)} v \, ds$$

For the third term :

$$- < (T^{(n)} u)\, \delta(s), v> = - \int_S v . T^{(n)} u \, ds$$

Therefore,

$$\int_V u . \Delta^* v \, dx = \int_V v . \Delta^* u \, dx + \int_S u . T^{(n)} v \, ds - \int_S v . T^{(n)} u \, ds \qquad (2.1.10)$$

which is the well known Betti identity. We see the equivalence between distribution theory and the classic formulation based on the Betti identity.

...

2.- Calculation of elementary solutions for the operator

One may list, among others, the following methods of calculation of elementary solutions :

- superposition (62)
- Fourier transformation (62)
- expression of δ in terms of plane waves (14)
- Hörmander's method

Hörmander's method is the best suited to calculation. If one symbolises by $P\frac{\partial}{\partial x}$ the matrix of a system of differential operators with constant coefficients and if Det $P(\xi) \neq 0$, according to the general theorem of Hörmander (17), the system of operators has the elementary solution :

$$E(x,0) = {}^{co}P\left(\frac{\partial}{\partial x}\right) e_x \qquad (2.2.1)$$

where ${}^{co}P\left(\frac{\partial}{\partial x}\right)$ is the matrix of cofactors of the matrix $P\left(\frac{\partial}{\partial x}\right)$ and e_x is the elementary solution of the operator Det $P\left(\frac{\partial}{\partial x}\right)$ i.e. the solution of :

$$\det P\left(\frac{\partial}{\partial x}\right) e_x = \delta \qquad (2.2.2)$$

For the particular case considered here, one has

$$P\left(\frac{\partial}{\partial x}\right) \equiv \Delta^*\left(\frac{\partial}{\partial x}\right) = \begin{vmatrix} \mu\Delta + (\lambda+\mu)\partial_1\partial_1 & (\lambda+\mu)\partial_1\partial_2 & (\lambda+\mu)\partial_1\partial_3 \\ (\lambda+\mu)\partial_2\partial_1 & \mu\Delta + (\lambda+\mu)\partial_2\partial_2 & (\lambda+\mu)\partial_2\partial_3 \\ (\lambda+\mu)\partial_3\partial_1 & (\lambda+\mu)\partial_3\partial_2 & \mu\Delta + (\lambda+\mu)\partial_3\partial_3 \end{vmatrix}$$

From which :

$$\text{Det } \Delta^* \left(\frac{\partial}{\partial x}\right) = \mu^2 (\lambda + 2\mu) \Delta^3$$

The solution e_x of

$$\mu^2 (\lambda + 2\mu) \Delta^3 e_x = \delta$$

is given in reference (62) as

$$e_x = - \frac{|x|^3}{3 \cdot 2^5 \cdot \eta \mu^2 (\lambda + 2\mu)}$$

and so 2.2.1 may be written

$$E_{ij}(x,0) = - \frac{1}{8\eta\mu(\lambda+2\mu)r} \left((\lambda + 3\mu) \delta_{ij} + (\lambda + \mu) \partial_i r \partial_j r \right) \quad (2.2.3)$$

where :

$$r = \sqrt{\sum_i x_i^2}$$

It is noteworthy that, in this method, it is only necessary to calculate the elementary solution for the scalar operator $\text{Det } P\left(\frac{\partial}{\partial x}\right)$, the other calculations consisting of algebraic manipulation.

Details of the derivation are given in reference (62)

In equation (2.2.2) the Dirac distribution δ is located at the origin. For any point $y \in S$ we have for $E_{ij}(x,y)$ the same expression as in 2.2.3 with $r = \sqrt{\sum_i (x_i - y_i)^2}$

Substituting for λ, μ in terms of E, ν and replacing $\partial_i r$ by its value $\frac{1}{r}(x_i - y_i)$ we have

$$E_{ij}(x,y) = - U_{ij}(x,y) = - \frac{(1+\nu)}{8\pi E(1-\nu) r} \left\{ (3-4\nu)\delta_{ij} + \frac{(x_i - y_i)(x_j - y_i)}{r^2} \right\} \quad (2.2.4)$$

3.- Calculation of displacement and stress at an interior point

Let us consider the interior problem for the three dimensional case, and suppose that u and t are known for all $x \in S$. Then the formula (2.1.9), which is simply Somigliana's identity gives u for all $x \in V$. That is :

$$u_i(x) = \int_V U_{ij}(x,y)\, f_j(y)\, dy - \int_S T_{ij}(x,y)\, u_j(y)\, dS_y + \int_S U_{ij}(x,y)\, t_j(y)\, dS_y \qquad (2.3.1)$$

for $x \in V$ where :

$$U_{ij}(x,y) = \frac{1+\nu}{8\pi E(1-\nu) r}\left\{(3-4\nu)\delta_{ij} + \frac{(x_i-y_i)(x_j-y_j)}{r^2}\right\} \qquad (2.3.2)$$

$$T_{ij}(x,y) = T_y^{(n_y)} U_{ij}(x,y) =$$

$$= \frac{1}{8\pi(1-\nu) r^2}\left\{(1-2\nu)\left[n_i(y)\frac{(x_j-y_j)}{r} - n_j(y)\frac{(x_i-y_i)}{r}\right] + \left[(1-2\nu)\delta_{ij} + 3\frac{(x_i-y_i)(x_j-y_j)}{r^2}\right] n_\lambda(y)\frac{(x_\lambda-y_\lambda)}{r}\right\} \qquad (2.3.3)$$

$T_y^{(n_y)}$ is given by 2.1.2 and $n(y)$ is the outward normal to S at y

For details of this derivation, see Lachat and Brebbia (62).

According to Hooke's law, the stress tensor σ_{ij} is given in terms of displacement by :

$$\sigma_{ij} = \lambda\, \delta_{ij}\, \partial_k u_k + \mu\,(\partial_i u_j + \partial_j u_i)$$

Using for u_i the expression (2.3.1) and defining :

$$D_{ijk}(x,y) = \lambda\delta_{ij}\partial_l U_{lk}(x,y) + \mu\partial_i U_{jk}(x,y) + \mu\partial_j U_{ik}(x,y)$$

$$S_{ijk}(x,y) = \lambda\delta_{ij}\partial_l T_{lk}(x,y) + \mu\partial_i T_{jk}(x,y) + \mu\partial_j T_{ik}(x,y)$$

one may obtain the result :

$$\sigma_{ij}(x) = \int_V D_{ijk}(x,y)\, f_k(y)\, dy + \int_S D_{ijk}(x,y)\, t_k(y)\, dS_y - \int_S S_{ijk}(x,y)\, u_k(y)\, dS_y \qquad (2.3.4)$$

where

$$D_{ijk}(x,y) = \frac{1}{8\pi(1-\nu)r^2}\left\{(1-2\nu)\left[\delta_{ij}\frac{(x_k-y_k)}{r} - \delta_{ik}\frac{(x_j-y_j)}{r} - \delta_{jk}\frac{(x_i-y_i)}{r}\right] - 3\frac{(x_i-y_i)(x_j-y_j)(x_k-y_k)}{r^3}\right\} \qquad (2.3.5)$$

$$S_{ijk}(x,y) = \frac{E}{8\pi(1-\nu^2)r^3}\left\{n_s(y)\frac{(x_s-y_s)}{r}\left[3(1-2\nu)\delta_{ij}\frac{(x_k-y_k)}{r} + 3\nu\left(\delta_{ik}\frac{(x_j-y_j)}{r} + \delta_{jk}\frac{(x_i-y_i)}{r}\right) - 15\frac{(x_i-y_i)(x_j-y_j)(x_k-y_k)}{r^3}\right]\right.$$

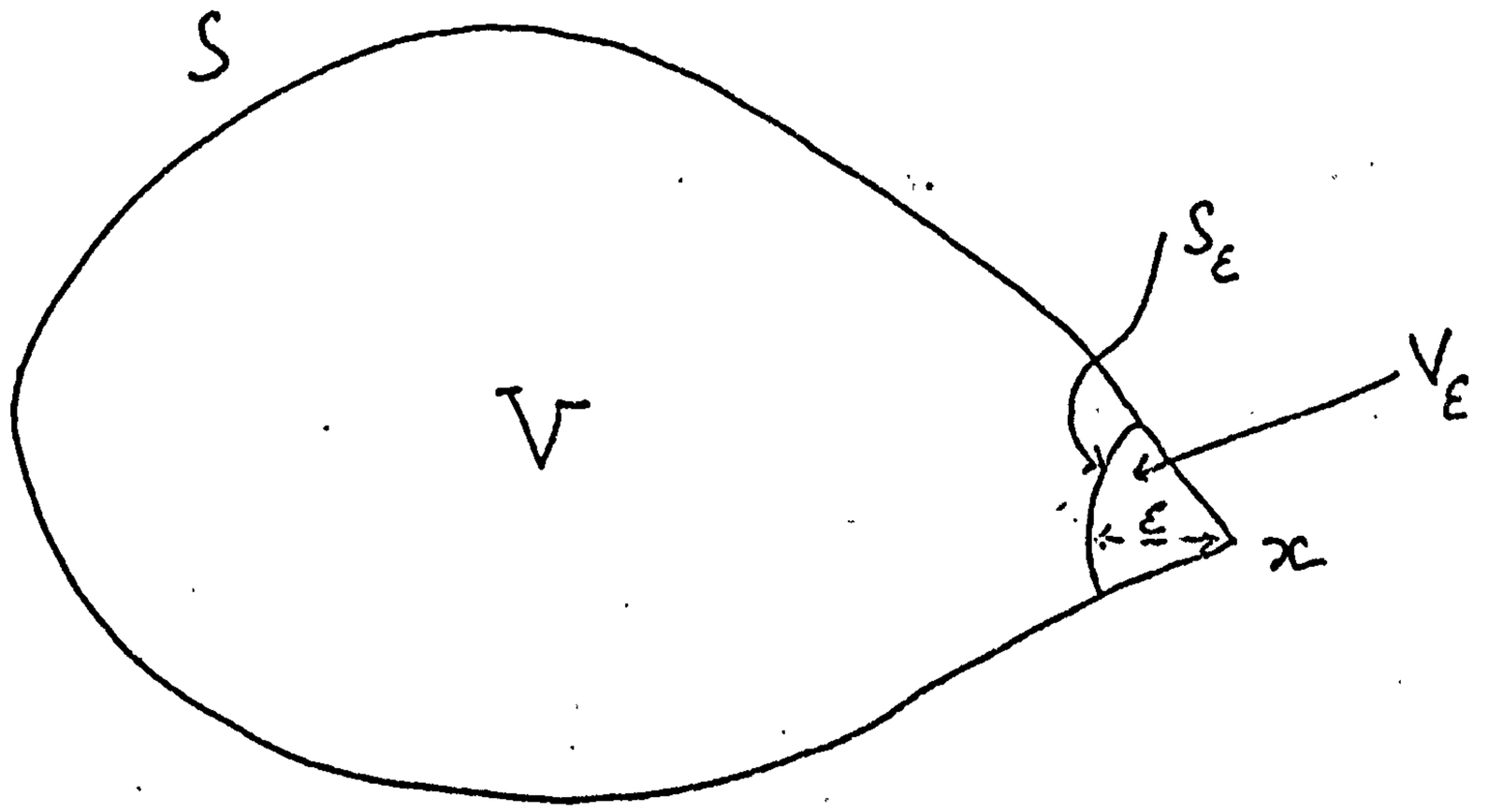
$$+ n_i(y)\left[3\nu\frac{(x_j-y_j)(x_k-y_k)}{r^2} + (1-2\nu)\delta_{jk}\right]$$

$$+ n_j(y)\left[3\nu\frac{(x_i-y_i)(x_k-y_k)}{r^2} + (1-2\nu)\delta_{ik}\right]$$

$$\left. + n_k(y)\left[3(1-2\nu)\frac{(x_i-y_i)(x_j-y_j)}{r^2} - (1-4\nu)\delta_{ij}\right]\right\}$$

4.- Calculation of displacement on the surface

Let us consider the case of zero body force ($f=0$ in equation (2.1.1)). The point x may be on a smooth surface, or at the intersection of several smooth surfaces, i.e. at an edge or corner :



Let u_i and v_i be displacement fields, and $\sigma_{ij}(u)$, $\sigma_{ij}(v)$ be the corresponding stress fields. If u and v are continuously differentiable in domain $V - V_\varepsilon$ and on its surface the divergence theorem may be used to give :

$$\int_{V-V_\varepsilon} \left\{ v_j \frac{\partial [\sigma_{jA}(u)]}{\partial y_A} + \frac{\partial v_j}{\partial y_A} \sigma_{jA}(u) \right\} dV_y$$

$$= \int_{S-(S\cap V_\varepsilon)+S_\varepsilon} v_j \, n_A(y) \, \sigma_{jA}(u) \, dS_y \qquad \ldots$$

and :

$$\int_{V-V_\varepsilon} \left\{ v_j \frac{\partial [\sigma_{jA}(u)]}{\partial y_A} - u_j \frac{\partial [\sigma_{jA}(v)]}{\partial y_A} \right\} dV_y$$

$$= \int_{S-(S\cap V_\varepsilon)+S_\varepsilon} \left\{ v_j \sigma_{jA}(u) - u_j \sigma_{jA}(v) \right\} n_A(y)\, dS_y \qquad (2.4.1)$$

Equation (2.4.1) is known as Betti's theorem for the domain $V - V_\varepsilon$

Let us replace v_j by $U_{ij}(x,y)$ (equation (2.3.1)), and let u_j be a solution of the Navier equations for the case of zero body force. Then the volume integral in equation (2.4.1) is zero. Now let us consider what happens as $\varepsilon \to 0$. Equations (2.4.1) may be rewritten.

$$\lim \varepsilon\to 0 \int_{S_\varepsilon} T_{ij}(x,y)\, u_j(y)\, dS_y + \lim \varepsilon\to 0 \int_{S-(S\cap V_\varepsilon)} T_{ij}(x,y)\, u_j(y)\, dS_y \qquad (2.4.2)$$

$$= \lim \varepsilon\to 0 \int_{S_\varepsilon} U_{ij}(x,y)\, t_j(y)\, dS_y + \lim \varepsilon\to 0 \int_{S-(S\cap V_\varepsilon)} U_{ij}(x,y)\, t_j(y)\, dS_y$$

Provided that the surface at x, or the surfaces that intersect at x, are Lyapounov smooth (58),

$$\lim \varepsilon\to 0 \int_{S_\varepsilon} T_{ij}(x,y)\, u_j(y)\, dS_y = C_{ij}(x)\, u_j(x)$$

where

$$C_{ij}(x) = \lim \varepsilon\to 0 \int_{S_\varepsilon} \left\{ T_{ij}(x,y) \left[u_j(y) - u_j(x) \right] + T_{ij}(x,y)\, u_j(x) \right\} dS_y \qquad (2.4.3)$$

Since $|T_{ij}| \leq \frac{K_1}{\varepsilon^2}$, $|u_j(y) - u_j(x)| \leq K_2 \varepsilon$ and the area of S_ε is not greater than $4\pi\varepsilon^2$, the first part of this integral tends to zero. This leaves

$$C_{ij}(x) = \lim \varepsilon\to 0 \int_{S_\varepsilon} T_{ij}(x,y)\, dS_y \quad \ldots$$

In the case in which the normal direction is continuous at x,

$$C_{ij}(x) = \frac{1}{2}\delta_{ij}$$

Again provided that the surface or surfaces at x are Lyapounov smooth, and since $u_j(y)$ is differentiable, the second integral on the left side of equation (2.4.2) tends to the principal value.

Since $|U_{ij}| \leq \frac{K_1}{\varepsilon}$, $|t_j(y)| \leq K_2$, $|S_\varepsilon| \leq 4\pi\varepsilon^2$,

$$\lim_{\varepsilon \to 0} \int_{S_\varepsilon} U_{ij}(x,y)\, t_j(y)\, dS_y = 0$$

The last integral always converges to a limit as $\varepsilon \to 0$, so

$$C_{ij}(x,y)\, u_j(x) + \int_S T_{ij}(x,y)\, u_j(y)\, dS_y = \int_S U_{ij}(x,y)\, t_j(y)\, dS_y \qquad (2.4.4)$$

where the principal value of the integral on the left side is taken. It is interesting to note that the value of $C_{ij}(x)$ would change if the shape of region of exclusion used to define the principal value is changed.

5.- Body forces

In the preceding section, the body forces f (equation (2.1.1)) are supposed to be zero, and an integral equation for the unknown functions u or t on S is derived. The general case may be treated by taking the displacement to be the sum of two components :

$$u = \hat{u} + \hat{\hat{u}} \tag{2.5.1}$$

The particular integral $\hat{u}$ is any function that satisfies

$$\Delta^* \hat{u} = - f \tag{2.5.2}$$

whilst the complementary function $\hat{\hat{u}}$ is chosen such that

$$\Delta^* \hat{\hat{u}} = 0 \tag{2.5.3}$$

and the sum $\hat{u} + \hat{\hat{u}}$ satisfies the boundary conditions for u

The particular integral may be taken as the volume potential

$$\hat{u}_i(x) = \int_V U_{ij}(x,y) f_i(y) \, dV_y \tag{2.5.4}$$

For special cases such as gravity and centrifugal force it is more convenient to use polynomials that satisfy equation (2.5.2)

...

6.- The plane problems

The displacement field is said to be plane, parallel to $x_3=0$ if $u_3=0$ and u_1, u_2 are functions of x_1 and x_2 only. In the linear theory, $\varepsilon_{ij} = \frac{1}{2}(\partial_i u_j + \partial_j u_i)$ so it may be deduced directly that at any point,

$$\varepsilon_{13} = \varepsilon_{33} = \varepsilon_{23} = 0$$

that is the tensor ε is everywhere a plane tensor. The Navier equations are :

$$\Delta^* u = \mu \Delta u_i + (\lambda+\mu)\, \partial_i \partial_j u_j = -f_i \quad i,j \in [1,2] \qquad (2.6.1)$$

It should be noted that σ is not, in general, a plane tensor. As for the three dimensional case, one denotes by Δ^* the operator

$$\Delta^* = \mu \Delta + (\lambda+\mu)\, \nabla \nabla .$$

The stress field is said to be plane, parallel to $x_3=0$ if σ is everywhere a plane stress tensor parallel to $x_3=0$ and its components σ_{ij} are independent of x_3. Consequently,

$$\sigma_{13} = \sigma_{23} = \sigma_{33} = 0$$

It follows that

$$\varepsilon_{13} = \varepsilon_{23} = 0 \qquad \varepsilon_{33} = -\frac{\lambda}{\lambda+2\mu}(\varepsilon_{11} + \varepsilon_{22})$$

The Navier equations for plane stress are :

$$\mu \Delta u_i + \frac{\mu(3\lambda+2\mu)}{\lambda+2\mu} \partial_i \partial_j u_j = -f_i \qquad (2.6.2)$$

$$i,j \in [1,2] \quad \ldots$$

One denotes by $\overline{\Delta}^*$ the operator

$$\overline{\Delta}^* = \mu \Delta + \frac{\mu(3\lambda+2\mu)}{\lambda+\mu} \nabla \nabla.$$

Now let

$$\Delta_a^* = \mu \Delta + a \nabla \nabla. \tag{2.6.3}$$

where

$$\Delta_a^* \begin{cases} \Delta^* & = \text{plane strain operator, for which } a = \lambda+\mu \\ \overline{\Delta}^* & = \text{plane stress operator, for which } a = \frac{\mu(3\lambda+2\mu)}{\lambda+2\mu} \end{cases}$$

The elementary solution for Δ_a^* may be calculated by Hörmander's method as in paragraph 2.2.

One obtains:

$$E_{ij}(x,y) = \frac{1}{4\pi\mu(a+\mu)} \left((a+2\mu)\delta_{ij} \log r - a\, \partial_i r\, \partial_j r \right) \tag{2.6.4}$$

where

$$r = \sqrt{\sum_i (x_i - y_i)^2}$$

From 2.6.4. we may obtain the expression of $U_{ij}(x,y)$ and $T_{ij}(x,y)$ in terms of E and ν as in paragraph 2.3 :

$$U_{ij}(x,y) = \frac{1+\nu'}{4\pi E'(1-\nu')} \left\{ (3-4\nu')\delta_{ij} \log r + \frac{(x_i-y_i)(x_j-y_j)}{r^2} \right\} \tag{2.6.5}$$

$$T_{ij}(x,y) = \frac{1}{4\pi(1-\nu')r} \left\{ n_s(y) \frac{(x_s-y_s)}{r} \left[(1-2\nu')\delta_{ij} + \frac{2(x_i-y_i)(x_j-y_j)}{r^2} \right] + (1-2\nu') \left[n_i(y) \frac{(x_j-y_j)}{r} - n_j(y) \frac{(x_i-y_i)}{r} \right] \right\} \tag{2.6.6}$$

where

$\nu' = \nu$ and $E' = E$ for plane strain

$\nu' = \frac{\nu}{1+\nu}$ and $E' = E(1-\nu'^2)$ for plane stress

...

The displacements at any point $x \in V$ and $x \in S$ are given by the relationships identical with (2.3.1) and (2.4.4).

The components of the stress tensor for $x \in V$ are determined using the same formula as (2.3.4), where :

$$D_{ijk}(x,y) = \frac{1}{4\pi(1-\nu')r}\left\{(1-2\nu')\left[\delta_{ij}\frac{(x_k-y_k)}{r} - \delta_{ik}\frac{(x_j-y_j)}{r} - \delta_{jk}\frac{(x_i-y_i)}{r}\right] - \frac{2(x_i-y_i)(x_j-y_j)(x_k-y_k)}{r^3}\right\} \quad (2.6.7)$$

$$S_{ijk}(x,y) = \frac{E'}{4\pi(1-\nu'^2)r^2}\left\{n_\lambda(y)\frac{(x_\lambda-y_\lambda)}{r}\left[2(1-2\nu')\delta_{ij}\frac{(x_k-y_k)}{r} + 2\nu'\left(\delta_{ik}\frac{(x_j-y_j)}{r} + \delta_{jk}\frac{(x_i-y_i)}{r}\right) - 8\frac{(x_i-y_i)(x_j-y_j)(x_k-y_k)}{r^3}\right]\right.$$
$$+ n_i(y)\left[2\nu'\frac{(x_j-y_j)(x_k-y_k)}{r^2} + (1-2\nu')\delta_{jk}\right]$$
$$+ n_j(y)\left[2\nu'\frac{(x_i-y_i)(x_k-y_k)}{r^2} + (1-2\nu')\delta_{ik}\right]$$
$$\left. + n_k(y)\left[2(1-2\nu')\frac{(x_i-y_i)(x_j-y_j)}{r^2} - (1-4\nu')\delta_{ij}\right]\right\} \quad (2.6.8)$$

where $\nu' = \nu \quad E' = E$ for plane strain

$\nu' = \frac{\nu}{1+\nu} \quad E' = E(1-\nu'^2)$ for plane stress

...

III - NUMERICAL FORMULATION OF THE PLANE PROBLEMS

1.- Review of previous formulations

In the formulations for the Laplace operator by Jaswon and Ponter (55) and by Symm (54), and the later formulations for elastostatics by Rizzo (29), Cruse (51), and Riccardella (52), the boundary is approximated by straight line segments, so for problems involving curved boundaries, either a large number of segments must be considered or there are large errors of geometrical representation.

Except in the work of Riccardella, the functions u and t are supposed to be constant over each boundary segment. The very approximate nature of this assumption requires that a large number of segments be considered even on straight boundaries.

The integral equation for smooth surfaces is used (e.g. equation (2.4.4)) with $C_{ij} = \frac{1}{2}\delta_{ij}$ this being satisfactory because the equation is written for the midpoint of each segment. Riccardella considers u and t to vary linearly, and writes the integral equations for the end points of each segment. The free term of the integral equation is modified to account for the possibility that adjacent segments are not in a straight line.

...

The results presented by Riccardella show that this formulation is superior to that in which functions are taken to be constant over each segment.

Cruse (49) further improves the efficiency of the method in some cases by representing only half or one quarter of a structure with symmetry. This is a feature which may usefully be retained in the present formulation.

Jaswon and Ponter, and Symm, use Simpson's Rule to integrate for the matrix coefficients, whereas Rizzo, Cruse and Riccardella integrate analytically the kernels over the straight line segments. The use of Simpson's Rule probably introduces considerable error, because the kernels vary rapidly as one argument approaches the other. Even the calculation of the stiffness matrices of finite elements, which involves only the integration of relatively smooth shape functions (65), is normally done by Gaussian quadrature formulae. Analytic integration is clearly the ideal solution for straight boundary segments with simple variation of u and t, but must be difficult or impossible to use for the more sophisticated segments and functional variations to be considered here.

Jaswon and Ponter, and Symm, report that the equations become unstable if the number of boundary segments exceeds a certain limit. The reason is not stated, but may be that the precision of floating point arithmetic during reduction of the system was insufficient, or that the system is inherently unstable because the unknowns (potential and normal derivative) are not of the same order. In the latter case, the problem may be solved by writing the system in nondimensional form. Cruse (9) finds that, in the case of elastostatics, stability is improved, indeed assured for the examples presented, by scaling the coefficients of traction unknowns by the shear modulus.

No author discusses the programming of the equation solution, other than to say that Gaussian elimination is used. The examples treated are quite small, and it is probable that the matrix is held in core during reduction. This approach is not suited to problems of practical size.

Once the equations are solved, the stress tensor at the surface may be calculated from a formula in terms of the kernels $\mathcal{D}$ and $\mathcal{S}$ (equations (2.6.7.), (2.6.8)), but the integrations involved are expensive to perform. A simpler and much more economical method, developed by Cruse (9), is to calculate the tensor from the traction on the tangent

plane and strain in that plane, the latter data being obtained by differencing or differentiating surface displacement.

2.- The parametric representation of geometry and functions

The boundary, which may consist of one or more separate parts, is represented by a sequence of straight or curved line elements, each of which has three nodes (fig. 3.2.1).

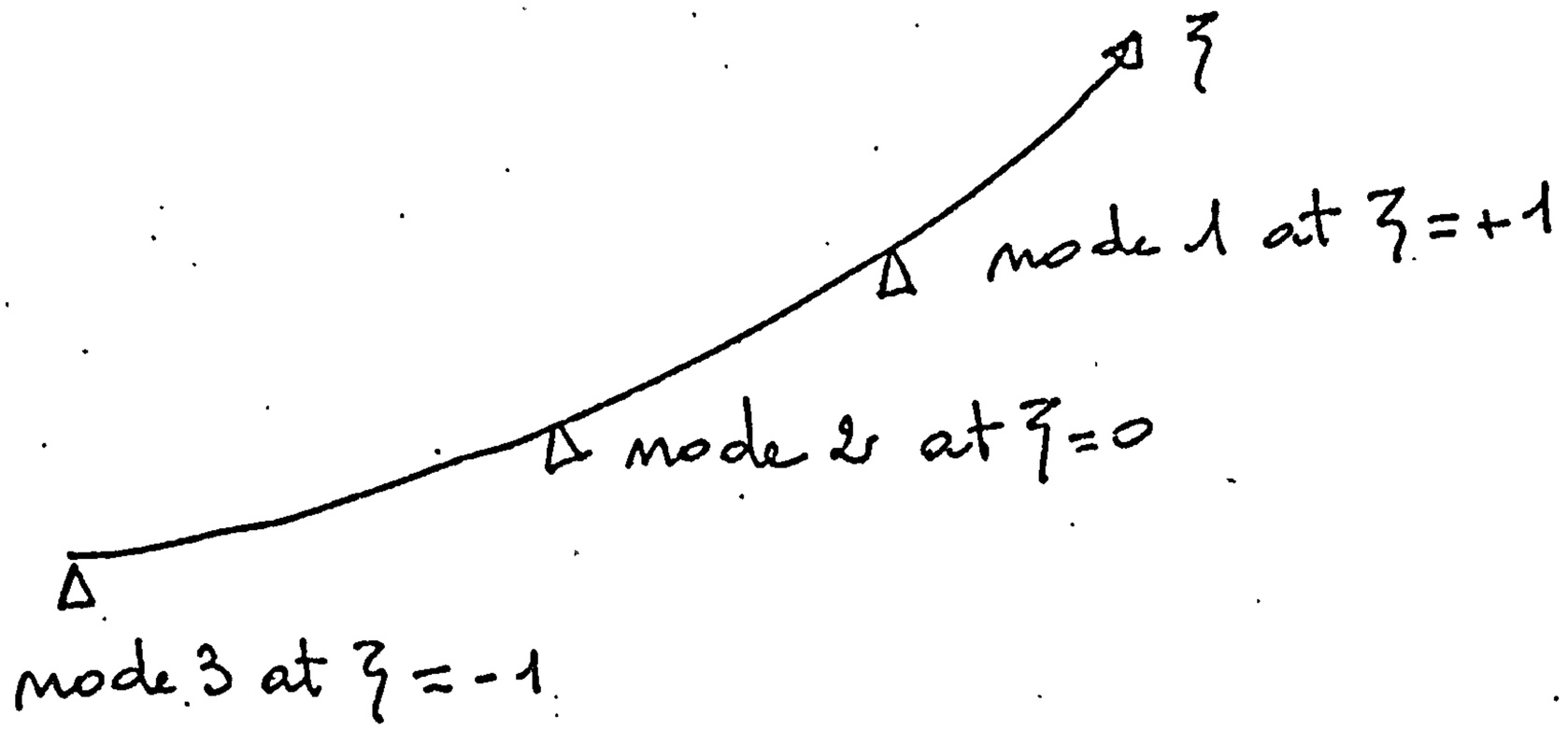


fig. 3.2.1. - a line element

The cartesian coordinates x_i of an arbitrary point of an element are defined in terms of the nodal cartesian coordinates x_i^a, $a \in \{1,2,3\}$ and shape functions $N^a(\xi)$:

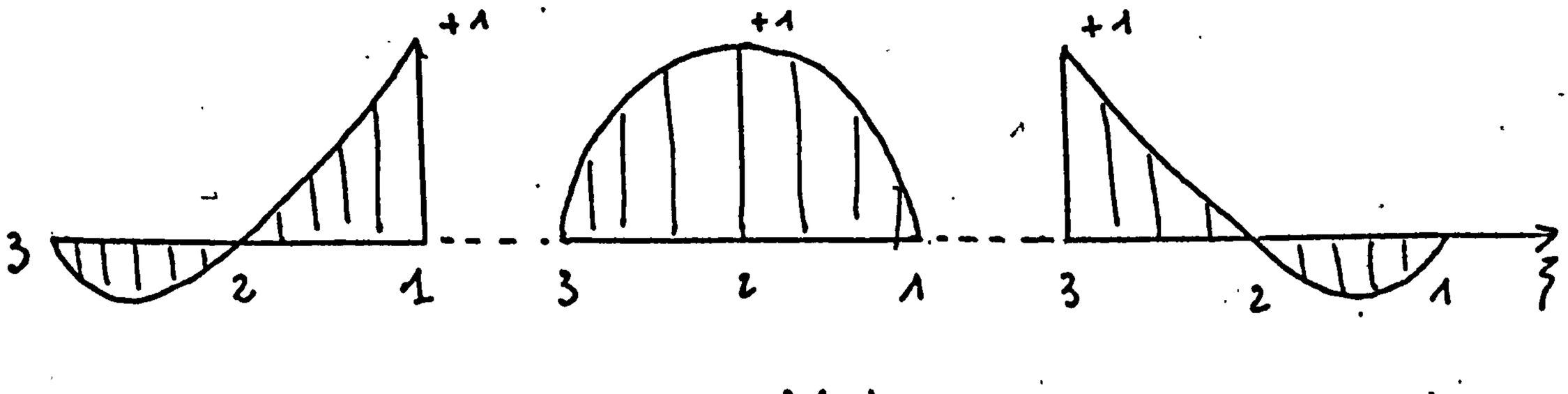
$$x_i(\xi) = N^a(\xi)\, x_i^a \qquad (3.2.1)$$

...

$\zeta \in [-1,1]$ being the intrinsic coordinate, and N the functions (Ergatoudis (67)) :

$$N^1(\zeta) = \frac{1}{2}\,\zeta(\zeta+1)$$
$$N^2(\zeta) = \frac{1}{2}\,\zeta(\zeta-1) \qquad (3.2.2)$$
$$N^3(\zeta) = 1-\zeta^2$$

The variation of these functions is shown in fig. 3.2.2.



$N^1(\zeta)$ $N^2(\zeta)$ $N^3(\zeta)$

fig. 3.2.2. - the geometric shape functions

By differentiating equation 3.2.1. with respect to ζ, a vector defining the tangent to the element is obtained :

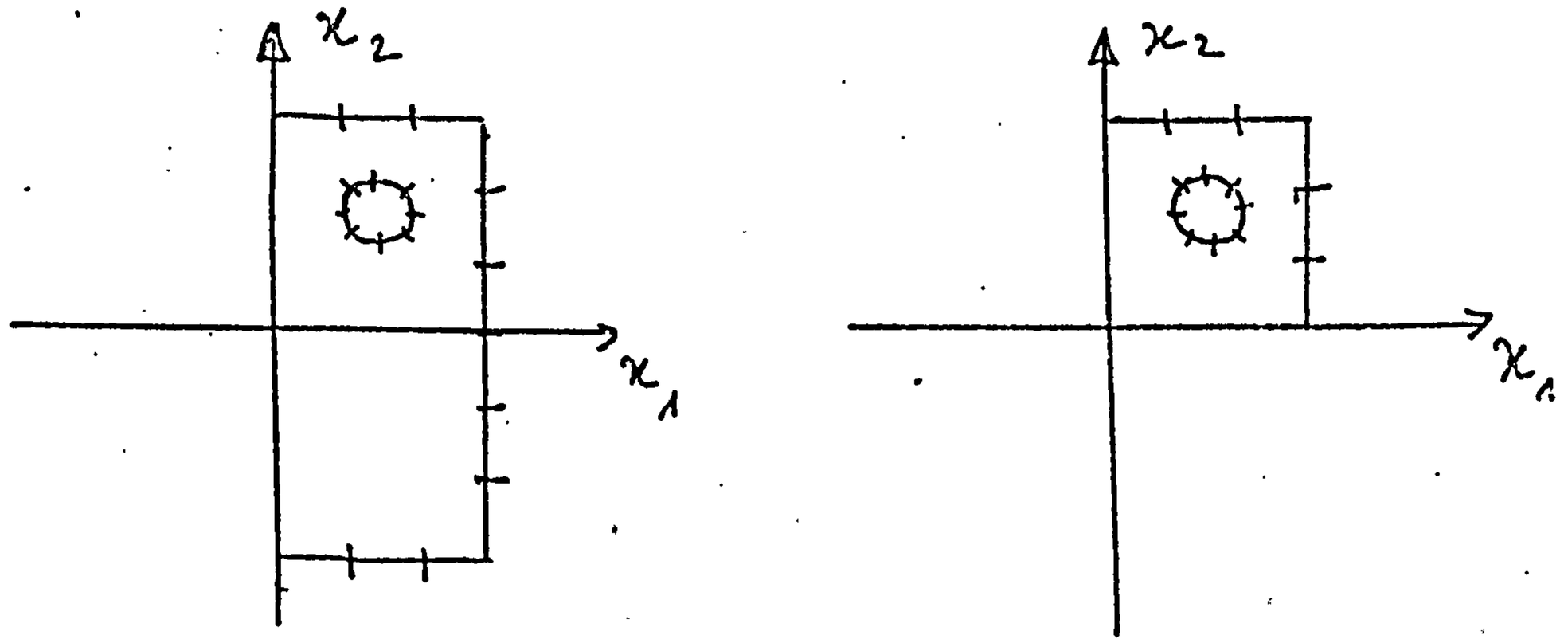
$$J_i(\zeta) = \frac{dx_i}{d\zeta} = \frac{dN^a(\zeta)}{d\zeta}\,x_i^a \qquad (3.2.3)$$

From this result may be calculated two functions of fundamental importance to the numerical representation of the integral equation : the Jacobian, which equals $|J|$;

...

and the normal to the element, required for the calculation of the kernels T and S, which is either $(\Delta_2, -\Delta_1)$ or $(-\Delta_2, \Delta_1)$ depending upon to which side of the element the elastic body lies.

In cases in which there is structural symmetry and symmetry of loads with respect to one or both of the cartesian coordinate axes, only half or one quarter of the boundary need be represented (fig. (3.2.3))



symmetry with respect to the axis x_2

symmetry with respect to both axes

fig. 3.2.3 - representation of symmetric structures

The functions u and t are considered to vary linearly, quadratically or cubically with respect to the intrinsic coordinate over each boundary element.

...

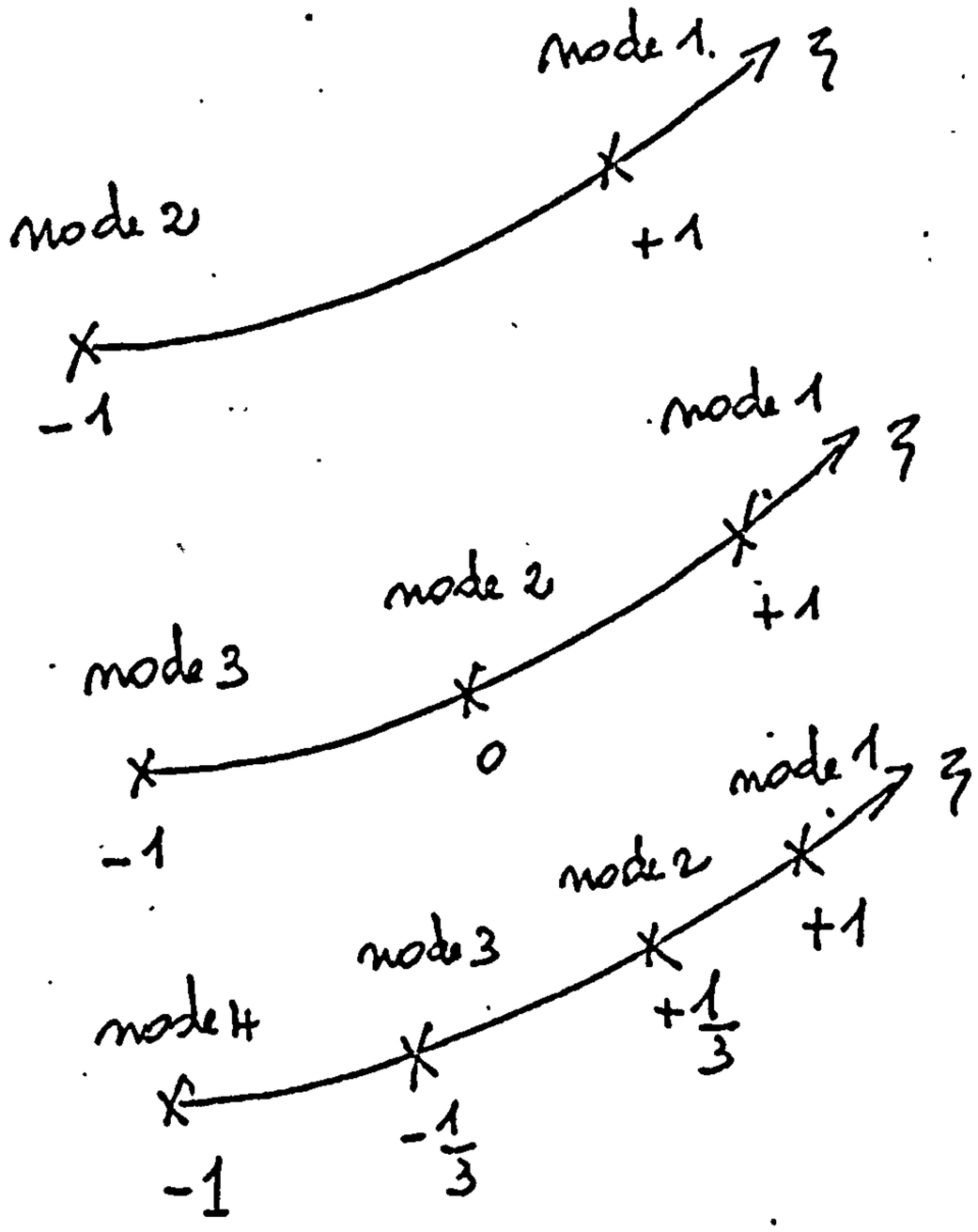


fig. 3.2.4 - nodes for the representation of functions

The value ϕ of a function at an arbitrary point of an element is defined in terms of its nodal values $\phi^a \; a \in \{1, \cdots, n\}$ (fig. 3.2.4) and the shape functions $M^a(\xi)$ corresponding to the variation chosen :

$$\phi(\xi) = M^a(\xi)\, \phi^a \qquad (3.2.4)$$

where M is defined as follows :

- linear variation

$$M^1(\xi) = \frac{1}{2}(\xi + 1)$$
$$M^2(\xi) = -\frac{1}{2}(\xi - 1) \qquad (3.2.5)$$

...

- quadratic variation (isoparametric case)

$$M^a(\zeta) = N^a(\zeta) \quad \text{(see equation 3.2.2)} \tag{3.2.6}$$

- cubic variation

$$M^1(\zeta) = \frac{9}{16}(\zeta+1)(\zeta^2 - \frac{1}{9})$$

$$M^2(\zeta) = \frac{27}{16}(\zeta + \frac{1}{3})(1-\zeta^2)$$

$$M^3(\zeta) = -\frac{27}{16}(\zeta - \frac{1}{3})(1-\zeta^2) \tag{3.2.7}$$

$$M^4(\zeta) = -\frac{9}{16}(\zeta - 1)(\zeta^2 - \frac{1}{9})$$

Let $m_{ij}, j \in \{1,2\}$ be the unit vector in the tangential direction (see equation 3.2.3) and m_{2j} be the unit normal. Let $\bar{u}_i$ be the displacement, $\bar{\varepsilon}_{ij}$ be the strain and $\bar{\sigma}_{ij}$ be the stress in the system of local coordinates so defined (see figure 3.2.5). Then the displacement in the tangential direction 1 is given by :

$$\bar{u}_1(\zeta) = M^a(\zeta)\, u_i^a\, m_{1i} \tag{3.2.8}$$

By differentiating with respect to ζ, an expression is obtained for strain in the tangential direction :

$$\bar{\varepsilon}_{11}(\zeta) = \frac{d\,M^a(\zeta)}{d\zeta}\, u_i^a\, m_{1i} \cdot \frac{1}{|J|} \tag{3.2.9}$$

Let us suppose that the nodal displacements u_i^a and the traction on the tangent plane at ζ are known, which is the case once the integral equation (2.4.4) is solved

...

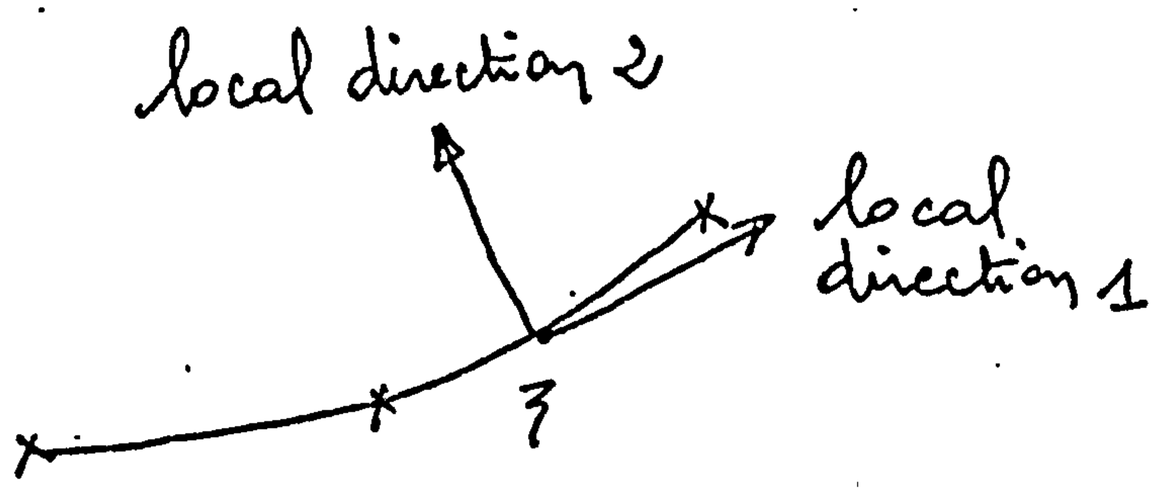


fig. 3.2.5 - local axes for calculation of stress

Then, using Hooke's law, the stress tensor may be calculated from the data already known :

$$\overline{\sigma_{11}} = \left(\frac{E'}{1+\nu'} \overline{\varepsilon_{11}} + \nu' \overline{t_2} \right) \Big/ (1-\nu')$$

$$\overline{\sigma_{12}} = \overline{\sigma_{21}} = \overline{t_1}$$

$$\overline{\sigma_{22}} = t_2$$

where, for plane strain :

$$\nu' = \nu \qquad E' = E$$

and for plane stress

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

and $\overline{t_i}$ is the traction in the local system of coordinates.

...

3.- Discretisation of the integral equation

The object of the discretisation is to approximate the integral equation by a system of simultaneous equations in unknowns associated with the element nodes used to define the variation of u and t (fig. 3.2.4). Let there be p elements each with n nodes. It is first necessary to create a global numbering $d(b, c)$, $b \in \{1, \cdots, p\}$, $c \in \{1, \cdots, n\}$ where b is the element and c the node. The numbers d vary between 1 and q where q is the number of distinct nodes, and define the order of solution of the simultaneous equations.

In practice, some of the nodes 1 to q are either partially or completely fixed, and the directions in which they are fixed do not necessarily coincide with the global coordinate directions. The integral equation (2.4.4) is rewritten in terms of the components of u and t in the fixed and free directions at the point x :

$$c'_{\alpha\beta}\, u'_\beta(x) + \int_S T'_{\alpha\beta}(x,y)\, u'_\beta(y)\, dS_y = \int_S U'_{\alpha\beta}(x,y)\, t'_\beta(y)\, dS_y \qquad (3.3.1)$$

where

$$u'_\alpha = l_{\alpha i}\, u_i \qquad \overline{T}'_{\alpha\beta} = l_{\alpha i}\, l_{\beta j}\, \overline{T}_{ij}$$

$$t'_\alpha = l_{\alpha i}\, t_i \qquad U'_{\alpha\beta} = l_{\alpha i}\, l_{\beta j}\, U_{ij} \qquad (3.3.2)$$

$$c'_{\alpha\beta} = l_{\alpha i}\, l_{\beta j}\, c_{ij}$$

In equation (3.3.2), $l_{\alpha i}$ are the direction cosines of the αth unknown at χ. Writing the equation (3.3.1) for each node, and substituting the parametric representations of t and u (3.2.4), the desired system of simultaneous equations is obtained :

$$C'_{\alpha\beta}(x^a)\, U'_\beta(x^a) + \sum_{b=1}^{p} \sum_{c=1}^{n} U'_\beta(x^{d(b,c)}) \int_{S_b} T'_{\alpha\beta}(x^a, y(\xi))\, M^c(\xi)\, J(\xi)\, d\xi$$

$$= \sum_{b=1}^{p} \sum_{c=1}^{n} t'_\beta(x^{d(b,c)}) \int_{S_b} U'_{\alpha\beta}(x^a, y(\xi))\, M^c(\xi)\, J(\xi)\, d\xi \qquad (3.3.3)$$

where S_b is the bth element, and $J(\xi)$ is the Jacobian $\frac{dS}{d\xi}$

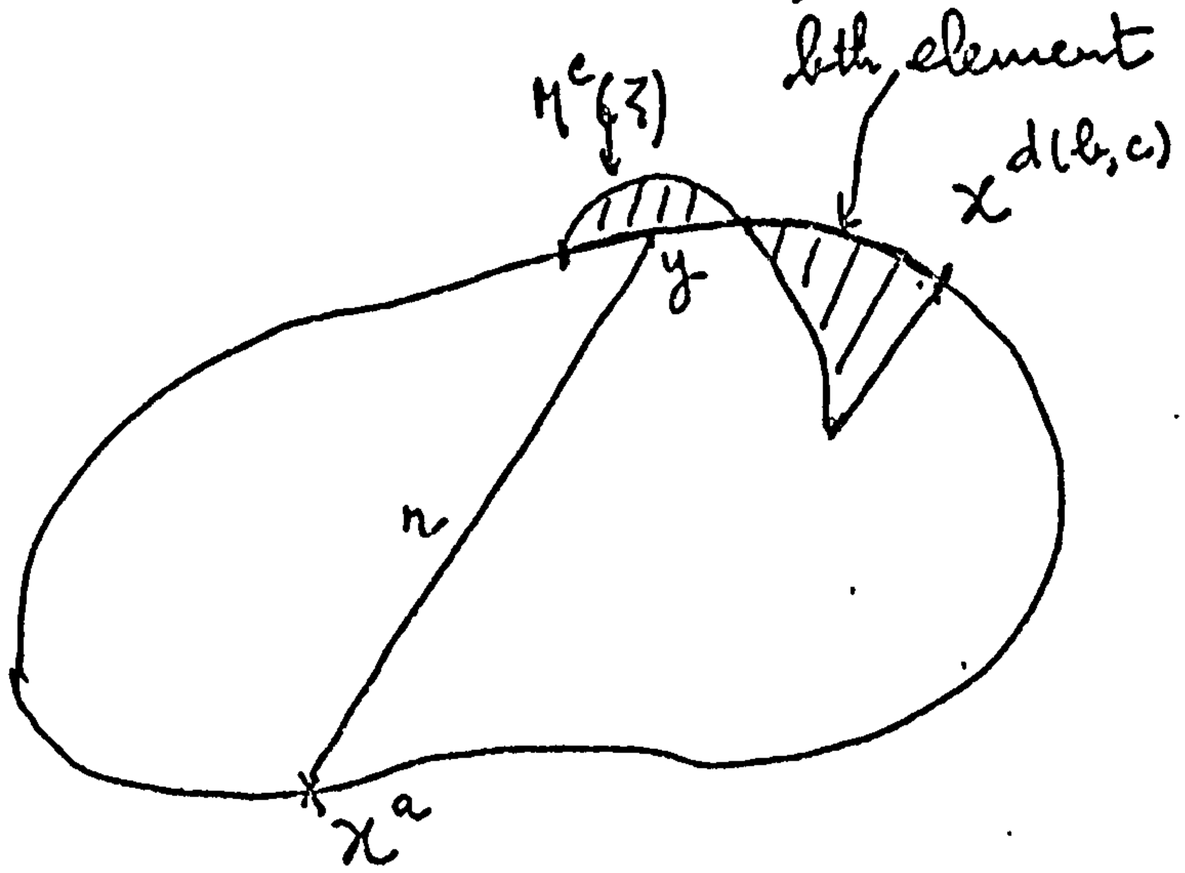


fig. 3.3.1 - parameters of equation 3.3.3

The equation coefficients are sums of integrals of kernel - shape function products. These integrals are evaluated using Gaussian quadrature formulae (Stroud and Secrest (60)). It is necessary to consider separately the case in which the node x^a is one of the nodes of the element to be integrated over, and that in which it is not. Where x^a is not a node of the element, the integrand varies smoothly over the interval and the Gaussian formulae with weight function equal 1.0 may be used :

$$\int_{-1}^{+1} f(\xi)\, d\xi \simeq \sum_{i=1}^{k} A_i f(\xi_i) \qquad (3.3.4)$$

This formula integrates exactly any polynomial of degree

$\leq 2k-1$

Let r be the distance between x and y (fig. 3.3.1). Where x^a is a node of the element, and $d(b,c) \neq a$, the integrand still varies smoothly because, whilst the kernel is $O(\log r)$ or $O\left(\frac{1}{r}\right)$, the shape function $M^c(\xi)$ is $O(r)$. Where $d(b,c) = a$ however, the procedure must be modified, because the integrand can no longer reasonably be approximated by a polynomial in ξ. To integrate $U'_{\alpha\beta} M^c$ (3.3.3), the Gaussian formulae with weight function $\log\left(\frac{1}{\eta}\right)$ is used :

$$\int_0^1 f(\eta) \log\left(\frac{1}{\eta}\right) d\eta \simeq \sum_{i=1}^{k} A_i f(\eta_i) \qquad (3.3.5)$$

...

This formula integrates exactly any polynomial of degree $\leq 2k-1$ multiplied by $\log \frac{1}{b}$, so can be expected to give good results. In the application of formula (3.3.5), it is necessary to carry out a linear transformation of the intrinsic coordinate to obtain the interval (0,1). Where x^a is not at an extremity of the element, two applications of the formula are required.

For the case $d(b,c) = a$, there exists no quadrature formula suitable for the calculation of the Cauchy principal value of the integral of $T'_{\alpha\beta} M^c$ (3.3.3). However, there is no need to evaluate separately such integrals and the coefficient $c'_{\alpha\beta}$ of the free term ; so the leading diagonal submatrix of coefficients of u'_β is calculated using the fact that the stress field corresponding to a rigid body translation is zero. Writing the integral equation for this case in global coordinates :

$$C_{ij}(x^a) + \sum_{b=1}^{p} \sum_{c=1}^{n} \delta_{ad} \int_{S_b} T_{ij}(x^a, y(\xi)) M^c(\xi) J(\xi) d\xi$$
$$= -\sum_{b=1}^{p} \sum_{c=1}^{n} (1-\delta_{ad}) \int_{S_b} T_{ij}(x^a, y(\xi)) M^c(\xi) J(\xi) d\xi \qquad (3.3.6)$$

where

$$\delta_{ad} = \begin{cases} 1 & \text{if } a = d \\ 0 & \text{if } a \neq d \end{cases}$$

...

Equation 3.3.6 gives the components of the leading diagonal submatrix in the global coordinate directions ; to obtain the components in the directions $l_{\alpha i}(x^a)$ the transformation (3.3.2) is used.

During the construction of each pair of equations, the integrals to be calculated directly are placed in the matrix and second member. All second members are calculated simultaneously.

Where $u'_\beta(x^d)$ is the unknown, the integral of $T'_{\alpha\beta} M^c$ is placed in the matrix whilst the integral of $U'_{\alpha\beta} M^c$ is multiplied by the known function $t'_\beta(x^d)$ and placed in the second member. Where $t'_\beta(x^d)$ is the unknown, the integral of $U'_{\alpha\beta} M^c$, sign reversed, is placed in the matrix, and that of $T'_{\alpha\beta} M^c$, sign reversed, is multiplied by the known function $u'_\beta(x^d)$ and placed in the second member.

Meanwhile, the global components of the leading diagonal submatrix (equation 3.3.6) are summed. When the integration is completed, this is transformed (equation 3.3.2) and placed in the equations. If $u'_\beta(x^a)$ is the unknown, the submatrix is placed on the leading diagonal, whilst if $t'_\beta(x^a)$ is unknown, it is multiplied by the known function $u'_\beta(x^a)$ and placed in the second member, sign reversed.

The equation coefficients are scaled, to obtain a non-dimensional and therefore numerically stable system. No scale factor is applied to the integrals of $T'_{\alpha\beta} M^c$, because these are already nondimensional ; the integrals of $U'_{\alpha\beta} M^c$ are multiplied by the modulus of elasticity and divided by the greatest dimension of the structure; given values of displacement are divided by the greatest dimension and given values of traction are divided by the modulus of elasticity.

The resulting system is that which would be obtained if the unit of distance were the greatest dimension and that of stress the modulus of elasticity. It is of course necessary, after solution, to multiply the calculated u by the greatest dimension and the calculated t by the modulus of elasticity.

If structural symmetry is used, certain unknowns are eliminated because they are zero. These unknowns are the displacements of and traction at nodes on a plane of symmetry, in the direction across that plane. In practice, the eliminations are effected by zeroising the corresponding matrix rows and columns and placing the value 1.0 on the leading diagonal.

4. - Reduction of the system of equations

The matrix is fully populated, and non symmetric. In general, there are several second members, each corresponding to a load case. By virtue of the scaling of coefficients, numerical stability is such that the system may be reduced by Gaussian elimination, without iteration on the residues, provided that the floating point arithmetic is done in IBM double precision or CDC 6600/7600 series single precision (about 15 decimal places). However, for all but the most trivial problems, the matrix is too large to retain in core. The program for plane elastic analysis is written to solve problems involving up to 150 nodes, or 300 equations ; although there do exist computers with enough core to hold the resulting 90,000 coefficients, the retention in core of such an array during reduction is undesirable because, the monopolisation of core by one or two large programs being inefficient, the price paid per second of central processing time is set higher for large programs than for small ones. In addition, it is undesirable to write a program that is incapable of expansion, or that will run only on a few large machines. It is therefore necessary to store the equations on a disk file, and order the reduction in such a way that the input-output time, which must also be paid for, is low. The block solver of the program for plane elastic analysis (Appendix 1,

...

subroutine IDSØL) is designed to satisfy these requirements.

In the plane analysis program, the matrix is written to a random accessfile, record length 900 coefficients. As many complete lines of matrix as possible are put in each record ; evidently there are always at least 3 equations per record, and the maximum wastage of space in a record is less than 25 %. The second members, for a maximum of 5 load cases, are in core ; the forward reduction of all load cases takes place at the same time as that of the matrix. Were the reduction of second members to be done afterwards, it would be necessary to create a file of pivots, which for a non-symmetric matrix are not numerically equal to the coefficients of the reduced matrix, as they are for the symmetric matrices obtained by the finite element method.

There are five arrays in core :

The "slow block" A (9000)

The "slow block output buffer" AØ (900)

The "fast block" FI (900)

The "fast block output buffer" FØ (900)

The second members BT (1500).

A flow chart of the block solver is shown in fig. 3.4.1.

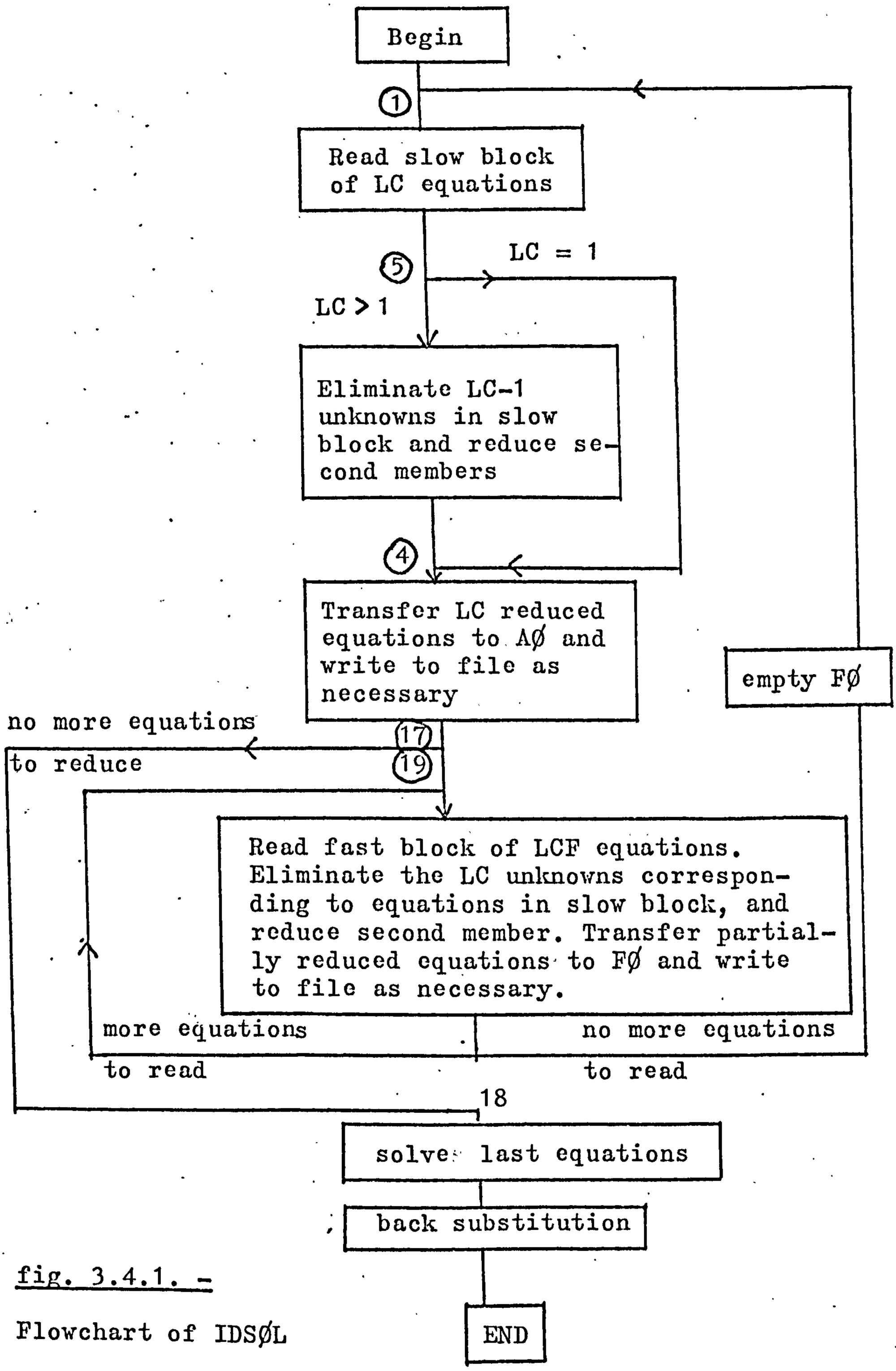
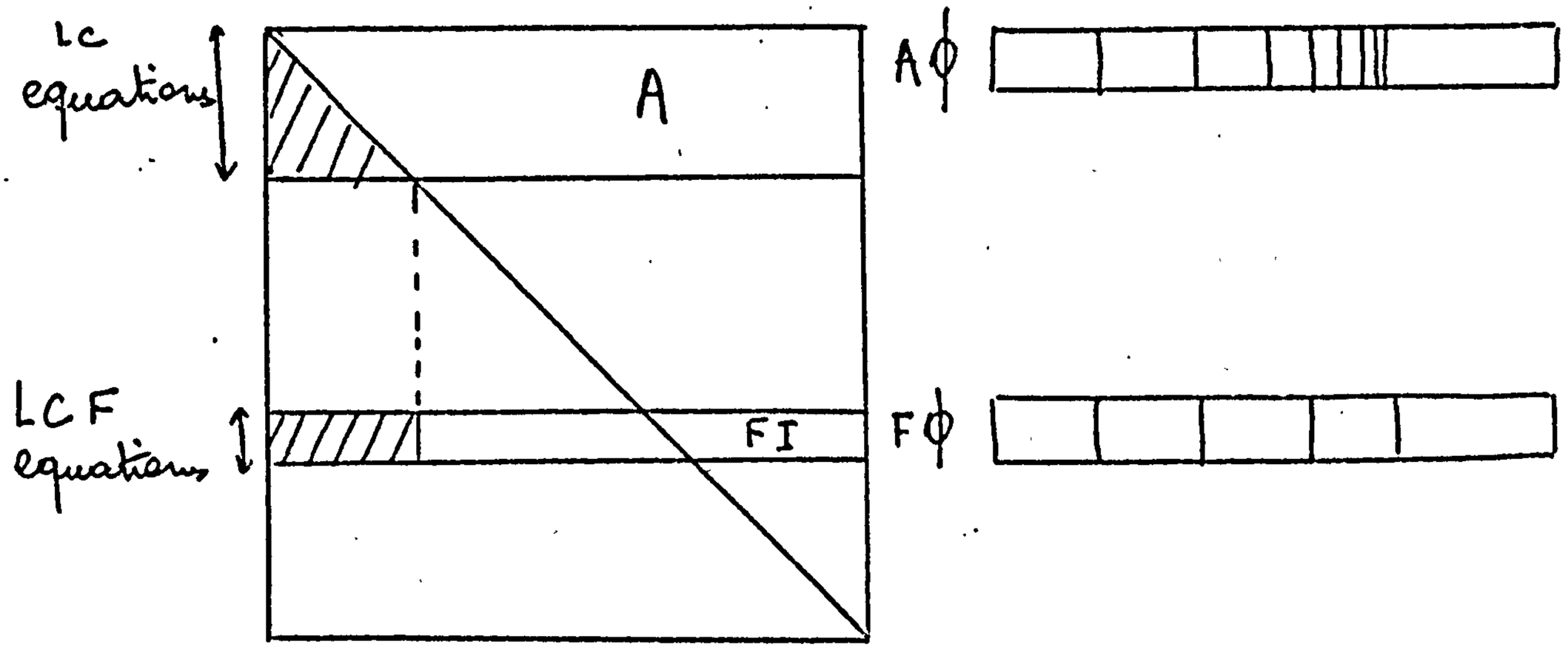


fig. 3.4.1. -

Flowchart of IDSØL

The principle of the block solver is best illustrated by diagrams of the matrix at two stages of reduction. In fig. 3.4.2 (a) is shown the data being treated at some moment during elimination of the unknowns corresponding to the first slow block.

a]



b]

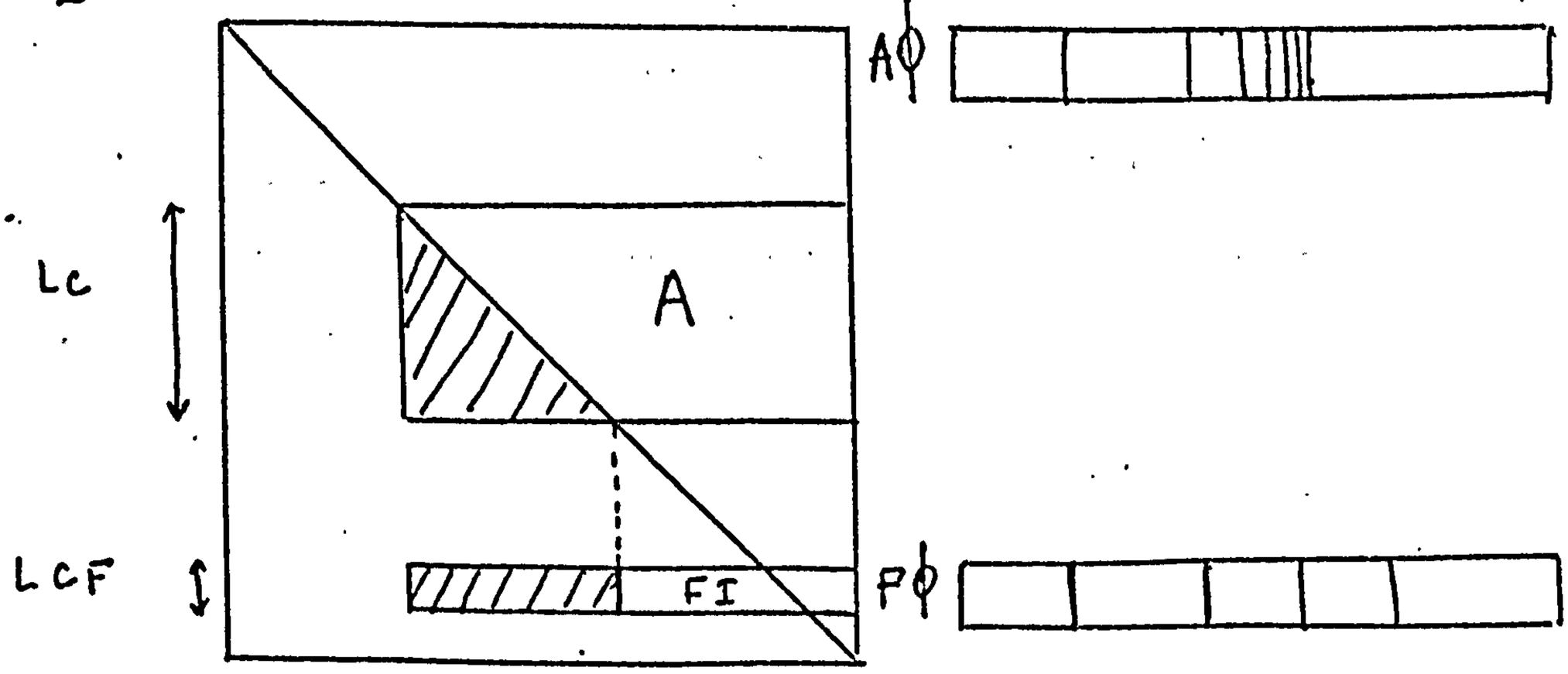


fig. 3.4.2 - elimination of unknowns

...

The slow block A has been read and reduced, and the reduced equations (unshaded area of A) have been written to the file via the buffer AØ. In general, there remain a few equations in AØ at any given time. The shaded area of A contains pivots for the elimination of the first Lc unknowns in subsequent equations. Several fast blocks have already been partially reduced, and some of the partially reduced equations, shorter than the unreduced ones, remain in the output buffer FØ. During the elimination for the fast block shown, the coefficients in the shaded area are eliminated, and afterwards the partially reduced equations (unshaded area of FI) are transfered to FØ. The latter array is emptied as necessary ; in general the reading of FI and writing of FØ take place asynchronously, and the number of equations per file record is increased. This reduces the input - output time.

In fig. 3.4.2 (b), the situation at some moment during elimination of the unknowns corresponding to the second slow block is shown. The slow and fast blocks contain more equations than previously.

Eventually, it is possible to put all the remaining equations into the slow block, and the forward reduction is completed by calculations within that block alone.

...

Back substitution is a routine process, involving only the reading of the file of reduced equations. Input-output time is low because each time the matrix is read and rewritten, several unknowns are eliminated. The block sizes shown, 9000 for the slow block and 900 for the fast block are those of the plane analysis program ; they may be changed at will, a reduction in core used resulting in an increase of input-output time and vice-versa.

5.- Calculation of stress and displacement at interior points

Prior to calculations at interior points, the global components $u_i(x^a)$, $t_i(x^a)$ of surface nodal displacements and tractions are calculated from the given values and the solution of the system of equations, using the transformations (3.3.2). For the calculation of stress and displacement at interior points, equations (2.3.1) and (2.3.4) are discretised in the same way as is the integral equation, except in that it is unnecessary to define local directions (equation 3.3.2) :

$$u_i(x) = \sum_{b=1}^{p} \sum_{c=1}^{m} \left\{ \int_{S_b} U_{ij}(x, y(\xi))\, M^c(\xi)\, t_j(x^{d(b,c)})\, J(\xi)\, d\xi - \int_{S_b} T_{ij}(x, y(\xi))\, M^c(\xi)\, u_j(x^{d(b,c)})\, J(\xi)\, d\xi \right\} \quad (3.5.1)$$

$$\sigma_{ij}(x) = \sum_{b=1}^{p} \sum_{c=1}^{m} \left\{ \int_{S_b} D_{ijk}(x, y(\xi))\, M^c(\xi)\, t_k(x^{d(b,c)})\, J(\xi)\, d\xi - \int_{S_b} S_{ijk}(x, y(\xi))\, M^c(\xi)\, u_k(x^{d(b,c)})\, J(\xi)\, d\xi \right\} \quad (3.5.2)$$

where M^c is the shape function (3.2.4), S_b is the bth element, and $J(\xi)$ the Jacobian

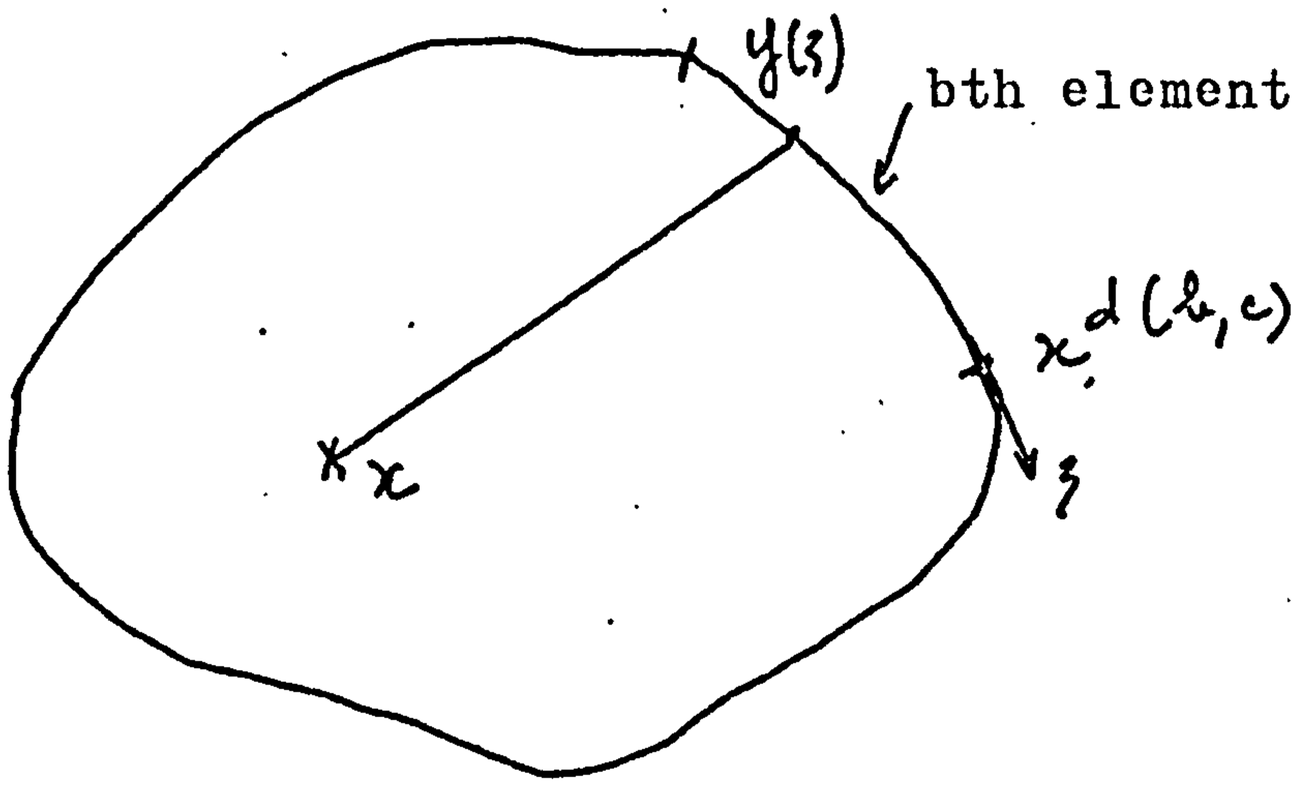


fig. 3.5.1 - parameters of equation 3.5.1.

For the evaluation of the integrals appearing in the equations (3.5.1 and 3.5.2), only the Gaussian quadrature formula with weight function 1.0 (equation 3.3.4) is required. To reduce central processing time, all load cases are treated at once and the results sorted afterwards.

6.- Examples and comparison with other methods

The plane analysis program has been used to analyse a number of problems of stress concentration and fracture mechanics. Most of the calculations have been done on a CDC 7 600. The program, of about 5 000 Fortran statements, is divided into a root segment and one primary overlay segment ; there are 10 overlays. 18 K of core is required for execution. The user can choose linear, quadratic or cubic variation of displacement and traction over each boundary segment, and 2,3, 4 or 5 Gauss integration points per segment. These options were included in the program in order to allow a study of relative efficiencies of various combinations of functional representation and precision of integration.

Three examples are shown here. They are :

a) the analysis of a gear tooth. This is a typical problem of calculation of stress concentrations in plane elasticity

b) the calculation of the stress intensity factor for a crack in a tensile test specimen. This illustrates the use of the method in plane fracture mechanics

...

c) the calculation of stresses near a rigid inclusion embedded in rubber. This demonstrates the applicability of the method to the analysis of stress in incompressible materials.

The results for the gear tooth are compared with finite element results, and those for the tensile test specimen are compared with both finite element and experimental results.

6.a) The gear tooth

The characteristics of the gear wheel considered are as follows :

number of teeth = 10

diametral pitch = 5.08

addendum modification = 0

modulus of elasticity = 21 000 daN/mm2

Poisson's ratio = 0.3

In the analysis, the loaded tooth (fig. 3.6.1.) is considered in isolation. The shaded sides are supposed to be fixed, and a line load of 40 daN/mm is applied as indicated,normal to the surface. A state of plane strain is assumed to exist, because the tooth is long compared with its width.

For the analysis by the finite element method the tooth is represented by 291 six-node isoparametric triangular elements (fig. 3.6.2), with a total of 630 nodes. In the finite element program the resulting system of 1 260 simultaneous equations is solved by a variable bandwidth Gaussian elimination algorithm, in which the matrix is held on a disc file and a sliding triangle of coefficients is resident in core.

...

In the analysis by the integral equation method, linear, quadratic and cubic variation of displacement and traction over each boundary segment were tried. For the linear and quadratic functional representation the boundary was represented by 33 segments (fig. 3.6.3), whereas for the cubic case it was represented by 33 segments and also by 13 segments (fig. 3.6.4).

Both programs were run on an IBM 360/75. Run statistics are shown in table (3.6.5). In the last column of the table is shown the calculated maximum stress near the root of the tooth, so that the precision of each analysis can conveniently be assessed. The precision is seen to improve with that of the functional representation, and the result for cubic variation, 33 segments, is probably better than that obtained by the finite element method. It should be noted that, had a program for automatic generation of finite element networks existed at the time, the quantity of data to be prepared for the finite element analysis would be of the same order as that for the integral equation analysis ; but in this case some additional computing time would be required The variation of the calculated principal stress at the surface in shown in fig. 3.6.6. No analysis gives sensible results near the line load (which is any case a fiction), so a part of the curve is missing.

...

The integral equation results just described were obtained with 5 Gauss points per segment. In fig. 3.6.7 is shown the convergence to a limit of the calculated stress as the precision of integration is increased. The results for the cubic case, 13 segments, are chosen for this test because they may be expected to be more sensitive to errors of integration than the others. It may be seen that there comes a point where it is impossible to extract further significant improvement by taking a higher order formula. A decrease of precision of integration was found not to yield great economies, so 5 Gauss points per segment became the standard choice for subsequent analysis.

The ratio input-output time/central processing time for the integral equation program was comparable with that for the finite element analysis, so the block sizes of the solver (section 3.4) are well chosen. To give an indication of roundoff error during solution, the integral equation program calculates the norm of the residues, divided by that of the second member, for each load case :

$$\varepsilon = \sqrt{\frac{\sum_{i=1}^{n} (a_{ij} x_j - b_j)^2}{\sum_{i}^{n} b_i^2}}$$

where a_{ij} is the matrix, b_i the second member and x_j the solution calculated by the program. In no case

...

considered does ε exceed 4.0×10^{-14}. IBM 360 series double precision arithmetic being done to 16 or 17 decimal places, this may be taken to show that the system of equations is very stable.

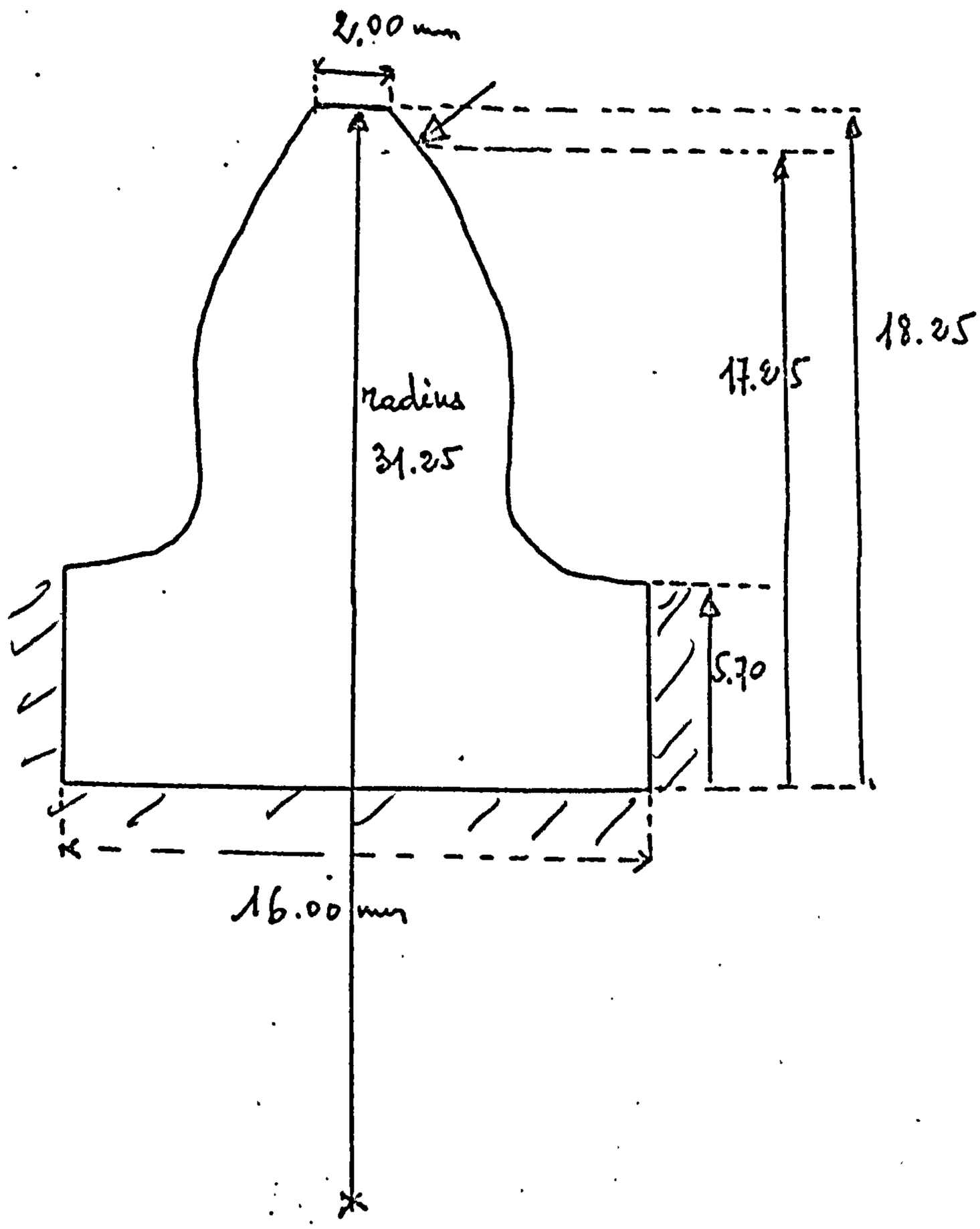
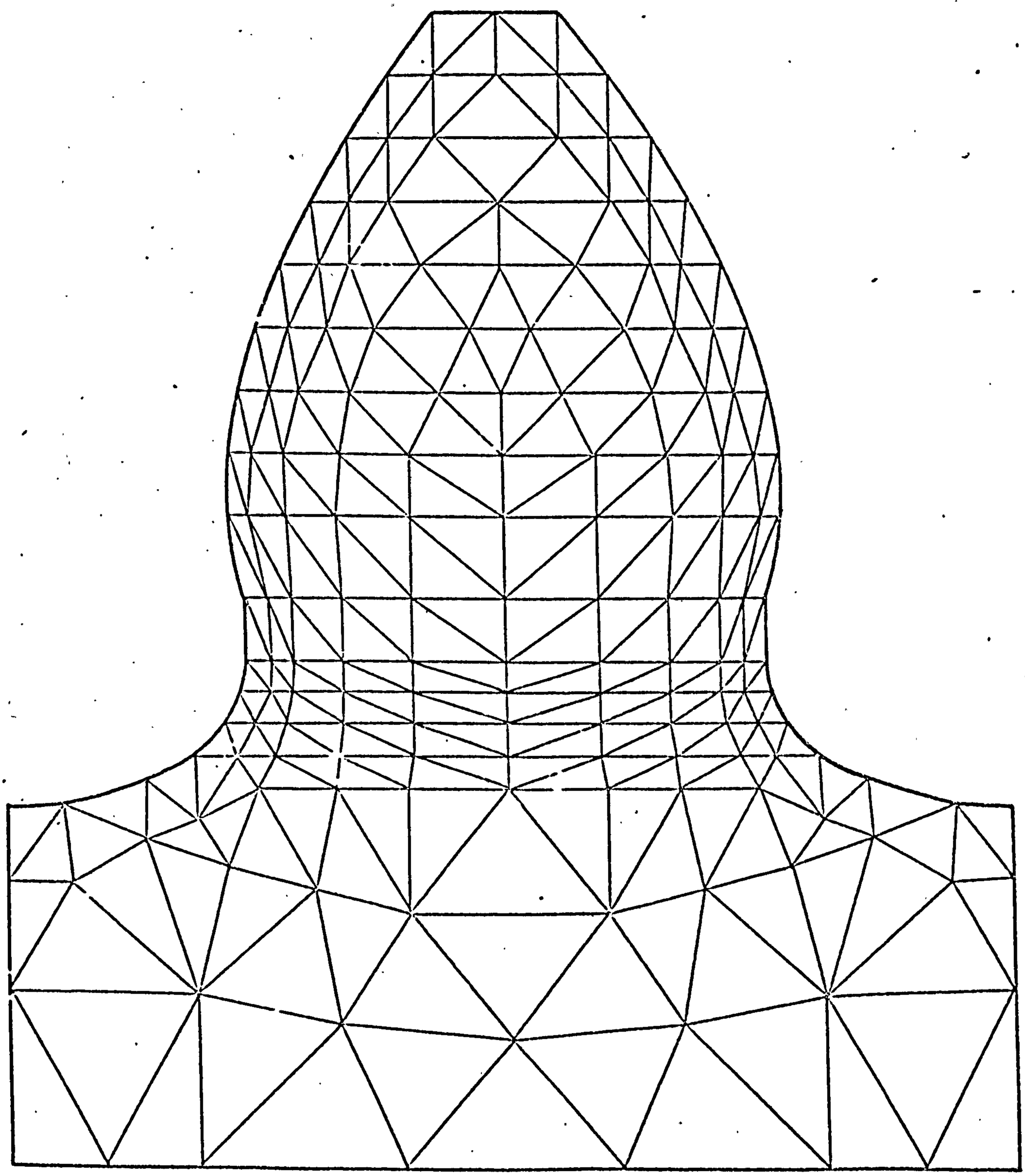


fig. 3.6.1. - idealisation of the gear tooth

...



291 elements - 630 nodes

Fig. 3.6.2 - F.E. mesh of the gear tooth

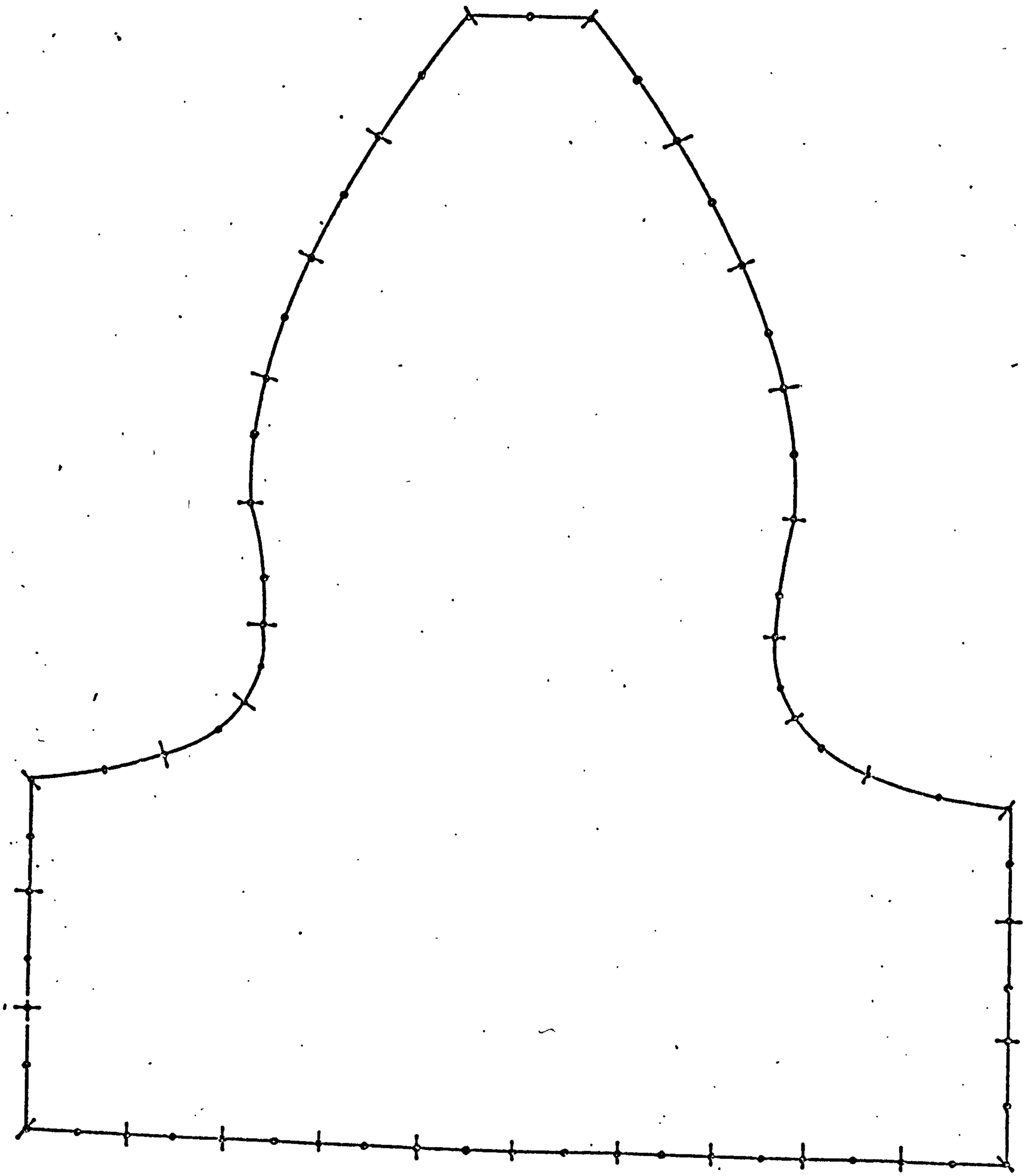


Fig. 3.6.3 - discretisation of the gear tooth. 33 segments

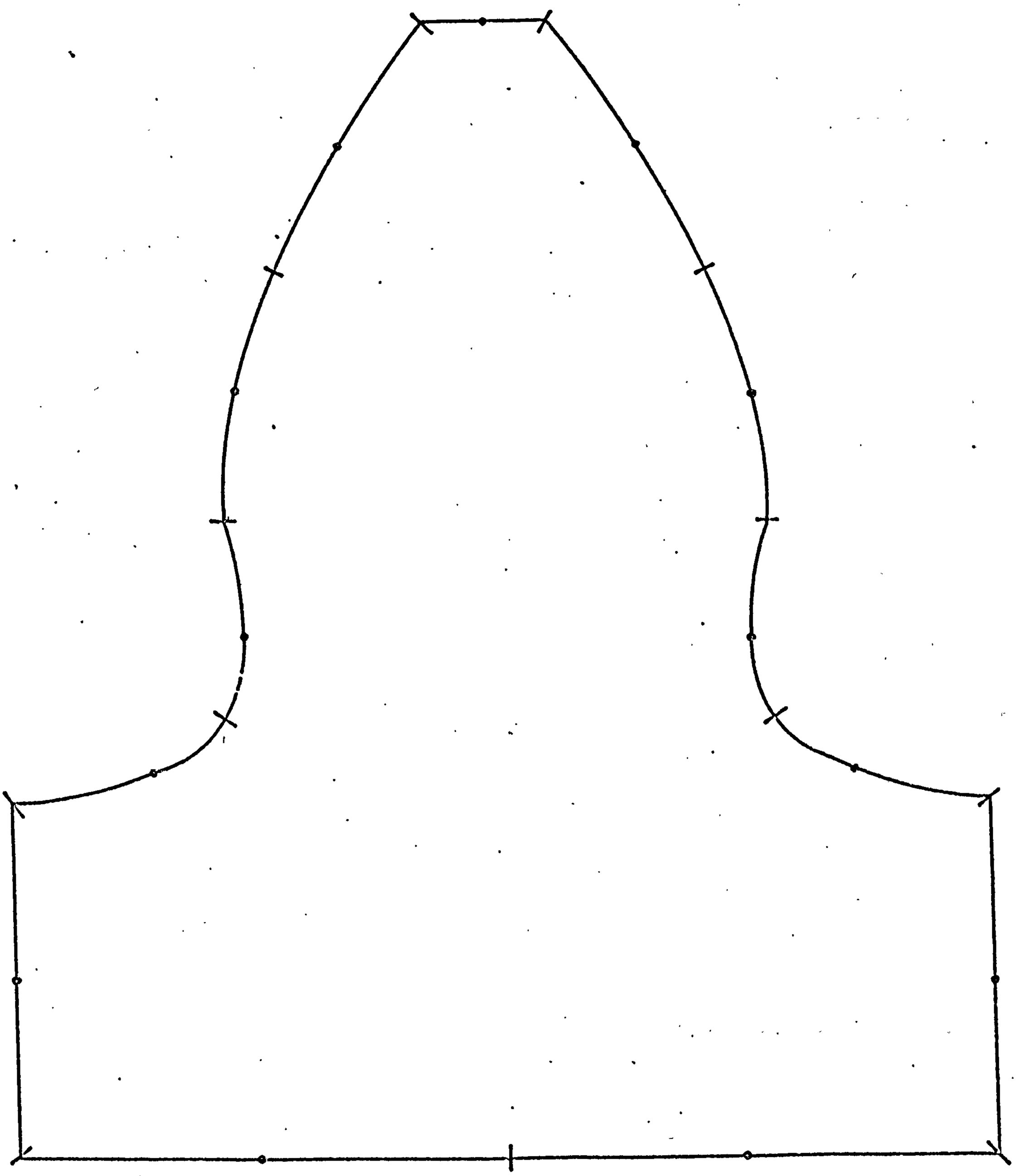


Fig. 3.6.4 - discretisation of the gear tooth : 13 segments

analysis		numbers of degrees of freedom	Program length (bytes)	run time i.o.+ c p u	data preparation time	numbers of cards of data	results maximum stress σ_{max}
F.E.		1 260	286K	173 s	24 h	500	- 43.0
Integral Equation 33 segments	linear	66	150K	22 s	3 h	35	- 42.0
	quadratic	132	150K	40 s	3 h	35	- 42.7
	cubic	198	150K	86 s	3 h	35	- 43.2
F.E.		1 260	286K	173 s	24 h	500	- 41.4
I.E. 13 segments cubic		78	150K	18s	2 h	31	- 38.1

table .6.5

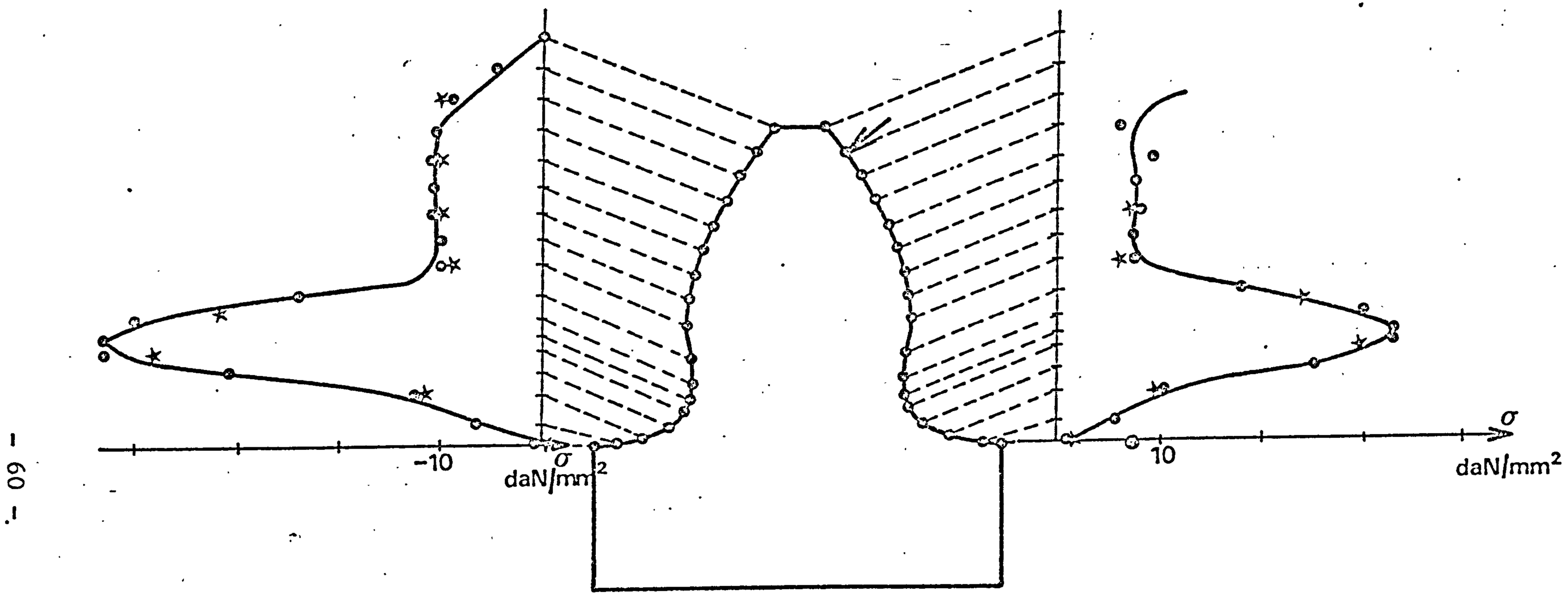
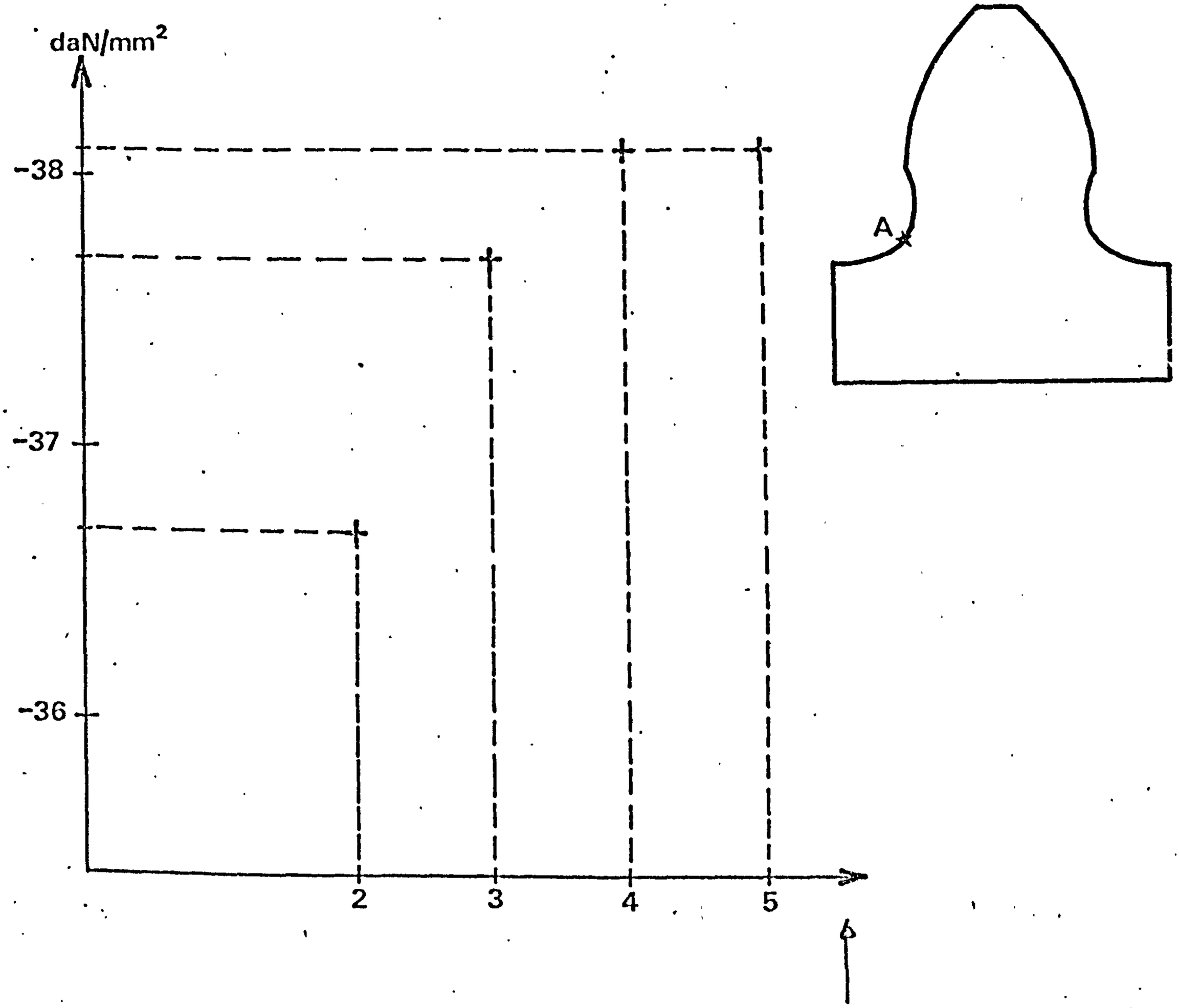


fig. 3.6.6 - variation of the principal stress at the surface



I.E. : 13 segments - cubic

Fig. 3.6.7 - convergence of results

6.b) The test specimen CT.15

The object of the analysis of the compact tensile test specimen CT 15 (fig. 3.6.8) is to calculate the variation of the stress intensity factor as a crack propagates along the plane of symmetry, according to the linear theory of fracture mechanics, and compare the results with those obtained experimentally and by the finite element method. Because the structure and the loads are symmetric, it is only necessary to consider one half of the specimen. A state of plane strain is supposed to exist, and seven different lengths of crack are considered. The elastic constants are the same as for the gear tooth, and the load of 2 000 daN acts inside the hole.

The stress intensity factor is calculated by two methods. In the first method, the stress intensity factor is obtained from the path independent integral J of Rice (70). The integral J is given by the relationship :

$$J = \int_{\Gamma} \left(w\,dy - t\,\frac{\partial u}{\partial x}\,ds \right) \qquad (3.6.2)$$

where w is the strain energy density :

$$w = \int_{0}^{\varepsilon} \sigma_{ij}\, d\varepsilon_{ij}$$

t and u are the traction and displacement along a contour Γ around the crack (fig. 3.6.9) ...

The stress intensity factor in mode I (64) may be obtained, in elastic plane strain, from the relation :

$$J = \frac{1-\nu^2}{E} K_I^2 \qquad (3.6.4)$$

In the second method, the stress intensity factor is calculated from the rate of change of potential energy with respect to crack length (64) :

$$K_I = \frac{P}{B\sqrt{W}} Y\left(\frac{a}{W}\right) \qquad (3.6.5)$$

where

$$Y\left(\frac{a}{W}\right) = \frac{B}{P}\left[\frac{E}{1-\nu^2}\frac{\partial U}{\partial\left(\frac{a}{W}\right)}\right]^{\frac{1}{2}} \qquad (3.6.6)$$

In equations 3.6.5 and 3.6.6 U is the potential energy per unit thickness, P the load, and B the thickness of the specimen. The network, for the finite element analysis, consisting of 223 elements and 505 nodes, is shown in fig. 3.6.10. For the analysis by the integral equation method, the boundary, including the hole, is represented by 28 segments with cubic functional variation. In fig. 3.6.11 is shown the discretisation for $\frac{a}{W} = 0{,}5$; certain charges are made for each crack length in order to group a number of very short segments around the crack tip, to represent the rapid functional variation in this zone.

...

In fig. 3.6.12 are shown the 27 cards of input data required to define the problem. Card A is the title and on card B are stated the choice of functional representation and integration ; whether the analysis is to be carried out if there are no errors of syntax in the data, or a data check only is to be performed ; and lastly the existence of symmetry with respect to coordinate axes. Cards C define the cartesian coordinates of nodes of segments on the outer contour. Cards D define the interior nodes at which results are required to calculate the integral J. Card E moves the origin of coordinates to the centre of the hole, and card F defines the cylindrical coordinates of the nodes of segments defining the hole. Card G defines the segments of the outer contour, and card H those defining the hole. Card I specifies that the segments on the plane of symmetry, and to the right of the end of the crack, be fixed in the vertical direction, and card J fixes one node in the horizontal direction. This is necessary because, for integral equations as for finite elements, the structure must be incapable of rigid body movements or the system of equationswill be singular. Card K is the load case title, cards L define the load, and card M marks the end of the data.

The results obtained by calculation of the integral J are compared with those obtained experimentally on 5 specimens by IRSID (64), and those calculated by the finite element method, in table 3.6.13. The results obtained by the calculation of J and by differentiation of the potential energy are compared with those given by the ASTM (71) and those calculated by the finite element method in fig. 3.6.14 and table 3.6.15. The results obtained by the calculation of J are about as close to the experimental results as those given by the finite element method, whereas the calculation of $\frac{\partial U}{\partial(\frac{a}{W})}$ gives results which are even closer. The difference between the accuracy of the two sets of integral equation results is probably due to appreciable error in the calculation of stress and displacement inside the body but near the surface ; in this case, the kernels D and S (equations 2.6.7 and 2.6.8) vary very rapidly over part of the surface and the 5 - point integration formula is not sufficiently accurate.

It is difficult to compare the program execution times; the finite element program took longer to execute than the integral equation program, but in the former case the same network (fig. 3.6.10) was used for all seven cracklengths considered, so over considerable areas the

...

elements were unnecessarily small. It is, however, arguable that it was the convenience of data preparation for the integral equation program that permitted the modifications for each crack length to be carried out reasonably quickly and therefore economically. In any case, on the CDC 7 600, only 3.780 seconds CP time, plus 0.665 seconds I/O time were required for execution of the integral equation program.

The large variation of boundary segment length in the discretisation (largest 11,25 mm, shortest 0,5 mm) does not induce numerical instability : the CDC 7 600 performs floating point arithmetic to about 15 decimal places, and the norm $\mathcal{E}$ (equation 3.6.1) for the case $\frac{a}{W} = 0.5$, for example, is 9.2×10^{-12}

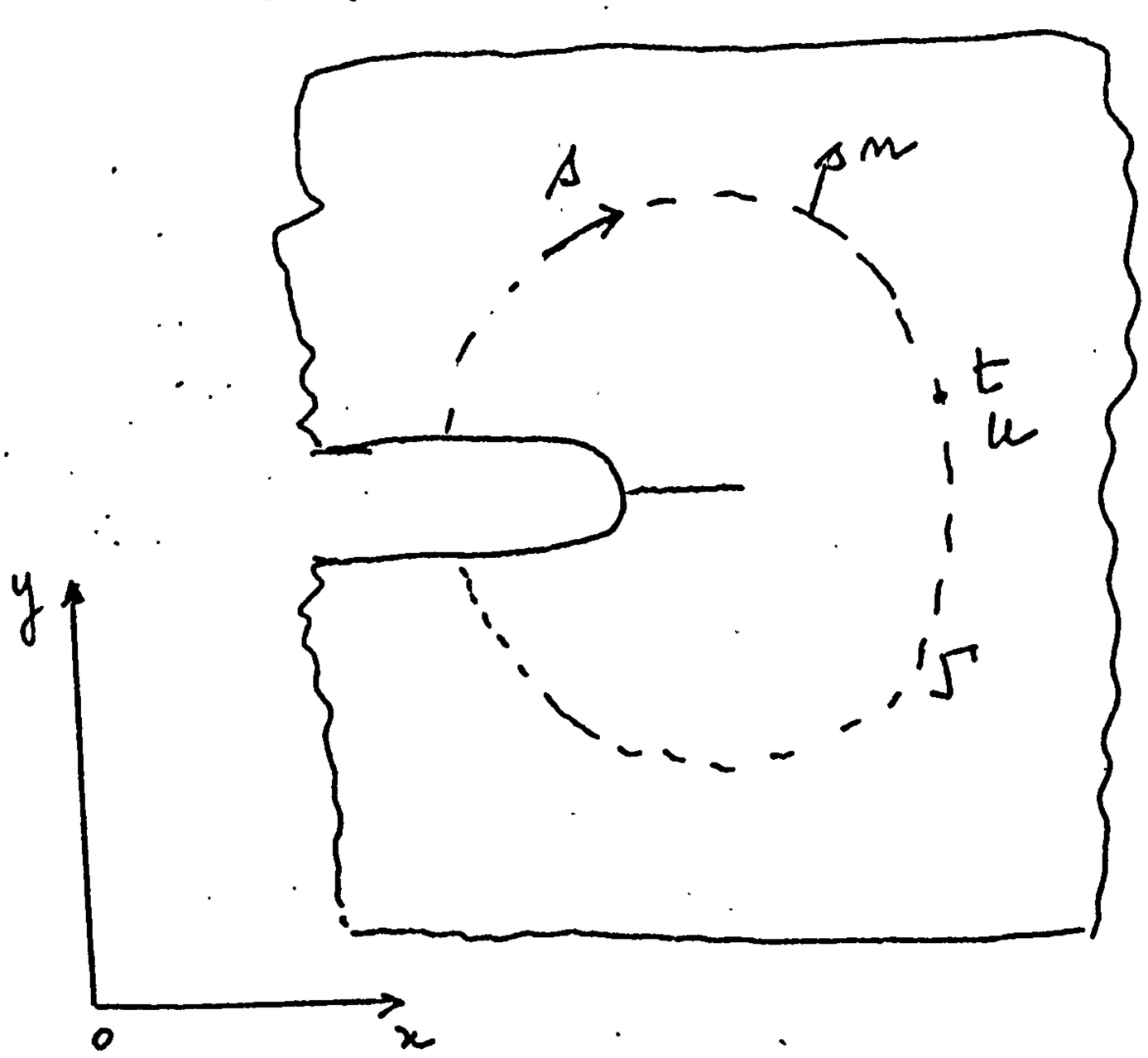
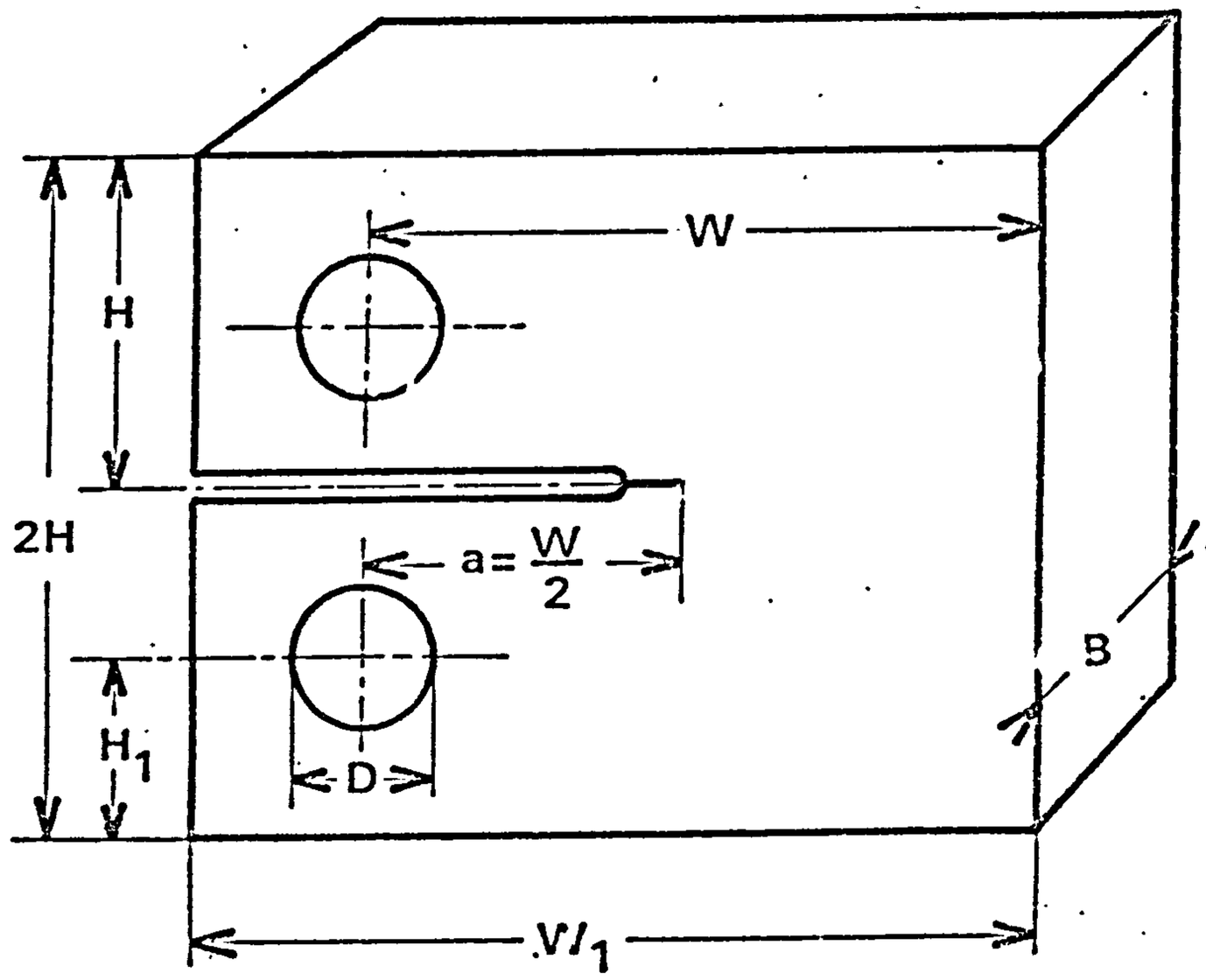


fig. 3.6.9 - the contour integral of Rice



W = 2B
a = B
H = 1,2 B
D = 0,5 B
W_1 = 2,5 B
H_1 = 0,65B

fig. 3.6.8 - the specimen CT 15

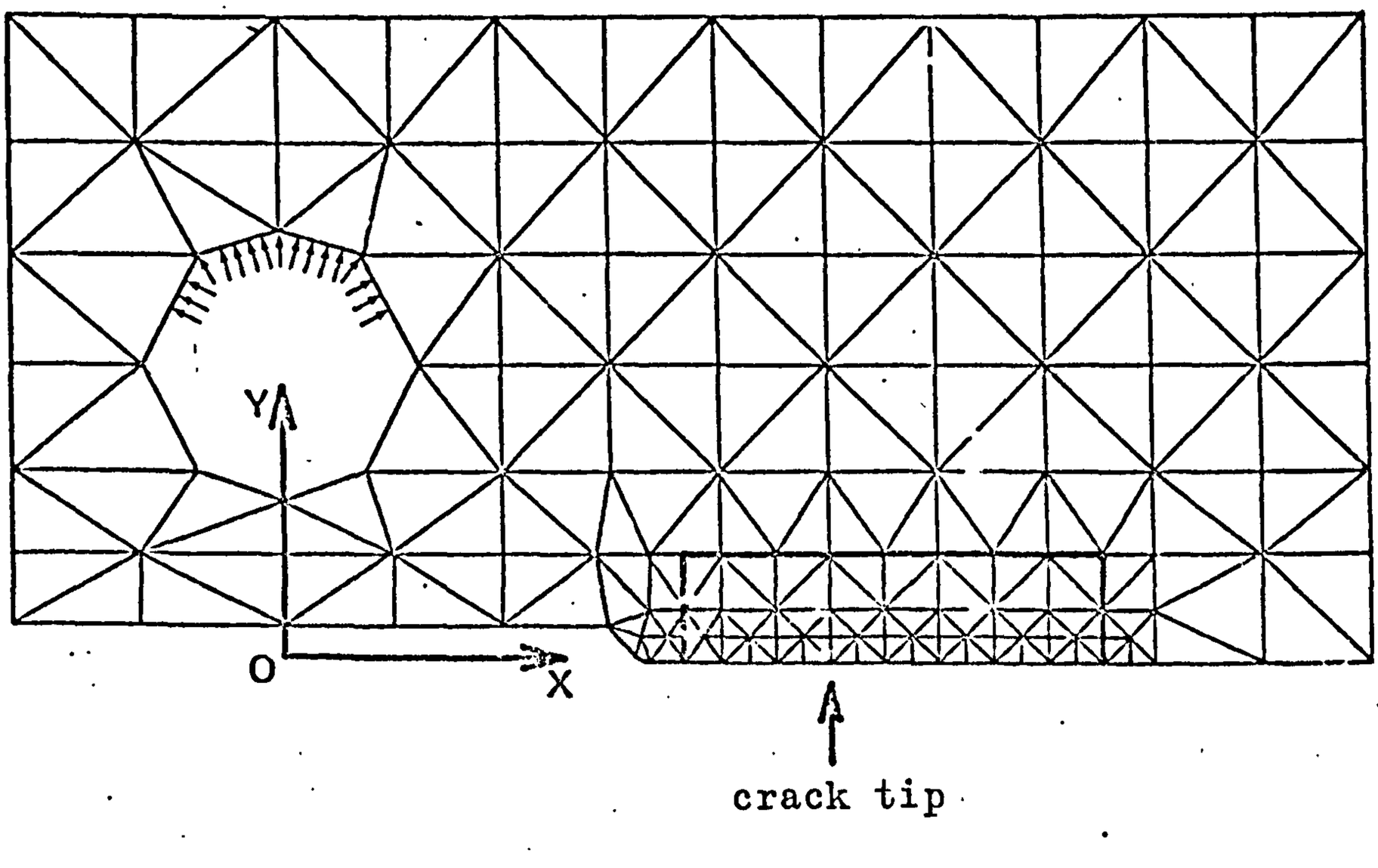


Fig. 3.6.10 - the finite element network

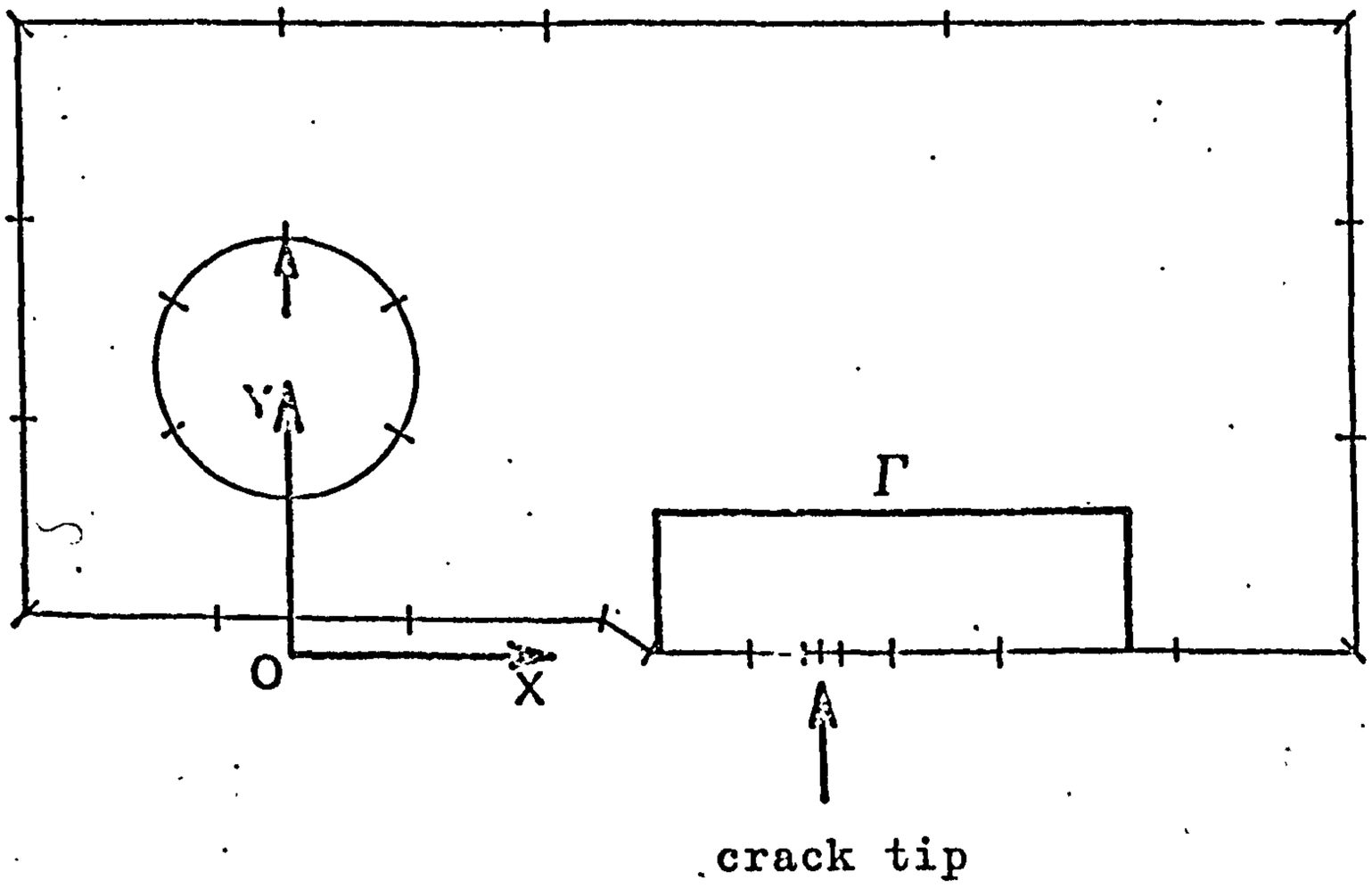


fig. 3.6.11 - the integral equation network

fig. 3.6.12 - data for the specimen CT 15

Kic hbar × $\sqrt{\text{mm}}$

specimen number	load at failure daN	I.R.S.I.D.	F.E.	I.E.
1	1 660	194	189	192
2	1 934	226	221	223
3	2 088	224	239	242
4	2 133	247	241	247
5	2 225	260	254	257

fig. 3.6.13

a/W \ Y	A.S.T.M.	F.E. (G)	I.E. (G)	I.E. (J)
0.40	7.323	6.943	7.295	6.993
0,45	8.337	8.321	8.253	8.235
0.50	9.603	9.651	9.538	9.505
0.55	11.260	11.183	11.242	11.161
0.60	13.540	13.574	13.530	13.429
0.65	16.778	16.618	16.523	16.500
0.70	21.427	21.436	21.430	20.951

fig. 3.6.15

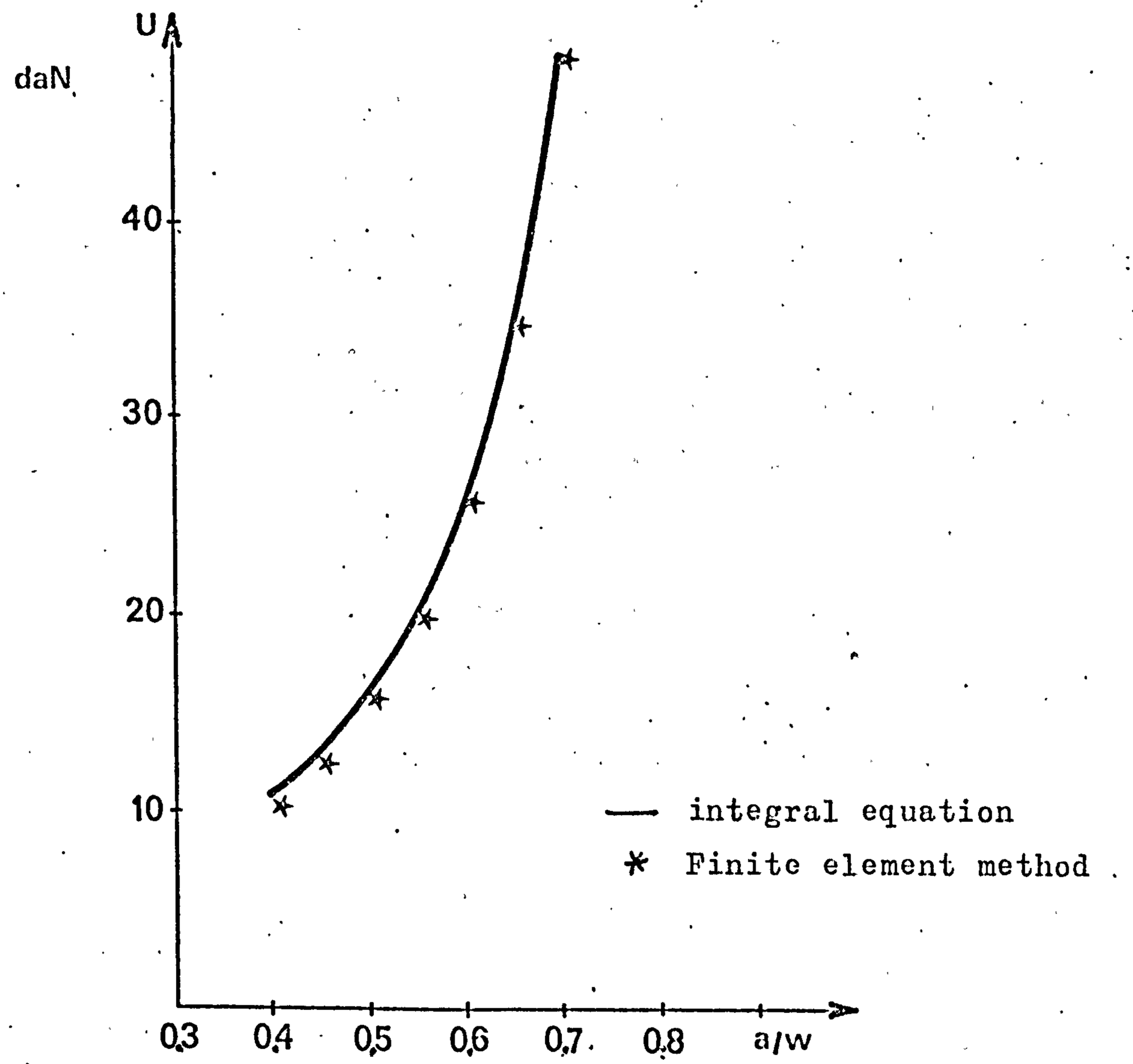


fig. 3.6.14 - Variation of potential energy per unit thickness

6.c) The rubber sheet with rigid inclusions

The final example is chosen to demonstrate the applicability of the method to the analysis of incompressible materials. The problem is to calculate stresses in a sheet of rubber 4.0 mm thick, with parallel steel wires of diameter 2.0 mm at 3.00 mm centres (fig. 3.6.16) when subjected to an overall imposed strain of 1.667×10^{-2} perpendicular to the wires. The modulus of elasticity is 2.0 daN/mm2 and two values of Poisson's ratio are considered : $\nu = 0.45$ et $\nu = 0.50$.

The rubber is considered to be in a state of plane strain ; and for the purposes of the analysis the wires are taken to be perfectly rigid. As the modulus of elasticity of steel is 21 000 daN/mm2, this is a reasonable assumption. There are vertical planes of symmetry passing between and by the centre of each wire, in addition to the horizontal plane of symmetry bisecting the sheet. It is therefore only necessary to consider the area marked ABCDE in fig. 3.6.16. The finite element program being incapable of treating the case $\nu = 0.50$, no finite element results are presented for comparison . Instead, to check precision, two sets of integral equation results are compared : those obtained by representing the contour by a relatively coarse discretisation, and those obtained by

...

taking a fine representation, with twice as many segments as in the coarse one. The two discretisations are shown in fig. 3.6.17. The cubic functional representation is chosen, and in order to avoid the need to discretise the sections BC and EA of the contour, the symmetries with respect to the axes X and Y are used (fig. 3.6.17). The section CD of the contour is subjected to a displacement of 0.025 mm in the X direction.

Some results of the analysis for $\nu = 0.45$ are shown in fig. 3.6.18(a). At the scale of the graph the results for 9 segments and those for 18 are indistinguishable. The norm of residues $\mathcal{E}$ (equation 3.6.1) is 8.0×10^{-14} for 9 segments, and 1.4×10^{-13} for 18. Execution of the program on the CDC 7600 required the following $\mathcal{CP}$ and I/O times :

9 segments 1.151 sec. $\mathcal{CP}$, 0.355 sec. I/O

18 segments 3.541 sec. $\mathcal{CP}$, 0.414 sec. I/O

The same results for $\nu = 0.5$ are shown in fig. 3.6.18 (b). In this case there is some difference between the results for 9 and 18 segments, amounting to not more than 5 % of the maximum stress. However, there is very little difference between the displacements : the vertical displacement at E is calculated to be $- 2.162 \times 10^{-2}$ mm

...

(9 segments) and -2.161×10^{-2} mm (18 segments). This disparity of precision is probably only the result of the inherent instability of calculation of stress for an incompressible material. The norm of residues $\mathcal{E}$ for $\nu = 0.5$ is of the same order as that for $\nu = 0.45$.

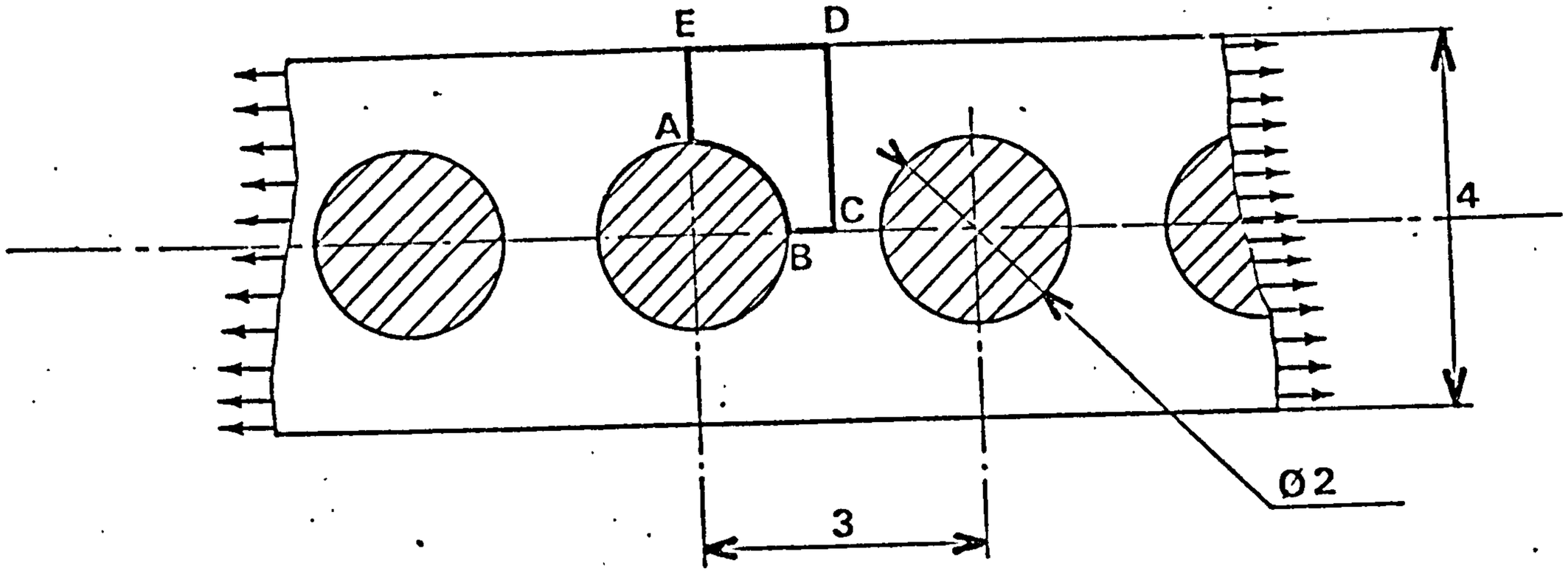
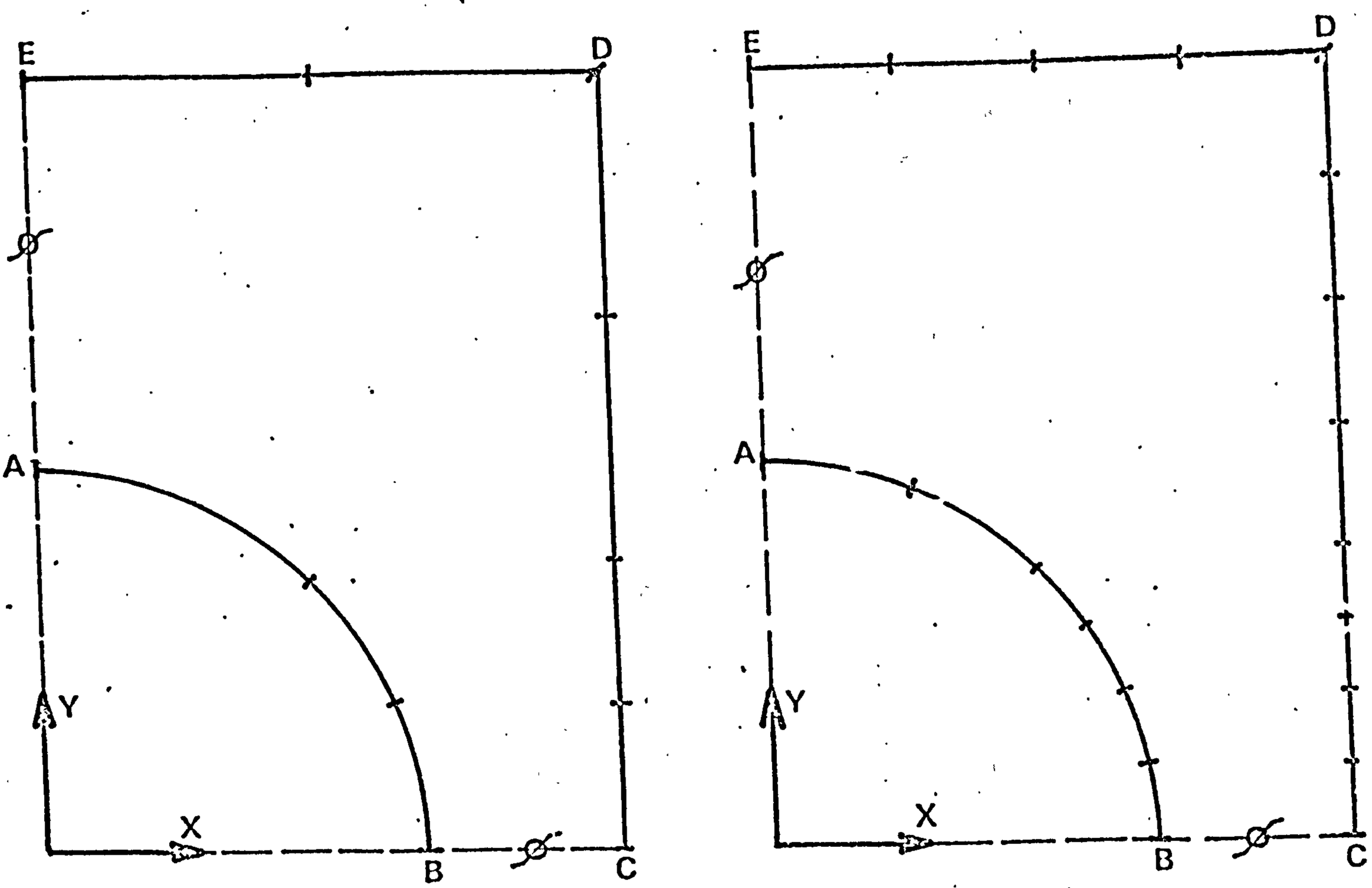


fig. 3.6.16 - sheet of rubber with embedded steel wires



a) coarse discretisation (9 segments) b) fine discretisation (18 segments

fig. 3.6.17 - discretisation for integral equation analysis

fig. 3.6.18 (a)

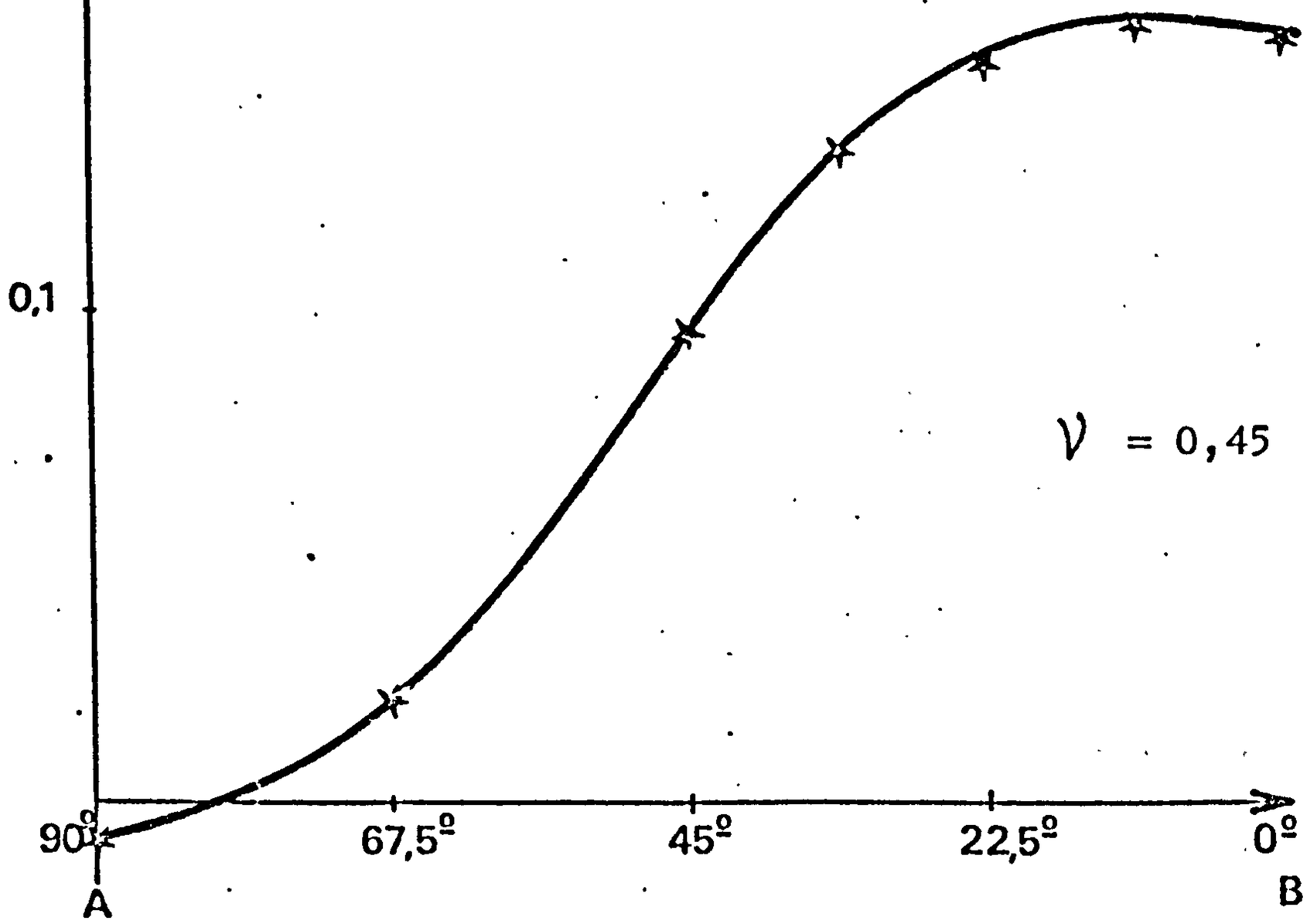


fig. 3.6.18 (b)

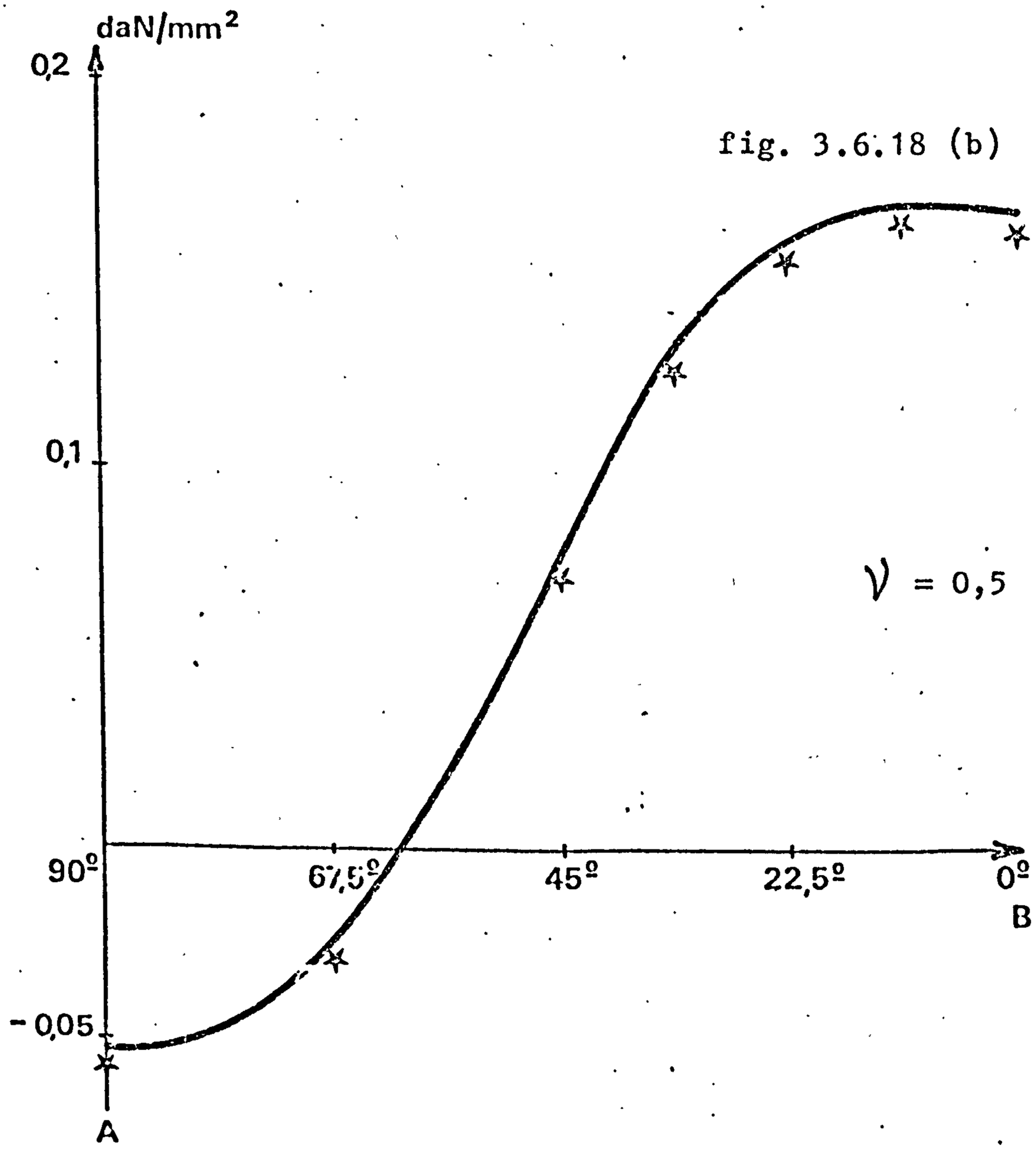


fig. 3.6.18 - variation of σ_{xx} along the curve AB

7.- Discussion

The strategy adopted for the numerical formulation of the plane problem incorporates certain techniques and manipulations, for which to date there exists no published information on efficiency. The parametric representation of geometry and functions, for example, although extensively used in finite element calculations, appears to be new to the field of research into boundary integral equations. From the experience obtained with the program for plane analysis, it is possible to make an assessement of these techniques ; this assessement can then be used as the basis of the numerical formulation in three dimensions.

The parametric representation of geometry facilitates the calculation of normals to the surface, required for the derivation of the kernel T, and of the stress tensor at the surface (section 3.2). In the formulation only quadratic variation of segment geometry was tried, whereas displacement and traction could be considered to vary linearly, quadratically or cubically with respect to the intrinsic coordinate. For the examples treated, no difficulty was found in adequately representing the contour by the parabolic segments ; in practice nearly all boundaries consist of straight lines and arcs of circles ; linear representation of geometry would be unsuitable for the circles, and progression to cubic

...

geometry would yield no improvement in accuracy for either straight lines or circles. In addition, it has been found from experience with finite elements that, if cubic geometry is used, small errors in the input data can lead to large distortions of the segment.

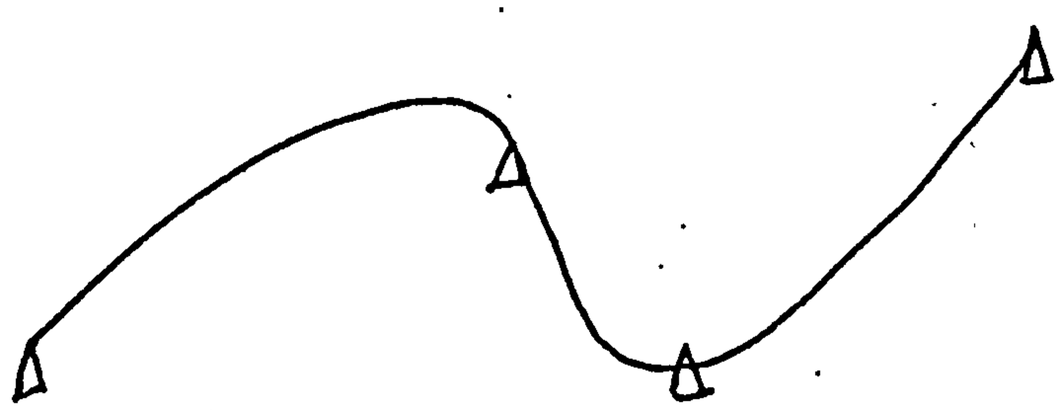


fig. 3.7.1 - segment with distorted cubic geometry

The provision for symmetry with respect to the coordinate axes (example 3.6.C) was effective in reducing both the quantity of input data required, and in most cases the execution time was reduced. In certain extreme cases, in which the reduction in the number of boundary segments thereby achieved is small, the extra integration necessary for construction of the matrix outweighs the advantage obtained by having a smaller system of equations to solve.

It is not clear, from the results of tests, which functional representation is the most economical. Sometimes, good results may be obtained cheaply by representing the boundary by a

...

large number of segments, and choosing linear variation, but the results so obtained are usually far from being uniformly accurate over the structure. The linear variation is incapable of representing certain rapid local variations. However, the option of linear variation was very useful during debugging of the program, because the diagnostics are simpler than for quadratic or cubic variation. Also, linear variation can be useful for testing a discretisation intended for quadratic or cubic variation. The quadratic and cubic functional variations appear to be more efficient than the linear, and give results of more uniform accuracy. The cubic representation is good in the case of bending.

In the discretisation of the integral equations (section 3.3) an approximation was made at corners of the structure at which the displacements of both the adjacent segments are given (fig. 3.7.2).

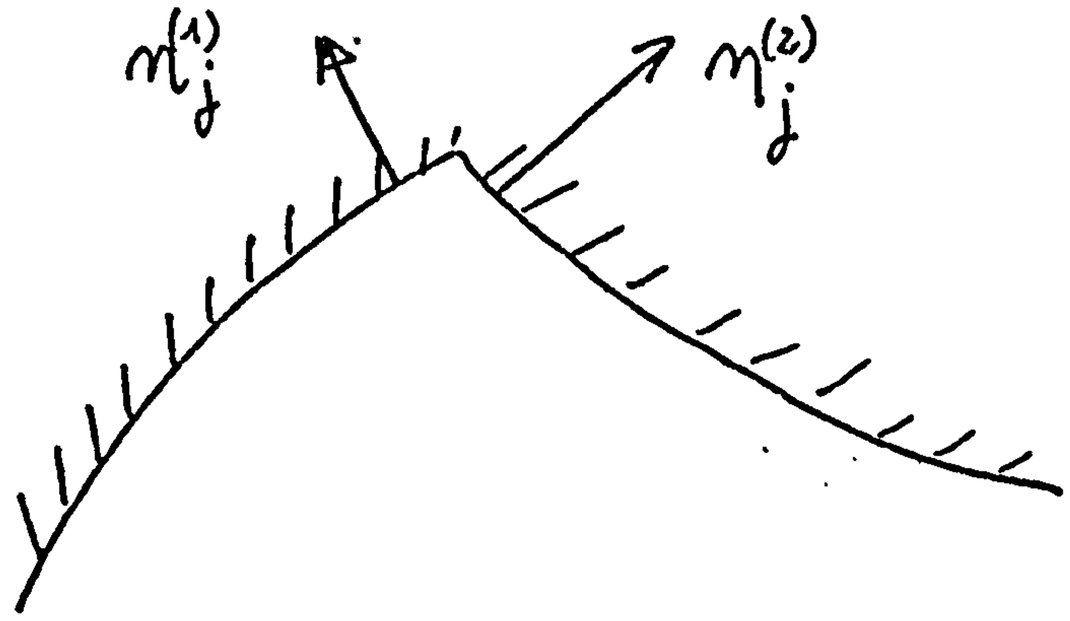


fig. 3.7.2 - corner with the adjacent segments fixed.

Let σ_{ij}, the stress tensor, be continuous at the corner. Then the limiting values of the tractions on each segment as the corner is approached are

$$t_i^{(1)} = \sigma_{ij}\, n_j^{(1)}$$
$$t_i^{(2)} = \sigma_{ij}\, n_j^{(2)} \qquad (3.7.1)$$

In general, $t_i^{(1)} \neq t_i^{(2)}$. But in the discretisation, it has been assumed that $t_i^{(1)} = t_i^{(2)}$, because the two limiting values of traction are not considered as separate unknowns. In most practical situations, such as that of the gear tooth considered in section 3.6 (a), this approximation does not lead to any error because, the given displacements of the adjacent segments being zero, $t_i^{(1)} = t_i^{(2)} = 0$. In the case in which the given displacements of the adjacent segments are not zero, local errors are introduced but the accuracy of results at points some distance from the fixed corner is not significantly affected.

It was noted (section 3.6. (b)) that stress and displacement at interior points near the boundary are not calculated very accurately, because of the rapid variation of the kernels D and S over the segments near the point, and the consequent inadequacy of the integration formula. Similar significant error must arise in the calculation of the matrix and second members, for example where there is rapid variation

...

of segment length. In the caseshown in fig.3.7.3 the kernel-shape function product varies very rapidly with respect to

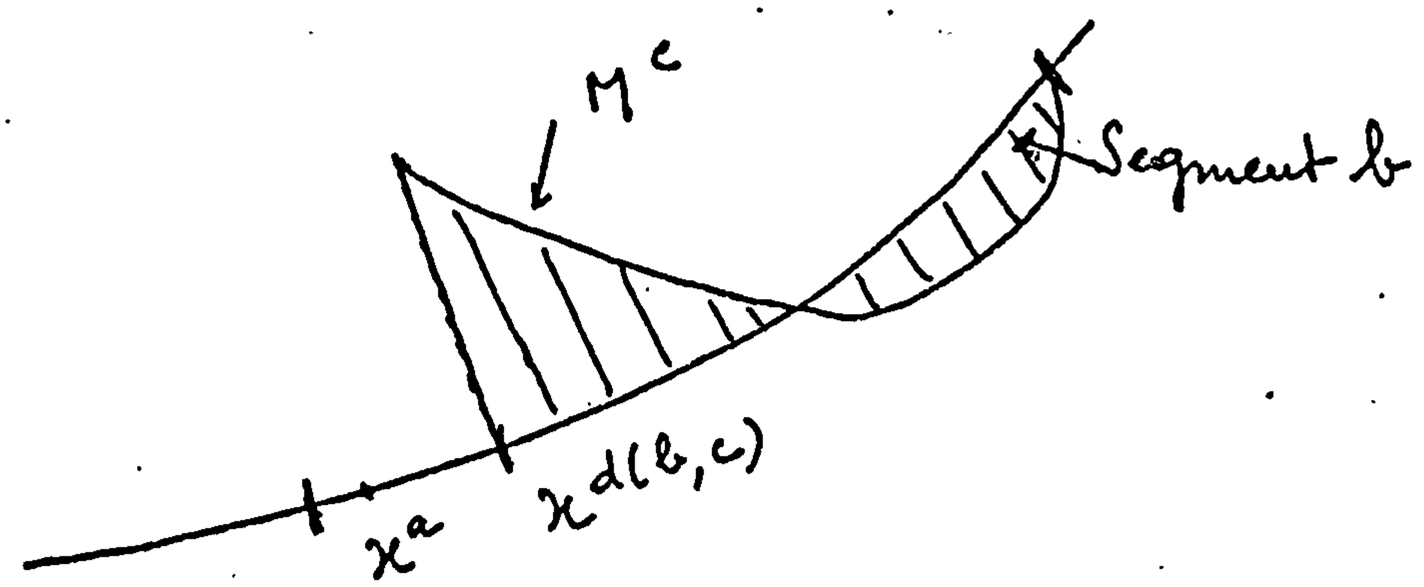


fig. 3.7.3 - critical case of integration

(notation equation 3.3.3)

the intrinsic coordinate. In the program, the number of integration points per segment is fixed beforehand ; if this number is set high enough to give good results for such a case, an unnecessary amount of computing time is spent on integrating over small segments, or segments distant from the singularity χ^a. It would therefore be desirable to include in the program,logic to calculate automatically, from upperbounds for error of integration, the formula to use, depending upon the length of the segment and its distance from the singularity χ^a. In this way, precise integration could be performed at low cost.

In the program for plane analysis, there is a contribution both to the matrix and to the second member for every term of the double sum , equation 3.3.3. However, they may

exist terms of this sum for which both of the integrals of kernel-density products should be placed in the second member. Let us consider the case in which $x^{d(b,c)}$ is the common node of two segments, one free in the direction β and the other fixed in that direction (fig. 3.7.4). Then the unknown in the direction β at x^d is the traction $t'_\beta(x^d)$, so the integral over S_b of the product $T'_{\alpha\beta} M^c$ must be multiplied by the known displacement $u'_\beta(x^d)$ and placed in the second member. However, although the traction $t'_\beta(x^d)$ is the unknown, the limiting value of the traction on S_b as x^d is approached is known, so the integral over S_b of the product $U'_{\alpha\beta} M^c$ should be multiplied by the known limiting value of traction and placed in the second member. The matrix coefficients of $t'_\beta(x^d)$ should consist of integrals over the fixed segment adjacent to x^d only.

The effect of this error of numerical formulation is twofold : the calculated value of stress at such points as x^d, fig. 3.7.4, is too low, and the structure as a whole is calculated to be a little stiffer than it really is.

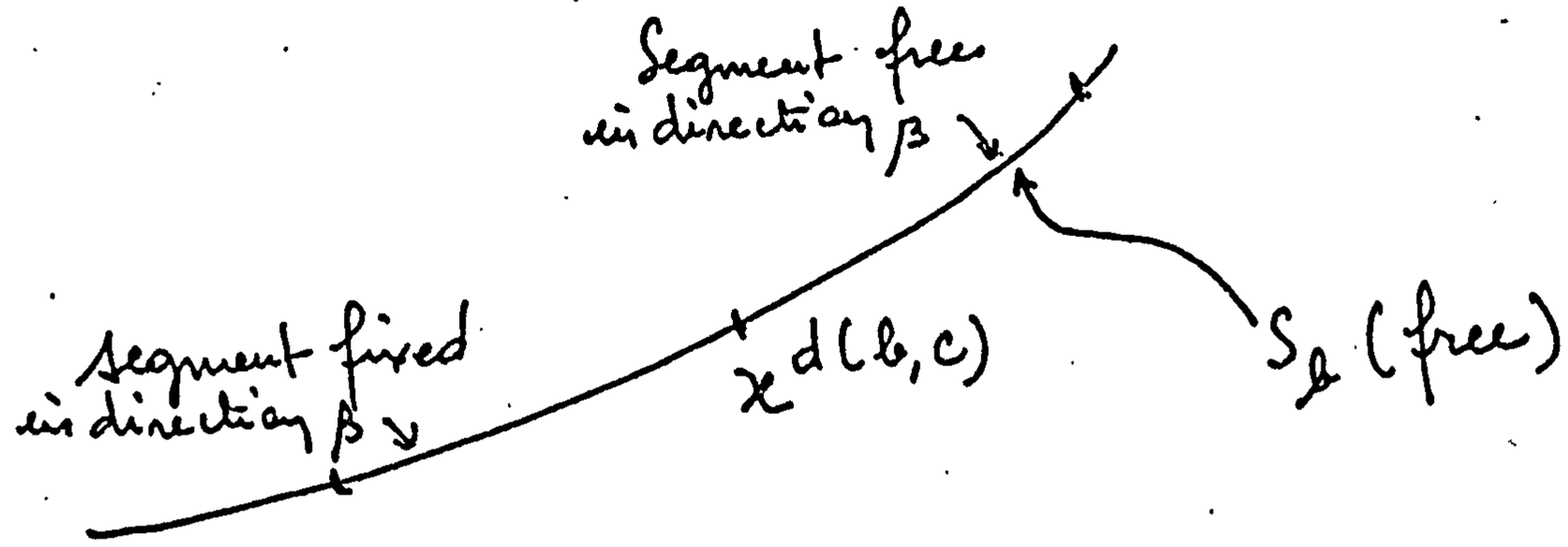


fig. 3.7.4 - node shared by fixed and free segment (notation equation 3.3.3) ...

As may be seen from the examples, (section 3.6), the system of equations is very stable, irrespective of the value of Poisson'sratio. For the systems solved, of order up to 200, the norm of residues (equation 3.6.1) is between 100 and 1 000 times the precision of floating-point arithmetic. Evidently the scaling of matrix coefficients to render them nondimensional solves the problem of numerical instability, noted by some authors. The residues are so small that iteration on them (Cruse, 9) will be unnecessary even for much larger systems.

The reduction of the equations becomes relatively expensive for systems of order $2q > 200$ (corresponding to about 33 cubic or 50 quadratic segments), because, whilst the time required for integration is proportional, roughly, to $(2q)^2$, that for reduction is proportional to $(2q)^3$.

Evidently this is a problem for the three-dimensional analysis, for which the order of the system of equations will be considerably greater than for two-dimensional problems. It is interesting to note that, in most cases, there are pairs of nodes of the structure which are relatively far from one another (fig. 3.7.5), and to which, in consequence, there correspond very small matrix coefficients (fig. 3.7.6).

...

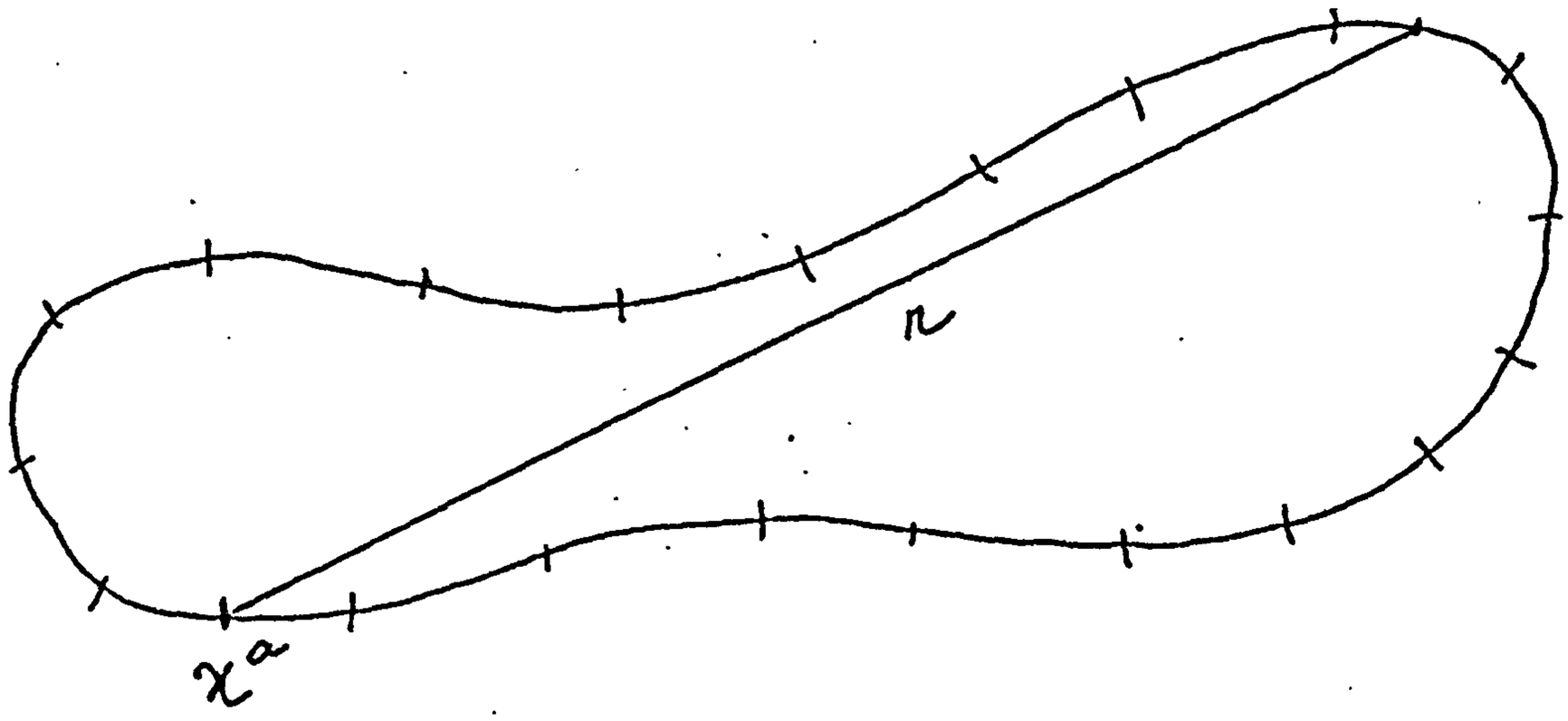


fig. 3.7.5 - elongated structure

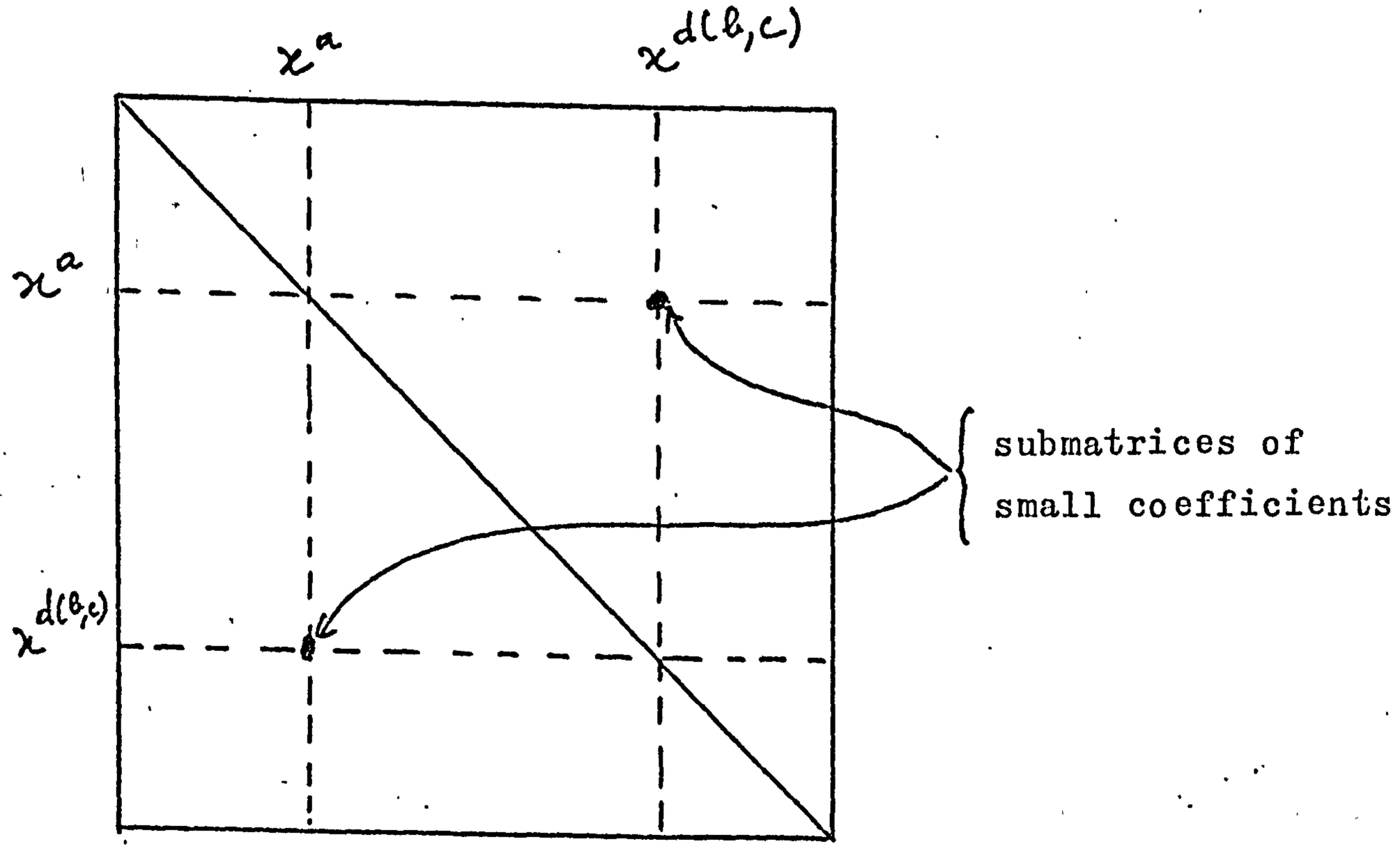


fig. 3.7.6 - matrix corresponding to
discretisation of fig. 3.7.5.

The coefficients are small because the kernels of the integral equation (equation 3.3.3) are $O\left(\frac{1}{r^2}\right)$ and $O\left(\frac{1}{r}\right)$. Evidently the processing of these small coefficients makes a considerable contribution to the high computing time for the construction and especially the reduction of large systems, for structures such as that shown in fig. 3.7.5. It would be desirable to devise a formulation in which such small coefficients are absent.

This can be done by dividing the structure into subregions, these being subdivisions of the elastic body, as shown in fig. 3.7.7., and writing the integral equation.

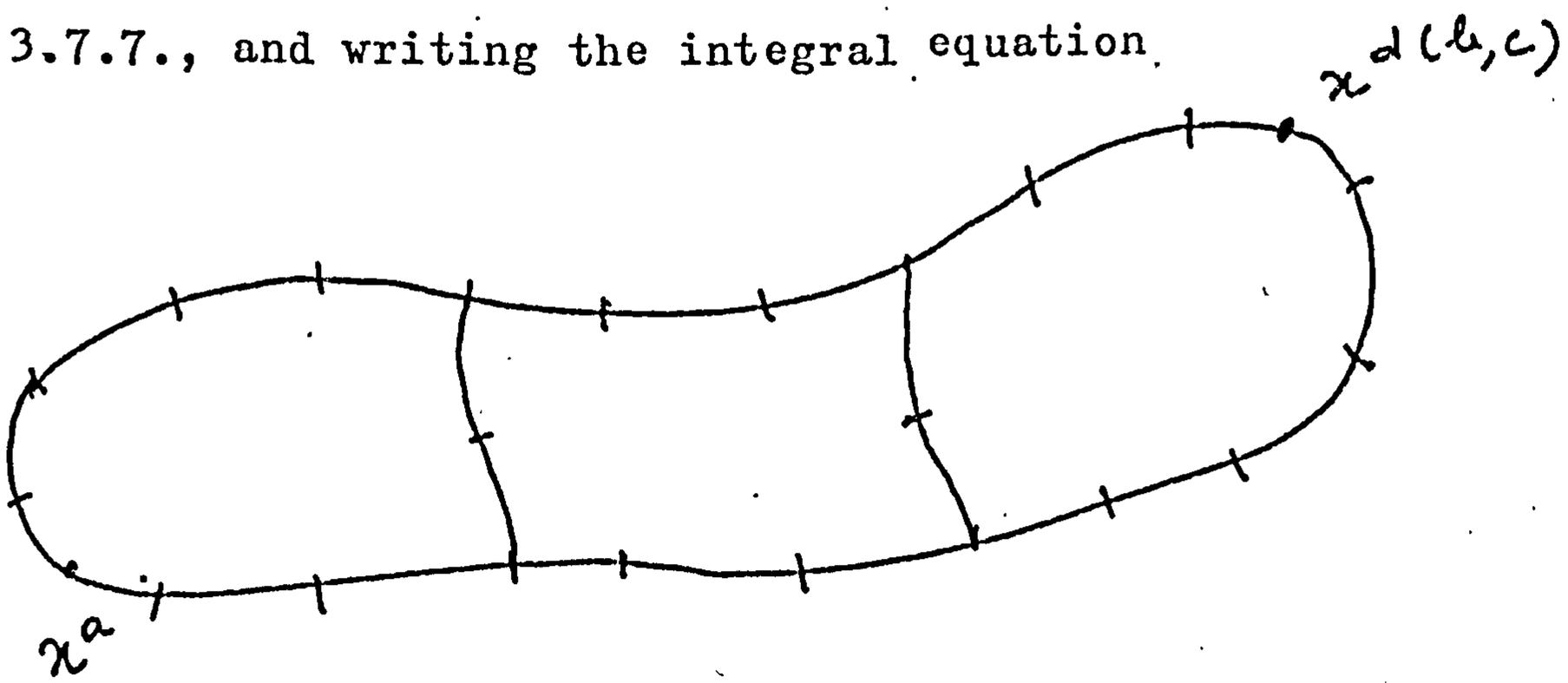


fig. 3.7.7 - elongated structure, divided into subregions

for each subregion . Using the discretisation described in section 3.3, a matrix of the form shown in fig. 3.7.8 is obtained .

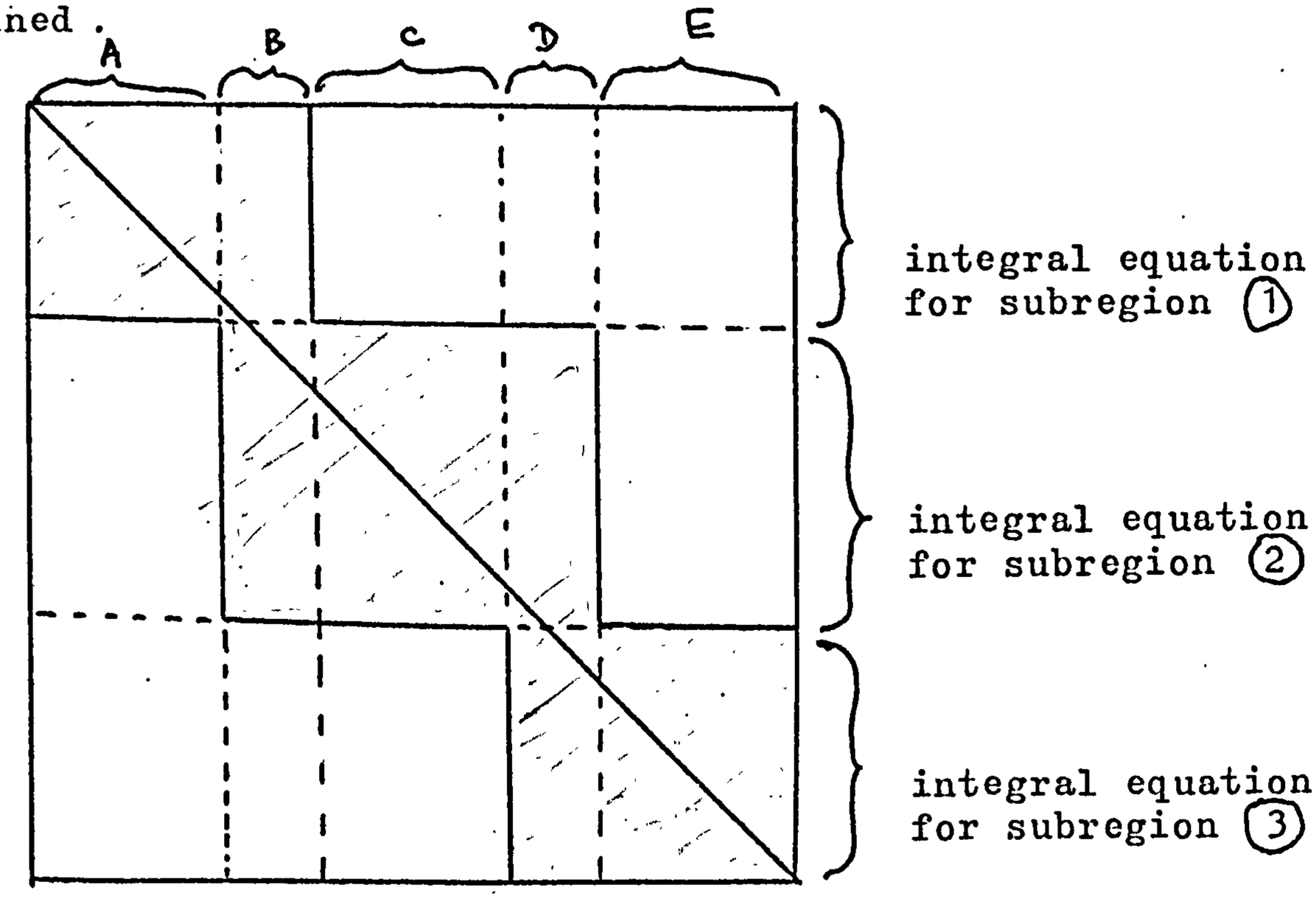


fig. 3.7.8 - matrix corresponding to discretisation of fig. 3.7.7.

In fig. 3.7.8., columns A correspond to nodes belonging to subregion (1) only, columns B to nodes belonging to subregions (1) and (2) , etc ... The matrix is now of banded form, the small coefficients shown in fig. 3.7.6 now being zero. There are fewer coefficients to calculate, and the time for reduction is no longer proportional to the cube of the order of the system. However, corresponding to nodes on the interfaces between subregions, some equations are added to the system. As an incidental benefit of division

...

into subregions, it becomes possible to analyse structures consisting of several different elastic materials.

IV - NUMERICAL FORMULATION OF THE THREE DIMENSIONAL PROBLEM

1.- Review of previous formulation

In previous formulations of the three dimensional problem, as for those of the plane problem, relatively simple numerical techniques have been used. The work of principal interest is that of Cruse (9,53). In the formulation presented in the earlier paper (9), the surface is represented by plane triangular elements, and it is supposed that u and t are constant over each element. The integral equation is written for the centroid of each element, and the resulting system of equations, after scaling of certain coefficients by the shear modulus, is solved by Gaussian elimination with iteration on the residues. Symmetries with respect to the coordinate axes may be taken into account, and as in the two dimensional analyses by Cruse, the stress tensor at the surface points is calculated from traction and tangential strain, this being obtained by differencing displacements of adjacent elements.

In the second paper (53) Cruse presents an improved formulation, still with plane triangular elements, but with linear variation over each element of u and t. The unknowns are now associated with the nodes, and the system of equa-

...

tions is obtained by writing the integral equation at these points. The integration is performed analytically, due respect being paid to the definition of the Cauchy principal value. The coefficient of the free term of the equation no longer necessarily equals $\frac{1}{2}\delta_{ij}$; Cruse calculates the free term implicitly, by considering rigid body translation, as is done in the formulation in two dimensions of chapter 3 (equation 3.3.6). The rest of the formulation is basically unchanged.

Cruse treats the example of a fracture specimen, and gives results obtained using the program with u and t constant over each segment, and using the improved program in two ways : by analysing the entire specimen, and by analysing only one half, taking account of symmetry. Results and run statistics are presented to show that the improved formulation is more efficient than the original. However, the run times are still rather high considering the relative simplicity of the problem and Cruse states in conclusion (53) that "Future efforts to improve the accuracy of the BIE method should concentrate on even higher order variation of the boundary data and curved boundary segments. Such effort has already had significant pay-off in two dimensional problems. The emphasis, as in finite elements, will shift to efficient algorithms for performing the necessary discrete integrations :numerically".

...

The object of the following formulation is to effect these improvements to the algorithm. The numerical methods are those used for the formulation of chapter 3, with the improvements suggested in the discussion of the results for plane elasticity.

2.- The parametric representation of geometry and functions

The surface is represented by n elements, each of which has either eight or six nodes (figures 4.2.1 and 4.2.2). Generally, the 8 - node quadrilateral elements are used, but where necessary, to represent complicated surfaces,

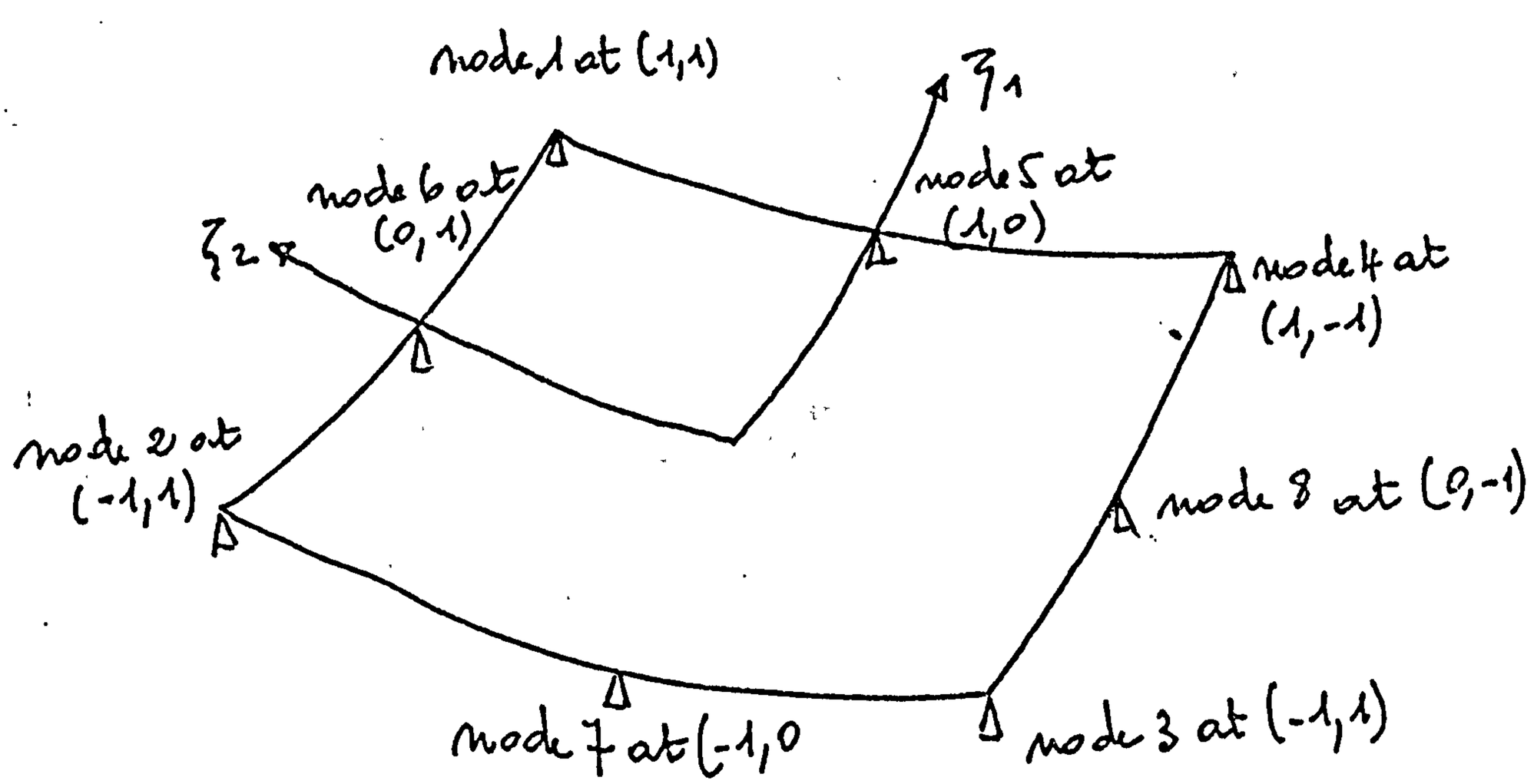


fig. 4.2.1 - quadrilateral surface element

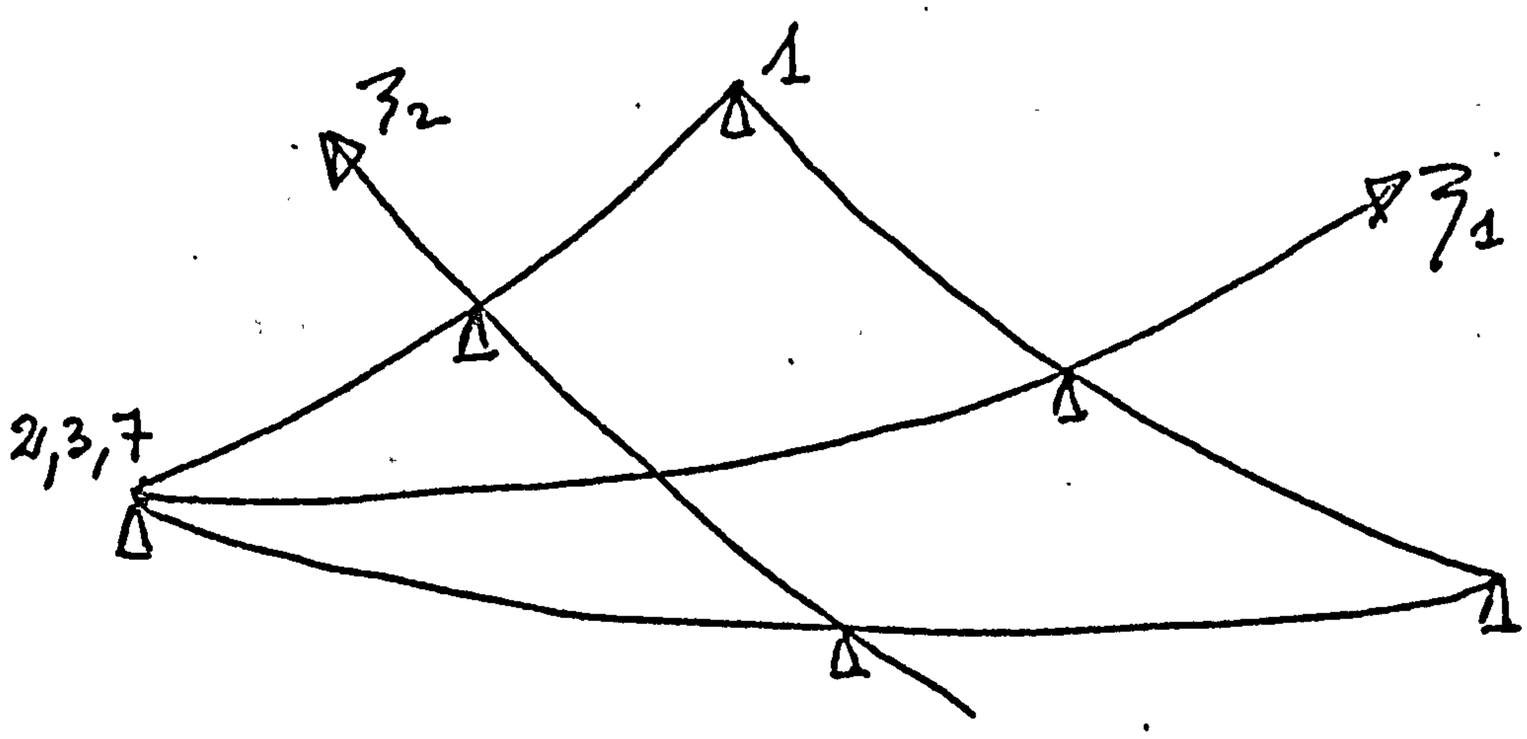


fig. 4.2.2 - triangular (degenerate) surface element

...

degenerate surface elements are incorporated in the network. In fig. 4.2.2 is shown only one of four ways in which a degenerate element may be defined; the end and mid - nodes of any side may be considered to be coincident.

The cartesian coordinates x_i of an arbitrary point of an element are defined in terms of the nodal coordinates x_i^a, $a \in \{1,\dots,8\}$ and shape functions $N^a(\xi)$

$$x_i(\xi) = N^a(\xi)\, x_i^a \qquad (4.2.1)$$

$\xi \in \{\xi_1, \xi_2\}$, $\xi_j \in [-1,+1]$ being the intrinsic coordinates, and N the functions (Ergatoudis (67)) :

$$N^1(\xi) = \frac{1}{4}(\xi_1+1)(\xi_2+1)(\xi_1+\xi_2-1)$$
$$N^5(\xi) = \frac{1}{2}(\xi_1+1)(1-\xi_2^2) \qquad (4.2.2)$$

the functions for other values of a being derived using symmetry. Evidently, an element may be flat, singly or doubly curved. The variation of these functions is shown in fig. 4.2.3.

...

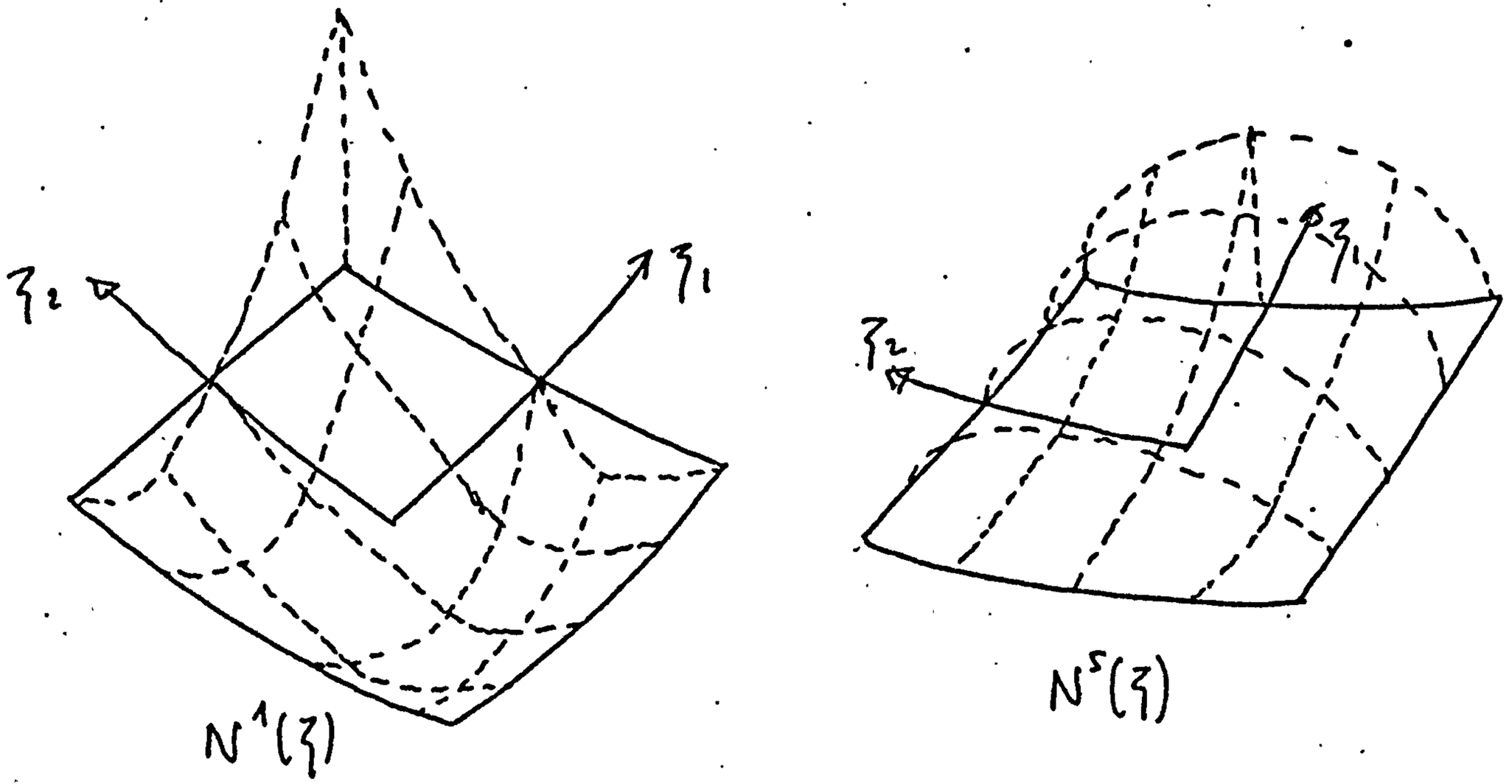


fig. 4.2.3 - the geometric shape functions N^1 and N^5

By differentiating equation 4.2.1. with respect to ξ_j, vectors tangent to the coordinate lines are obtained :

$$J_{ij} = \frac{\partial x_i}{\partial \xi_j} = \frac{\partial N^a(\xi)}{\partial \xi_j} x_i^a \qquad (4.2.3)$$

A vector normal to the element is obtained by taking the vector product of the two vectors defined by equation 4.2.3, and the Jacobian is the modulus of this vector product. This procedure breaks down at the singularity of intrinsic coordinates occurring in a degenerate element (e.g. the line $\xi_1 = -1$, fig. 4.2.2), but in this case it is sufficient to calculate the normal a little way towards the centre of the element, to obtain a value precise enough for practical purposes.

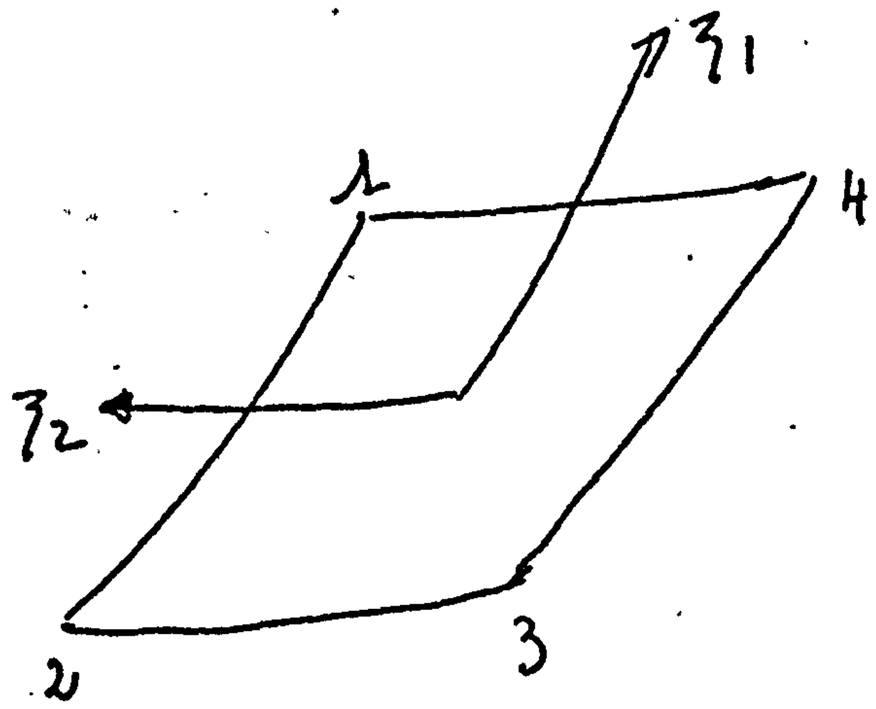
Symmetries with respect to coordinate axes may be treated in the same way as for the plane problem (section 3.2).

...

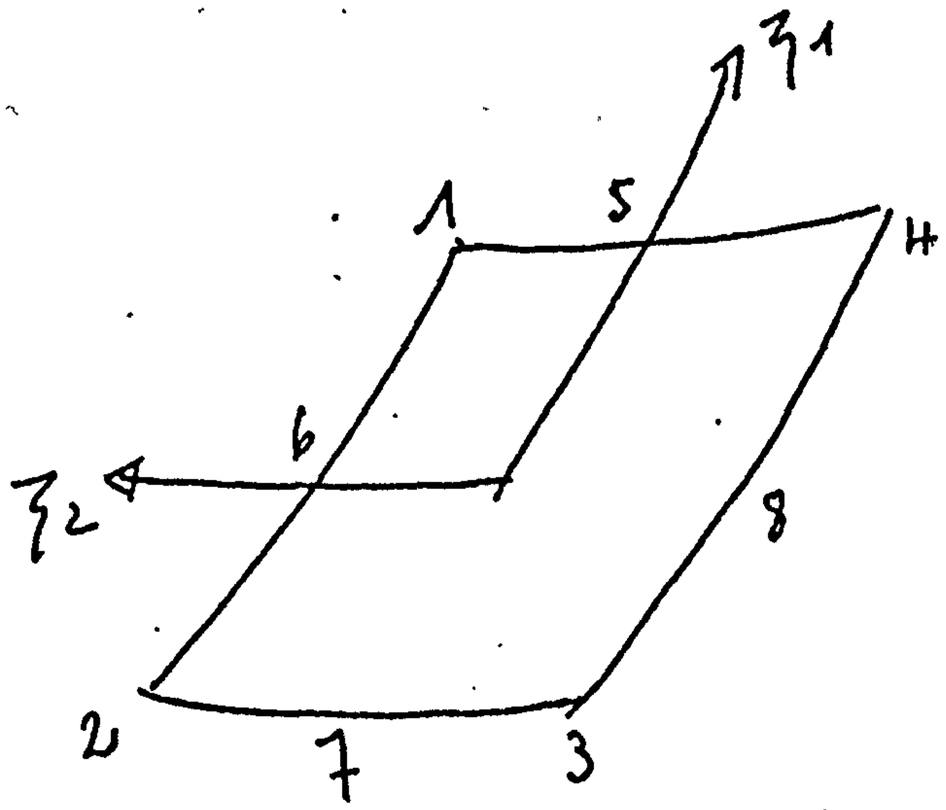
The functions u and t are considered to vary linearly, quadratically or cubically with respect to the intrinsic coordinates over each surface element. The value ϕ of a function at an arbitrary point of an element is defined in terms of its nodal values ϕ^a, $a \in \{1,\dots,n\}$ (fig. 4.2.4) and the shape functions $M^a(\zeta)$ corresponding to the variation chosen:

$$\phi(\zeta) = M^a(\zeta)\,\phi^a \qquad (4.2.4)$$

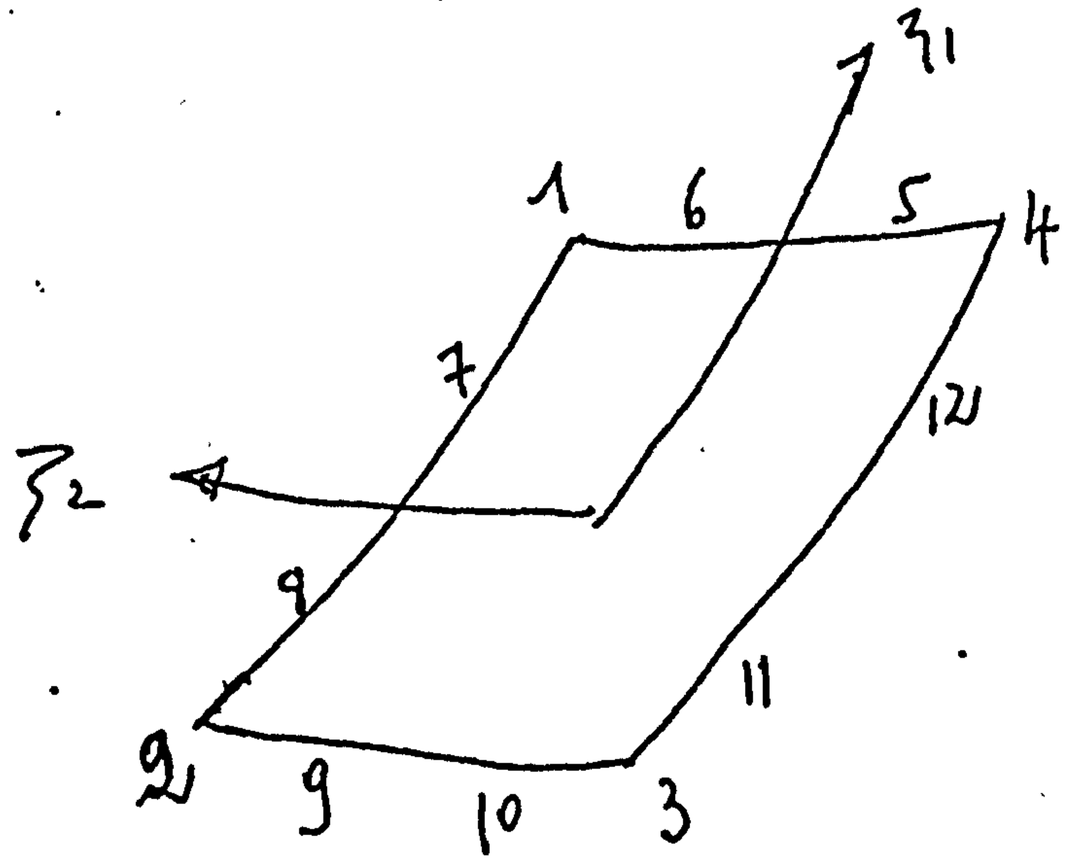
...



linear variation



quadratic variation



cubic variation

fig. 4.2.4 - nodes for the representation of functions

...

$M^a(\zeta)$ is defined as follows :

- linear variation

$$M^1(\zeta) = \frac{1}{4}(\zeta_1+1)(\zeta_2+1) \qquad (4.2.5)$$

- quadratic variation

$$M^2(\zeta) = N^a(\zeta) \quad \text{(see equation 4.2.2)} \qquad (4.2.6)$$

- cubic variation

$$M^1(\zeta) = \frac{1}{32}(\zeta_1+1)(\zeta_2+1)\left[9(\zeta_1^2+\zeta_2^2)-10\right]$$
$$M^2(\zeta) = -\frac{27}{32}(\zeta_1+1)(1-\zeta_2^2)(\zeta_2-\frac{1}{3}) \qquad (4.2.7)$$

The shape functions for other values of a are obtained using symmetry.

Let m_{1j}, $j \in \{1,2,3\}$, be the unit vector in the coordinate direction ζ_1, m_{2j} be the unit vector in the coordinate direction ζ_2 (see equation 4.2.3), and m_{3j} be the unit normal. In general, the directions m_{1j} and m_{2j} are not orthogonal. Let μ_{ij} be the direction cosines of the orthogonal system of axes, obtained by taking $\mu_{1j} = m_{1j}$, $\mu_{3j} = m_{3j}$ and taking μ_{2j} to be the vector product of m_{1j} and m_{3j}. Let $\bar{u}_i$ be the displacements, $\bar{\varepsilon}_{ij}$ be the strain, $\bar{\sigma}_{ij}$ be the stress and $\bar{t}_i$ be the traction in the orthogonal local system defined by μ_{ij} (see figure 4.2.5)

...

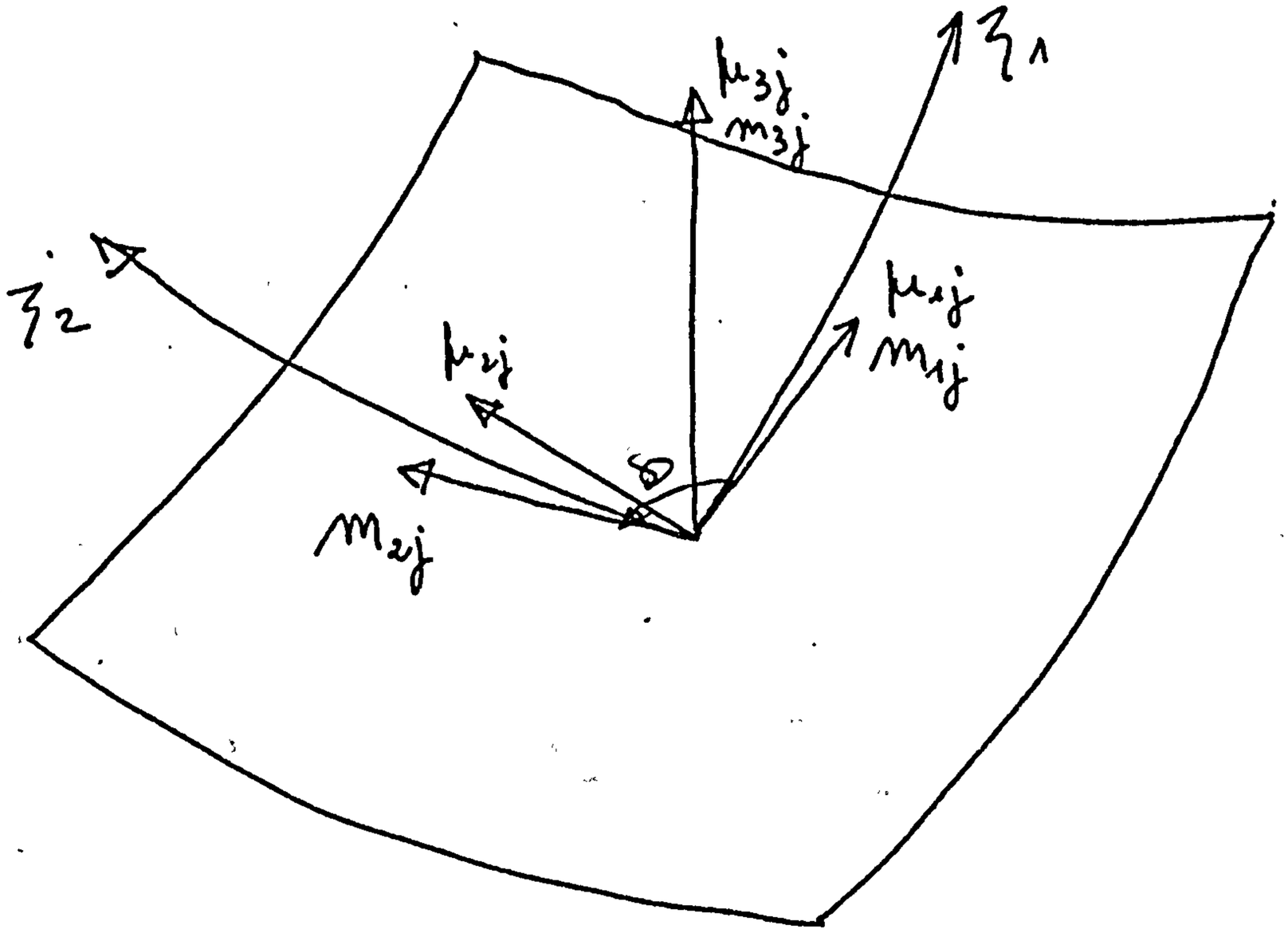


fig. 4.2.5 - local axes for calculation of stress

Then the displacement in the orthogonal tangential direction i, $i \in \{1,2\}$, is given by

$$\overline{u}_i(\xi) = M^a(\xi)\, u^a_\Lambda \,|\mu_{i\Lambda} \qquad \Lambda \in \{1,2,3\} \tag{4.2.8}$$

By differentiating with respect to ξ, an expression is obtained for tangential strain in the orthogonal system

...

$$\overline{\varepsilon_{i1}}(\xi) = \frac{\partial M^a(\xi)}{\partial \xi_1} \frac{u_s^a \mu_{is}}{|\Delta_{i1}|} \tag{4.2.9}$$

and

$$\overline{\varepsilon_{i2}}(\xi) = -\frac{\partial M^a(\xi)}{\partial \xi_1} \frac{u_s^a \mu_{is} \cos\theta}{|\Delta_{i2}| \sin\theta} + \frac{\partial M^a(\xi)}{\partial \xi_2} \frac{u_s^a \mu_{is}}{|\Delta_{i2}| \sin\theta} \tag{4.2.10}$$

where θ is the angle shown in figure 4.2.5. Then, using Hooke's law, the stress tensor may be calculated from the traction and displacement :

$$\overline{\sigma_{11}} = \frac{\nu}{1-\nu} \overline{t_3} + \frac{E\nu}{1-\nu^2}\left[\overline{\varepsilon_{11}} + \overline{\varepsilon_{22}}\right] + \frac{E}{1+\nu} \overline{\varepsilon_{11}}$$

$$\overline{\sigma_{12}} = \overline{\sigma_{21}} = \frac{E}{2(1+\nu)} \overline{\varepsilon_{12}}$$

$$\overline{\sigma_{22}} = \frac{\nu}{1-\nu} \overline{t_3} + \frac{E\nu}{1-\nu^2}\left[\overline{\varepsilon_{11}} + \overline{\varepsilon_{22}}\right] + \frac{E}{1+\nu} \overline{\varepsilon_{22}}$$

$$\overline{\sigma_{33}} = \overline{t_3}$$

$$\overline{\sigma_{32}} = \overline{\sigma_{23}} = \overline{t_2}$$

$$\sigma_{31} = \overline{\sigma_{13}} = \overline{t_1} \tag{4.2.11}$$

...

3.- Subregions

The elastic body is considered to consist of several homogeneous subregions $R^{(k)}$ the elastic properties of which may differ (fig. 4.3.1). The integral equation is written

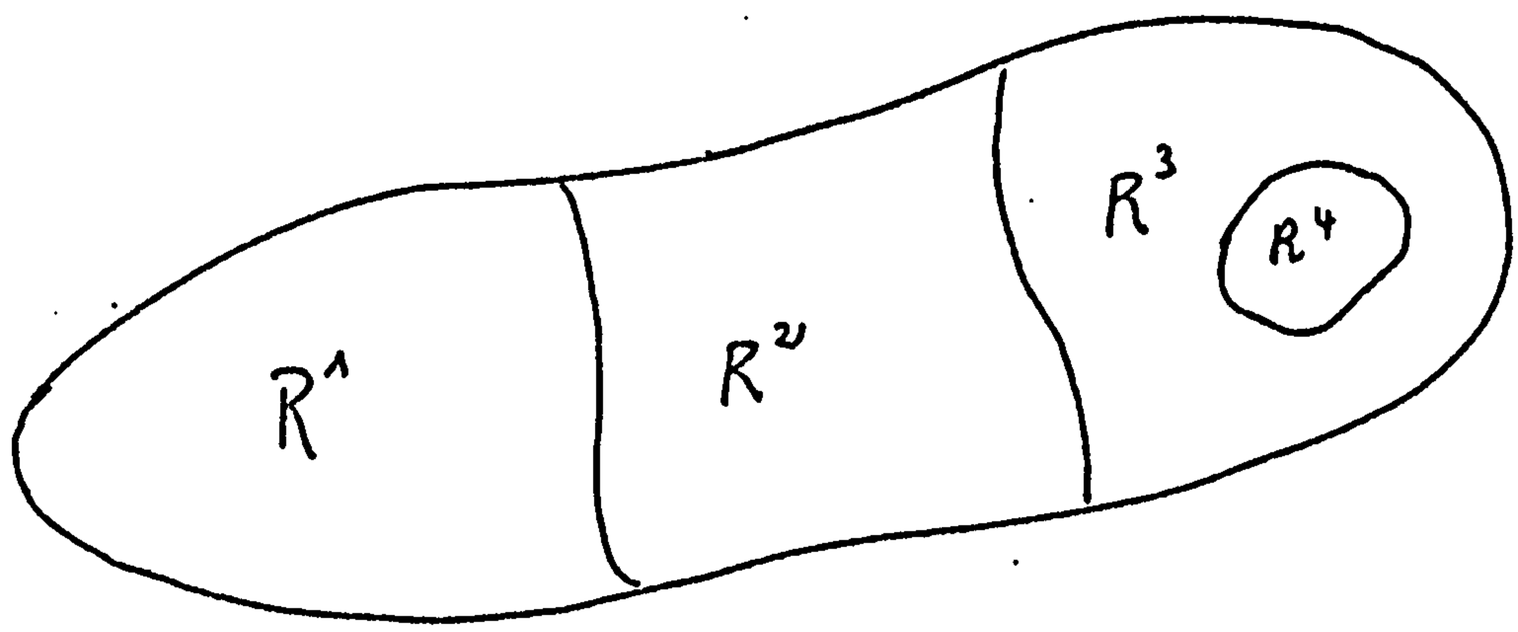


fig. 4.3.1 - elastic body divided into subregions

for each subregion :

$$C_{ij}^{(k)}(x)\, u_j^{(k)}(x) + \int_{S^{(k)}} T_{ij}^{(k)}(x,y)\, u_j^{(k)}(y)\, dS_y = \int_{S^{(k)}} U_{ij}^{(k)}(x,y)\, t_j^{(k)}(y)\, dS_y \qquad (4.3.1)$$

where $S^{(k)}$ is the surface of $R^{(k)}$ and the superscript k indicates that a function is calculated from the elastic properties of $R^{(k)}$ and the outward normal to $S^{(k)}$. To equations 4.3.1 are added the compatibility and equilibrium conditions across interfaces :

...

$$u_i^{(k)}(x) = u_i^{(l)}(x)$$

$$t_i^{(k)}(x) = -t_i^{(l)}(x) \qquad (4.3.2)$$

where $x \in S^{(k)} \cap S^{(l)}$

Where there is body force, the solution is taken for each subregion as the sum of a particular integral $\hat{u}^{(k)}$ and complementary function $\hat{\hat{u}}^{(k)}$ (see section 2.5)

$$u^{(k)} = \hat{u}^{(k)} + \hat{\hat{u}}^{(k)} \qquad (4.3.3)$$

The compatibility and equilibrium conditions across interfaces (equation 4.3.2) give the following relationship between the complementary functions for $R^{(k)}$ and $R^{(l)}$

$$\hat{u}_i^{(k)} + \hat{\hat{u}}_i^{(k)} = \hat{u}_i^{(l)} + \hat{\hat{u}}_i^{(l)}$$

$$\hat{t}_i^{(k)} + \hat{\hat{t}}_i^{(k)} = -\left(\hat{t}_i^{(l)} + \hat{\hat{t}}_i^{(l)}\right) \qquad (4.3.4$$

where $\hat{t}$ and $\hat{\hat{t}}$ are the tractions corresponding to the displacement fields $\hat{u}$ and $\hat{\hat{u}}$ respectively. It may be seen that in general the complementary functions $\hat{\hat{u}}^{(k)}$ and $\hat{\hat{u}}^{(l)}$ are not equal on the interface $S^{(k)} \cap S^{(l)}$

To calculate stress and displacement at a point inside a subregion, equations 2.3.1 and 2.3.4 are written for that subregion only.

4.- Discretisation of the integral equations

Let there be $p^{(k)}$ elements and $q^{(k)}$ distinct nodes on the surface $S^{(k)}$, $k \in \{1 \cdots z\}$. A global numbering $d^{(k)}(b,c)$, $b \in \{1, \ldots, p^{(k)}\}$, $c \in \{1, \ldots, r\}$ is created for each subregion, the nodes being numbered in the following order :

- nodes on $S^{(1)}$ only
- nodes on $S^{(1)} \cap S^{(2)}$
- nodes on $S^{(1)} \cap S^{(3)}$, and not already numbered
- ⋮
- nodes on $S^{(1)} \cap S^{(z)}$
- nodes on $S^{(2)}$ only
- nodes on $S^{(2)} \cap S^{(3)}$, and not already numbered
- ⋮
- nodes on $S^{(z)}$ only

In this way a banded matrix is obtained.

As in the two-dimensional case (section 3.3), the integral equation is rewritten in terms of the components of u and t in the fixed and free directions at the point x:

$$C_{\alpha\beta}^{\prime(k)}\, u_\beta^{\prime(k)}(x) + \int_{S^{(k)}} T_{\alpha\beta}^{\prime(k)}(x,y)\, u_\beta^{\prime(k)}(y)\, dS_y = \int_{S^{(k)}} U_{\alpha\beta}^{\prime(k)}(x,y)\, t_\beta^{\prime(k)}(y)\, dS_y \qquad (4.4.1)$$

where

$$u'^{(k)}_{\alpha} = l_{\alpha i}\, u^{(k)}_{i}$$
$$t'^{(k)}_{\alpha} = l_{\alpha i}\, t^{(k)}_{i}$$
$$C'^{(k)}_{\alpha\beta} = \lambda_{\alpha i}\, \lambda_{\beta j}\, C^{(k)}_{ij}$$

$$\overline{T}'^{(k)}_{\alpha\beta} = \lambda_{\alpha i}\, \lambda_{\beta j}\, \overline{T}^{(k)}_{ij}$$
$$U'^{(k)}_{\alpha\beta} = \lambda_{\alpha i}\, \lambda_{\beta j}\, U^{(k)}_{ij}$$

(4.4.2

and $\lambda_{\beta i}\, l_{\beta j} = \delta_{ij}$

In equation 4.4.2, $l_{\alpha i}$ are the direction cosines of the unknowns at χ. Writing the equation 4.4.1 for the node $x^a \in S^{(k)}$ and substituting the parametric representation of t and u (4.2.4), the following equations are obtained

$$C'^{(k)}_{\alpha\beta}(x^a)\, u'^{(k)}_{\beta}(x^a) + \sum_{b=1}^{h^{(k)}} \sum_{c=1}^{n} u'^{(k)}_{\beta}(x^{d^{(k)}}) \int_{S^{(k)}_b} \overline{T}'^{(k)}_{\alpha\beta}(x^a, y(\xi))\, M^c(\xi)\, J(\xi)\, d\xi$$
$$= \sum_{b=1}^{h^{(k)}} \sum_{c=1}^{n} t'^{(k)}_{\beta}(x^{d^{(k)}}) \int_{S^{(k)}_b} U'^{(k)}_{\alpha\beta}(x^a, y(\xi))\, M^c(\xi)\, J(\xi)\, d\xi \qquad (4.4.3$$

where $\int_b^{(k)}$ is the bth element of the surface $S^{(k)}$ and $J(\xi)$ is the Jacobian.

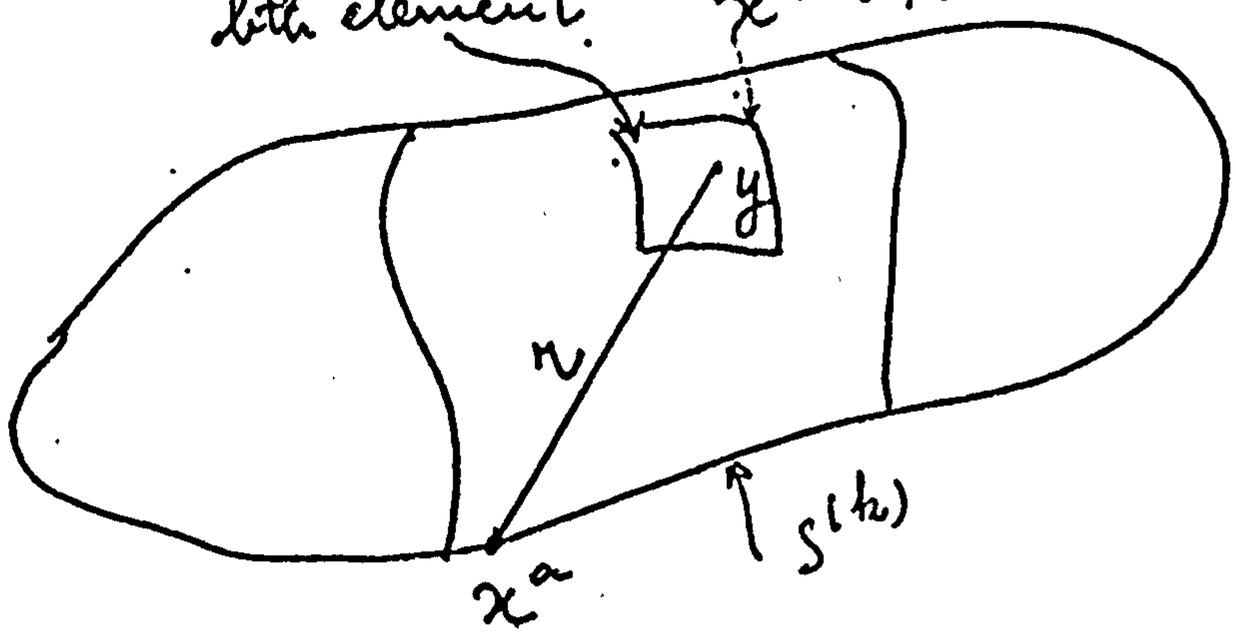


fig. 4.4.1 - parameters of equation 4.4.3 ...

The integrals of kernel-shape function products appearing in equation 4.4.3 are evaluated using Gaussian quadrature formulae(60), the integration scheme being chosen according to the rapidity of variation of the integrand. $\lambda^{(L)}$ and $\lambda^{(H)}$ are the lowest and highest orders of Gauss formula available for use in the procedure, and where necessary the element is divided into subelements, over each of which formulae within this range are sufficiently precise. As for the two-dimensional analysis (section 3.3), the case in which the node χ^a is one of the nodes of the element $\int_h^{(k)}$ and that in which it is not are considered separately.

In the latter case, Gaussian formulae with weight function 1.0 are used :

$$\int_{-1}^{+1}\int_{-1}^{+1} f(\zeta)d\zeta \simeq \frac{1}{\chi_1\chi_2}\sum_{i=1}^{\chi_1}\sum_{j=1}^{\chi_2}\sum_{k=1}^{\lambda_1(i,j)}\sum_{l=1}^{\lambda_2(i,j)} A_k^{\lambda_1(i,j)} A_l^{\lambda_2(i,j)} f(\zeta) \qquad (4.4.4)$$

where (see fig. 4.4.2) the argument ζ is calculated as follows :

$$\zeta_1 = -1 + \frac{1 - 2i + \eta_k^{\lambda_1(i,j)}}{\chi_1}$$

$$\zeta_2 = -1 + \frac{1 - 2j + \eta_l^{\lambda_2(i,j)}}{\chi_2} \qquad (4.4.5)$$

The functions A_t^Δ and η_t^Δ appearing in equations 4.4.4 and 4.4.5 are the weight function and offset of the t th Gauss point of the formula order Δ .

...

To calculate, for a given position of the singularity x^a and a given element $S_k^{(k)}$ the values of χ_1 and χ_2 (number of subelements each way) and $\lambda_1(i,j), \lambda_2(i,j)$ (order of integration formula for subelement (i, j)) necessary to give satisfactory precision, the upper bounds given by Stroud and Secrest (60),

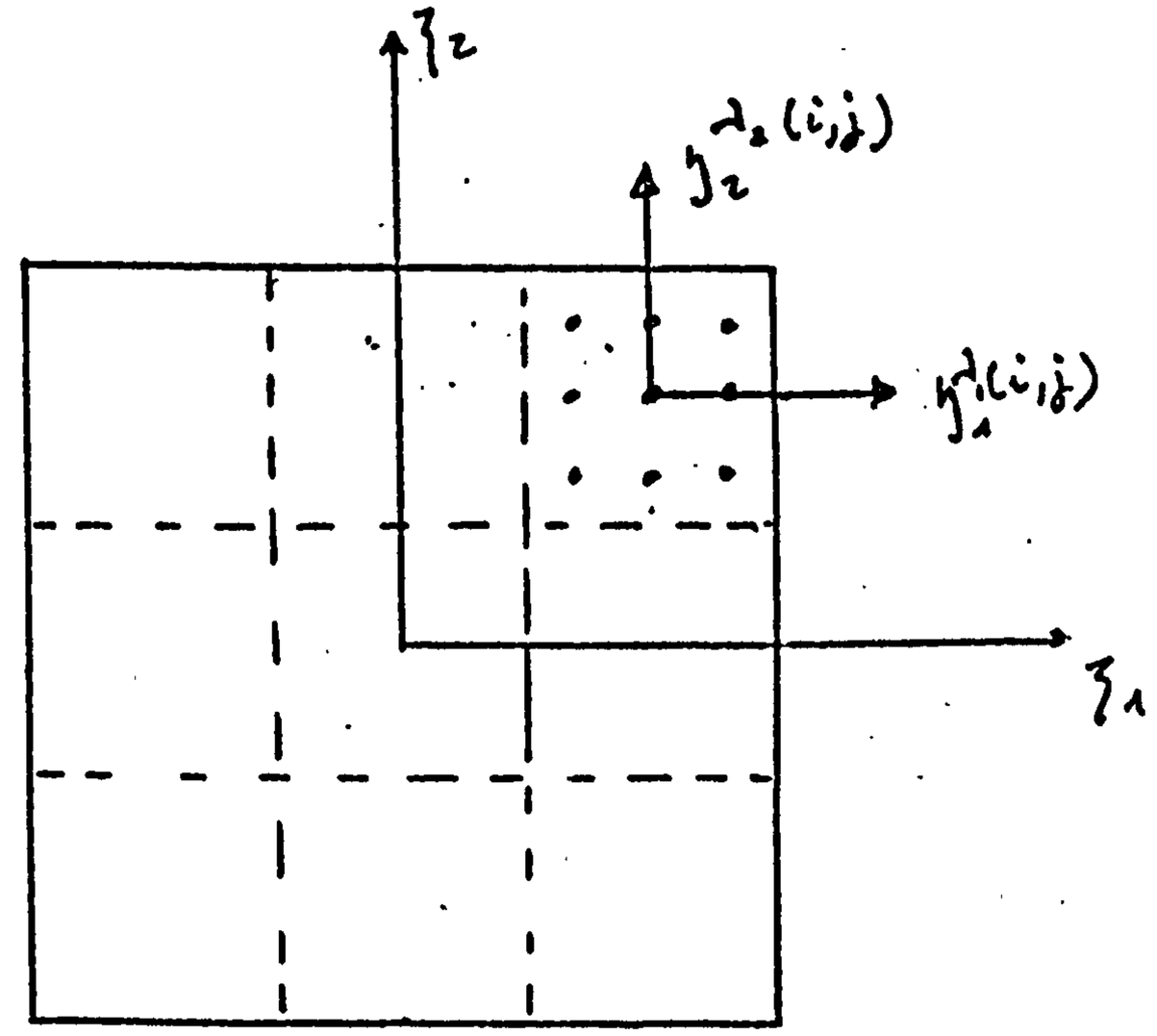


Fig. 4.4.2 - example of integration scheme for regular integrand

for error in terms of derivatives of the integrand are used. Stroud gives upper bounds for error in terms of the 2 n th derivative of the integrand, n being the order of the integration formula :

$$\left| \int_{-1}^{1} f(x)\,dx - \sum_{k=1}^{n} A_k f(x_k) \right| \leq e\, M \qquad (4.4.6)$$

where $\left| \frac{d^{2n} f}{dx^{2n}} \right| \leq M.$

The function e is tabulated by Stroud and Secrest. ...

For all practical purposes, one may write

$$e = \frac{G}{2^{2n}\, 2n!} \tag{4.4.7}$$

where $G \simeq 4.0$. The same authors generalise this result to the multidimensional case ; for two dimensions they obtain

$$\left| \int_{-1}^{+1}\int_{-1}^{+1} f(x)\,dx_1\,dx_2 - \sum_{k=1}^{n_1}\sum_{l=1}^{n_2} A_k B_l f(x_k, x_l) \right| \leq 2 \sum_{i=1}^{2} e_i M_i \tag{4.4.8}$$

where $\left| \frac{\partial^{2n_i} f}{\partial x_i^{2n_i}} \right| \leq M_i,\ i \in \{1,2\}$ $\quad e_i \simeq \frac{G}{2^{2n_i}\, 2n_i!}$

It would be impractical to calculate the preceding upper bounds for the functions appearing in equation 4.4.3, so, in the procedure for choice of the integration scheme, the function $\frac{1}{r^2}$ (see fig. 4.4.1) is taken to be representative of the integrands. This is reasonable because it is, in critical cases, the most rapidly varying component of the kernel T. The kernel U varies more slowly, and the shape functions and Jacobian should present no problem either.

Even the calculation of derivatives with respect to intrinsic coordinates of $\frac{1}{r^2}$ is impractical, and further simplifications are made.

...

Let us consider the transformation from the coordinates used for the integration, η_i, to the global cartesian coordinates. Provided the element is not badly distorted, the rate of change of the Jacobian for this transformation is low compared with the rate of change of $\frac{1}{r^2}$. In these circumstances, the calculation of error bounds may reasonably be performed supposing the Jacobian $\frac{\partial \Delta}{\partial \eta_i}$ where Δ = arc length, to be constant. Then

$$\frac{\partial^{2n_i}\left(\frac{1}{r^2}\right)}{\partial \eta_i^{2n_i}} = \frac{\partial^{2n_i}\left(\frac{1}{r^2}\right)}{\partial \Delta^{2n_i}} \left(\frac{\partial \Delta}{\partial \eta_i}\right)^{2n_i} \qquad (4.4.9)$$

Let us assume that the $2n_i$th derivative with respect to arc length is never greater than the $2n_i$th derivative with respect to r at the nearest point of the subelement to x^a. Then

$$\left| \frac{\partial^{2n_i}\left(\frac{1}{r^2}\right)}{\partial \eta_i^{2n_i}} \right| \leq \frac{(2n_i+1)!}{R^{2n_i+2}} \left(\frac{\partial \Delta}{\partial \eta_i}\right)^{2n_i} \qquad (4.4.10)$$

where R is the minimum distance. Let us adopt the following criteria :

1.- the upper bound for error shall be proportional to the product of $\frac{1}{R^2}$ and the area of the subelement

2.- as nearly as possible, $e_1 M_1 = e_2 M_2$ (equ. 4.4.8), i.e. the integration is equally precise in each direction.

...

Let the Jacobian for the transformation from the coordinates η_i to global cartesian coordinates, which is assumed to be constant, be J. Then

$$\int_{\text{subelement}} \frac{1}{r^2}\, dA = \int_{-1}^{1}\int_{-1}^{1} \frac{J}{r^2}\, d\eta_1\, d\eta_2$$

Then by equations 4.4.8 and 4.4.10, the error $\mathcal{E}$ is given by

$$\mathcal{E} = \frac{2GJ}{R^2} \sum_{i=1}^{2} (2n_i+1)\, \alpha_i^{2n_i} \qquad (4.4.11)$$

where

$$\alpha_i^{2n_i} = \left(\frac{ds}{d\eta_i}\right)^{2n_i} \qquad (4.4.12)$$

Then, according to the criteria just proposed, n_i should be chosen such that :

$$(2n_i+1)\, \alpha_i^{2n_i} \leq K \qquad (4.4.13)$$

where K is a constant, to be chosen in the light of experience. The parameters $\chi_1, \chi_2, \lambda_1(i,j)$ and $\lambda_2(i,j)$ of equation 4.4.4 are, therefore, calculated as follows :

1.- the minimum distance R for the whole element is determined .

2.- χ_1 and χ_2 are chosen such that, for the subelement containing the nearest point, inequality 4.4.13 is satisfied whilst $n_i \leq \lambda^{(H)}$, the highest order of Gauss formula available. This gives $\lambda_1(i,j), \lambda_2(i,j)$ for the subelement nearest to x^a.

...

3.- the corresponding functions for all other elements are now calculated, from their minimum distances from x^a and inequality 4.4.13.

The effect of this procedure is to concentrate the integration points around the singularity, as shown in fig. 4.4.3. The lower limit $\lambda^{(L)}$ on order of Gauss formula is imposed so that, where the kernels vary very slowly, enough points are retained to integrate accurately the shape functions.

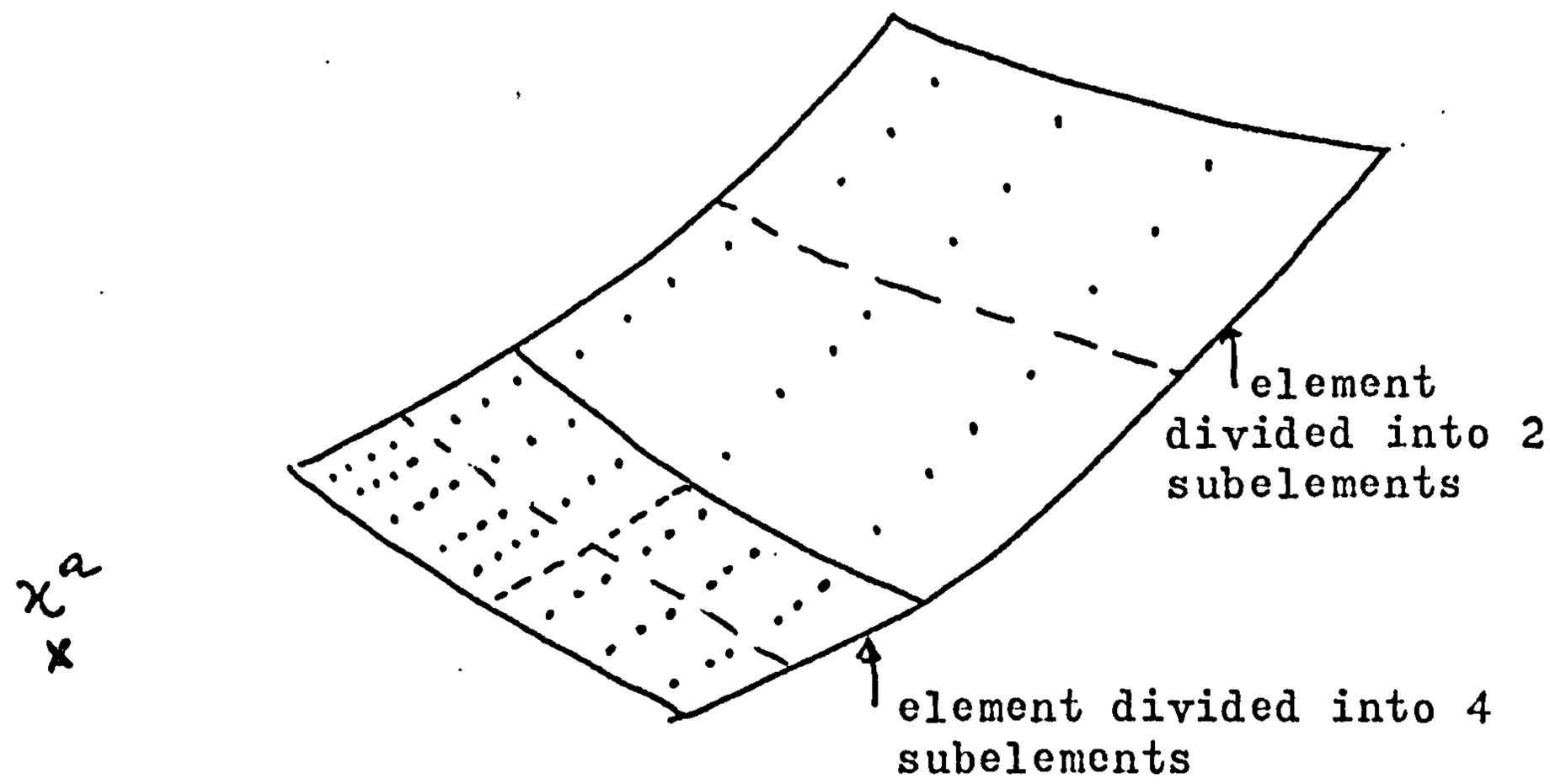


fig. 4.4.3 - typical distribution of integration points.

In the case in which x^a is a node of the element $\int_b^{(k)}$; the element is divided into triangles, each with a vertex at x^a and with a side of the element as the base opposite that vertex (fig. 4.4.4.). With each

...

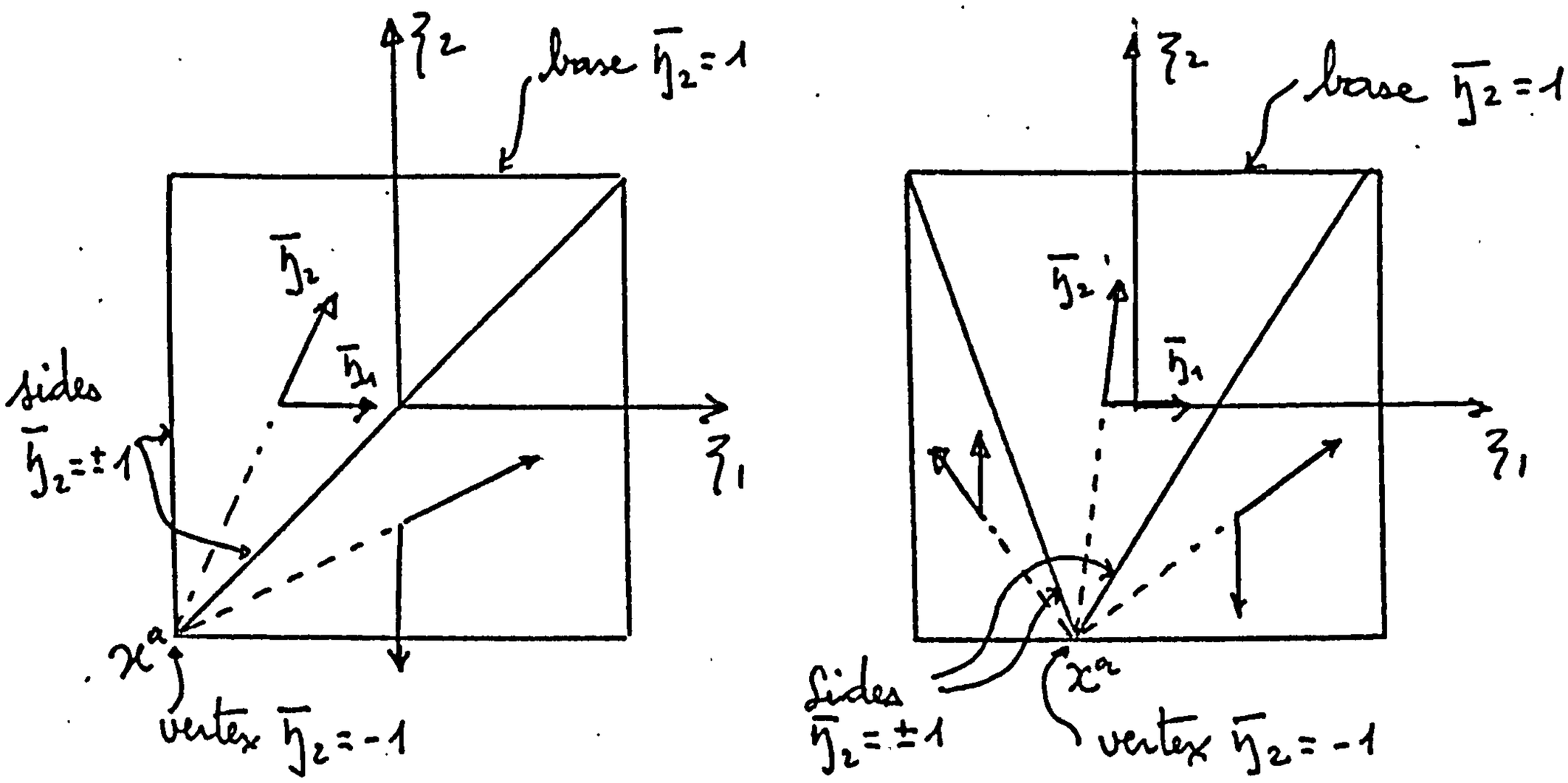


fig. 4.4.4 - examples of division into triangles (nondegenerate element).

triangle are associated the triangle coordinates $\bar{\eta}_i$ (fig. 4.4.4). In the case of nondegenerate elements, or degenerate elements with x^a on the degenerate side, the coordinates $\bar{\eta}_i$ are defined in terms of ζ_i by the relationships

$$\zeta_i(\bar{\eta}) = L^a(\bar{\eta})\,\zeta_i^{(a)} \tag{4.4.14}$$

where $L^a(\bar{\eta})$ are the linear shape functions defined by equation 4.2.5, and $\zeta_i^{(a)}$ are the intrinsic coordinates of the vertices of the triangle, as shown in fig. 4.4.5. For a nondegenerate element, $\zeta_i^{(3)} = \zeta_i^{(4)}$; for a degenerate element with x^a on the degenerate side, $\zeta_i^{(3)}$ and $\zeta_i^{(4)}$ are the limiting values as x^a is approached along the sides $\bar{\eta}_1 = -1$ and $\bar{\eta}_1 = +1$ respectively, and $\zeta_i^{(3)} \neq \zeta_i^{(4)}$. The Jacobian $\frac{\partial \zeta_i}{\partial \bar{\eta}_j}$ is given by

$$\frac{\partial \zeta_i}{\partial \bar{\eta}_j} = \frac{\partial L^a(\bar{\eta})}{\partial \bar{\eta}_j}\,\zeta_i^{(a)} \tag{4.4.15}$$

...

Let us consider what happens as the functional singularity x^a is approached. In the case in which the element is not degenerate, $\left|\frac{\partial \xi_i}{\partial \bar{\eta}_j}\right|$ is $O(\bar{\eta}_2+1)$ and consequently the Jacobian $\frac{dA}{d\bar{\eta}}$ is $O(\bar{\eta}_2+1)$ as well.

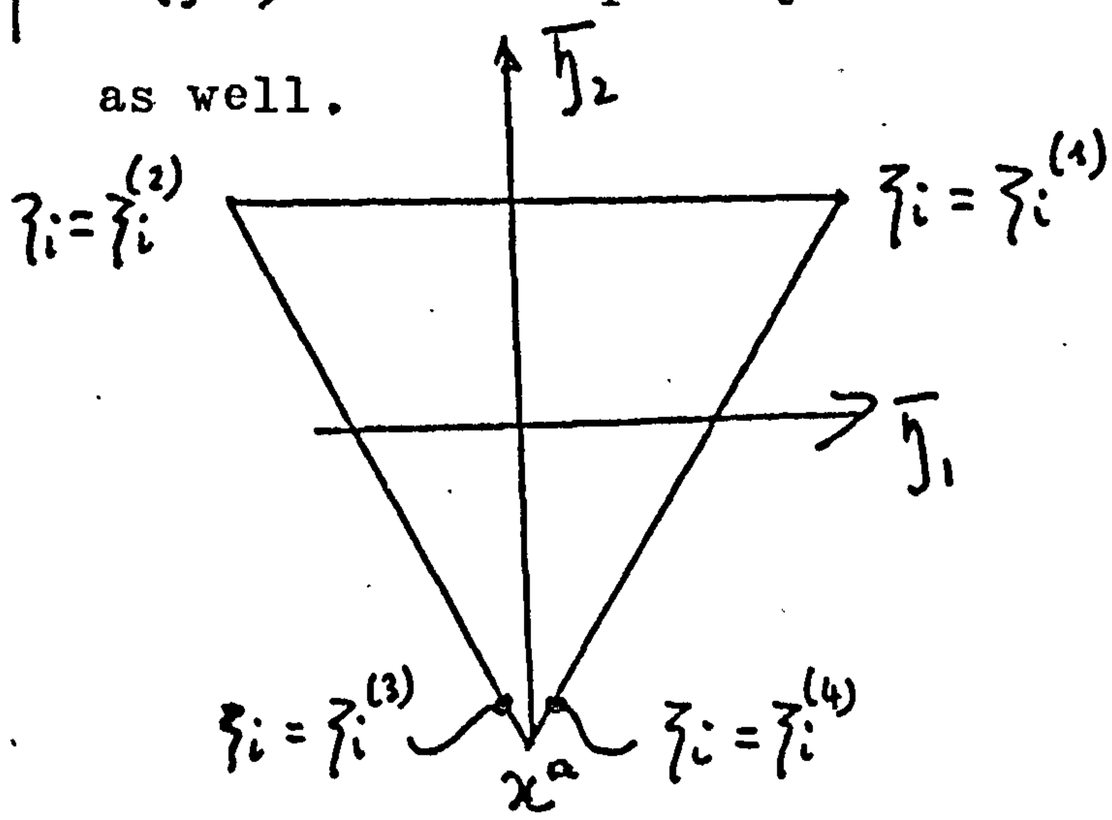
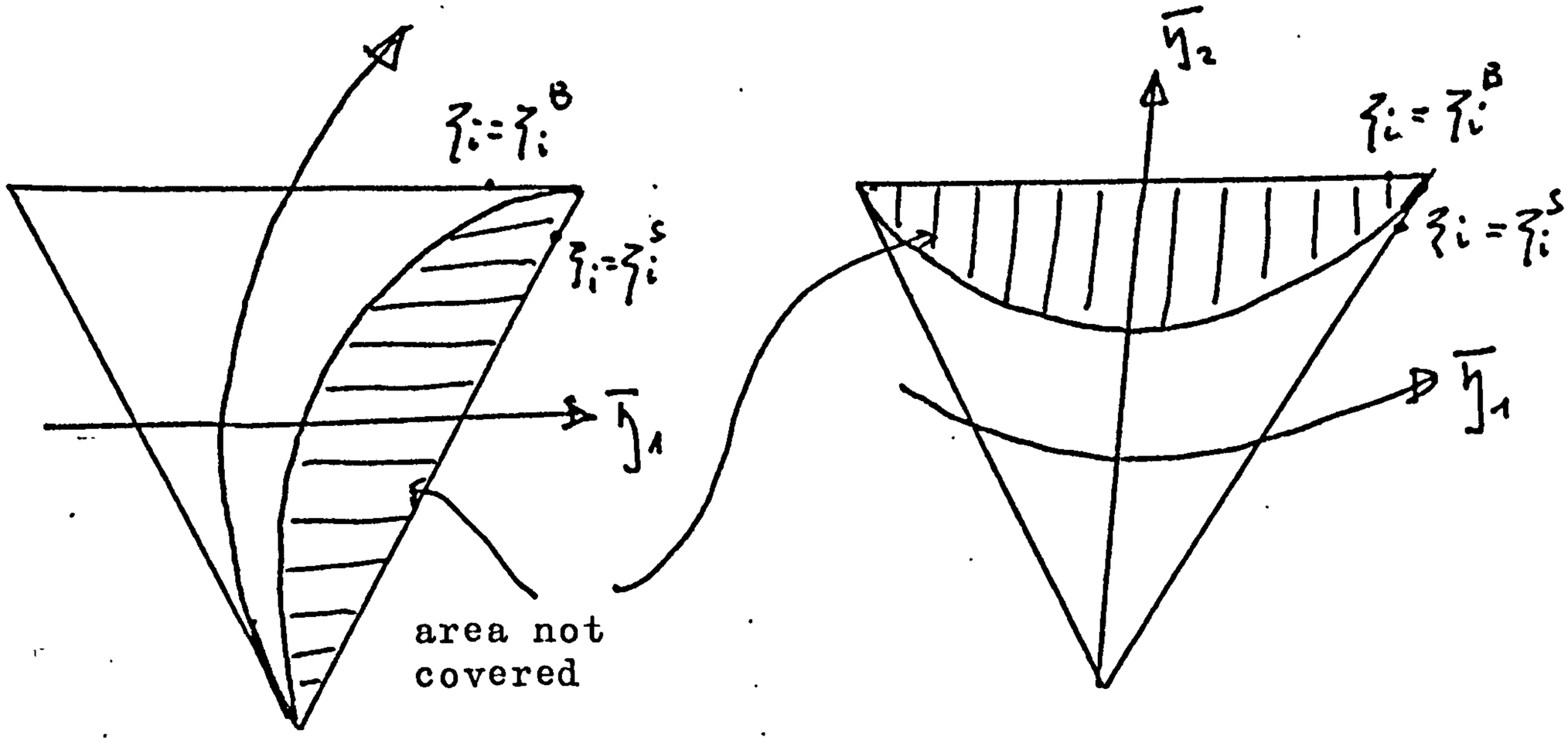


fig. 4.4.5 - definition of triangle coordinates by linear transformation

Where the element is degenerate and x^a is on the degenerate side, $\left|\frac{\partial \xi_i}{\partial \bar{\eta}_j}\right|$ is $O(1)$, and again $\frac{dA}{d\bar{\eta}}$ is $O(\bar{\eta}_2+1)$

In the case in which the element is degenerate, but x^a is not on the degenerate side, the transformation 4.4.14 is not used because, whatever value be adopted for ξ_i at the vertex on the degenerate side, there exist points in the triangle to which there correspond no $\bar{\eta}_i$, $-1 \leq \bar{\eta}_i \leq 1$ (see fig. 4.4.6). In fig. 4.4.6, ξ_i^B and ξ_i^S are the limiting values of ξ_i as the point (1,1) is approached along the base and sides respectively. To obtain a transformation covering the whole triangle, a combination of the two transformations $\psi_i^{(k)}$ shown in fig. 4.4.6 is used :

...



a) transformation $\zeta_i = \psi_i^{(1)}$

obtained by putting

$$\zeta_i^{(1)} = \zeta_i^B$$

b) transformation $\zeta_i = \psi_i^{(2)}$

obtained by putting

$$\zeta_i^{(1)} = \zeta_i^S$$

fig. 4.4.6 - transformation 4.4.14 with degenerate side at the point (1,1)

$$\zeta_i(\bar{\eta}) = \psi_i^{(k)}(\bar{\eta})\, f^{(k)}(\bar{\eta}) \qquad k \in \{1,2\} \tag{4.4.16}$$

where

$$f^{(1)}(\bar{\eta}) = \frac{1}{2}\,(1+g(\bar{\eta}))$$

$$f^{2}(\bar{\eta}) = \frac{1}{2}\,(1-g(\bar{\eta})) \tag{4.4.17}$$

and

$$g(\bar{\eta}) = \frac{\bar{\eta}_2 - \bar{\eta}_1}{2 - \bar{\eta}_1 - \bar{\eta}_2} \tag{4.4.18}$$

...

Let $\bar{\eta}_2 = 1$. Then $g(\bar{\eta}) = 1$ and $\zeta_i = \psi_i^{(1)}$. Now let $\bar{\eta}_1 = 1$ Then $g(\bar{\eta}) = -1$ and $\zeta_i = \psi_i^{(2)}$. Along the lines $\bar{\eta}_i = -1$, $\psi_i^{(1)} = \psi_i^{(2)}$ and so ζ_i has the desired values on all four sides of the square $-1 \leq \bar{\eta}_i \leq +1$

The Jacobian $\frac{\partial \zeta_i}{\partial \bar{\eta}_j}$ is given by the product formula

$$\frac{\partial \zeta_i}{\partial \bar{\eta}_j} = \psi_i^{(k)}(\bar{\eta}) \frac{\partial f^{(k)}(\bar{\eta})}{\partial \bar{\eta}_j} + \frac{\partial \psi_i^{(k)}(\bar{\eta})}{\partial \bar{\eta}_j} f^{(k)}(\bar{\eta}) \qquad (4.4.19$$

It may be readily be shown that :

$$\frac{\partial f^{(1)}}{\partial \bar{\eta}_1} = \frac{\bar{\eta}_2 - 1}{(2 - \bar{\eta}_1 - \bar{\eta}_2)^2} \qquad \frac{\partial f^{(2)}}{\partial \bar{\eta}_1} = -\frac{\partial f^{(1)}}{\partial \bar{\eta}_1}$$

$$\frac{\partial f^{(1)}}{\partial \bar{\eta}_2} = \frac{1 - \bar{\eta}_1}{(2 - \bar{\eta}_1 - \bar{\eta}_2)^2} \qquad \frac{\partial f^{(2)}}{\partial \bar{\eta}_2} = -\frac{\partial f^{(1)}}{\partial \bar{\eta}_2} \qquad (4.4.20$$

Let us consider the behaviour of the Jacobian $\frac{\partial \zeta_i}{\partial \bar{\eta}_j}$ as x^a is approached. The first component of the left side of 4.4.19 is $\theta(\bar{\eta}_2 + 1)$ because $\frac{\partial f^{(2)}}{\partial \bar{\eta}_j} = -\frac{\partial f^{(1)}}{\partial \bar{\eta}_j}$ and, since x^a is not on a degenerate side, $(\psi_i^{(1)} - \psi_i^{(2)})$ is $\theta(\bar{\eta}_2 + 1)$. The function $\frac{\partial \psi_i^{(k)}}{\partial \bar{\eta}_1}$ is $\theta(\bar{\eta}_2 + 1)$ (see equation (4.4.15). Hence $\frac{dA}{d\bar{\eta}}$ is $\theta(\bar{\eta}_2 + 1)$ as is the case for the transformation 4.4.15.

A similar transformation exists for the case in which the degenerate side is at (-1, 1).

In all cases, therefore, the Jacobian $\frac{dA}{d\bar{\eta}}$ is $\theta(\bar{\eta}_2 + 1)$

In addition it is known that the kernel T is $\theta\left(\frac{1}{(\bar{\eta}_2+1)^2}\right)$ and U is $\theta\left(\frac{1}{\bar{\eta}_2+1}\right)$.

Where $d(b,c) \neq a$, the shape function $M^c(\xi)$ is $\theta(\bar{\eta}_2+1)$. Therefore, the integrand, being the product of the kernel U or T, shape function M and the Jacobian $\frac{dA}{d\bar{\eta}}$ is $\theta(1)$ near the functional singularity at x_i^a, and it is permissible to use Gauss formulae with weight function = I.0 to integrate over $\chi_1 \chi_2$ subelements (i,j), with each of which is associated the coordinates defined by

$$\bar{\eta}_1 = 1 + \frac{1 - 2i + \eta_1}{\chi_1}$$

$$\bar{\eta}_2 = 1 + \frac{1 - 2j + \eta_2}{\chi_2} \qquad (4.4.21)$$

Where $d(b,c)=a$, the shape function $M^c(\xi)$ is $\theta(1)$. To integrate the product of the kernel U, the function M and the Jacobian, the same procedure as for $d(b,c)\neq a$ may be used. But there exists no quadrature formula suitable for the calculation of the product of T, M and $\frac{dA}{d\bar{\eta}}$, because the product is $\theta\left(\frac{1}{\bar{\eta}_2+1}\right)$. As for the two-dimensional case (section 3.3), the Cauchy principal value and the coefficient of the free term are evaluated indirectly from the following relationship between global components :

$$C_{ij}^{(k)}(x^a) + \sum_{b=1}^{p^{(k)}} \sum_{c=1}^{m} \delta_{ad} \int_{\zeta_b^{(k)}} \overline{T}_{ij}^{(k)}(x^a; y(\xi))\, M^c(\xi)\, J(\xi)\, d\xi \qquad (4.4.22)$$

$$= - \sum_{b=1}^{p^{(k)}} \sum_{c=1}^{m} (1-\delta_{ad}) \int_{\zeta_b^{(k)}} T_{ij}^{(k)}(x^a; y(\xi))\, M^c(\xi)\, J(\xi)\, d\xi \quad \cdots$$

For both $d(b,c) \neq a$ and $d(b,c) = a$ (kernel U only), therefore, the integrals are calculated as follows :

$$\int_{-1}^{+1}\int_{-1}^{+1} f(\xi)\,d\xi \simeq \sum_{k=1}^{H} \sum_{i=1}^{\chi_1} \sum_{j=1}^{\chi_2} \sum_{k=1}^{\lambda_1(i,j)} \sum_{l=1}^{\lambda_2(i,j)} A_k^{\lambda_1(i,j)} A_l^{\lambda_2(i,j)} f\left(\xi\left(y_k^{\lambda_1(i,j)}, y_l^{\lambda_2(i,j)}\right)\right) \bar{J}(\xi) \qquad (4.4.23)$$

where H is the number of triangles into which the element is split (see fig. 4.4.4). For a degenerate element, $H=1$

A and y are the Gauss weight and offsets.

The argument ξ is calculated from equations 4.4.21 and either 4.4.14 or 4.4.16, and the Jacobian $\bar{J}$ may be calculated from equations 4.4.21 and either 4.4.15 or 4.4.19

The procedure for the calculation of χ_1, χ_2, $\lambda_1(i,j)$ and $\lambda_2(i,j)$ is the same as for the case in which x^a is not a node of the element (see equations 4.4.13), except in that the distance R is taken to be the minimum distance from x^a to a point on the base, and $\lambda_1(i,j)$, $\lambda_2(i,j)$ are taken to be independent of j (see fig. 4.4.7)

...

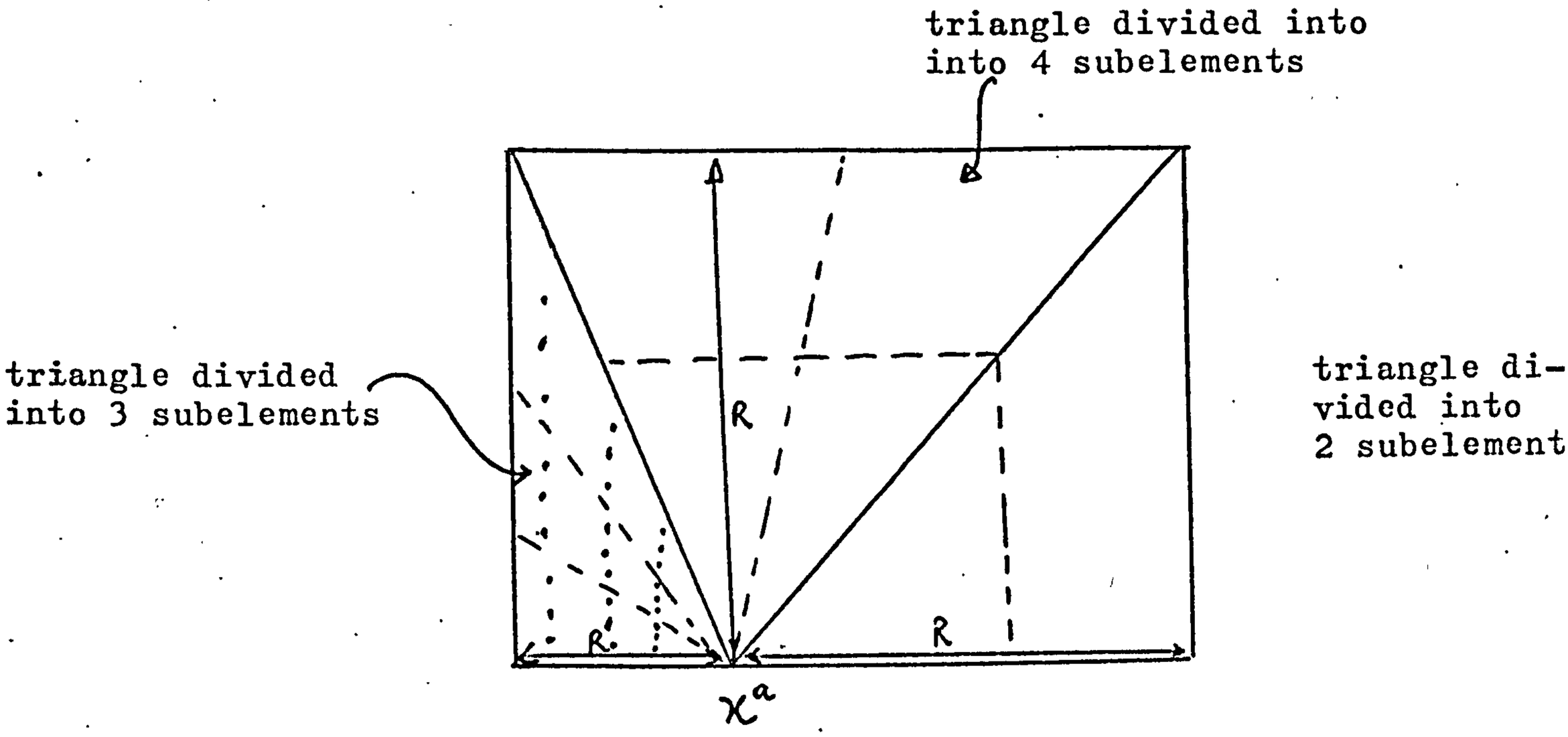


fig. 4.4.7 - typical distribution of integration points near singularity

The equations 4.4.3 for each subregion and the equilibrium conditions 4.3.4, form a linear system which can be solved for the unknown functions on the surface and interfaces.

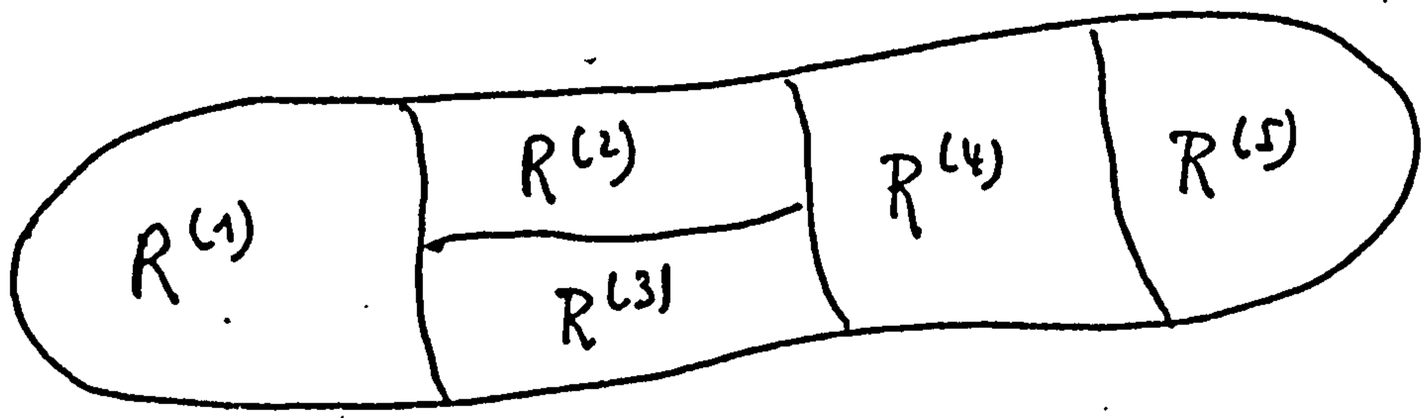
In practice it is unnecessary actually to include equations 4.3.4 in the system ; it is sufficient to include as unknowns $\hat{t}_i^{(k)}$ or $\hat{t}_i^{(l)}$, and either $\hat{u}_i^{(k)}$ or $\hat{u}_i^{(l)}$, and, for the traction unknowns to change the signs of certain contributions to the matrix and second member.

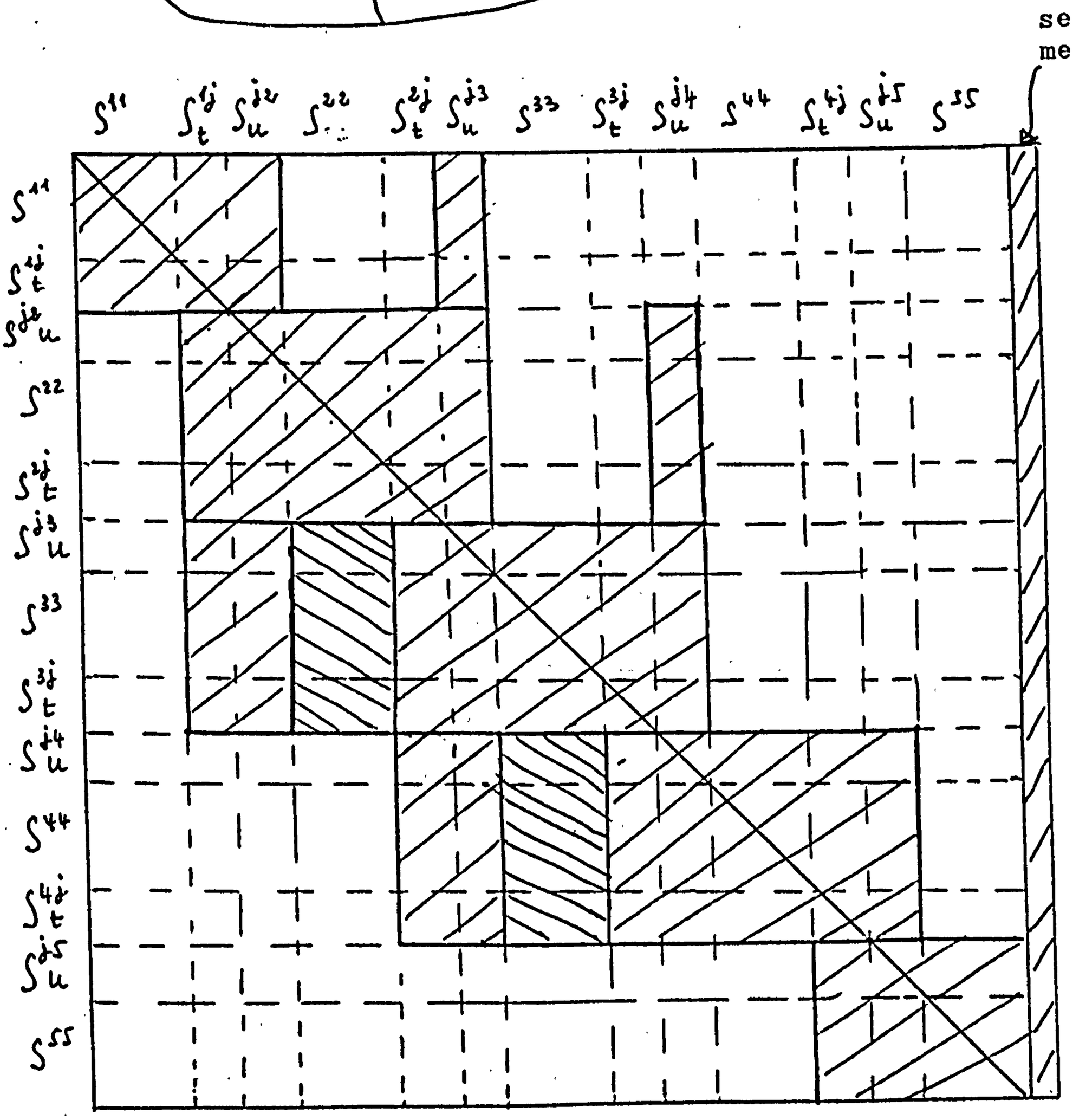
The unknowns are numbered in the following order (see fig. 4.4.8) :

...

S^{11}	– unknowns (u or t) at nodes on $S^{(1)}$ only	) equation
$S^{12}t$	– tractions on the interface $S^{(1)} \cap S^{(2)}$	) for first
$S^{13}t$	– tractions on the interface $S^{(1)} \cap S^{(3)}$	) subregion
$S^{12}u$	– other unknowns (u or t) at nodes on the interface $S^{(1)} \cap S^{(2)}$	) equation
$S^{13}u$	– other unknowns (u or t) at nodes on the interface $S^{(1)} \cap S^{(3)}$, not already numbered	) for second
S^{22}	– unknowns (u or t) at nodes on $S^{(2)}$ only	)
$S^{23}t$	– tractions on the interface $S^{(2)} \cap S^{(3)}$	) subregion
$S^{24}t$	– tractions on the interface $S^{(2)} \cap S^{(4)}$	)
$S^{23}u$	– other unknowns (u or t) at nodes on the interface $S^{(2)} \cap S^{(3)}$, not already numbered	) equation
$S^{24}u$	– other unknowns (u or t) at nodes on the interface $S^{(2)} \cap S^{(4)}$, not already numbered	) for third
S^{33}	– unknowns (u or t) at nodes on $S^{(3)}$ only	) subregion

...





unreduced
matrix and second member

areas filled
during reduction

note: for S^{ij}, $j \geq i$

fig. 4.4.8 - typical system of equations ...

As in the two-dimensional case (section 3.3), certain unknowns at nodes in planes of symmetry, if these are taken into account in the geometric representation, are zero and so are eliminated from the system. In addition, it is necessary to substitute for certain unknowns in the sets $S^{ij}u$, before reduction of the system. These unknowns are tractions on the surface of the body, introduced when a node shared by an interface and the surface (fig. 4.4.9) is considered to be fixed in some direction.

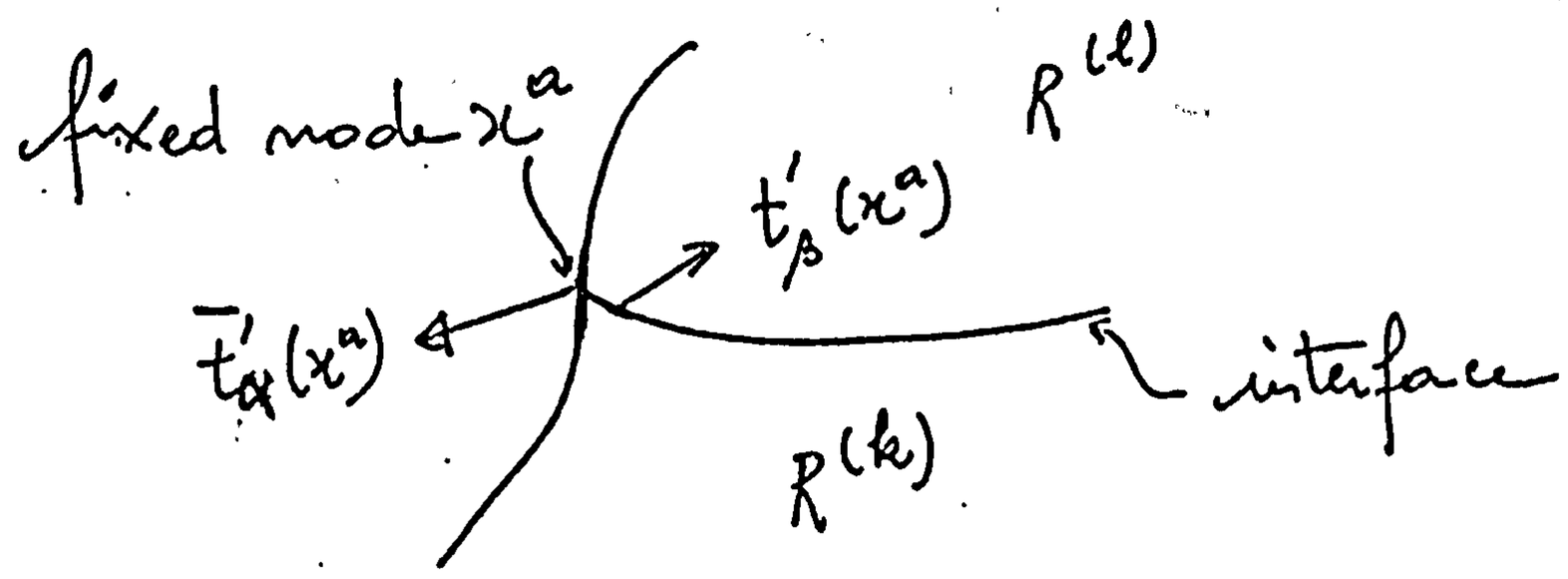


Fig. 4.4.9 - fixed node shared by surface and interface

By supposing the stress tensor to be continuous in either of the subregions $R^{(k)}$ or $R^{(l)}$ as x^a is approached and using Hooke's Law, the surface traction $\bar{t}'_\alpha(x^a)$ may be expressed in terms of the interface tractions $t'_\beta(x^a), \beta \in \{1,2,3\}$ and the tangential strain of the interface. In the numerical formulation, this tangential strain may be expressed in terms of the nodal displacements of the adjoining interface elements (see equations 4.2.8, 4.2.10), and so the unknown $\bar{t}'_\alpha(x^a)$ may be expressed in terms of other unknowns appearing in the system of equations. ...

As in the two-dimensional case, there are contributions to the matrix and the second member for each term of the double sum, equation 4.4.3. As explained in section 3.7. the integral over a free element of the shape function for a fixed node, multiplied by the kernel U (see fig. 4.4.10), must be multiplied by the known limiting value of traction as the fixed node is approached and placed in the second member. There are, of necessity, certain derogations

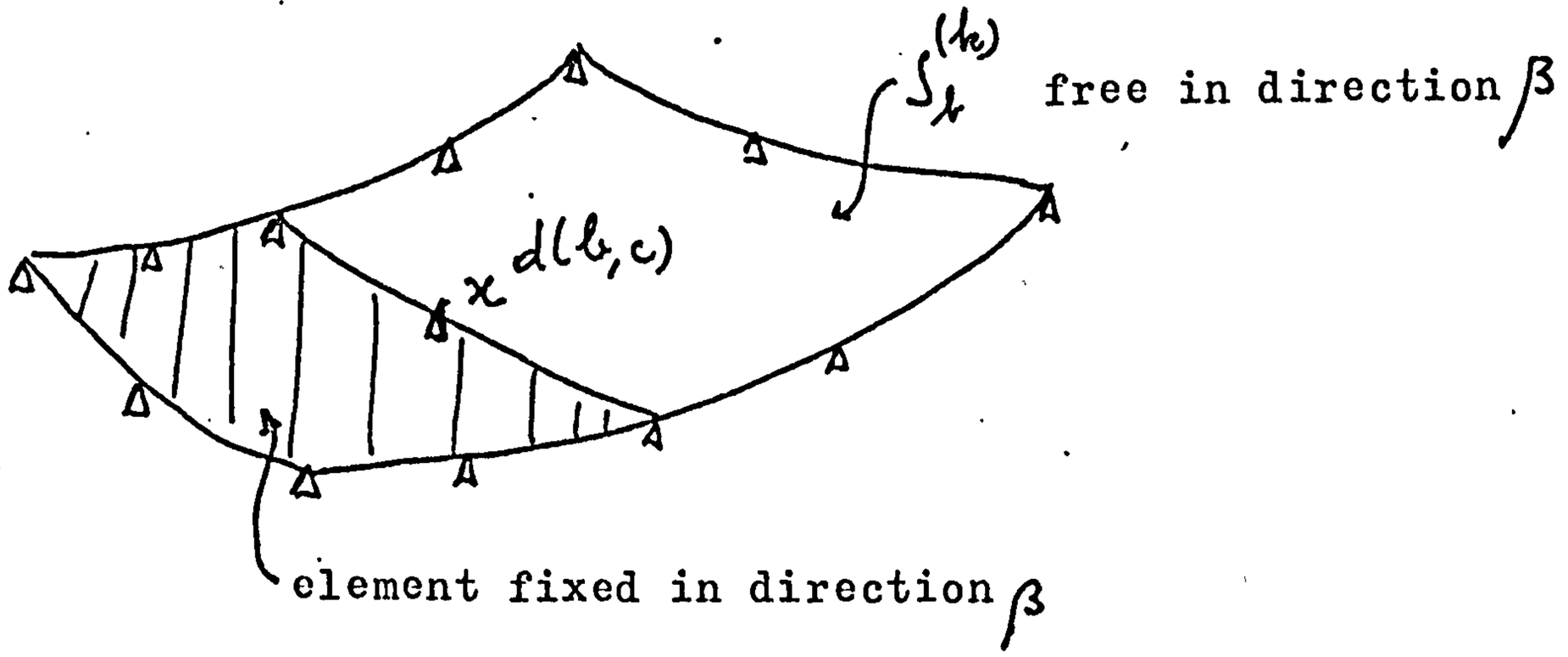


fig. 4.4.10 - case in which $\int U'_{\alpha\beta} M^c$ is multiplied by known traction and placed in second member

of this rule. Where it is specified that an isolated node is fixed, or that the common side of two elements is fixed, the procedure just described would lead to the presence in the system of equations of traction unknowns to which there correspond zero surface areas. In this case, the shape functions associated with the fixed node or the fixed line are multiplied by U and placed in the matrix, as in the two-dimensional analysis. Examples of such sets of shape

...

functions, for the case of quadratic functional variation, are shown in fig. 4.4.11.

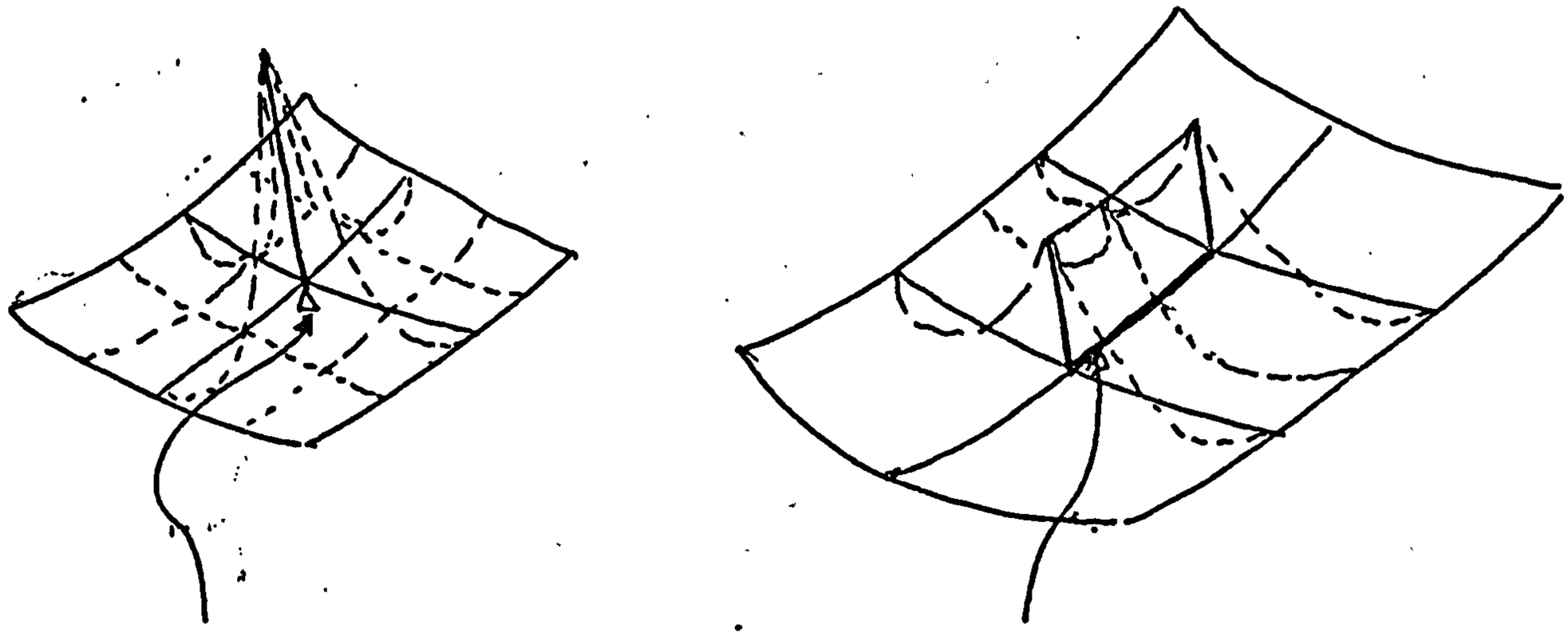


fig. 4.4.11 - shape functions associated with fixed nodes and lines.

Similarly, point and line loads are converted to equivalent tractions, multiplied by the integral of $U'_{\alpha\beta} M^c$ for the functions M^c shown in fig. 4.4.11, and placed in the second member. This "spreading" of point and line loads is not unrealistic because concentrated forces are impossible anyway.

During the integration for a node χ^a, the global components of the leading diagonal submatrix (equation 4.4.22) are summed. When the integration is completed, this is transformed (equation 4.4.2) and placed in the equations.

The equation coefficients are scaled, as previously, to give a numerically stable system (see section 3.3). For the analysis with subregions, the unit of distance is taken to be the greatest dimension of a subregion, and that of stress is taken to be the average modulus of elasticity.

This system of equations is reduced by Gaussian elimination, as described in chapter 5, and then the global components $u_i^{(k)}(x^a)$, $t_i^{(k)}(x^a)$ are calculated, using the transformation 4.4.2.

...

5.- Calculation of stress and displacement at interior points

For points on interfaces, the results may be calculated directly from the solution of the system of equations. To calculate stress and displacement at a point x inside the subregion k, the equations 2.3.1 and 2.3.4 for that subregion only are discretised in the same way as is the integral equation :

$$u_i^{(k)}(x) = \sum_{b=1}^{p^{(k)}} \sum_{c=1}^{m} t_j^{(k)}(x^d) \int_{S_b^{(k)}} U_{ij}^{(k)}(x, y(\xi)) M^c(\xi) J(\xi) d\xi$$

$$- \sum_{b=1}^{p^{(k)}} \sum_{c=1}^{m} u_j^{(k)}(x^d) \int_{S_b^{(k)}} T_{ij}^{(k)}(x, y(\xi)) M^c(\xi) J(\xi) d\xi \qquad (4.5.1)$$

$$\sigma_{ij}^{(k)}(x) = \sum_{b=1}^{p^{(k)}} \sum_{c=1}^{m} \left\{ t_s^{(k)}(x^d) \int_{S_b^{(k)}} D_{ijs}^{(k)}(x, y(\xi)) M^c(\xi) J(\xi) d\xi \right.$$

$$\left. - u_s^{(k)}(x^d) \int_{S_b^{(k)}} S_{ijs}^{(k)}(x, y(\xi)) M^c(\xi) J(\xi) d\xi \right\} \qquad (4.5.2)$$

The integrals of kernel-shape function products appearing in equations 4.5.1 and 4.5.2 are calculated using Gaussian quadrature formulae (60), the integration scheme being chosen in the same way as for the integral equation (see section 4.4.). The point x is not on the surface of the subregion k, so only the formula 4.4.4. is required. To calculate the number of subelements and order of integration formulae, the upper bounds given by Stroud and Secrest (60) are again used (equation 4.4.8). The function $\frac{1}{r^3}$, where r is the distance

...

between the two arguments of the kernels, is taken to be representative of the integrands, because this is,in critical cases, the most rapidly varying component of either kernel. The same approximations are made as previously (section 4.4), and in the notation of equations 4.4.11,

$$\mathcal{E} = \frac{2GJ}{R^3} \sum_{i=1}^{2r} \frac{(2n_i+1)(2n_i+2)}{2} \alpha_i^{2n_i} \qquad (4.5.3)$$

Using the same criteria as previously, n_i should be chosen such that :

$$\frac{1}{2}(2n_i+1)(2n_i+2)\,\alpha_i^{2n_i} \leq K \qquad (4.5.4)$$

where K is a constant.

Where there is body force, the integration just described gives only the complementary function $\hat{\hat{u}}_i^{(k)}$ (equation 4.3.4), and it is necessary to add the particular integral $\hat{u}_i^{(k)}$ to obtain the desired result.

...

V - PROGRAMMING

For a sophisticated method of calculation to be useful, it is essential that not only the numerical formulation but also the programming be carried out with care. For example, were it not for the development of sliding triangle, frontal and block solvers for symmetric banded systems of equations, the finite element method would be very inefficient In this chapter are described the overall structure of the program for three dimensional analysis, and the solutions adopted to programming problems more or less peculiar to the boundary integral equation method.

1.- Objectives

The program is designed so that, for problems of practical size,

a) it is easy to use

b) the cost of execution is low

c) it can easily be converted to run on machines other than that for which it was originally written.

...

The attainment of the first of these objectives depends upon the manner in which data is read and checked, and the presentation of results. Restrictions on the order of presentation of data, and on node and element numbering, for example, must be kept to a minimum, and it is essential that as much data as possible (e.g. normals to the surface) be calculated automatically by the program rather than entered by the user. The program must check the data read, and continue doing so no matter how many errors have been encountered.

The cost of execution is kept low both by attention to the detail of calculations, and by a global strategy in which core and files are efficiently used. The latter point is of vital importance because, the quantity of information to be handled being too great to be held in core, extensive use of files must be made. As will be seen later, the order in which certain calculations are done is dictated by the need for efficient file access, as indeed is the case for the equation solver of the program for two-dimensional analysis (section 3.4).

Absolute machine independence is incompatible with efficency, but a program can be written so that a minimum of change is required when converting to another machine. There are four principal areas of interest : input data decoding, the

...

relative length of reals and integers, file structure, and program length. The decoding routines provided by computer manufacturers are often insufficiently robust, and even where there is satisfactory error recovery it may be difficult to incorporate this into the overall error recovery scheme of the program. The efficient use of core requires that real and integer arrays be equivalenced, so to allow simple conversion a standard ratio length of real/length of integer must be chosen. In the specification of files, only those file structures that are permissible in all large computer systems should be used. Finally, the program must not be so long that it is confined to a few very large machines. It is in any case desirable to limit program length because of its influence on cost per second of computing time.

2.- General characteristics of the program

The problem size is subject to the following limits. The letters L, Q and C denote the cases of linear, quadratic and cubic functional variation respectively.

(a) 1 200 geometric nodes and interior points

(b) 400 surface and interface elements

(c) 10 subregions

(d) 150 (L), 75 (Q) or 50 (C) elements on the surface of any one subregion

(e) 900 equations in the set $S^{ii}_{u} \cup S^{ii}_{i} \cup S^{ii}_{E}$ (see fig. 4.4.8), for any subregion $R^{(i)}$

(f) 1 200 coefficients per equation in the shaded area of the matrix, fig. 4.4.8

(g) 4 800 equations in all

(h) between 2.56×10^6 and 3.84×10^6 matrix coefficients, depending upon the amount the unusable space on the matrix file (see section 5.4)

(i) 640 points with which are associated unknowns (as distinct from geometric nodes) with constraints upon displacement

(j) 5 load cases.

The program is written in Fortran for the CDC 6 600/7 600 series of computers, in such a way that conversion for the IBM 360/370 series would not be difficult. To this end, and also for efficient use of core, integer arrays are generally packed 4 coefficients per 60 -bit word. This allows the storage of integers up to 32K, which is ample, and the IBM equivalent would simply be 2 - byte (16 bit) integers and double precision (64 bit) reals. The CDC 7 600 Large Core Memory is not used. Variable - length direct access file records, permissible in CDC 7 600 Fortran, are not used because there exists no equivalent in IBM Fortran.

...

In the interest of efficiency, multidimensional arrays are not used.(this is less important where optimising compilers are available, but it is preferable to be sure that adressing is done in the most efficient possible way rather than hope that the optimiser arrives at the right conclusions).

The program consists of over 80 subroutines, comprising 10 000 cards of Fortran. The root segment contains, in addition to the main program, about 30 of these subroutines ; the rest are arranged as 13 overlays, each of which is loaded only once. The first six overlays, half the program as measured by number of cards, read, check and generate data. The rest construct and reduce the system of equations, calculate data at interior points and print the results. In appendix 1(are reproduced listings of the two overlays which construct and reduce the system, and supporting routines. The occupation of corc is shown in fig. 5.2.1

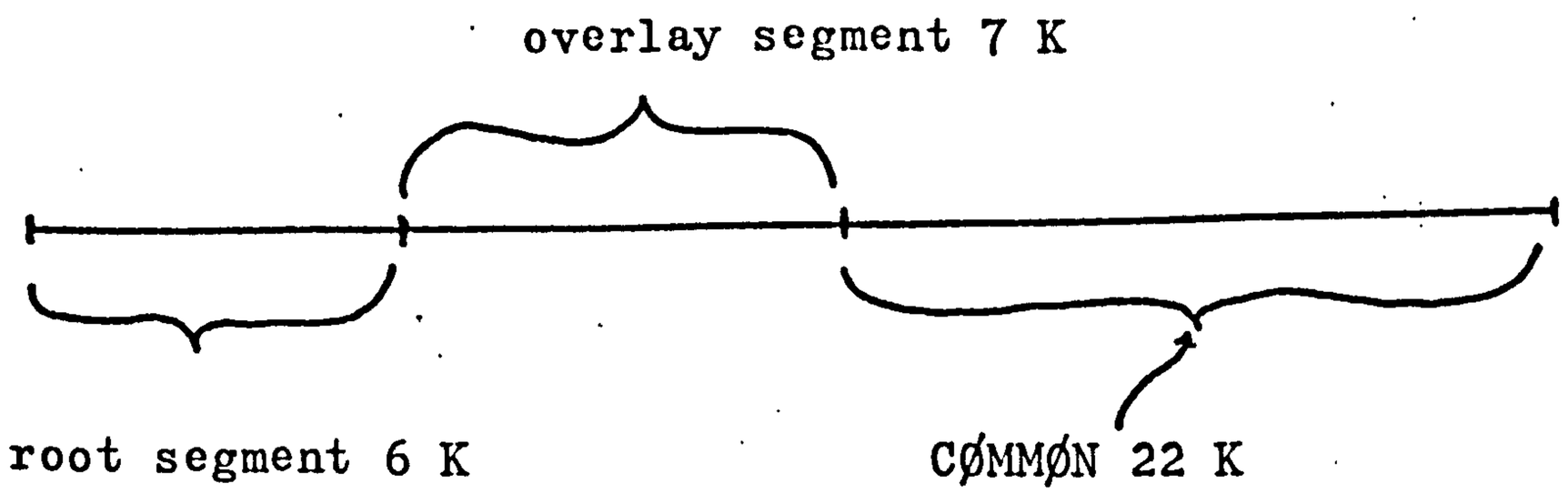


fig. 5.2.1 - occupation of CDC 7600 core by the program.

3.- Reading,checking and generation of data

Data is read in the following order (see example, section 6.2) :

(a) title and options	)
(b) coordinates of geometric nodes and interior points	) 1 st overlay
(c) nodes of surface and interface elements	) - 2 nd overlay
(d) definition of subregions	) - 3 rd overlay
(e) constraints on displacement	) - 5 th overlay
(f) loads	) - 6 th overlay

The options under (a) above include symmetry with respect to global cartesian coordinate axes, and whether it is desired to check the data only or check the data and, if no errors are found , carry out the analysis. The program can read the cartesian, cylindrical polar or spherical coordinates of nodes with respect to any system of axes. There are interpolation facilities for the generation of nodes along straight lines,circles or spirals. There are no restrictions on the order in which nodes are numbered ; there may be gaps in the numbering, and whilst there may not be more than 1 200 nodes, the maximum permissible node number is 2 400. The coordinates of nodes at midpoints of straight element sides need not be specified. ...

The second overlay includes automatic generation facilities for the easy specification of nodes of elements. The only restriction on numbering is that no element number may exceed 400.

The third overlay reads the elastic properties of each subregion $R^{(k)}$, and given one element and its orientation (node numbering anticlockwise or clockwise viewed from outside $R^{(k)}$) for each separate part of the surface $S^{(k)}$ finds the set of elements $S_l^{(k)}$, $l \in \{1, \ldots, f^{(k)}\}$, representing that surface. The program records the orientation of each element, to enable the outward normal to be calculated. A hole in the surface is detected by the program .

The fourth overlay finds and numbers all the degrees of freedom, and writes the sequential file LUR, one record per subregion, of coordinates XL of points with which are associated degrees of freedom and the numbering NL of the degrees of freedom associated with each surface or interface element.

The fifth overlay reads constraints on the displacement of elements, lines and nodes, and merges this data into tables of constraints on the displacement of each point with which is associated degrees of freedom : the arrays constructed are NCϕ, the number of constraints at a point, and IDCϕ, DCϕ, which give the direction cosines of degrees

...

of freedom (equation 4.4.2). In addition, information on the fixity of elements is added to the information on file LUR (see discussion on free element next to fixed node, section 4.4).

The sixth overlay reads loads, by which is meant both known nonzero tractions and known nonzero displacements. The user may specify tractions on elements, forces distributed over lines and concentrated at nodes, gravitational and centrifugal loading, and nonzero displacements of elements, lines and nodes. Forces distributed over lines and concentrated at nodes are distributed as described in section 4.4. The program, in addition to performing the distribution, adjusts the traction on each adjacent element so that the resultant of the equivalent tractions coincides in position as well as in magnitude with that of the applied load. The procedure is rather the reverse of that for finite elements (displacement method), in which tractions are represented by equivalent nodal forces.

Loads are read by load case, but for maximum economy all load cases are treated simultaneously during construction and reduction of the system of equations, and calculations for interior points. In addition, the table of known functions at surface points, even for a single load case, is too large to be held in core. The data is therefore stored on a direct access file ; this file is later read

(by the seventh overlay), the data rearranged and placed on the sequential file LUA, which is the file of known functions and discontinuities read during construction of the matrix (section 5.4).

All data is read in the format (A4, 76A1), the field A4 corresponding to a keyword. The decoding subroutine decodes the remaining 76 characters, provided the keyword is valid, without using E, F or I conversions. The procedure is therefore crashproof ; if illegal data is found, a flag is set and control is returned to the calling subroutine. The program as a whole detects errors at three levels :

(a) decoding

(b) single card

(c) completeness and compatibility of data sets consisting of more then one card.

At level (b), the validity of data on just one card is considered (e.g. upper limit on node number), whilst at level (c) the compatibility of data on a card with data read previously is checked (e.g. existence of a node of an element, existence of an element to be loaded, limit on number of load cases). There are three error flags :

...

JØHN : denotes error found by overlays 1-4, i.e.in the specification of geometry

JØE : denotes error found by overlay 5, i.e. in the specification of constraints on displacement

KAY : denotes error found by overlay 6, i.e. in the specification of loads.

Checks at levels (a) and (b) are continued no matter what errors have already been found. Checks at level (c), however, are suppressed if it is indicated by the flags JØHN or JØE that the data to be checked against is faulty. If, when all the data has been read, any of the three flags are set, execution is terminated. About 100 different error messages can be printed ; the system of flags is intended to ensure that the user is given the greatest possible amount of information about errors in the data during one run of the program.

...

4.- Construction and reduction of the system of equations

Prior to construction of the system, the sequential files LUA, LUG and LUE are created. To file LUA are written the given values and discontinuities of the complementary function for each surface element of each subregion. The data for a subregion begins at the beginning of a record, and as many records of 1 440 words as necessary are written for each subregion. In fig. 5.4.1, is shown the particular case

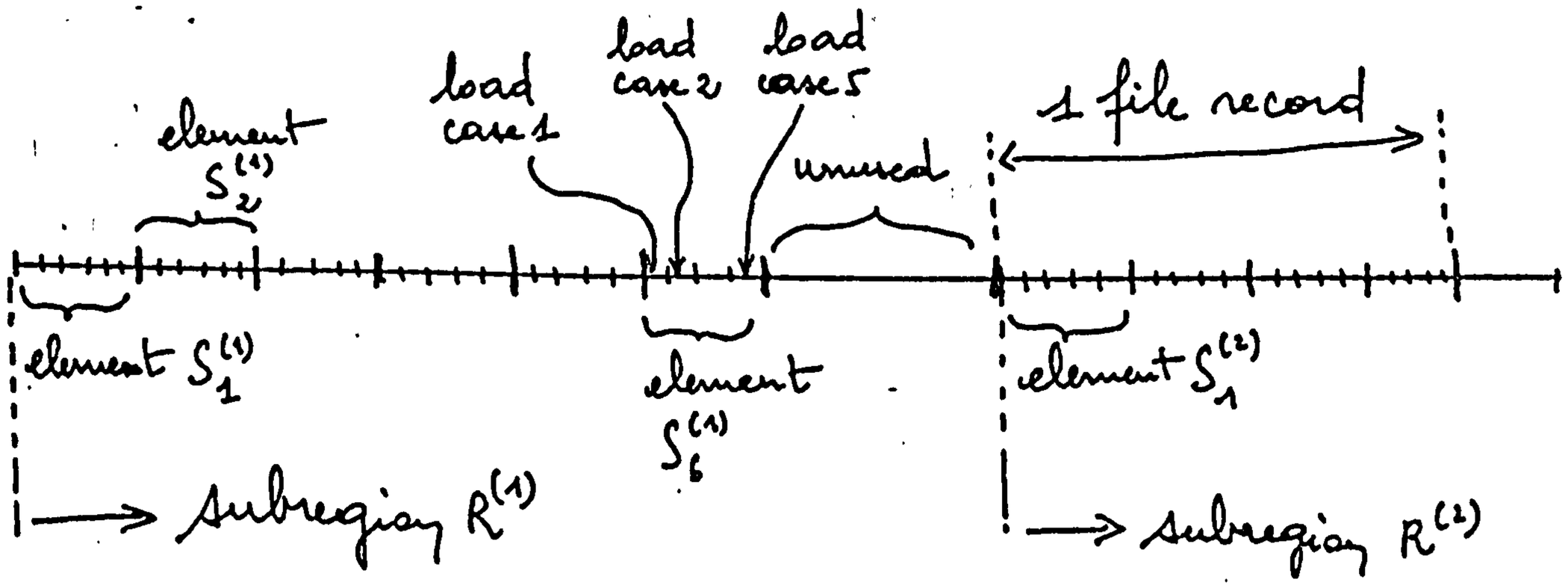


fig. 5.4.1 - structure of file LUA (record length = 1440 words

of cubic functional variation, 5 load cases and 6 elements on the surface of the first subregion. The program always puts the data for as many elements as possible into each record, according to the choice of functional variation and number of load cases.

...

File LUG holds the coordinates of, and outward normal and Jacobian at, each of the integration points for the 3 x 3 Gauss formula, 1 subelement (see integration scheme, section 4.4). This information is filed because the 3 x 3 formula is the most commonly used, and it is desirable to avoid repeated geometric calculations. The format of file is the same as LUA (fig. 5.4.1), except in that the record length is 1008, and instead of load cases there are 2^N reflections, where N is the number of symmetries with respect to global coordinate axes.

File LUE holds factors for the elimination from the system of surplus traction unknowns at points shared by an interface and the surface (section 4.4). Substitutions are made in terms of the functions over every interface element adjoining the point, the data on the file being arranged in the order in which the unknowns to be eliminated appear in the table NL (file LUR). The format of file LUE is the same as that of file LUA, except in that the record length is 500, instead of elements there are unknowns to be eliminated, and instead of load cases there are adjacent interface elements.

The subroutine ITFØR of overlay 8 (for listing, see Appendix 1) reads the files LUR, LUA, LUG and LUE, constructs equations and writes them to the direct access file LUM and to

...

the sequential file LUS. The formats of files LUM and LUS are identical : as many equations as possible are put into each record of 2 410 words ; the first equation for each subregion begins at the beginning of a record (fig. 5.4.2.)

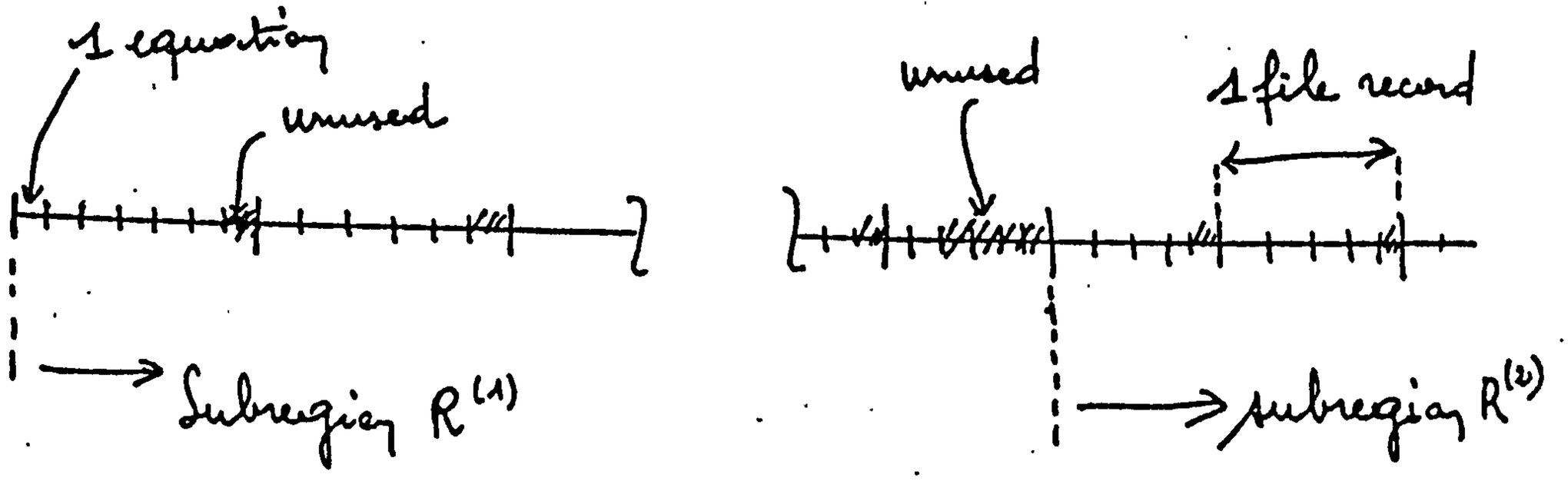


fig. 5.4.2. - structure of files LUM, LUS (record length = 2 410 words)

The maximum length of an equation is 1 205 coefficients, so the wastage of space on files LUM, LUS cannot exceed 33 % (ignoring unused space at the end of each subregion).

ITFØR consists of a series of nested loops, the outermost of which is over subregions. For each subregion, before construction of the equations, the data relevant to that subregion on files LUA, LUG and LUE is copied to files LUB, LUC and LUF respectively. The equations are

...

assembled in blocks of three LUM records each. For each block, the files LUB, LUC and LUF are read once from beginning to end, and all equations in the block are constructed simultaneously. The procedure is essential if the input time for the three files is to be kept within reasonable limits.

The subroutine ITSØL of overlay 9 (for listing, see Appendix 1) reduces the system of equations held on file LUM. The procedure is the same as that for the two-dimensional analysis(section 3.4), except in that the second members are on the file LUM (which simplifies the forward reduction), and addressing is more complicated. ITSØL leaves the solution on a sequential file, one record per subregion, rather than on the file LUM in place of the second members.

After the solution has been calculated, it is multiplied by the copy of the unreduced matrix on the file LUS and the result subtracted from the second members, to calculate the residues. The norm of the residues is calculated and printed for each load case.

...

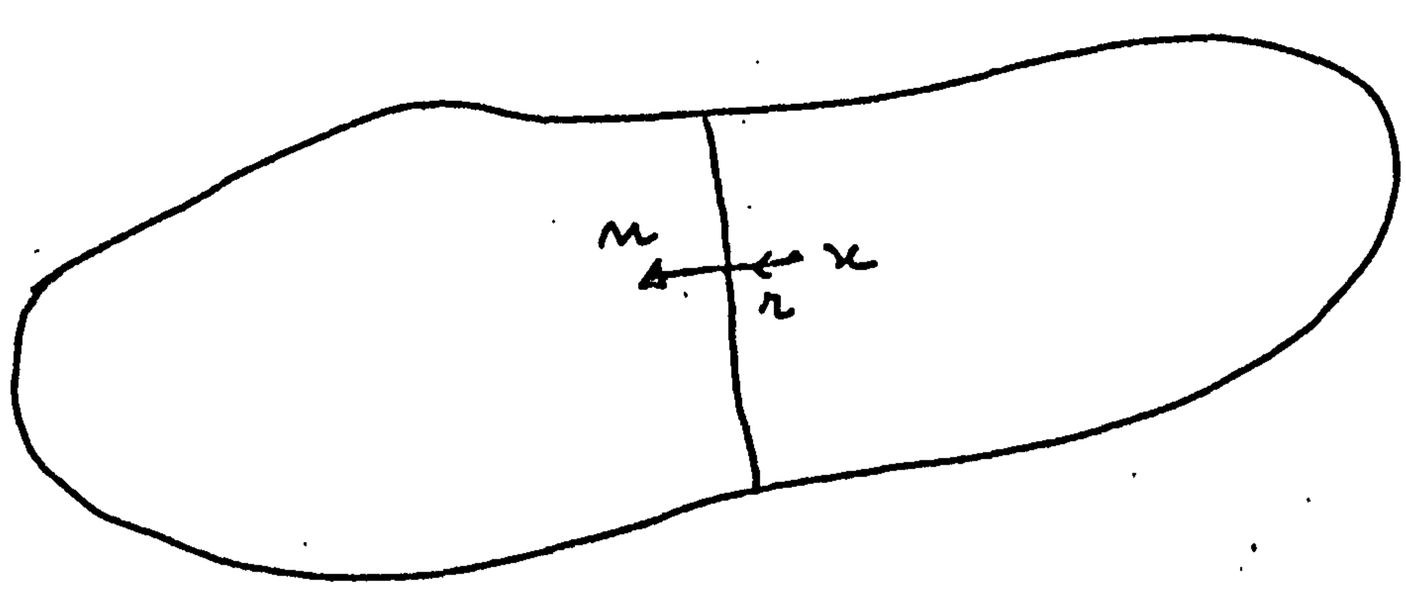
5.- Calculation of results at points inside subregions

Before calculation of results at interior points, two preliminary operations are performed. Firstly, file LUA is rewritten. The data on LUA is merged with the solution, and placed on another file ; this file and file LUE are then read, the eliminated tractions at points shared by the surface and an interface are calculated and added to the data, and the components of displacement and traction in the global coordinate directions are calculated. These global components are rewritten to file LUA, in the format shown in fig. 5.4.1.

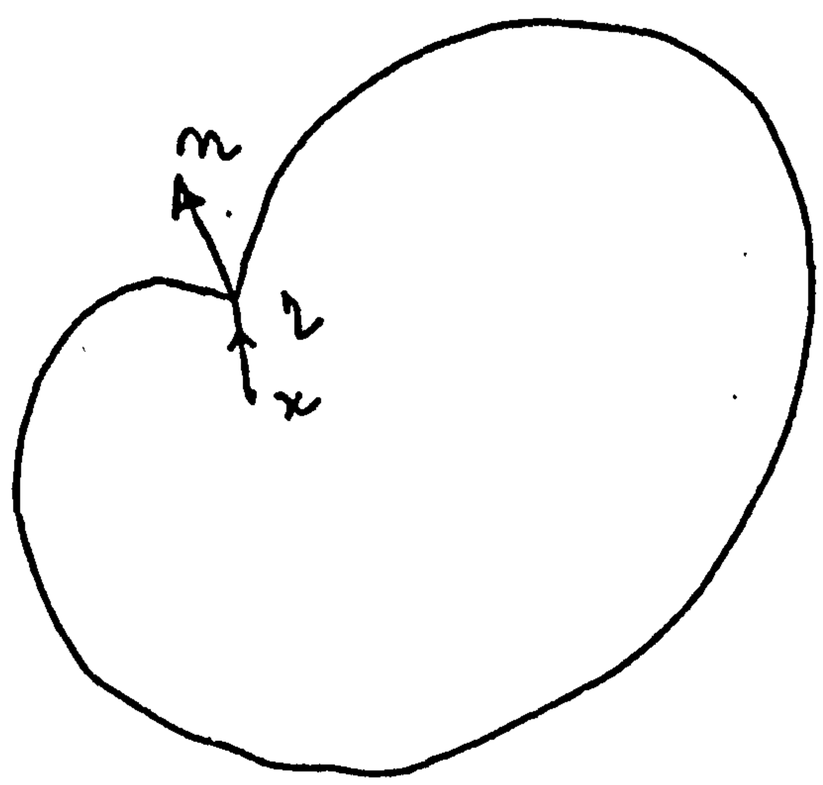
Secondly, the program determines in which subregion each interior point lies. This it does by finding, for each interior point, the nearest point on any surface or interface element, and calculating, for each subregion on the surface of which this point lies, the inner product of the outward normal and the vector joining the interior and surface points (fig.5.51 (a)). Where the normal at the nearest point is discontinuous (fig. 5.5.1 (b)), the average of the limiting values is taken. If the point is found not to be inside any subregion, a diagnostic is printed and calculations for that point are suppressed.

...

The integration for displacement and stress at interior points is carried out in the same way as is the construction of equations. For each subregion, the relevant data is copied from files LUA and LUG to files LUB and LUC.



(a) normal continuous at nearest point



(b) normal discontinuous at nearest point.

fig. 5.5.1 - location of interior point x.

Stress and displacement are calculated, for all load cases simultaneously, in blocks of 32 nodes each. For each block, the files LUB and LUC are read once from beginning to end, and integrations are performed for all nodes simultaneously. The results for each block are sorted by load case and placed on a direct access file, so that, after all the results have been calculated, they can be printed by load case.

VI - EXAMPLES

Three examples of three-dimensional analysis are presented here. They are :

(a) the analysis of a thick cylinder. This is intended principally as a test of the integration scheme.

(b) the analysis of a pipe connection. This is a typical calculation of stress concentrations in a mechanical component.

(c) the calculation of the stress intensity factor for a crack in a rolling mill cylinder.

The results for the thick cylinder are compared with finite element results and the exact solution, and those for the flange are compared with finite element and experimental results.

6-a) The thick cylinder

The characteristics of the thick cylinder considered in the first part of this example are as follows :

internal diameter = 20 mm

external diameter = 40 mm

modulus of elasticity = 21 000 daN/mm2

Poisson'ratio = 0,3

The cylinder is subjected to an internal pressure of 2.0 daN/mm2, and is considered to be in a state of plane strain with respect to the axial direction. A 90º sector, of height 40 mm, is analysed ; the discretisation of the surface is shown in fig. 6a1. There are two subregions. The finite element discretisation (20 - node isoparametric elements) is shown in fig. 6a2.

In table 6a3 are shown the calculated and exact solutions, for five values of radius. For both analyses by the integral equation method, the constant K appearing in inequality 4.4.13 in taken such that there will be at least two Gauss points for an interval which subtends an angle of one radian at the functional singularity, for the calculation of equation coefficients. In the following discussion, this is referred to as "precision of integration 2.0". It may be seen that, whilst the results obtained with linear functional variation are less accurate than the finite element results, those obtained with quadratic variation are as good, for displacements, and better, for stresses, than the finite element results. In the case of quadratic functional variation, the norm of residues is 1.9×10^{-11} (CDC 7600), so the system of equations is numerically stable. The execution time of 38 seconds for quadratic variation is high compared with the execution time

...

7.5 seconds for the analysis by the finite element method, but the problem of the thick cylinder is a small one for which the time required to solve the equations is heavily outweighed by that required to construct the system and especially to integrate around the singularity of the kernels U and T. The significance of this is that the integration time, especially that for integration around the singularity, climbs relatively slowly as problem size increases.

In the second part of this example, the internal diameters 20 mm, 30 mm, 36 mm and 38 mm are considered, to test the stability and precision of the algorithm as surface elements become long and narrow and very close to one another compared with their dimensions. Cubic functional variation is adopted for this test ; the discretisation for internal diameter 38 mm is shown in fig. 6.a.4. The precision of integration 2.0 is retained. The execution times and norms of residues are shown in table 6.a.5. The increase in execution time with internal diameter is due to an increase in the number of integration points required to obtain the requested precision. The variations of exact and calculated hoop stresses are shown in fig. 6.a.6. The error in calculated stresses remains below 3 % of the maximum value of stress even at internal diameter = 36 mm,

...

at which stage there are surface elements with length/ breadth ratio 10. At internal diameter = 38 mm, i.e. thickness = 1 mm, the error is still less than 10 %. This shows that the algorithm for choice of integration scheme continues to work well even in extreme cases.

E = 20000

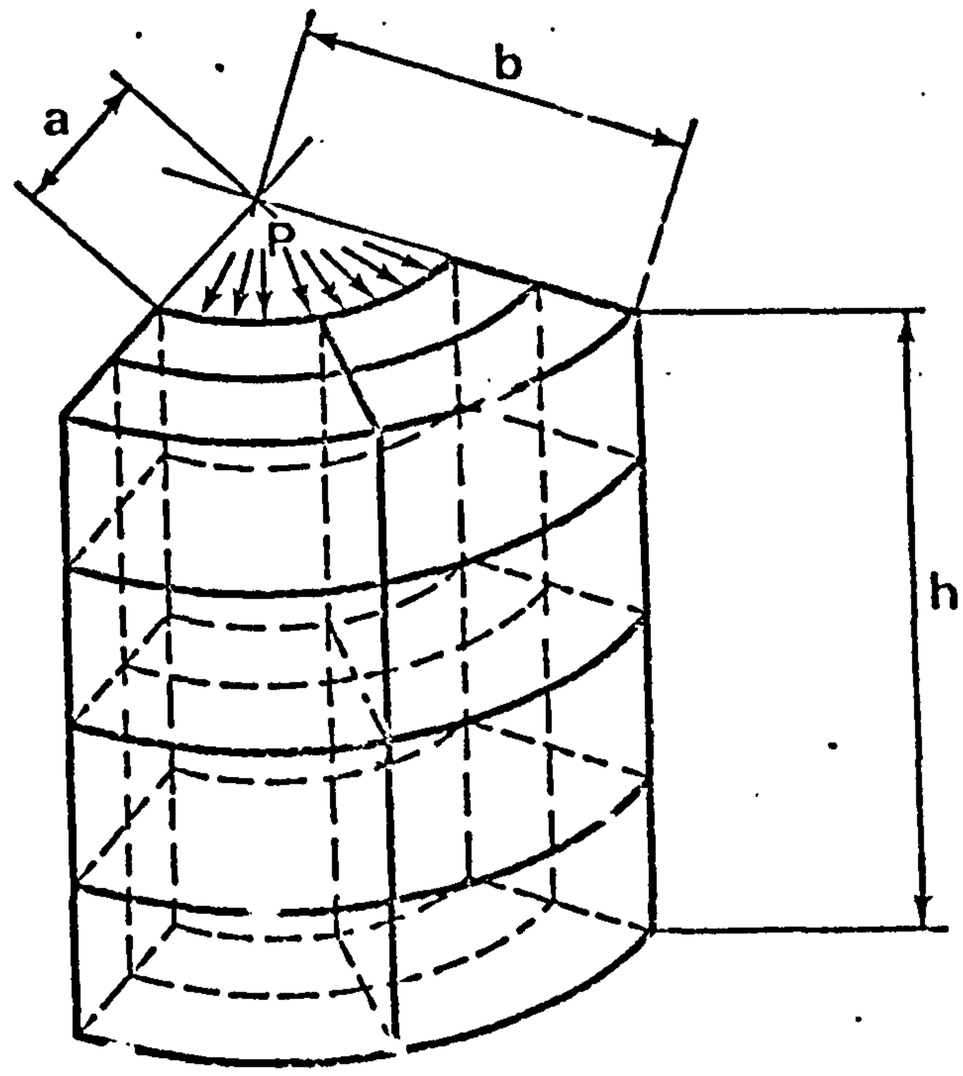
V = 0.3

a = 10.0

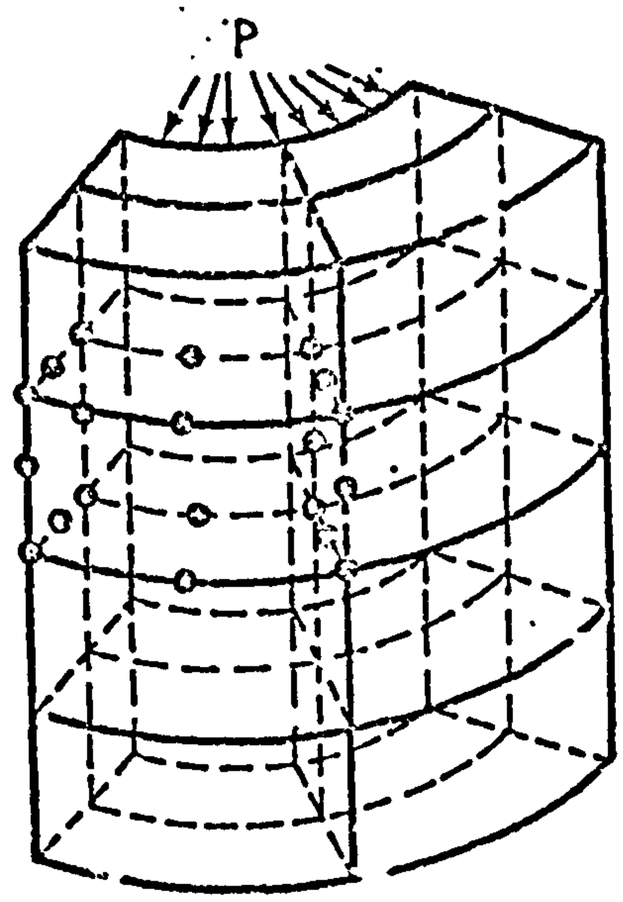
b = 20.0

h = 40.0

P = 2.0



6.a.1 - Integral equation network of the thick cylinder



6.a.2 - Finite element mesh

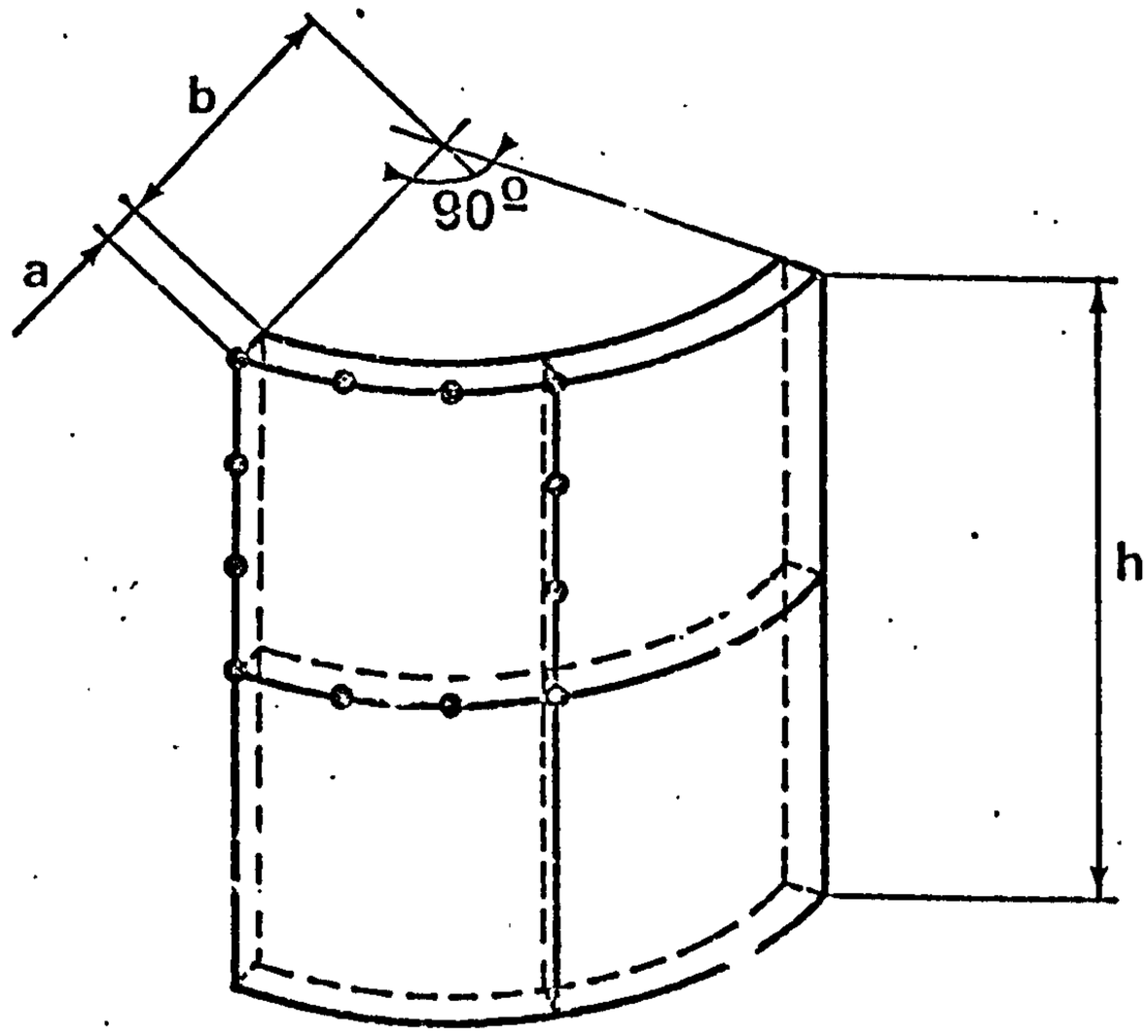
Function	Radius	Exact Solution	Finite Element	integral equations	
				linear	quadratic
radial displacement	10.0	1.904	1.905	1.818	1.905
	12.5	1.602	1.600	1.568	1.600
	15.0	1.415	1.414	1.319	1.414
	17.5	1.293	1.292	1.234	1.291
	20.0	1.212	1.212	1.150	1.211
radial stress	10.0	-2.00	-1.74	-1.38	-1.85
	12.5	-1.04	-1.16	-1.32	-1.13
	15.0	-0.52	-0.36	-0.18	-0.41
	17.5	-0.20	-0.24	-0.12	-0.22
	20.0	0.00	0.07	0.11	0.04
hoop stress	10.0	3.33	3.44	2.71	3.35
	12.5	2.37	2.33	2.01	2.37
	15.0	1.85	1.92	1.52	1.85
	17.5	1.54	1.52	1.43	1.54
	20.0	1.33	1.36	1.21	1.33

Table 6.a.3 - Comparison between calculated and exact solutions

a = 1.00

b = 19.0

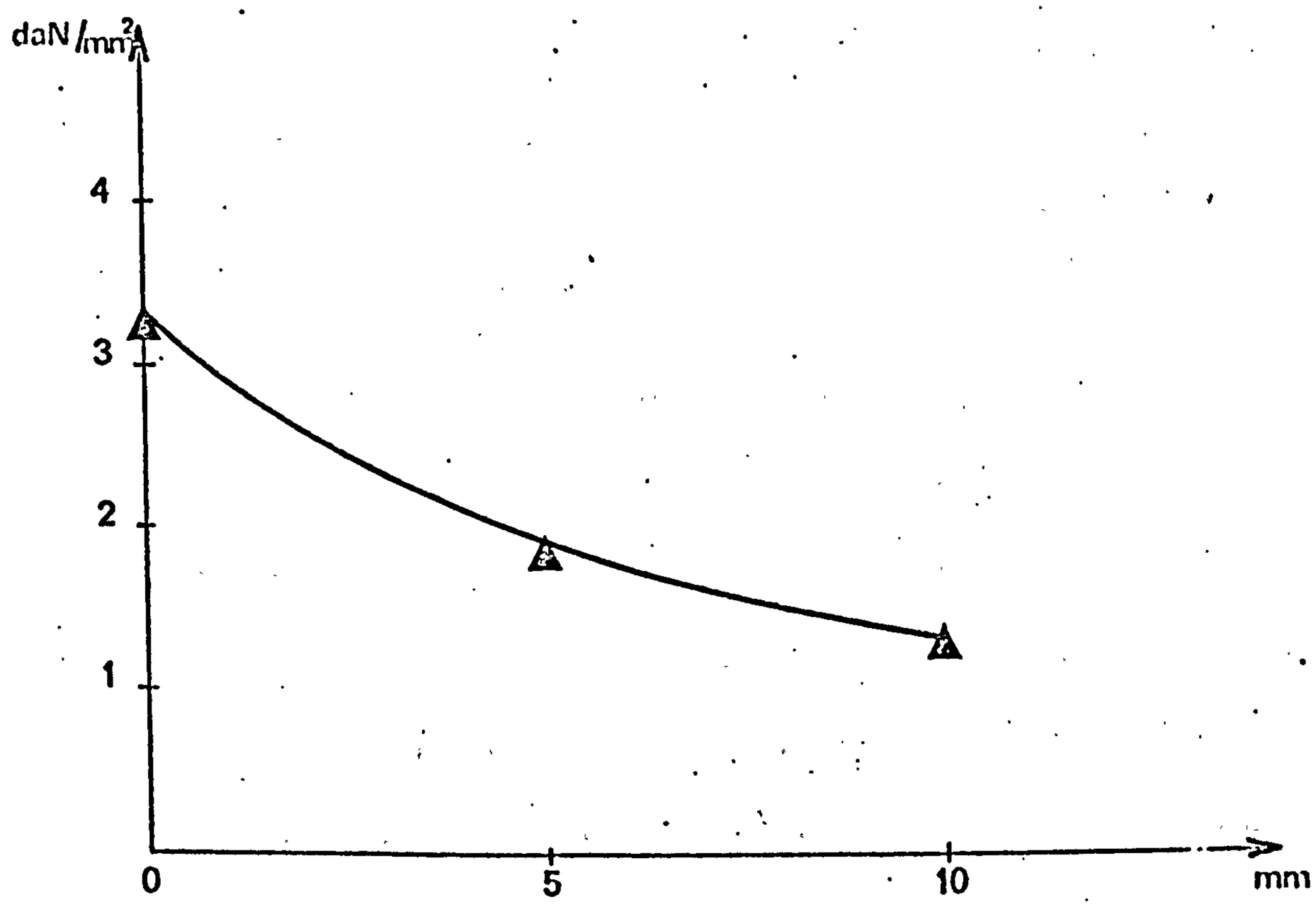
h = 40.0

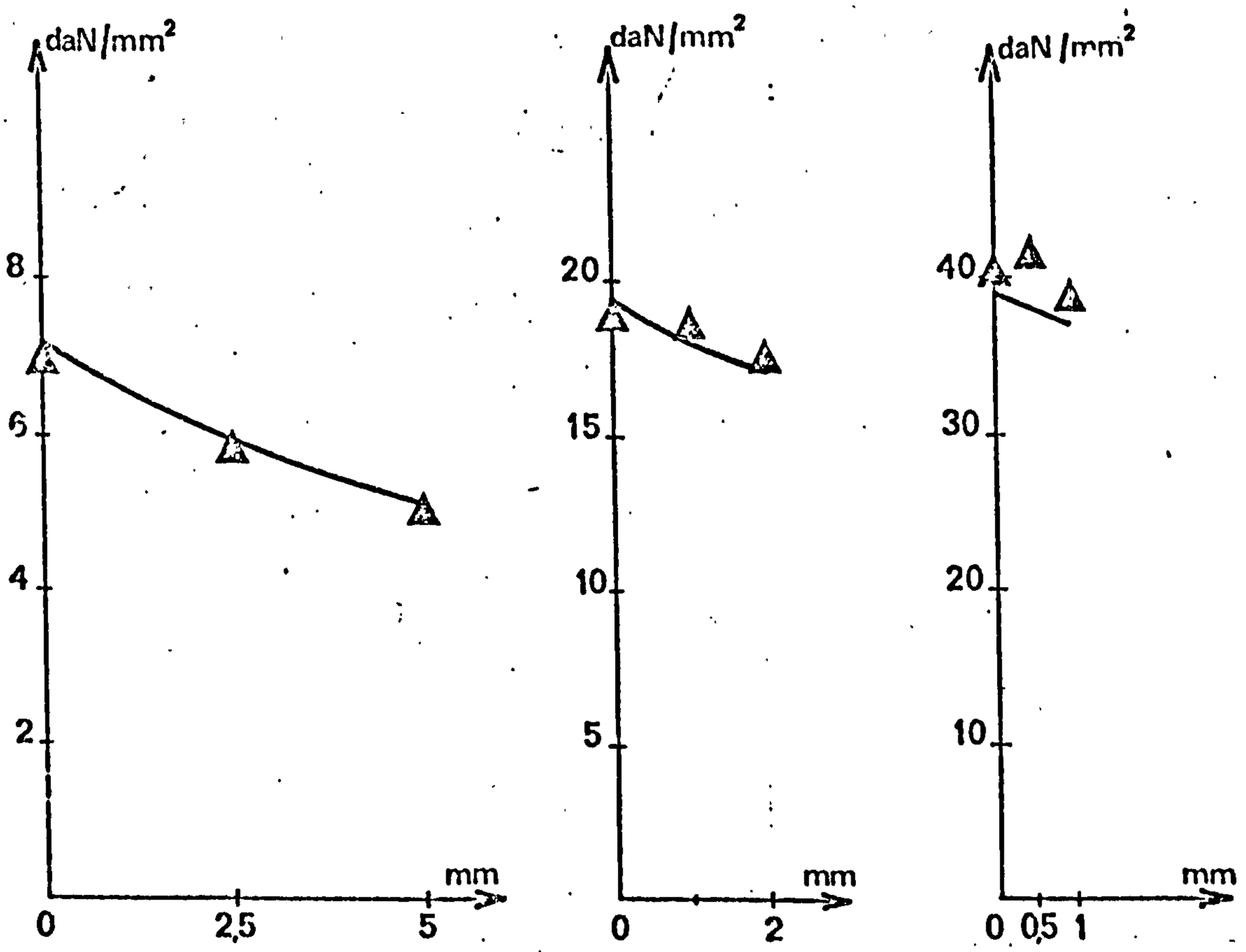


6.a.4 - Integral equation network for internal diameter 38 mm

internal diameter mm	execution time	norm of residues
20	21 s	4.5×10^{-13}
30	23 s	1.4×10^{-12}
36	35 s	4.0×10^{-12}
38	50 s	7.8×10^{-12}

table 6.a.5





6.a.6 - Variations of exact and calculated hoop stress

6-b) <u>The pipe connection</u>

The flange to be analysed is the flange B3 which was the subject of experimental and theoretical studies at the Centre Technique des Industries Mécaniques, France (72). The flange (fig. 6.n.1) is designed to connect two pipes carrying fluid under pressure ; the arrangement is symmetric with respect to the plane of the joint, so only one half need be considered. The twelve connecting bolts being equally spaced, there are also planes of symmetry passing through pairs of holes, and between holes, and only a 15º sector need be considered. For the purposes of numerical analysis, only 100 mm of the pipe are taken into account (fig. 6.b.2). The dimensions of the flange are shown in fig. 6.b.2. The elastic constants are :

modulus of elasticity = 21 000 daN/mm2

Poisson's ratio = 0.3

Two load cases are considered : bolt tension of 5000 daN /bolt, and bolt tension plus an internal pressure of 0.45 daN/mm2.

For the analysis by the integral equation method, the sector of flange shown in fig. 6b2 is divided into four subregions, and the symmetry with respect to the plane equidistant from two holes is used to avoid the need to discretise that plane. The discretisation, consisting of 57 surface and interface elements of which three are degenerate, and 180 nodes, is shown in fig. 6b3 (a). The 80 cards of input data required to define the discretisation are shown in fig. 6b3 (b). The interpretation of data is similar to that for the plane problem (section 3.6 and especially fig. 3.6.12). The precision of integration is specified on the second card ; it is expected that when sufficient experience has been gained a default option will be provided for this parameter. The elastic properties may be defined on the second card or on the cards SØUS which define the subregions ; values given on the latter cards override any data on the second card.

The finite element network, consisting of 64 twenty-node isoparametric elements and 501 nodes, is shown

...

in fig. 6b.4 . Over 200 cards of data are required to define the discretisation.(the data generation facilities of the finite element program are comparable in sophistication with those of the integral equation program). The finite element program used incorporates a variable bandwidth sliding triangle algorithm for the reduction of the system of equations. The central processing time (CDC 7 600) required for the finite element analysis is 37 seconds.

The execution time for the analysis by the integral equation method, with quadratic functional variation and precision of integration 2.0 is 57.6 seconds, or 65 % greater than that for the finite element analysis. The input-output time for both analyses is very low. The norms of residues for the two load cases are 1.3×10^{-10} and 6.9×10^{-11}. In fig. 6b.5 to 6b.12 are shown the variation of meridonal and circumferential stresses over the outer and inner surfaces, for the two load cases. The results are presented as follows :

...

6.b.5	–	outer surface, on symmetry passing through 2 holes	bolt tension only
6.b.6	–	outer surface, on symmetry passing between holes	
6.b.7	–	inner surface, on symmetry passing through 2 holes	
6.b.8	–	inner surface, on symmetry passing between holes	
6.b.9	–	as 6.b.5	bolt tension + internal pressure
6.b.10	–	as 6.b.6	
6.b.11	–	as 6.b.7	
6.b.12	–	as 6.b.8	

The integral equation, finite element and experimental results all show, as would be expected, that there is a strong stress concentration in the radius between the plate and the cone, and a weaker stress concentration at the junction of the cone and the pipe. The stress in the radius remains sensibly constant under different loading conditions, whereas that at the junction of the cone and the pipe varies greatly with pressure. It is therefore to be expected that, under repeated loading and unloading, the flange would fail in fatigue at the junction with the pipe, rather than at the radius. The state of stress is shown to be practically axisymmetric everywhere but in the plate. ...

However, certain of the calculated and experimentally determined results must be regarded with scepticism. In both numerical analyses the radius is represented by only one element, and, more seriously, the junction with the pipe is represented as a corner rather then as a radius. Undoubtedly, in reality, there was a radius between the cone and the pipe, but there exists no data on this so it could not be modelled numerically. The results at the junction are quite meaningless, and the fact that the value obtained by the finite element method is higher than that given by the integral equation method is due to the differing fineness of networks. By putting more and smaller finite elements or surface elements near the junction, results of any magnitude could be obtained. The experimental results at points where stress varies rapidly are no more reliable than the numerical results, because the strain guages are of significant length compared with the dimensions of the region of stress concentration.

The flange B3 was the first practical problem to be analysed by the integral equation method, and the discretisation of the surface (fig. 6b3) is probably not ideal. The results could be improved by moving certain nodes towards the junction with the pipe. The problem is relatively expensive to solve by the integral equa-

...

tion method because the surface area/volume ratio is high, and the finite element method is economical because the bandwidth is small. In the analysis by integral equations, most of the computing time is spent integrating for the matrix, whereas in the analysis by the finite element method the reduction of the system is the longest calculation. However, by comparing with the analysis of the thick cylinder, it can already be seen that the integral equation method becomes more economical as problem size increases. In fig. 6b.14 is reproduced a typical page of results.

An analysis was performed with the same discretisation but with precision of integration 3.0 instead of 2.0, to observe the effect of the latter parameter on accuracy and execution time. The effect on accuracy turns out to be negligible, the calculated stresses changing by not more than 0.1 daN/mm2. The execution time, however, increases to 71 seconds. Evidently, precision of integration 2.0 is nearer the optimum than is 3.0 This may not apply to the case of cubic functional variation, however, and it could be that the precision of integration must be increased with problem size.

Analyses were carried out using the same discretisation as before, but with linear and cubic functional variation.

...

statistic or result	linear	quadratic	cubic
execution time, secs.	15	57	166
norm of residues	5.7×10^{-12}	6.9×10^{-11}	4.3×10^{-10}
stress in radius (merid.	10.1	11.1	11.3
daN/mm2 (circ.	3.76	4.27	4.35
stress at pipe (merid.	2.80	4.10	5.30
junction, daN/mm2 (circ.	5.32	6.43	6.85

fig. 6b13 - effect of functional variation on results

In table 6b13, are shown some typical results, for the second load case. The convergence to a limit may clearly be seen, for all results except the meridional stress at the pipe junction, which, as has already been noted, is undefined. The maximum stress in the radius, calculated using quadratic variation, must in fact be very near the exact value, and even the analysis with linear variation gives this stress to within about 10 %, for an execution time of only 15 seconds. As the system of equations increases in size, the norm of residues also increases, but the system is still very stable.

...

Finally, an analysis was carried out, using quadratic functional variation, but which only two subretions. The calculated stresses in this case are within 0.1 daN/mm2 of those previously obtained, so the introduction of interfaces does not seriously affect accuracy. The execution time for the analysis with two subretions is 76 seconds, or about 30 % higher than with four subregions.

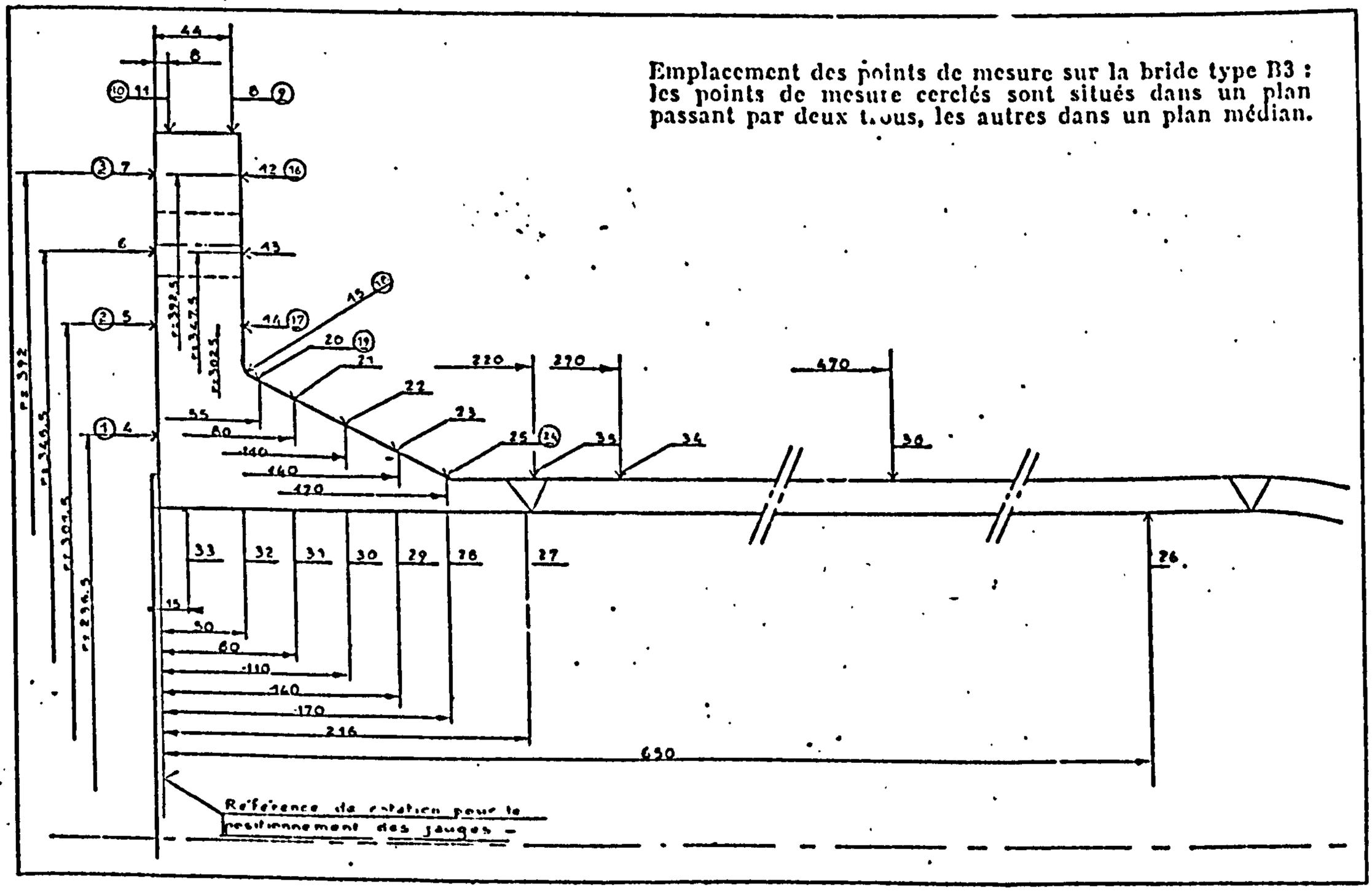


fig. 6.b.1 - Section of the flange B3

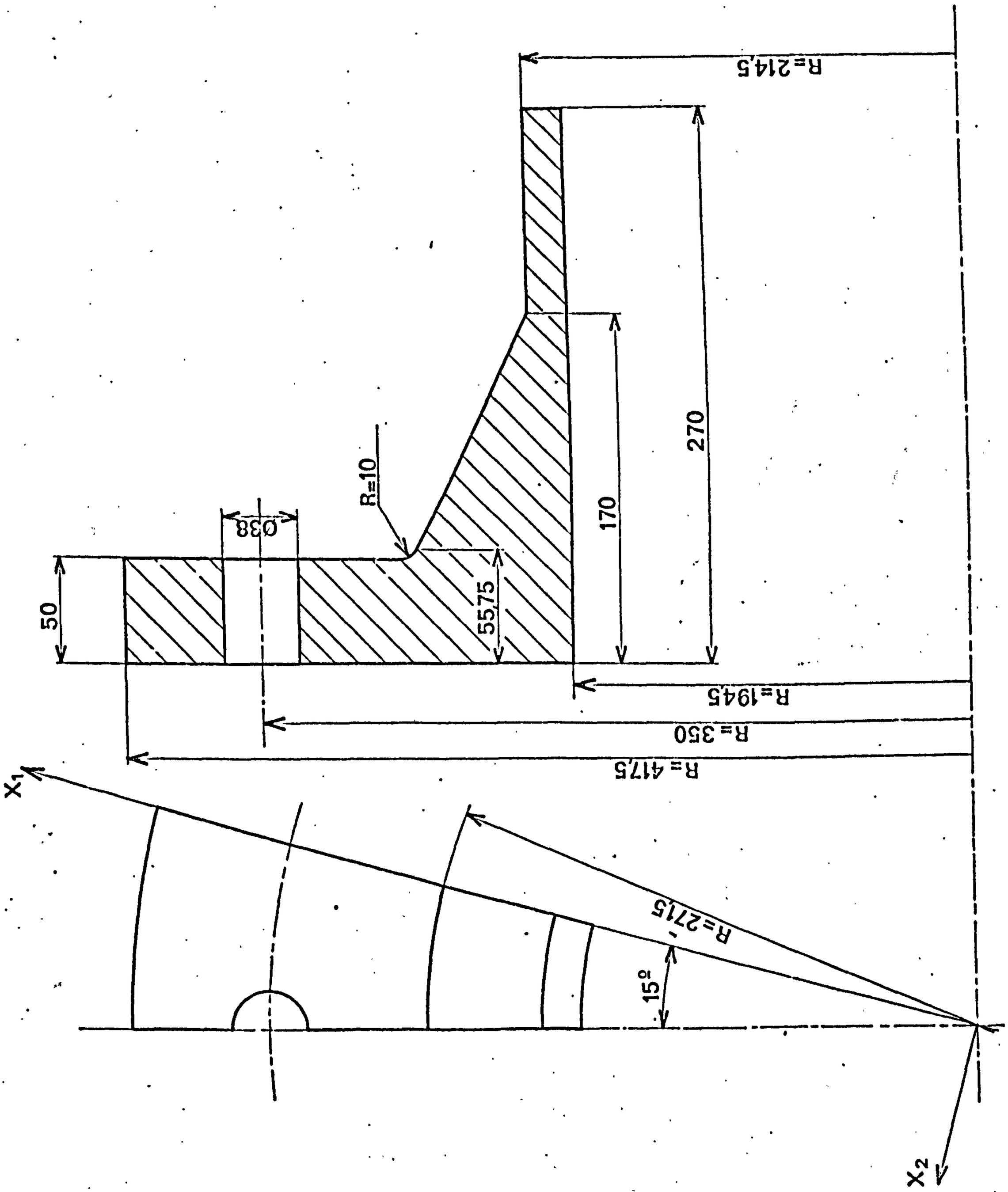


fig. 6.b.2 - the dimensions of the flange

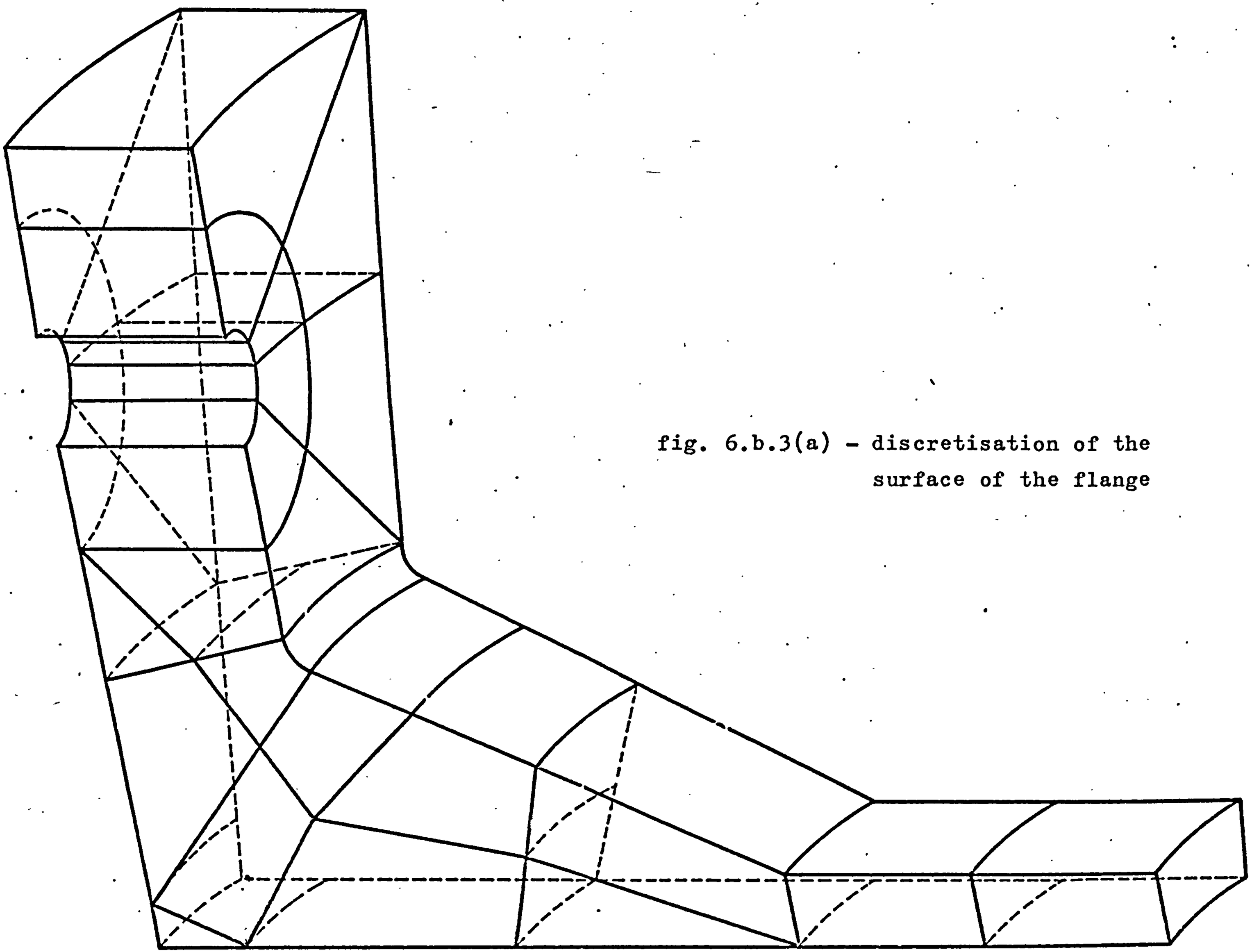


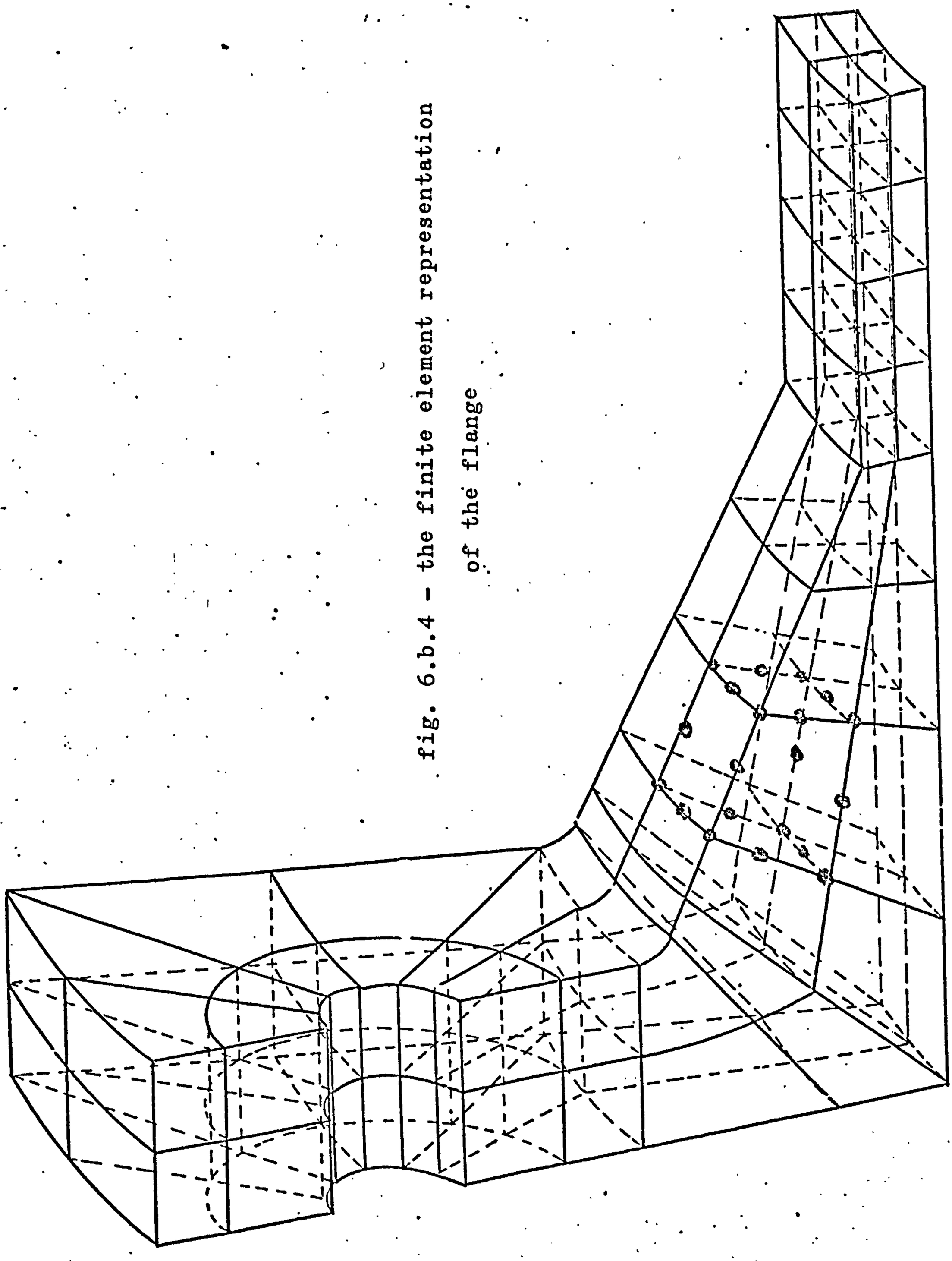
fig. 6.b.3(a) - discretisation of the surface of the flange

```
            ETUDE DE LA BRIDE R3 PAR LA METHODE DES EQUATIONS INTEGRALES
      OUI       21000.          0.3        0.006                         2.0          P    2
      CYL     5  194.5          0.0       270.0       15  194.5       15.0      270.0      5
      CYL     1  214.5          0.0       270.0       11  214.5       15.0      270.0      5
      CYL   205  194.5          0.0       170.0      215  194.5       15.0      170.0      5
      CYL   201  214.5          0.0       170.0      211  214.5       15.0      170.0      5
      CAR *   5                                      205                                 100
      CAR *  10                                      210                                 100
      CAR *  15                                      215                                 100
      CAR *   1                                      201                                 100
      CAR *   6                                      206                                 100
      CAR *  11                                      211                                 100
      CYL   309  194.5          0.0        95.0      319  194.5       15.0       95.0      5
      CYL   305  219.5          0.0       102.5      315  219.5       15.0      102.5      5
      CYL   301  244.5          0.0       110.0      311  244.5       15.0      110.0      5
      CYL   409  194.5          0.0        22.5      419  194.5       15.0       22.5      5
      CYL   405  228.0          0.0        52.0      415  228.0       15.0       52.0      5
      CYL   401  259.5          0.0        80.0      411  259.5       15.0       80.0      5
      CYL   439  194.5          0.0         0.0      449  194.5       15.0        0.0      5
      CYL   509  204.5          0.0         0.0      519  204.5       15.0        0.0      5
      CYL   505  252.0          0.0        40.0      515  252.0       15.0       40.0      5
      CYL   501  271.5          0.0        55.75     511  271.5       15.0       55.75     5
      CYL   551  275.25         0.0        51.50     561  275.25      15.0       51.50
      CYL   609  270.0          0.0         0.0      619  270.0       15.0        0.0      5
      CYL   605  275.0          0.0        26.0      615  275.0       15.0       26.0      5
      CYL   601  280.5          0.0        50.0      611. 280.5       15.0       50.0      5
      CYL   581  417.5          0.0        50.0      571  417.5       15.0       50.0     -5
      CYL   589  417.5          0.0         0.0      579  417.5       15.0        0.0     -5
      CAR * 581                                      601                                  10
      CAR * 589                                      609                                  10
      ROT        0.9659258 0.258819         0.0          -0.258819 0.9659258        0.0
      ORIG       338.07403 90.58665         0.0
      CYL   719    47.5       -180.0        0.0      679    47.5        0.0        0.0     -5
      CYL   711    47.5       -180.0       50.0      671    47.5        0.0       50.0     -5
      CYL   819    19.0       -180.0        0.0      779    19.0        0.0        0.0     -5
      CYL   811    19.0       -180.0       50.0      771    19.0        0.0       50.0     -5
      ELE   200   201     4   301   311   315   305
      ELE   350               1    11    15     5
      ELE     1     6   100     1   101   111    11
      ELE    21    22   100     5   105   115    15
      ELE    23               205   309   319   215
      ELE    24               309   409   419   319
      ELE    25               409   439   449   419
      ELE    26               439   509   519   449
      ELE    27               509   609   619   519
      ELE    31    36   100    11   111   115    15
      ELE    41               215   315   319   215
      ELE    42               419   519   449   419
      ELE    44    46   100   315   415   419   319
      ELE    48               719   619   615   719
      ELE    50    53   -10   611   601   701   711
      ELE    60    63   -10   711   701   801   811
      ELE    70    73   -10   619   609   709   719
      ELE    80    83   -10   719   709   809   819
      ELE    91    92   -20   599   591   691   699
      ELE   100   102   -20   711   811   819   719
      ELE   120   123   -10   819   811   801   809
      ELE   150               611   711   719   615
      ELE   160               571   581   589   579
      ELE   300   301    .4   601   605   615   611
      SOUS    1                                                                         -200
      SOUS    2                                                                          200


      SOUS    3                                                                         -300
      SOUS    4                                                                           91
      F1  E   31-0.258819 0.9659258      0.0        36                                     1
      F1  E   41-0.258819 0.9659258      0.0
      F1  E   44-0.258819 0.9659258      0.0        46                                     1
      F1  E   42-0.258819 0.9659258      0.0        48                                     6
      F1  E  150-0.258819 0.9659258      0.0
      F1  E   92-0.258819 0.9659258      0.0
      F1  E  100-0.258819 0.9659258      0.0       102                                     2
      F1  L   27 0.0        0.0          1.0              41
      CHA          EFFORT  DE  SERRAGE
      PEU            -0.8390
                60    61    62    63
      CHA          EFFORT  DE  SERRAGE  ET  PRESSION
      PEU            -0.8390
                60    61    62    63
      PEU            -0.45
                21    22    23    24    25    26
      PEU            +2.08
               350
      FIN
```

fig. 6.b.3(b) - Input data for integral equation program

fig. 6.b.4 - the finite element representation of the flange



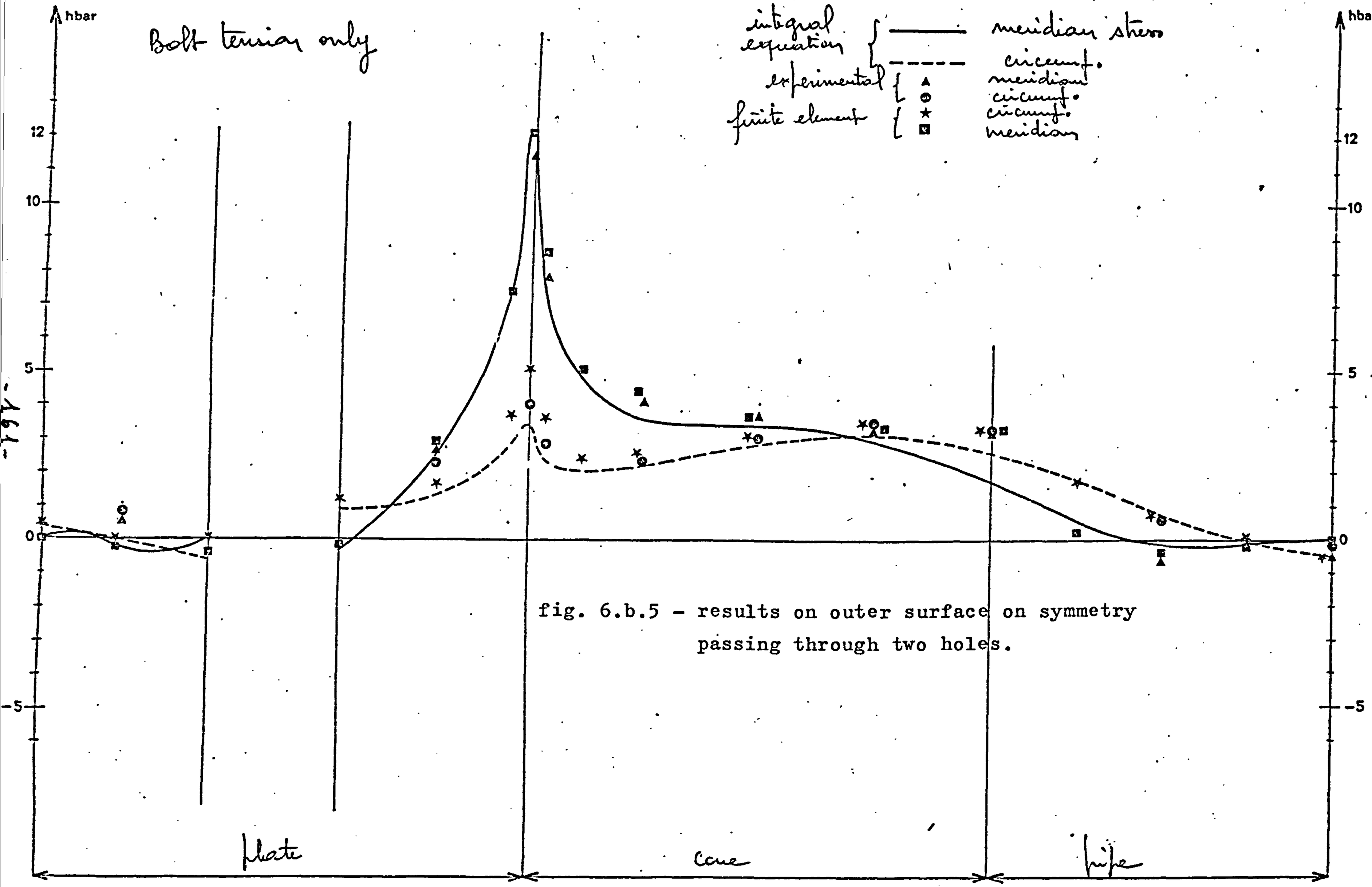


fig. 6.b.5 - results on outer surface on symmetry passing through two holes.

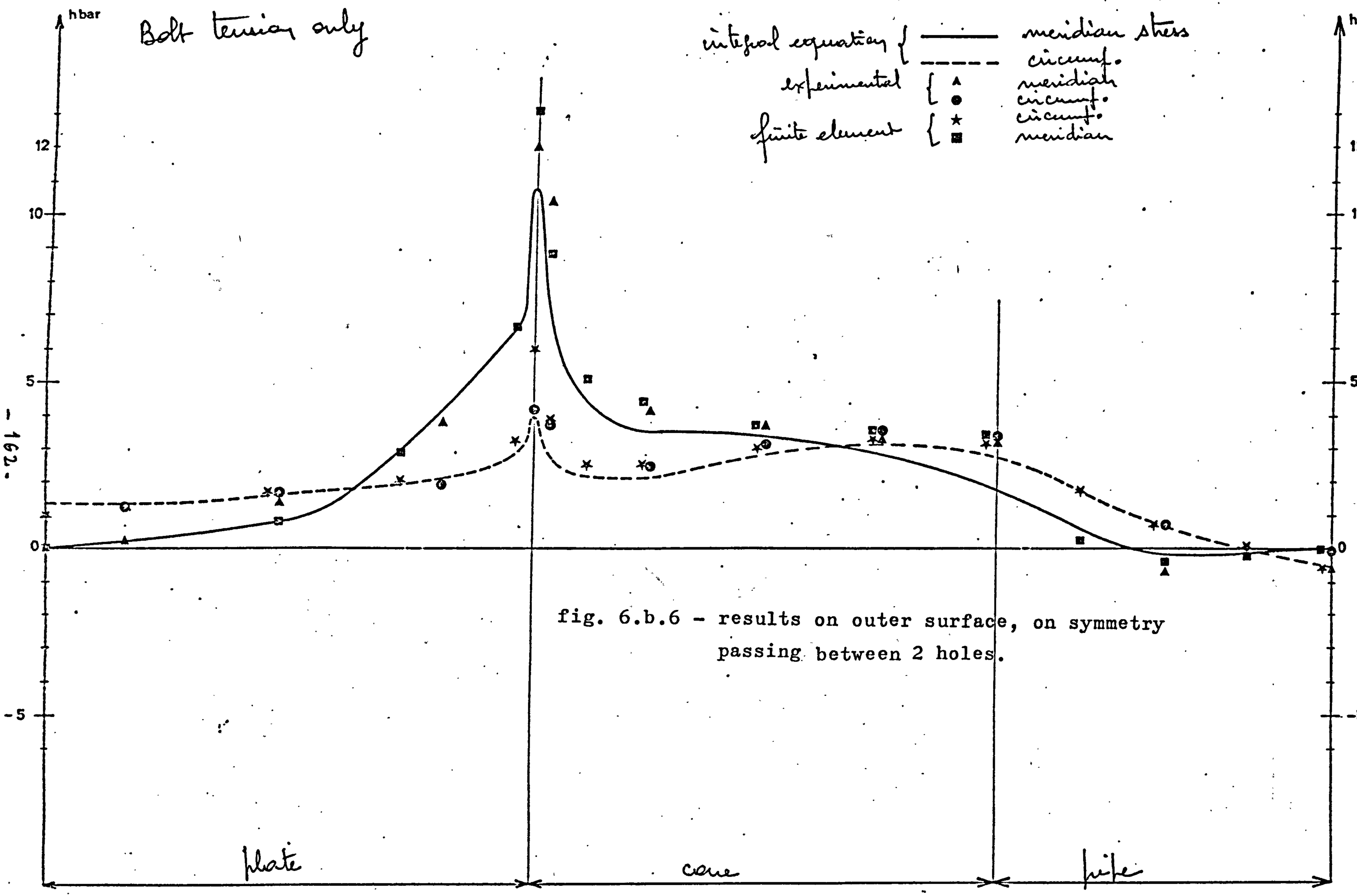


fig. 6.b.6 - results on outer surface, on symmetry passing between 2 holes.

Bolt tension only

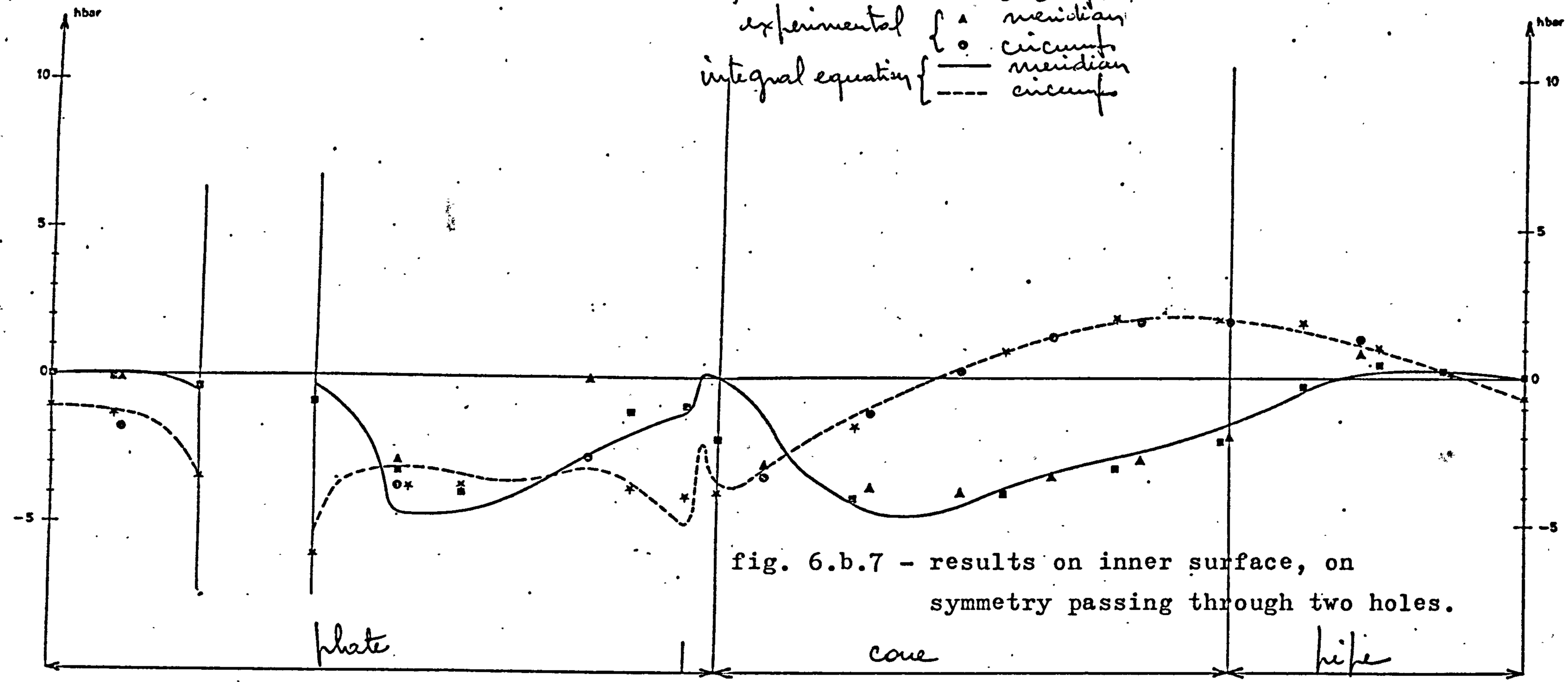


fig. 6.b.7 - results on inner surface, on symmetry passing through two holes.

Bolt tension only

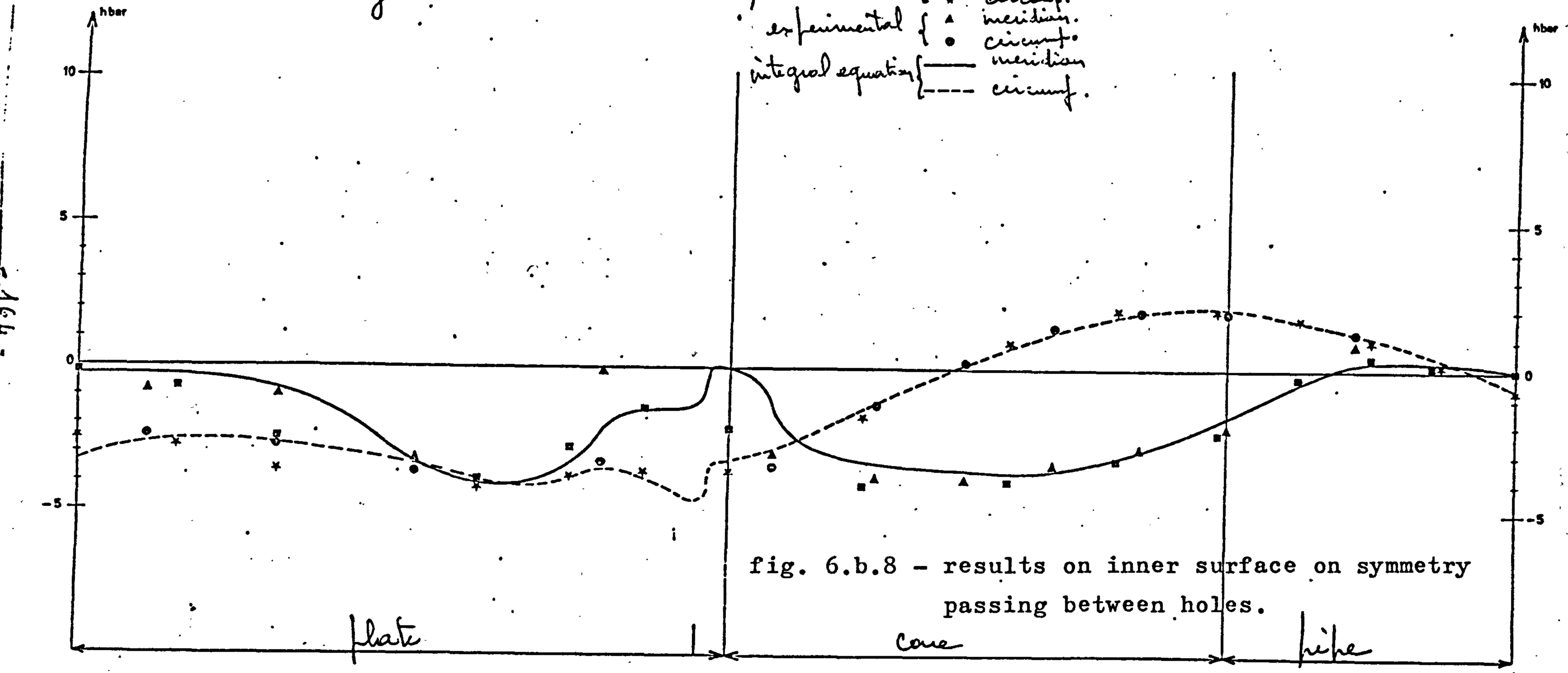


fig. 6.b.8 - results on inner surface on symmetry passing between holes.

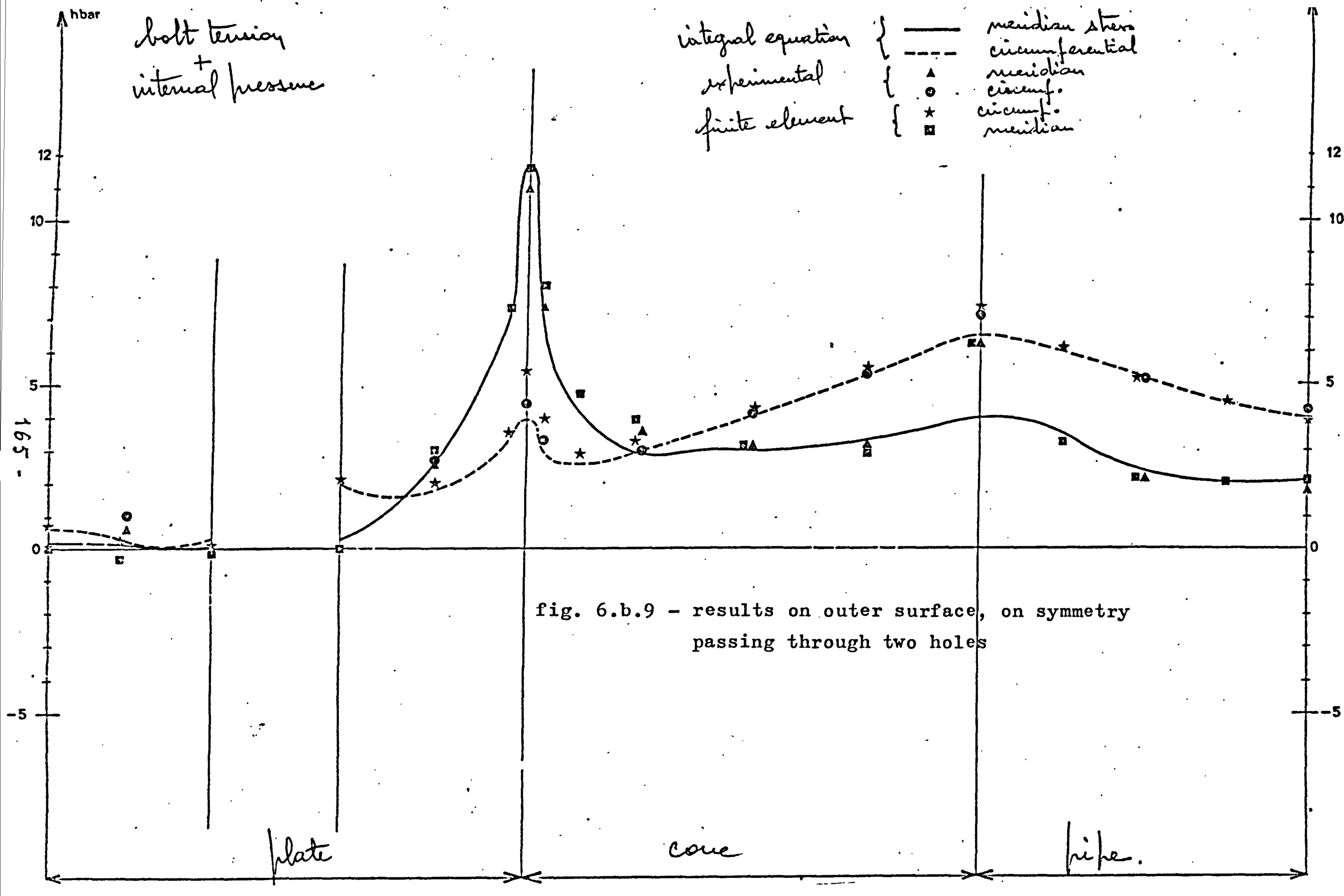


fig. 6.b.9 - results on outer surface, on symmetry passing through two holes

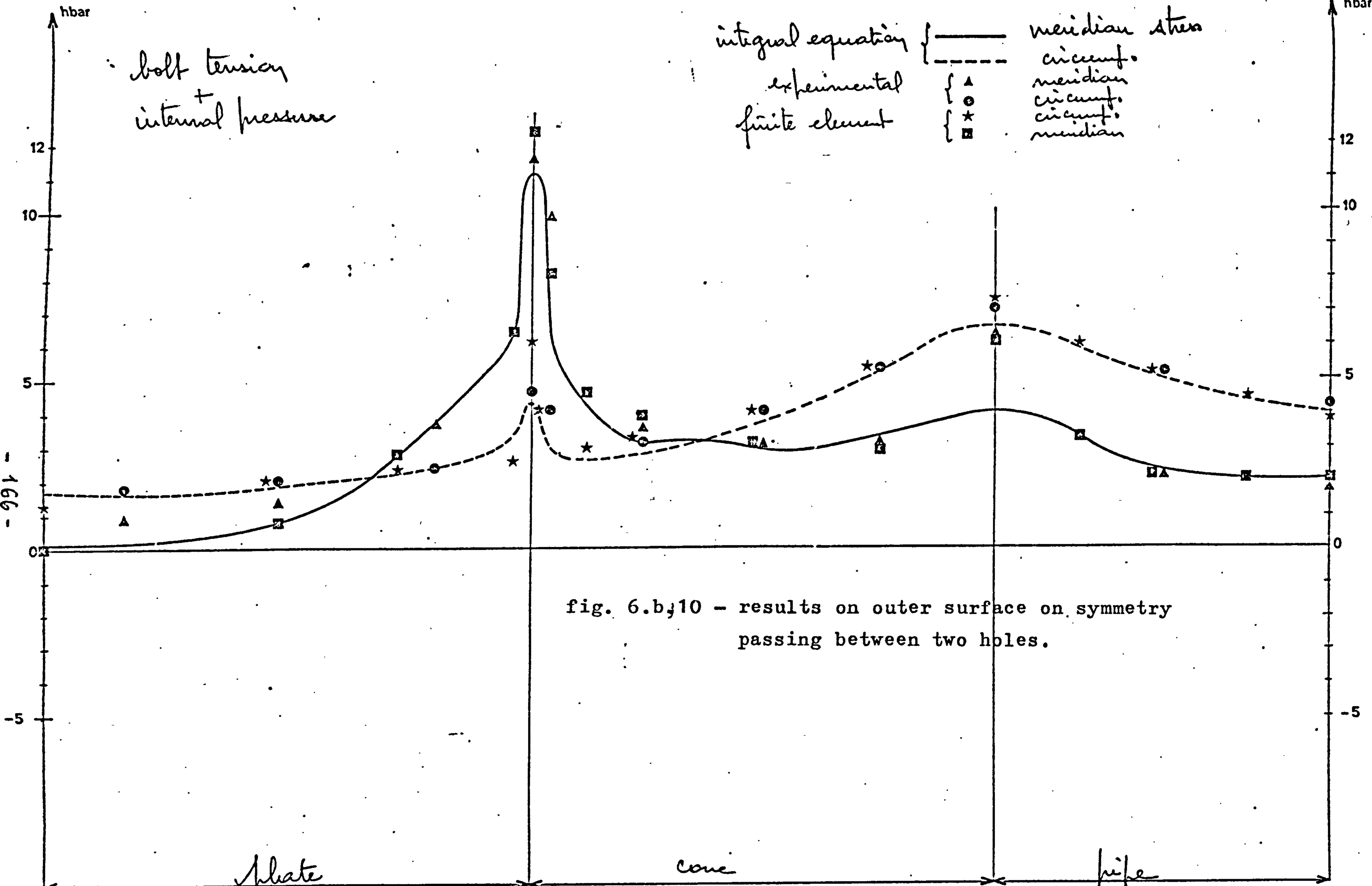


fig. 6.b,10 - results on outer surface on symmetry passing between two holes.

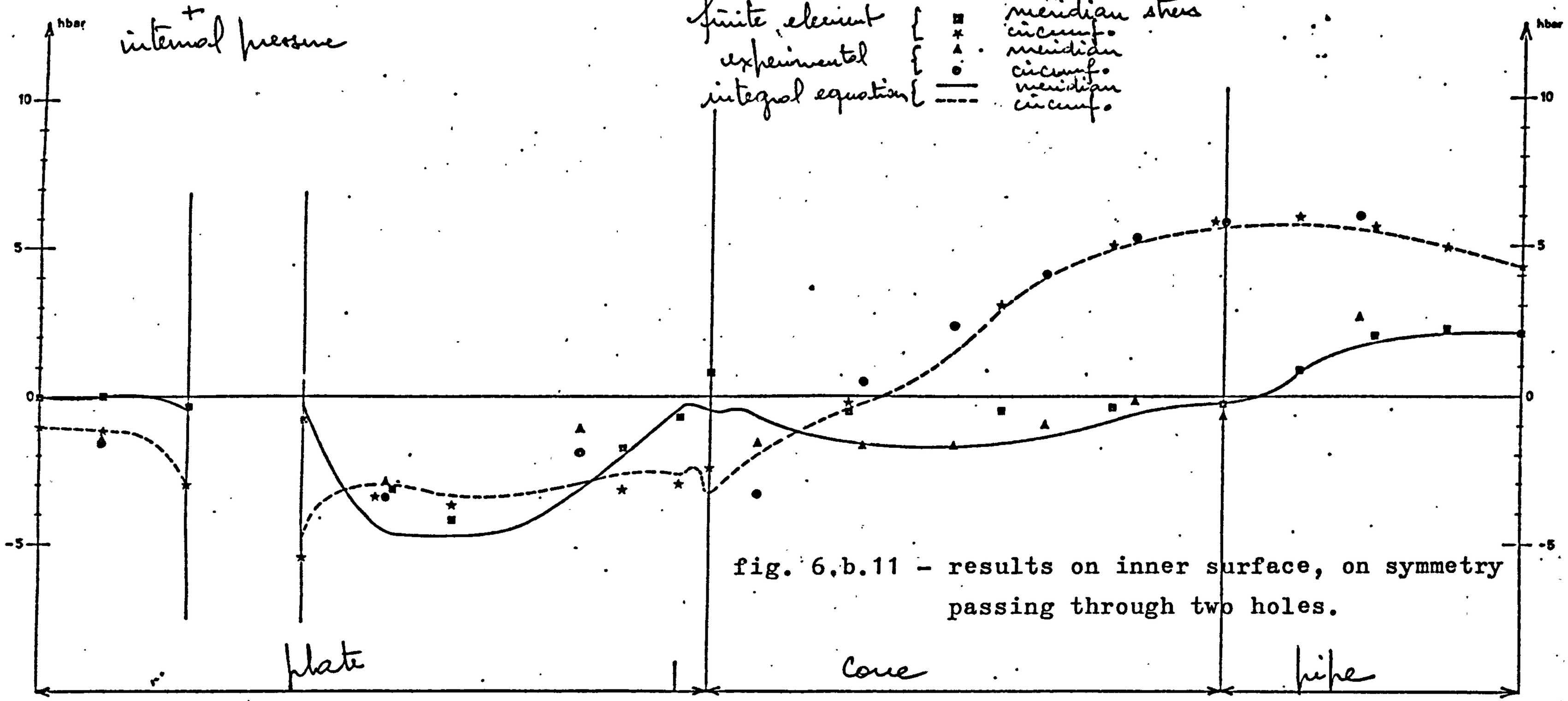


fig. 6.b.11 - results on inner surface, on symmetry passing through two holes.

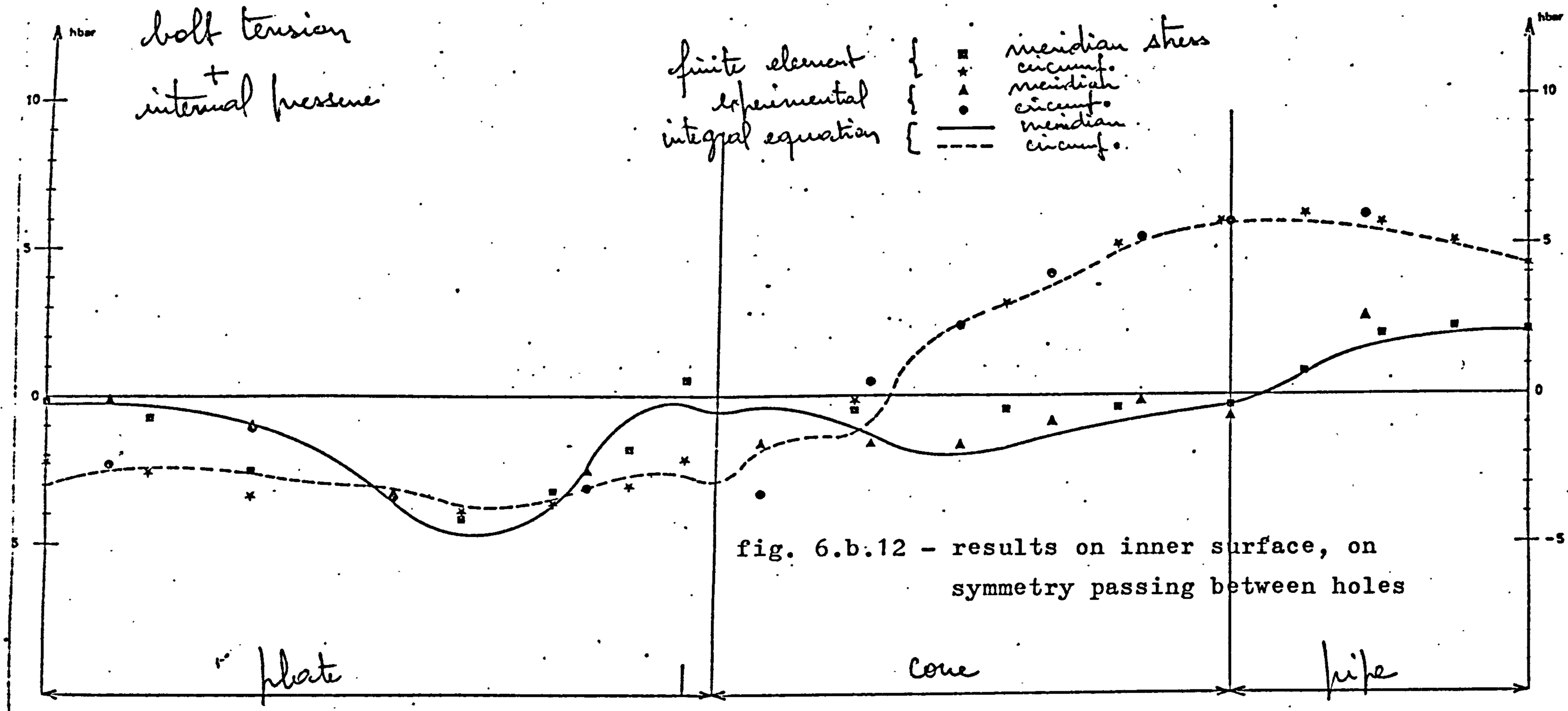


fig. 6.b.12 - results on inner surface, on symmetry passing between holes

EFFORT DE SERRAGE ET PRESSION

DEPLACEMENTS ET CONTRAINTES AUX NOEUDS SOUS-REGION 2 PAGE 6

NOEUD	DEPLACEMENTS U1 / U2 / U3	TENSEUR DES CONTRAINTES 11	12	13 / 22 / 23 / 33	CONTRAINTES PRINCIPALES, ET COSINUS DIRECTEURS DES DIRECTIONS PRINCIPALES				CRITERES VON MISES ET TRESCA	NOEUD
511	1.256E-02	2.68E+00	-6.41E-02	-2.81E+00	6.86E+00	-.55	-.15	.82	5.44E+00	511
	3.365E-03		2.02E+00	-7.59E-01	2.94E+00	-.26	.97	.00	3.11E+00	
	-4.292E-02			4.84E+00	6.50E-01	-.79	-.21	-.57		
513	8.110E-03	2.07E+00	7.07E-02	-9.56E-01	2.84E+00	.77	.21	-.61	1.78E+00	513
	2.173E-03		1.83E+00	-2.56E-01	1.61E+00	-.26	.97	.00	1.03E+00	
	-3.605E-02			1.54E+00	7.68E-01	.59	.16	.80		
514	-2.451E-02	-3.22E-01	2.36E-01	0.	1.3[illegible]E-01	0.00	0.00	1.00	2.54E+00	514
	-3.222E-03		-2.55E+00	0.	-2.85E-01	.99	.13	-0.00	1.36E+00	
	0.			1.[illegible]E-01	-2.58E+00	-.13	.99	0.00		
515	2.600E-03	1.18E+00	1.45E-01	5.23E-01	1.35E+00	.90	.24	.36	8.81E-01	515
	6.966E-04		6.24E-01	8.56E-02	5.45E-01	-.26	.97	-.00	4.93E-01	
	-2.929E-02			4.99E-01	3.68E-01	-.35	-.09	.93		
517	-1.102E-02	-3.60E-01	2.69E-01	3.69E-01	1.39E-01	.64	.17	.75	1.30E+00	517
	-2.954E-03		-1.29E+00	9.88E-02	-6.29E-01	.72	.19	-.67	7.50E-01	
	-1.425E-02			-2.02E-01	-1.36E+00	-.26	.97	-.00		
519	-2.390E-02	-6.25E-01	5.41E-01	2.84E-02	-4.18E-02	.05	.00	1.00	2.43E+00	519
	-5.404E-03		-2.52E+00	-2.38E-02	-4.82E-01	.97	.26	-.05	1.31E+00	
	0.			-4.31E-02	-2.66E+00	-.26	.97	.0.		
551	1.206E-02	7.85E+00	0.	-5.02E+00	1.11E+01	.84	0.00	-.54	9.66E+00	551
	0.		4.27E+00	-0.	4.27E+00	-0.00	1.00	-0.00	5.53E+00	
	-4.675E-02			3.20E+00	5.14E-04	.54	0.00	.84		
559	-2.568E-02	-2.41E+00	0.	0.	-1.19E-10	0.00	0.00	1.00	2.80E+00	559
	0.		-3.07E+00	0.	-2.41E+00	1.00	-0.00	-0.00	1.53E+00	
	-2.102E-02			0.	-3.07E+00	0.00	1.00	-0.00		
561	1.163E-02	8.39E+00	1.21E+00	-4.35E+00	1.16E+01	.82	.22	-.53	9.05E+00	561
	3.116E-03		4.22E+00	-1.17E+00	3.69E+00	-.26	.97	.00	4.99E+00	
	-4.698E-02			4.42E+00	1.58E+00	.52	.14	.85		

fig. 6.b.14 – Typical page of results

6(c) <u>The rolling mill cylinder</u>

The cylinder is one of a pair, supported by bearings at the ends and loaded (by the plate being rolled) along the central portion of its length. In fig. 6c1 is shown the idealisation of the problem ; the bearings are represented by simple supports on a circle defined by the intersection of the surface of the bearing and a vertical plane bisecting that surface.

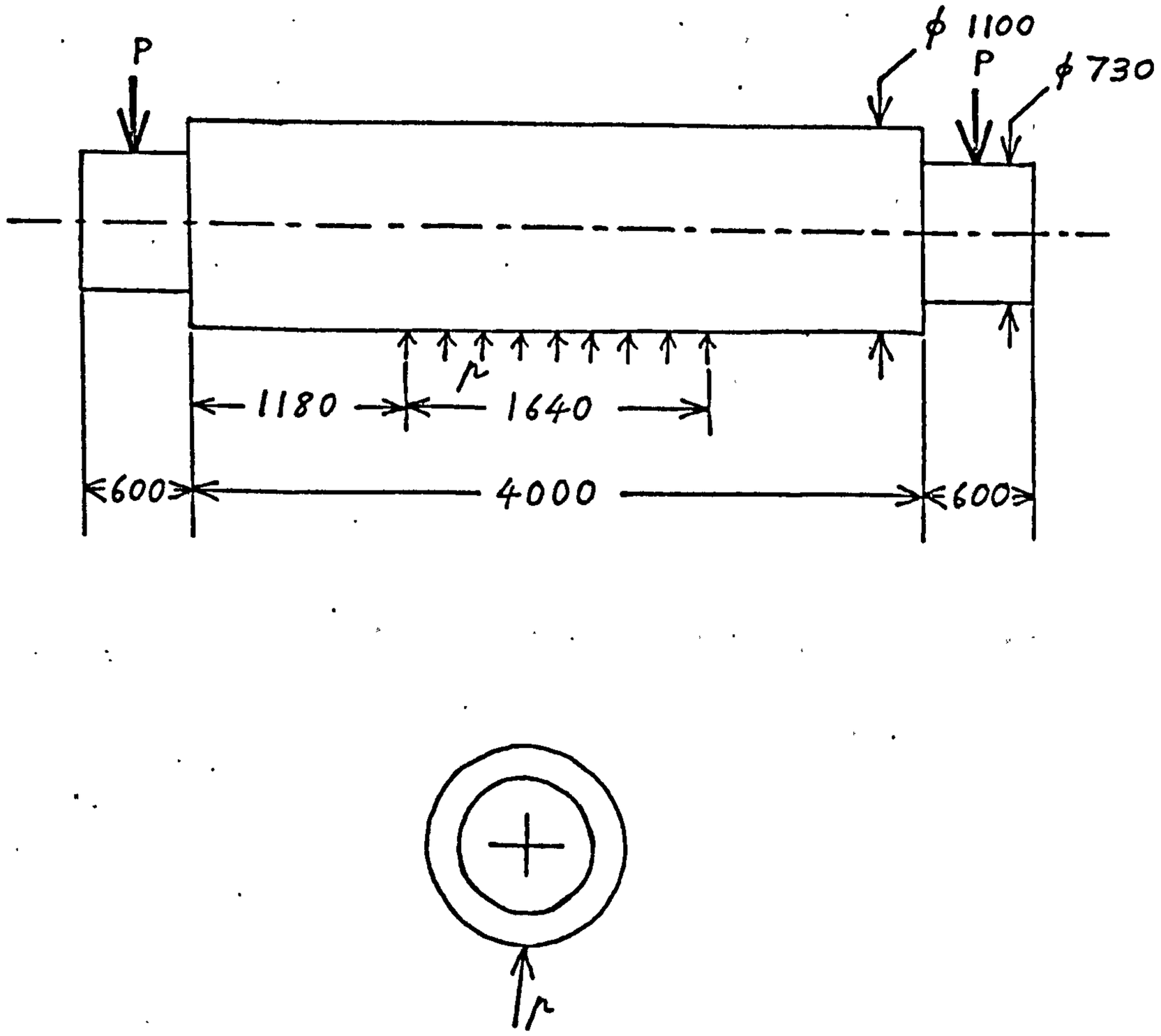


fig. 6c1 – idealisation of the rolling mill cylinder

In the analysis presented here, the effect of temperature differential is ignored. The load is in reality distributed over a finite area of the cylinder ; in the analysis it is supposed that the load is distributed along the line through which the resultant passes. As may be seen from fig. 6c1, the resultant does not pass through the axis of the cylinder, so there is torsion as well as bending.

The object of the analysis is to determine the admissibility of vertical cracks located at the centre of the span. This is of great interest to steelmakers because it would be useful to know at what stage during the propagation of such a crack it is advisable to replace a cylinder. It is not necessary to withdraw a cylinder from service immediately a crack appears ; but if the cylinder fails whilst in operation, considerable damage is done to other components such as the bearings. For the purposes of analysis, the crack tip is supposed to be circular (fig. 6c2).

Here, only one value of crack depth and crack tip radius is considered ; to obtain the required information, it is necessary to carry out analyses of several different combinations of depth and radius, and determine according to same crack propagation criterion at what stage the crack becomes critical.

The modulus of elasticity is taken to be 21 000 daN/mm2, and Poisson's ratio0.3. The intensity of the line load is nearly constant, 850 daN/mm, and the length of the perpendicular from the axis of the cylinder to the resultant is 40 mm. There is therefore very little torsion and the critical case may be taken to be that in which the crack is diametrically opposed to the load point. The crack considered here is of depth 165 mm and

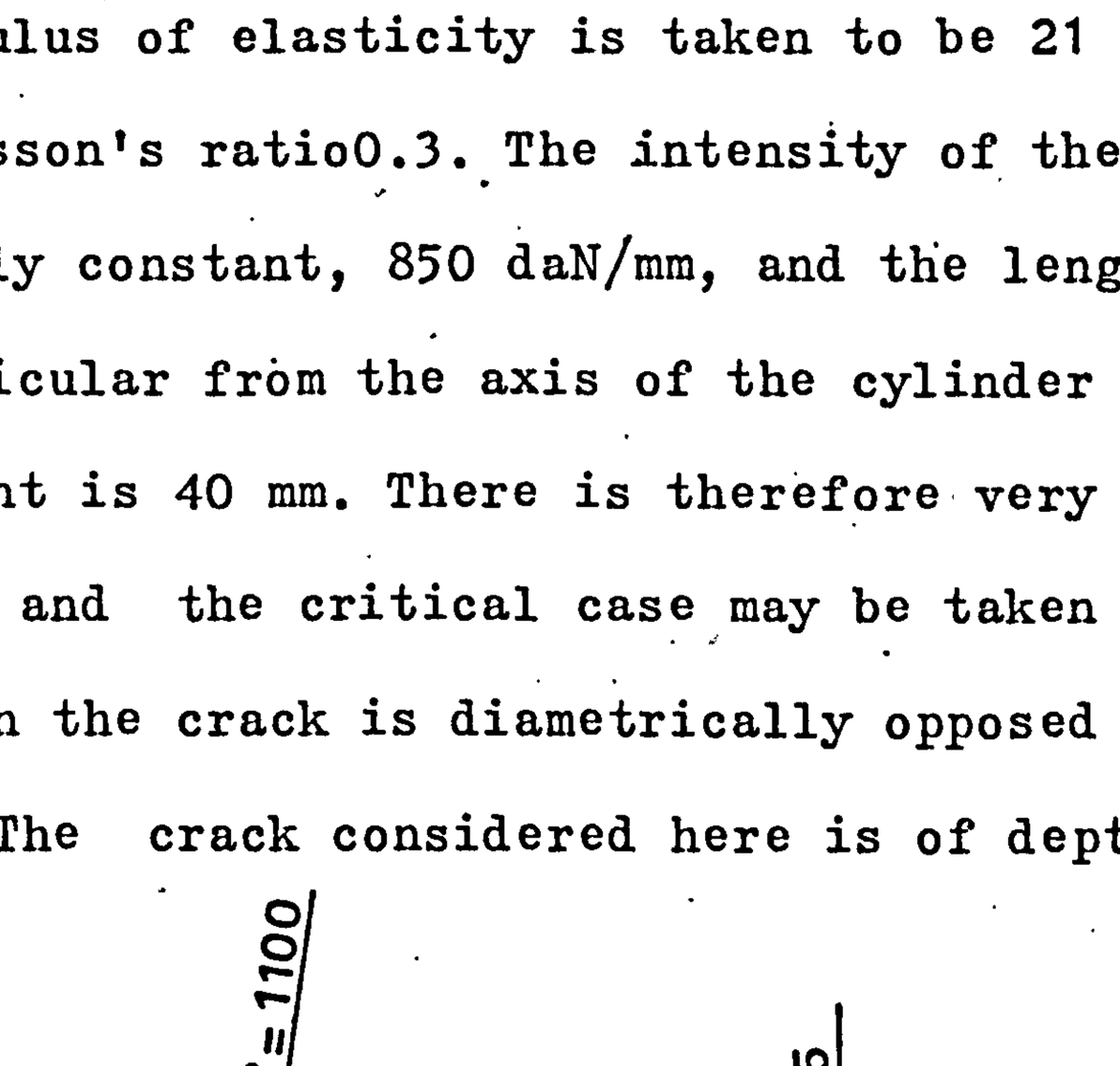


fig. 6.2 - dimensions of the crack considered.

radius 1100 mm. The cylinder is divided into 5 subregions (fig. 6.c.3), the fifth containing the crack. The discretisation of the surface of each subregion is shown in fig. 6.c.4 - 6.c.5. There are 289 nodes and 117 elements, many of the elements being degenerate. About 120 cards of data are required to define this discretisation. Quadratic functional variation is chosen, and the precision of integration is taken to be 2.0.

The execution time (CDC 7 600) for this problem is 120 seconds central processing plus 6 seconds of input/output time. The norm of residues is 1.2×10^{-10} The program calculates the vertical displacement at midspan to be 2.5 mm. This compares well with the deflection 1.9 mm obtained by the finite element method for the uncracked cylinder. The finite element analysis, incidentally, took 88 seconds (c.p. time), and so it is certain that, with a network representing the crack to an accuracy comparable with that of the integral equation discretisation, a finite element analysis would cost much more than the analysis presented here.

In fig. 6.c.7 is shown the variation of calculated direct stress on the plane containing the crack as the

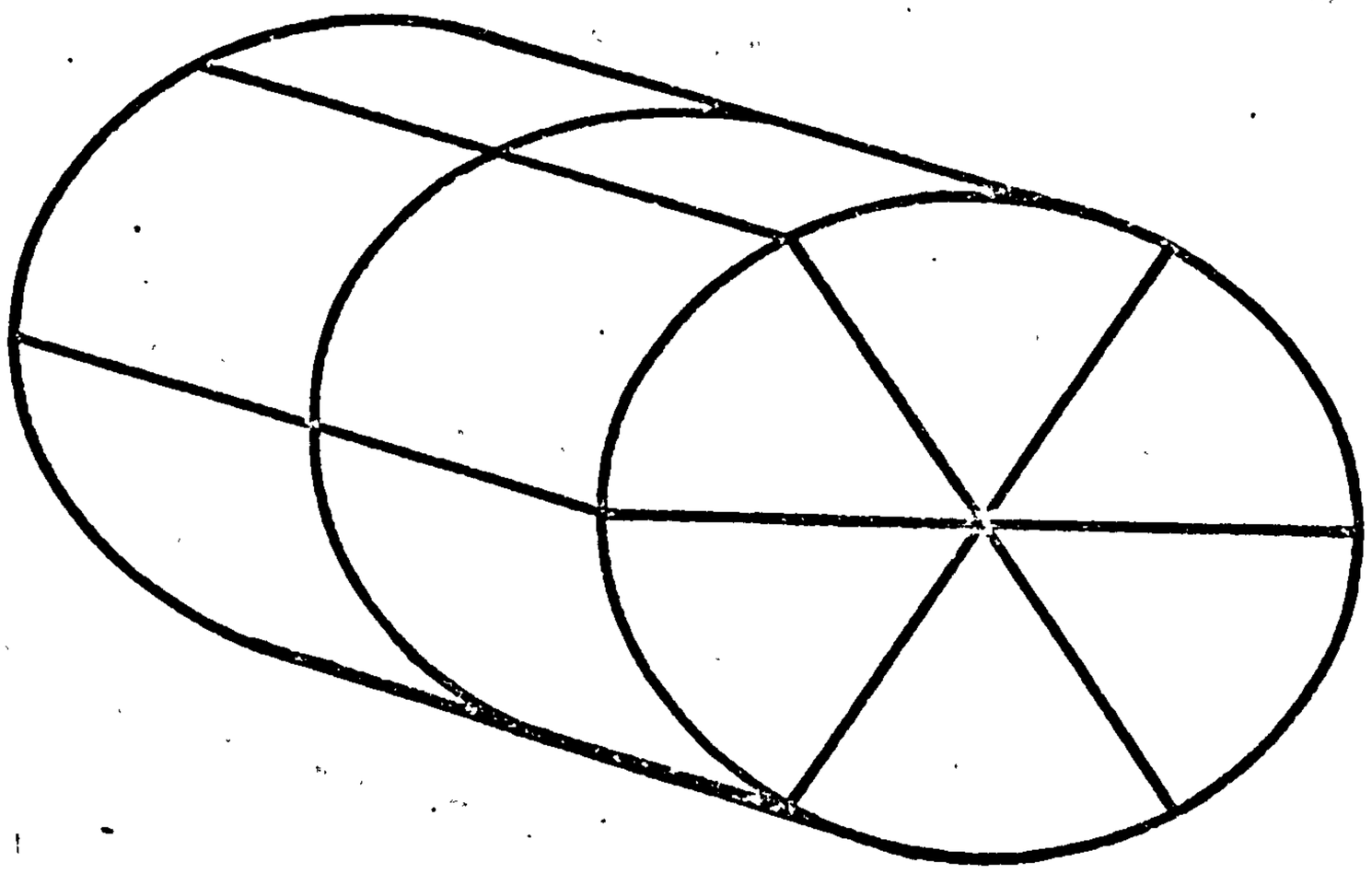
...

crack tip is approached, along five lines in that plane. From these results may be calculated the stress intensity factor, according to the linear theory of fracture mechanics:

$$K_I = \lim_{r \to 0} \sigma \sqrt{2\pi r}$$

The variation of stress intensity factor along the crack tip is shown in fig. 6.c.8.

fig. 6.c.3 - Division of the cylinder into subregions.



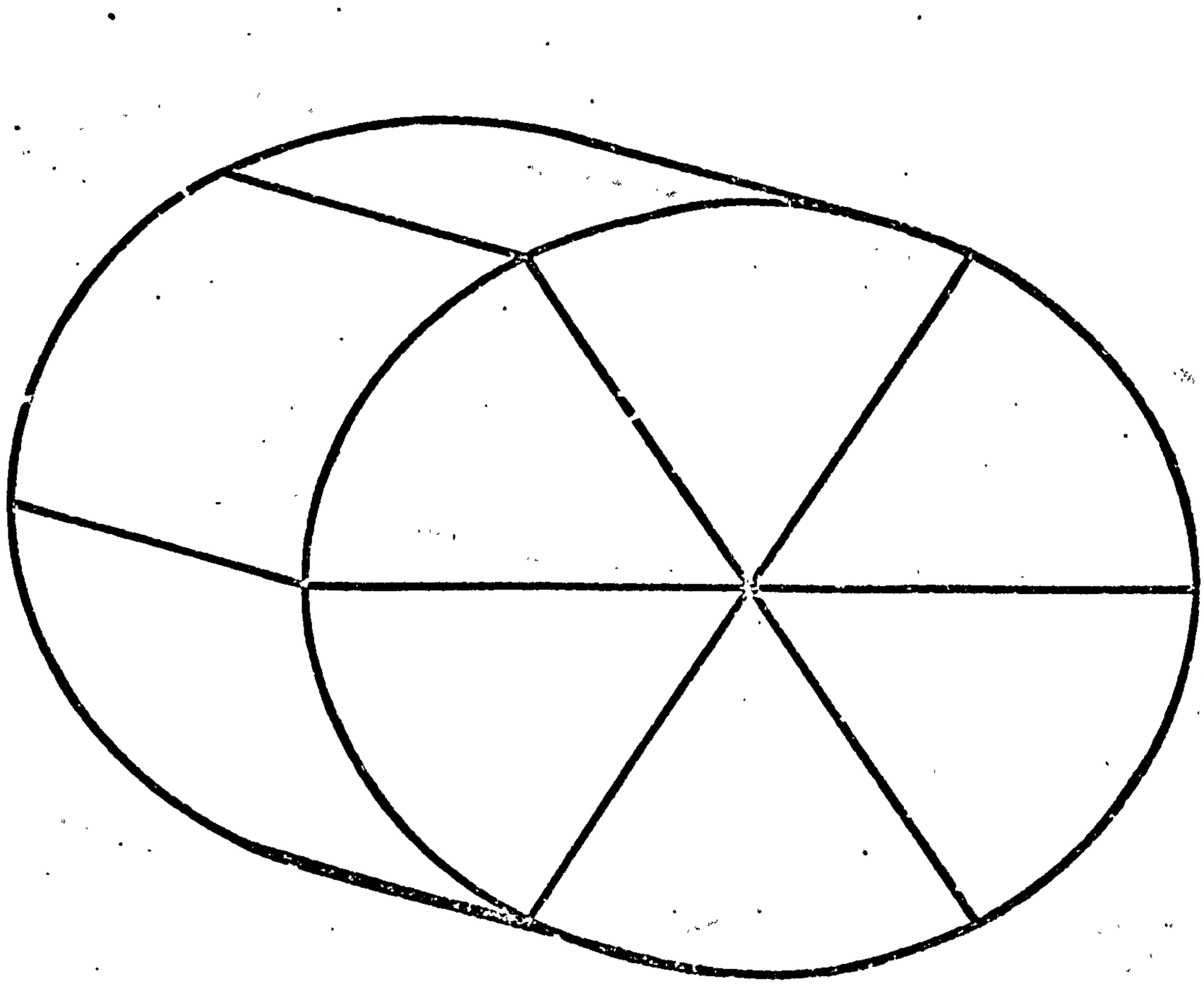


fig. 6.c.4 - first and second subregions

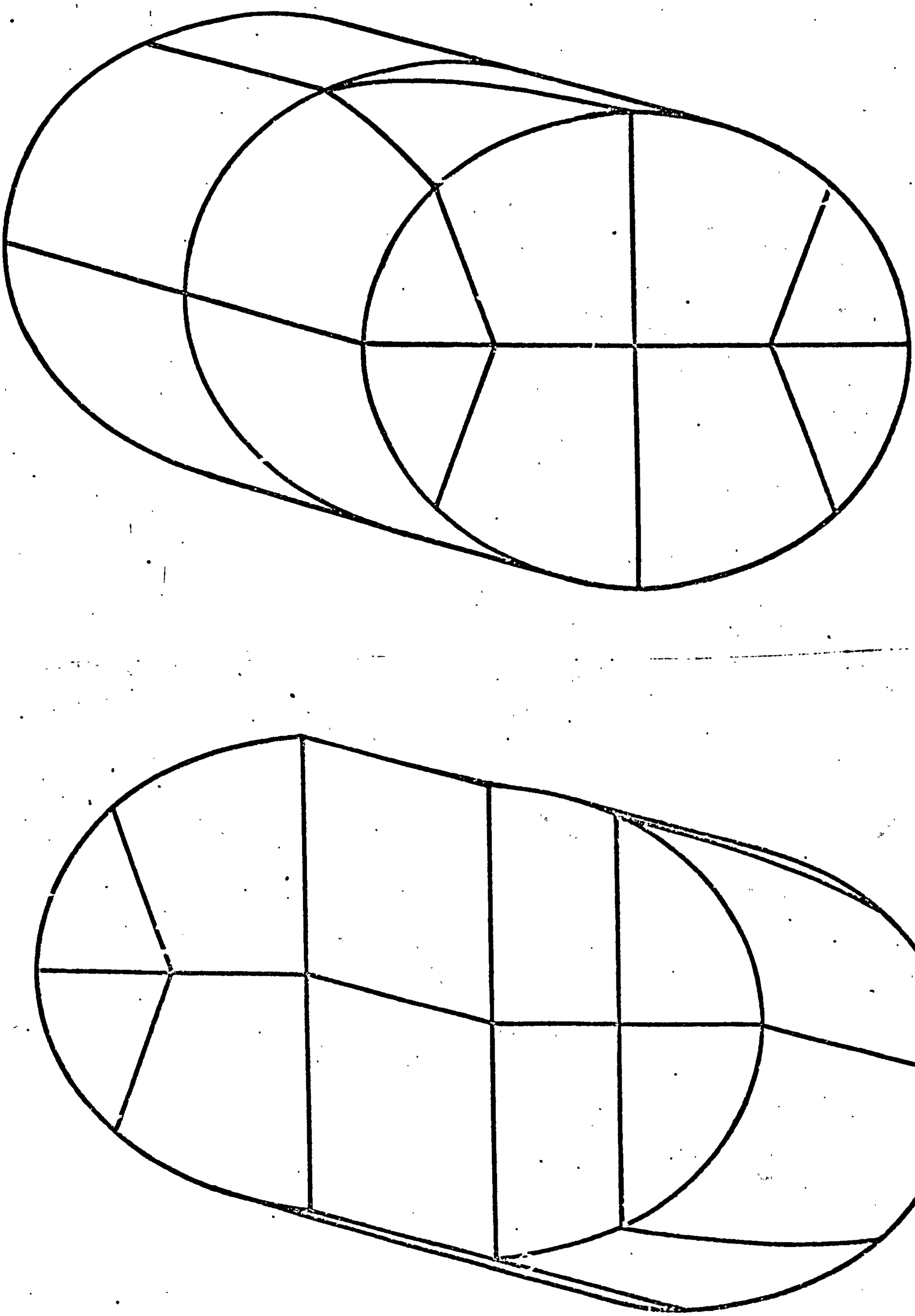


fig. 6.c.5 - third and fourth subregions

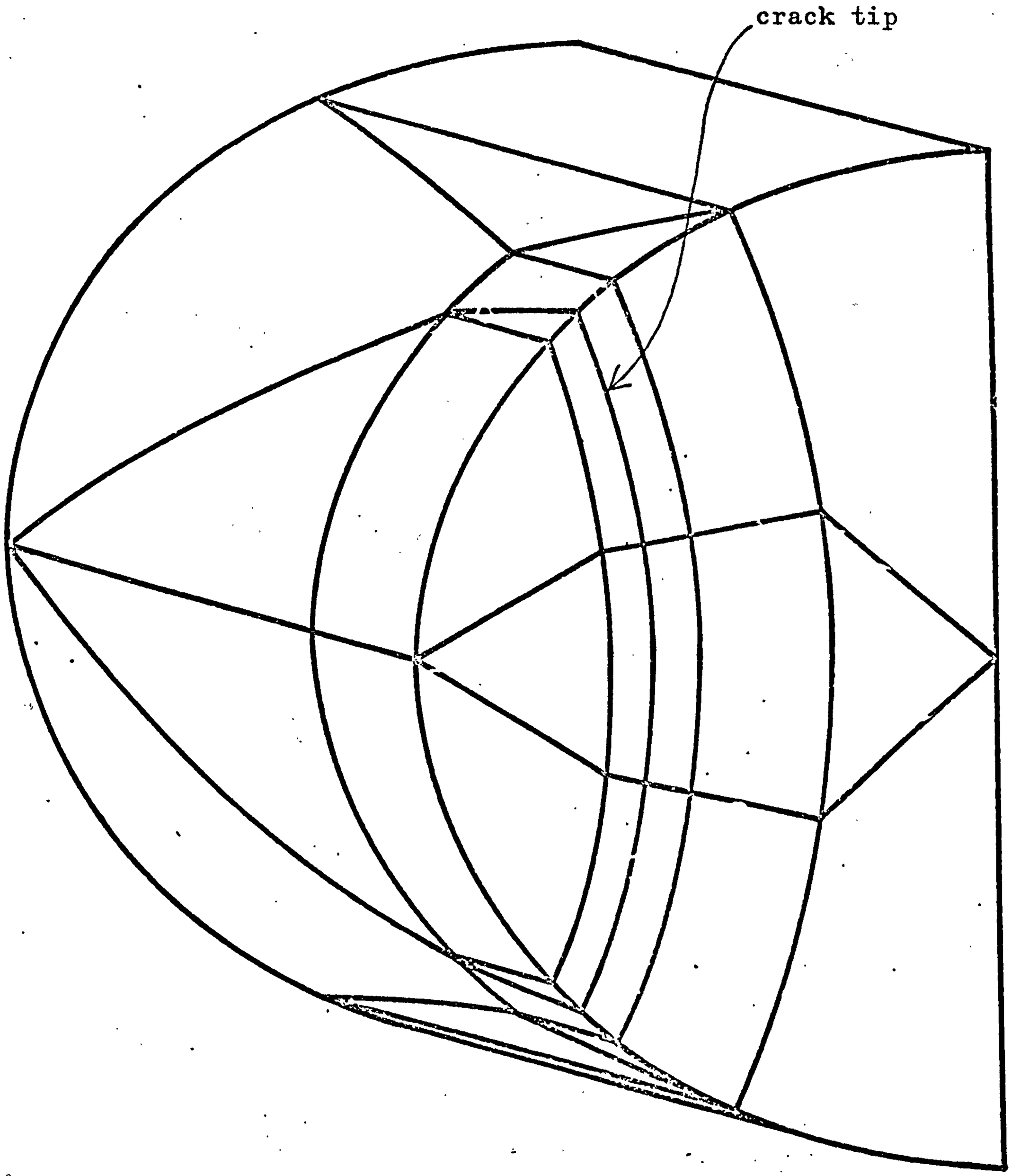


fig. 6.c.6 - fifth subregion

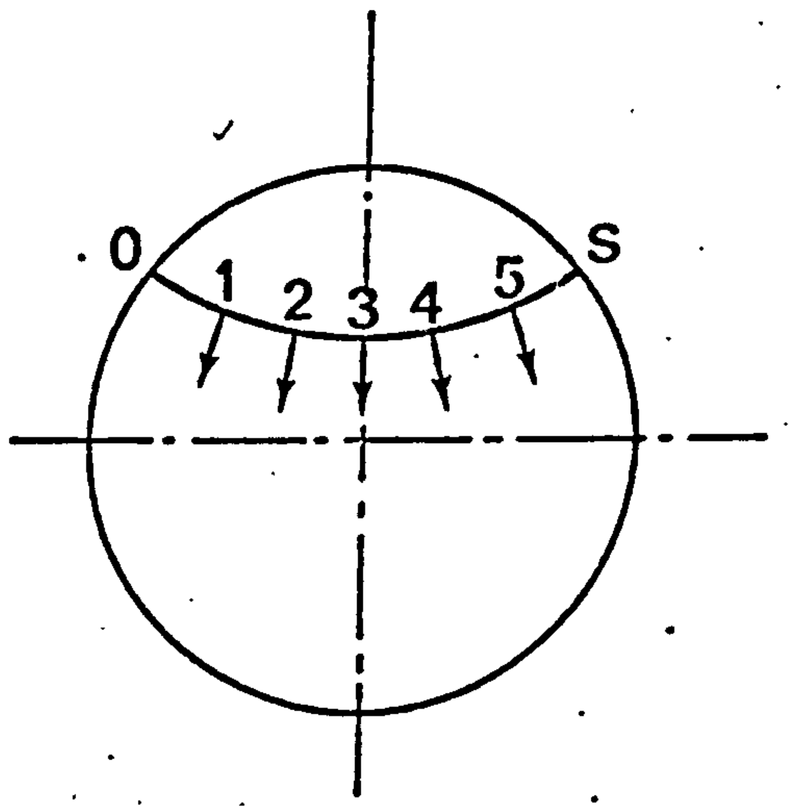


fig. 6.c;7 - variation of the calculated direct stress σ_{zz}

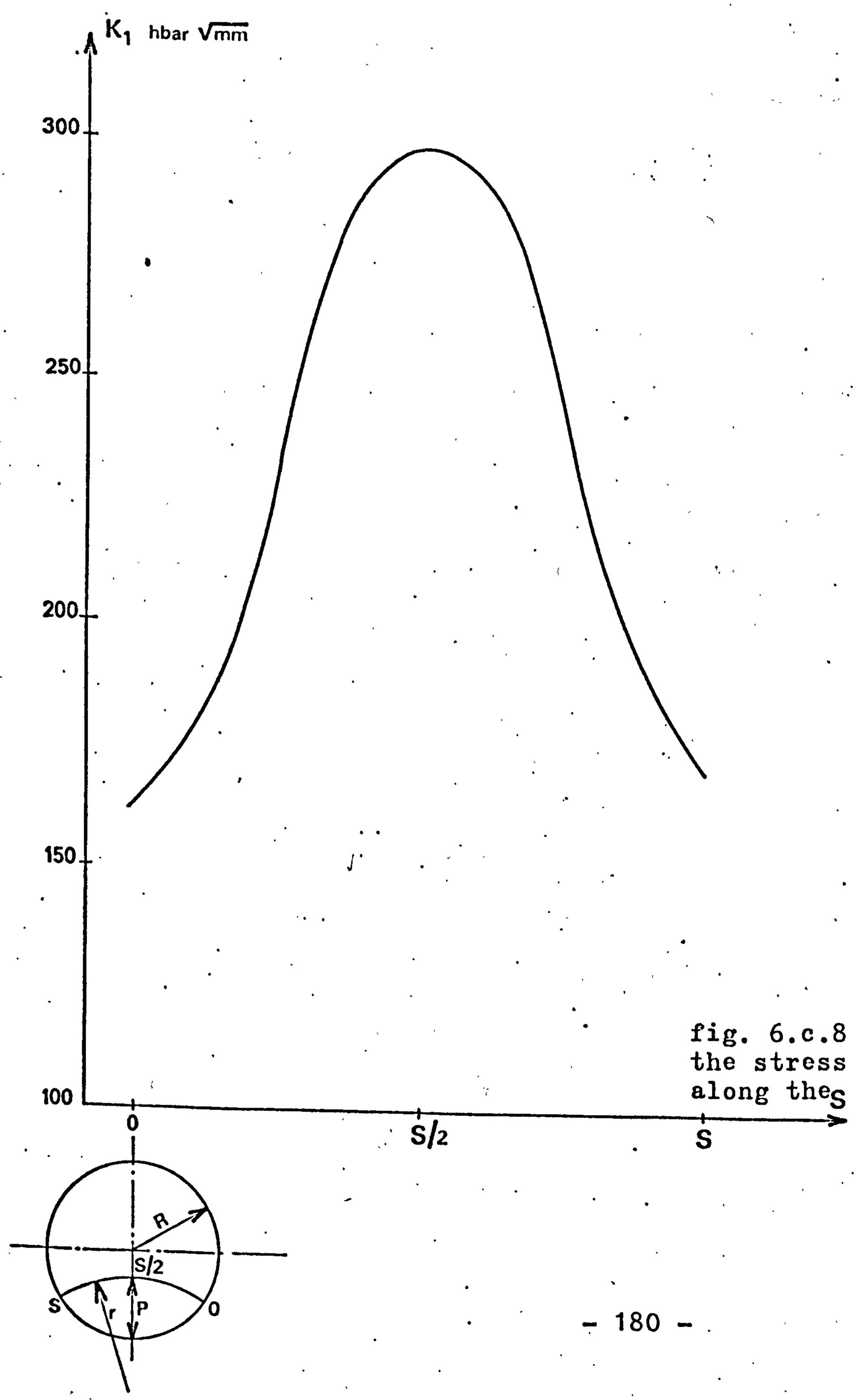


fig. 6.c.8 - variation of the stress intensity factor along the crack tip

VII - DISCUSSION

The formulation of the three-dimensional problem is in several respects an improvement upon that of the plane problem (chapter 3). The refined integration scheme works well. The analysis of the cylindrical shell (section 6 (a)) shows that good accuracy is maintained even in extreme cases in which the functional singularity is very close to the element being integrated over. This permits the representation of the surface by elements with aspect ratio (length/breadth) as high as 10, and therefore in many practical situations a reduction in the number of elements required adequately to represent a structure. In three-dimensional finite element discretisations, it is generally inadvisable to exceed an aspect ratio of 3.

The division of the elastic body into subregions results in significant economies, without appreciable deterioration of accuracy, and also allows inhomogeneous bodies to be analysed. It would be possible, by eliminating all the traction unknowns from the system for each subregion before assembly of the global system, to create a stiffness matrix for each subregion. The stiffness matrices could then be assembled as in the finite element formulation. This strategy is, however, undesirable because in the present formulation, by retaining the traction unknowns at points on interfaces, better continuity across interfaces is assured.

...

The representation in the three-dimensional formulation of constraints on displacement is a considerable improvement, and it is this above all that permits the use of coarser discretisations than could be used for plane analyses. The improvement consists in changes in the allocation of integrals of kernel-shape function products to the matrix and second members, and also in allowance for the possibility of non-orthogonal fixed and free directions at a node. This case, which does not occur in the examples of Chapter 6, would arise for example at the intersection of two non-orthogonal planes of symmetry, and results in $\lambda \neq \ell$ in equation 4.4.2.

The discontinuity of traction at points at which the displacement is given and the normal is discontinuous (see Section 3.7 and especially fig. 3.7.2) is not taken into account in the formulation of the three-dimensional problem, and it might be interesting to see what is the effect upon accuracy of allowance for this factor. In the example of Section 6(c), this situation arises at points on the interface between the fourth and fifth subregions, but the points in question being near the neutral plane for bending, the corresponding traction unknowns are nearly zero and so the effect of the approximation is not seen.

It is evident from the run statistics for the thick cylinder, flange and rolling mill cylinder that the execution time increases more slowly with problem size than for analyses by the finite element method. For nearly all practical problems, an integral

equation analysis requires less execution time than the equivalent finite element analysis, provided the results at not too many interior points are required. In practice, it is the results on the surface that are of primary interest, and results at interior points can often be obtained free, by judicious positioning of interfaces. For the rolling mill cylinder, for example, a finite element analysis in which the crack is represented to an accuracy comparable with that of the integral equation analysis would require about 5 minutes of CDC 7 600 execution time, instead of the 2 minutes for the integral equation analysis. This is for a discretisation with less than 300 nodes ; for more complicated problems, towards the limit of the integral equation program (1200 nodes), the advantage over finite elements would be greater still. For all problems, there is less data to prepare for an analysis by the integral equation method than for an analysis by the finite element method.

There is still scope for reduction of execution time. The choice of integration formula, which is a long calculation in itself, could be speeded up, and it might be worthwhile to calculate before construction of the matrix the geometric data for the 4 x 4 integration formula as well as the 3 x 3, and use 4 x 4 where the program calculates that the 3 x 4, 4 x 3 or 4 x 4 formulae are necessary.

Immediate further developments should include the plotting at least of the input data, to facilitate checking. The graphical representation of the results of three-dimensional analyses is difficult, and might be left until later. It may be desirable to introduce special crack-tip elements, to represent the stress singularity. The next stage is the analysis of thermal stresses. The internal temperature distribution itself could be calculated by the integral method, and the results taken as input data for the elastic analysis. The necessary logic for analysis of the effect of volume distributions, that is, the representation of the solution for each subregion as the sum of a particular integral and a complementary function, already exist in the program for three dimensional analysis. There would be some volume integration to do, but the order of the system of equations would be unchanged.

The improved integration scheme and division into subregions could be incorporated in the program for two-dimensional analysis without difficulty. Next to be considered should be axisymmetric problems,for which the kernels are expressed in terms of Bessel functions, and plate bending. The extension to thin shells would be more difficult, but, although the kernels for the boundary integral equation might be expensive to calculate, this development would be worthwhile because the integral method may well give good results in cases in which the accuracy of finite element analyses is poor. It should

prove possible, by using the concept of subregions, to analyse assemblies of plates, shells etc., and to include finite elements in the discretisation to represent beams, for example. The development of such a system would be slow, because of the programming difficulties involved . Just for the three-dimensional problem, it is difficult for a single researcher to cope with the numerical analysis and programming ; to develop a system, a team of researchers would be required.

The generalisation to the analysis of time-dependent phenomena presents no insurmountable problem. Kupradze (20) gives fundamental solutions for elastodynamics, which could be used to give systems of equations similar to those for elastostatics.

A. Laplace transform with respect to time gives equations that may be solved by the methods described here. The inverse transformation being unstable for large values of time, it would be necessary to divide the interval of time into increments of duration such that the inverse transform is stable at the end of the step.

Material anisotropy, inhomogenity and nonlinearity present problems of a different order, and it is not certain that the entire range of problems soluble by the finite element method can be solved economically by the integral method. It is to be hoped that at least the problem of a permeable anisotropic elastic material, a fair approximation to soil, can be considered.

VIII - CONCLUSIONS

The parametric representation of element geometry and of functions provides a good basis for the numerical representation of boundary integral equations. Variation of order of integration formula according to the rapidity of variation of the kernels, and special procedures for integration around the functional singularity, are essential to the calculation of consistently accurate results at reasonable cost. The division of the elastic body into subregions yields important economies ; indeed, without subregions the analysis of large three-dimensional problems would not be feasible. It appears, from the results of the few tests carried out so far, that for most three-dimensional problems of practical size, the execution time required for an analysis by the integral equation method, using subregions, is lower than that required for an equivalent analysis by the finite element method ; the advantage of the integral method increases with the complexity of the problem. More tests must, however, be carried out before the circumstances in which the method is advantageous can be defined with precision. Fewer cards of input data are required for an integral equation analysis than for the equivalent finite element analysis. The numerical representation and programming are difficult, and it is not certain that the entire range of problems soluble by the finite element method can be solved economically by the integral method. The optimal

...

algorithm is probably one in which both formulations are used, according to their suitability.

REFERENCES

ARSAC

(1) - Transformation de FOURIER et théorie des distributions - Paris, DUNOD, 1961.

BOURBAKI

(2) - Espaces vectoriels topologiques - Fascicules XV et XVIII, 2ème édition - Paris - HERMANN, 1966.

(3) - Intégration - Fascicule XIII, 2ème édition corrigée. Paris, HERMANN, 1965.

(4) - Topologie générale : espaces fonctionnels - Fascicule X, Paris, HERMANN, 1961.

COURANT-HILBERT

(5) - Methods of Mathematical Physics - 2 volumes - New York, Interscience Publishing 8th Edition 1970.

CRUSE T.A.

(6) - The Direct Potential Method in Three Dimensional Elastostatics. Depart. of Mech. Engineering - Carnegie Institute of Technology - Pittsburgh.

CRUSE T.A. and RIZZO F.

(7) - A Direct Formulation and Numerical Solution of the General Transient Elastodynamic Problem I, J. Math. Anal. Appl. 22, April 1968.

CRUSE T.A. and RIZZO F.

(8) - A Direct Formulation and Numerical Solution of the General Transient Elastodynamic Problem II, J. Math. Anal. Appl. 22, May 1968.

CRUSE T.A.

(9) - Numerical Solutions in Three Dimensional Elastostatics. Int. J. Sol. Struct. , Vol. 5 - 1969.

GAGNON

(10) - Distribution Theory of Vector Fields - Amer. J. of Physics Vol. 38, Nº 7, 1970.

GARNIR

(11) - Sur la formulation des problèmes aux limites dans la théorie des distributions - Bull. Société Royale des Sciences de Liège, 1951.

(12) - Sur les distributions résolvantes des opérateurs de la physique mathématique. Bull. Société Royale des Sciences de Liège, 1951.

...

GEL'FAND et SHILOV

(13) - Fourier Transform of Rapidly Increasing Functions, and Questions of Uniqueness of the Solution of Cauchy's Problem, 1953.

(14) - Les distributions vol. I, II et III, Paris, DUNOD 1962, 1964, 1965.

GERMAIN

(15) - Théorie des milieux continus. Paris. DUNOD.

HADAMARD

(16) - Le problème de Cauchy et les équations aux dérivées partielles linéaires hyperboliques - Paris, HERMANN, 1932.

HORMANDER

(17) - On the Theory of General Partial Differential Operators, Acta. Mat. Vol. 94 - (1955).

(18) - On the Division of Distributions by Polynomials - 1958.

(19) - Linear Partial Differential Operators - Springer - Berlin, 1963.

KUPRADZE

(20) - Dynamical Problems in Elasticity, Progress in Solid Mechanics, Vol. III, edited by I.N. Sneddon and R. Hill. Wiley (1963).

...

LAVOINE

(21) - Calcul symbolique des distributions et pseudo-fonctions

CNRS - Paris - 1959.

LAX

(22) - On the Cauchy Problem for Hyperbolic Equations, and the Differentiability of Solutions of Elliptic Equations. Communications of Pure and Applied Math. (1955).

LIGHTHILL

(23) - An Introduction to Fourier Analysis and Generalised Functions. Cam. Univer. Press, New York, 1958.

LIONS

(24) - Problèmes aux limites en théorie des distributions - Acta. Mat. 94 (1955).

LOVE

(25) - A Treatise on the Mathematical Theory of Elasticity - Dover (1944).

MALGRANGE

(26) - Equations aux dérivées partielles à coefficients constants. Equations avec second membre - CN AC.SC Paris 1954.

MIKLHLIN

(27) - Multidimensional Singular Integrals and Integral Equations - Pergamon Press (1965).

MUSKHELISHVILI

(28) - Some Basic Problems of the Mathematical Theory of Elasticity, 4 th Edition - Noordhoff (1953).

RIZZO

(29) - An Integral Equation Approach to Boundary Value Problems of Classical Elastostatics. Q. Appl. Math. 25,83 (1967).

(30) - A Formulation and Solution Procedure for the General Non-homogenous Elastic Inclusion Problem. Int. J. Solids Structures Vol 4, 1968.

SCHWARTZ

(31) - Méthodes mathématiques pour les sciences physiques.

(32) - Théorie des distributions - Paris - HERMANN - 1966.

(33) - Distributions à valeurs vectorielles - Annales de l'Institut FOURIER - 1959.

(34) - Equations aux dérivées partielles - Séminaires, Institut H. POINCARE, Paris, 1955.

I. FREDHOLM

(35) - Solution d'un problème fondamental de la théorie de

l'élasticité .Arkiv fur Mathematik Astronomie und Fysik, 2, 1-8 (1905).

GUIRAUD

(36) - Equations à intégrales principales, étude suivie d'une application 3ème série, 51 (1934) Annales de l'Ecole Normale Supérieure.

(37) - Théorie du bruit balistique provoqué en atmosphère non homogène par le vol d'un avion supersonique . Note technique ONERA Nº 79 (1964).

SCHWARTZ

(38) - Analyse : topologie générale et analyse fonctionnelle - HERMANN Paris 1970.

BOGGIO

(39) - Sull'Integrazione di Alcune Equizioni Lineari Alle Derivate Parziali, Anu. Mat. Ser. III, 8 (1903).

WEYL

(40) - The Method of Orthogonal Projection in Potential Theory · Duke Journ. Vol. 7 (1940), p. 411. 444.

LICHTENSTEIN

(41) - Neur Entwicklung der Theorie partieller Differential-gleichungen zweiter Ordnung vom elliptischen Typus. Encyklopädie der Mathematischen Wissenschaften, DB.II3, Haft Leipzig, 1924.

FICHERA

(42) - Cours d'analyse mathématique . Paris 1956.

CRUSE T.A. and VANBUREN W.

(43) - Three Dimensional Elastic Stress Analysis of a Fracture Specimen With an Edge Crack. Int. J. Fract. Mech. 7.1.15 (1971).

MARINESCU

(44) - Sur les solutions des équations de l'élasticité asymetrique. Int. J. Eng. Sci. 1968.

IESAN

(45) - Existence Theorems in Micropolar Elastostatics. Int. J. Eng. Sci. Vol. 9 - 1971.

SWEDLOW J.L. and CRUSE T.A.

(46) - Formulation of Boundary Integral Equations for Three Dimensional Elasto-plastic Flow.
Int. Journal Solids Structures, 1971, Vol. 7

CRUSE T.A.

(47) - Application of the Boundary Integral Equation Method to Three Dimensional Stress Analysis.
Computers and Structures, Vol. 3 - 1973.

CRUSE T.A.

(48) - Application of the Boundary Integral Equation Method to Three Dimensional Stress Analysis.
National Symposium of Computerized Structural Analysis and Design, George Washington University, March 27-29.1972.

CRUSE T.A.

(49) - Numerical Evaluation of Elastic Stress Intensity Factors by the Boundary Integral Equation Method.
The Winter Annual Meeting of the ASME, Nov. 1972.

CRUSE T.A. and BESUNER P.M.

(50) - Residual Life Prediction for Surface Cracks in Complex Structural Details.
AIAA Journal of Aircraft, August 1974.

CRUSE T.A.

(51) - Application of the Boundary Integral Equation Solution Method in Solid Mechanics.
Variational Methods in Engineering - Dept. Civil Eng. Univ. Southampton Sept. 1972.

RICCARDELLA P.C.

(52) - An Improved Implementation of the Boundary Integral Technique for Two Dimensional Elasticity Problems.
Carnegie Institute of Technology - Pittsburgh Sept. 1972.

CRUSE T.A.

(53) - An Improved Boundary Integral Equation Method for Three Dimensional Elastic Stress Analysis.
Computers and Structures, Vol. 4, 1974.

SYMM G.T.

(54) - Integral Equation Methods in Potential Theory
Proc. Roy. Soc. Ser. A. 275 - 1963.

JAWSON M.A. and PONTER A.R.

(55) - An Integral Equation Solution of the Torsion Problem
Proc. Roy. Soc. Ser. A. 273 - 1963.

BARONE M.R. and ROBINSON A.R.

(56) - Determination of Elastic Stresses at Notches and Corners by Integral Equations.
Int. J. Solids Structures, Vol. 8, 1972.

MENDELSON A.

(57) - Boundary Integral Methods in Elasticity and Plasticity
NASA TN D-7418 Nov. 1973.

WATSON J. O.

(58) - The Analysis of Thick Shells with Holes, by Integral Representation of Displacement.
Ph. D Thesis, Univ. Southampton, 1972.

MASSONNET C.E.

(59) - Numerical use of Integral Procedures.
Stress Analysis ed. By O.C. Zienkiewicz - Mc Graw Hill

STROUD A.H. and SECREST D.

(60) - Gaussian Quadrature Formulas
Prentice - Hall N.Y. 1966

MUSKHELISHVILI N.I.

(61) - Singular Integral Equations
Noordhoff, Groningen, 1953.

LACHAT J.C. and BREBBIA C.

(62) - Résolution du problème élastostatique par la méthode des équations intégrales.
University of Southampton, June 1972.

DUBOIS M. and LACHAT J.C.

(63) - The Integral Formulation of Boundary Value Problems.
Variational Methods in Engineering, Univ. of Southampton, 1972.

BOISSENOT J.M., LACHAT J.C. and WATSON J.O.

(64) - Etude par équations intégrales d'une éprouvette C.T.15
Revue de Physique Appliquée, T.9, Juillet 1974.

ZIENKIEWICZ O.C.

(65) - The Finite Element Method in Engineering Science
Mc Graw Hill - London 1971.

TOTTENHAM H. and BREBBIA C.

(66) - Finite Element Techniques in Structural Mechanics.
Proceedings of a seminar at the University of Southampton
April 1970.

ERGATOUDIS J.G.

(67) - Isoparametric Finite Elements in Two and Three Dimensional Stress Analysis.
Ph. D Thesis University of Swansea 1968.

KINOSHITA N. and MURA T.

(68) - On the Boundary Value Problem of Elasticity
Res. Rep. Fac. Eng. Meiji Univ. Nº 8, 1956.

MIKHLIN S.G.

(69) - Integral Equations
Pergamon 1957.

RICE J.R.

(70) - A Path Independent Integral and the Approximate Analysis of Strain Concentration by Notches and Cracks. J. of Applied Mechanics - June 1968.

A.S.T.M. Standards

(71) - Standard Method of Test for Plane Strain Fracture Toughness of Metallic Materials. May 1972. part. 31 ; designation : E 399-72.

BOUTILLON J.P. - LACHAT J.C. and PICOLLIER G.

(72) - Jonctions par brides : détermination des contraintes et des déformations - Mémoires Techniques du CETIM Nº 6 octobre 1970.

APPENDIX 1

In this appendix are presented the Fortran subroutine IDSØL, the equation solver of the program for two-dimensional analysis, the subroutines ITFØR and ITSØL, which construct and reduce the system of equations for three-dimensional analysis, and subroutines called by ITFØR and ITSØL. The order of presentation is

IDSØL
ITFØR
ITSØL
ITSSB
ITMKE
ITMAT
ITSHA
ITSHB
ITUNS
ITNLG
ITNCØ
ITBUF
ITNØR
ITVPR
ITTRA
ITIPR
ITZER
ITSET
ITTRS
ITSEI
ITPIK
ITGEØ
ITUNV
ITMIN
ITNCH
ITDCØ
ITSSA
ITSWØ
ITDRW

```
      SUBROUTINE IDSOL
C     ALL LOAD CASES ARE TREATED SIMULTANEOUSLY
C     CHOICE OF UNITS FOR ANALYSIS GIVES WELL-CONDITIONED MATRIX
      DIMENSION FI(900),FO(900)
      COMMON/FIXE/AYM(18),NLIBD,NEBC,NEBD,NCC,
     1 LUS,LUM,LBLM,NBLM,JMAT,MMAT
      INTEGER AYM
      COMMON/CHANGE/BT(1500),A(9000),AO(900)
      EQUIVALENCE (ICUE,AO(900))
      DATA NBLO/10/
      DATA LUO/6/.
      LBUF=NBLM*LBLM
C     INCREMENTS AND SLOW BLOCK FILE PARAMETERS
      JA=NLIBD
      JALO=JA-1
      JB=JA+1
      IAI=1
      IAO=1
C     EQNS READ TO DATE AND SLOW OUTPUT BUFFER CONTROL
      LA=0
      LB=1
    1 CONTINUE
C     READ SLOW BLOCK = LD SOL BLOCKS
      LC=MINO(NBLO*MMAT,JA-LA)
      LCLO=LC-1
      LD=LCLO/MMAT+1
      JC=1
      JD=LBLM
      DO 2 IA=1,LD
      JE=JC
      JF=JD
      DO 3 IB=1,NBLM
      IR=(JF-JE)+1
      CALL READMS(LUM,A(JE),IR,IAI)
      JE=JE+LBLM
      JF=JF+LBLM
      IAI=IAI+1
    3 CONTINUE
      JC=JC+JMAT
      JD=JD+JMAT
    2 CONTINUE
C     LONGEST EQUATION SLOW BLOCK, LEADING DIAGONAL INCREMENT
      LE=JA-LA
      LED=JB-LA
      IF (LCLO) 4,4,5
C     REDUCTION OF SLOW BLOCK
    5 CONTINUE
      JC=1
      JIA=LA+1
      DO 6 IA=1,LCLO
C     ELIMINATE THE UNKNOWN JIA
      ELT=A(JC)
      IF ( ABS(ELT)-1.0E-6) 999,999,8
  999 WRITE (LUO,100) JIA
  100 FORMAT (   71H UN COEFFICIENT DIAGONAL DE LA MATRICE TRIANGULAIRE
     1EST ZERO, EQUATION ,I3)
      STOP
```

```
      8 CONTINUE
        ELT=1.0E0/ELT
        JD=JC+LE
        JIB=JIA+1
        DO 9 IB=IA,LCLO
C       MODIFY THE EQUATION JIB
        ELTA=A(JD)*ELT
C       THE MATRIX
        JE=JD+1
        JF=JC+1
        DO 10 IC=JIA,JALO
        A(JE)=A(JE)-ELTA*A(JF)
        JE=JE+1
        JF=JF+1
     10 CONTINUE
C       THE SECOND MEMBER FOR EACH LOAD CASE
        JE=JIB
        JF=JIA
        DO 11 IC=1,NCC
        BT(JE)=BT(JE)-ELTA*BT(JF)
        JE=JE+JA
        JF=JF+JA
     11 CONTINUE
        JD=JD+LE
        JIB=JIB+1
      9 CONTINUE
        JC=JC+LED
        JIA=JIA+1
      6 CONTINUE
C       MOVE REDUCED EQUATIONS TO OUTPUT BUFFER, WRITE AS NECESSARY
      4 CONTINUE
        JC=LE-1
        JD=1
        JE=LE
        DO 12 IA=1,LC
        JF=LB+JC
        IF (JF-LBUF) 13,14,14
C       WRITE SLOW OUTPUT BUFFER
C       ICUE IS START OF LAST EQN
     14 CONTINUE
        ICUE=LB-JC-2
        JF=1
        JG=LBLM
        DO 15 IB=1,NBLM
        IS=(JG-JF)+1
        CALL WRITMS(LUM,AO(JF),IS,IAO)
        JF=JF+LBLM
        JG=JG+LBLM
        IAO=IAO+1
     15 CONTINUE
        LB=1
     13 CONTINUE
C       TRANSFER ONE EQUATION
        DO 16 IB=JD,JE
        AO(LB)=A(IB)
        LB=LB+1
     16 CONTINUE
```

```
      JC=JC-1
      JD=JD+LED
      JE=JE+LE
   12 CONTINUE
C     EQNS READ TO DATE
      LAF=LA+LC
      IF (LAF-JA) 17,18,18
C     REDUCTION OF REST OF THE SYSTEM
   17 CONTINUE
C     FAST BLOCK FILE PARAMETERS
      IFI=IAI
      IFO=IAO
C     FAST OUTPUT BUFFER CONTROL
      LBF=1
      LBG=LC+1
      JCF=JALO-LAF
   19 CONTINUE
C     READ FAST BLOCK = 1 SOL BLOCK
      LCF=MIN0(MMAT,JA-LAF)
      JC=1
      JD=LBLM
      DO 20 IA=1,NBLM
      IT=(JD-JC)+1
      CALL READMS(LUM,FI(JC),IT,IFI)
      JC=JC+LBLM
      JD=JD+LBLM
      IFI=IFI+1
   20 CONTINUE
      JC=1
      JIA=LA+1
      DO 21 IA=1,LC
C     ELIMINATE THE UNKNOWN JIA
      ELT=A(JC)
      IF ( ABS(ELT)-1.0E-6) 555,555,41
   41 ELT=1.0E0/ELT
      JD=IA
      JIB=LAF+1
      DO 22 IB=1,LCF
C     MODIFY THE EQUATION JIB
      ELTA=FI(JD)*ELT
C     THE MATRIX
      JE=JD+1
      JF=JC+1
      DO 23 IC=JIA,JALO
      FI(JE)=FI(JE)-ELTA*A(JF)
      JE=JE+1
      JF=JF+1
   23 CONTINUE
C     THE SECOND MEMBER FOR EACH LOAD CASE
      JE=JIB
      JF=JIA
      DO 24 IC=1,NCC
      BT(JE)=BT(JE)-ELTA*BT(JF)
      JE=JE+JA
      JF=JF+JA
   24 CONTINUE
      JD=JD+LE
```

```
      JIB=JIB+1
   22 CONTINUE
      JC=JC+LED
      JIA=JIA+1
   21 CONTINUE
C     MOVE REDUCED EQUATIONS TO OUTPUT BUFFER,WRITE AS NECESSARY
      JD=LBG
      JE=LE
      DO 25 IA=1,LCF
      JF=LBF+JCF
      IF (JF-LBUF) 26,26,27
C     WRITE FAST OUTPUT BUFFER
   27 CONTINUE
      JF=1
      JG=LBLM
      DO 28 IB=1,NBLM
      IU=(JG-JF)+1
      CALL WRITMS(LUM,FO(JF),IU,IFO)
      JF=JF+LBLM
      JG=JG+LBLM
      IFO=IFO+1
   28 CONTINUE
      LBF=1
   26 CONTINUE
C     TRANSFER ONE EQUATION
      DO 29 IB=JD,JE
      FO(LBF)=FI(IB)
      LBF=LBF+1
   29 CONTINUE
      JD=JD+LE
      JE=JE+LE
   25 CONTINUE
      LAF=LAF+LCF
      IF (LAF-JA) 19,30,30
C     EMPTY FAST BUFFER
   30 CONTINUE
      JC=1
      JD=LBLM
      DO 31 IA=1,NBLM
      IV=(JD-JC)+1
      CALL WRITMS(LUM,FO(JC),IV,IFO)
      JC=JC+LBLM
      JD=JD+LBLM
      IFO=IFO+1
   31 CONTINUE
      LA=LA+LC
      MMAT=LBUF/(JCF+1)
      JMAT=MMAT*(JCF+1)
      IAI=IAO
      GOTO 1
C     SOLVE LAST EQUATION
   18 CONTINUE
      JC=LB-1
      ELT=AO(JC)
      IF ( ABS(ELT)-1.0E-6) 555,555,33
   33 CONTINUE
      ELT=1.0E0/ELT
```

```
      JD=JA
      DO 34 IA=1,NCC
      BT(JD)=BT(JD)*ELT
      JD=JD+JA
   34 CONTINUE
C     BACKWARD PASS
      JB=2
      JD=JA
      DO 35 IA=1,JALO
      JC=JC-JB
      JD=JD-1
      IF (JC) 36,36,37
   36 CONTINUE
      IAO=IAO-NBLM
      IAI=IAO
      JE=1
      JF=LBLM
      DO 38 IB=1,NBLM
      IW=(JF-JE)+1
      CALL READMS(LUM,AO(JE),IW,IAI)
      JE=JE+LBLM
      JF=JF+LBLM
      IAI=IAI+1
   38 CONTINUE
      JC=ICUE
   37 CONTINUE
      JE=JC+1
      JF=JD+1
C     CALCULATE THE UNKNOWN JD
      ELT=1.0E0/AO(JC)
      JG=JD
      JH=JF
      DO 39 IB=1,NCC
C     SUM OVER UNKNOWNS ALREADY CALCULATED
      JI=JE
      JJ=JH
      SUM=0.0E0
      DO 40 IC=1,IA
      SUM=SUM+AO(JI)*BT(JJ)
      JI=JI+1
      JJ=JJ+1
   40 CONTINUE
      BT(JG)=(BT(JG)-SUM)*ELT
      JG=JG+JA
      JH=JH+JA
   39 CONTINUE
      JB=JB+1
   35 CONTINUE
      RETURN
  555 WRITE(LUO,556)JIA
  556 FORMAT(   71H UN COEFFICIENT DIAGONAL DE LA MATRICE TRIANGULAIRE
     1ST ZERO, EQUATION ,I3)
      STOP 556
      END
```

```
      SUBROUTINE ITFOR
C     CONSTRUCTS THE MATRIX AND SECOND MEMBERS
      DIMENSION XL(900),NL(300),VAC(3),
     1 ILIB(4),LIB(36),LINK(4),IPQP(4),IPQN(4),
     2 IXLS(4),XLS(48),GOF(20),GWT(20),IGOF(6),IDER(10),
     3 KSYMB(3),LSYM(3),NLT(24),IAQ(24),NCAT(72),NCBT(72),DCA(108),
     4 VACA(3),VACB(3),VACC(3),XA(24),XB(24),
     5 UVL(108),TVL(108),UTL(108),TTL(108),UT(9),TT(9),
     6 NR(2),NI(200),XI(4),XC(3),XD(9),UN(3),XAI(16),
     7 JDCO(20),TS(180),DD(300),KDCO(20),NLU(24),NLV(3)
      DIMENSION BUFC(1008),IBUF(500),
     1 NCHT(72),ISU(3),JAQ(13),KAQ(13)
      DIMENSION YI(2),DERO(2),DER(8),YAI(10),XIH(2),DERH(4)
      COMMON/FIXE/IDENT(8),JDENT(40),ITEST,JOHN,IFLAG(8),
     1 NOMX,NUMXC,PINO,NONO,NONE,NONOC,NONOD,NONOE,NONOF,NONC,NOPAR(5),
     2 NSYM,ISYM(3),NXB,NEB,NSUB,JSUB(10),YM(10),PR(10),DEN(10),
     3 NSQ(10),NSL(10),ISUB(10),INK(100),IEQH(10),IEQL(10),
     4 JEQD(10),JEQF(10),JEQL(10),KEQD(10),KEQF(10),KEQL(10),
     5 LEQD(10),LEQF(10),LEQL(10),NLIB,LUA,LUB,LUR,LBLM,NBLM,LENR,
     6 ECH,YMO,LCO,NCC,LUD(14),KGRAV(50),KSYM(10),
     7 LUAL,LUAN,LUS,PIN,JOKER,LUMD(10),
     8 MOTS(10),MMAT(10),JMAT(10),MUG(10),MUE(10),LUG,LUGL,NOMO,
     9 LUE,LUEL,LUEN,LUEI,LUEJ
      COMMON/CHANGE/X(3600),JEL(100),KEL(800),NS(200),
     1 DCO(5760),NCO(400),IDCO(400),BUFR(1200),BUFB(1440),
     2 A(7230),BUFF(500)
      EQUIVALENCE (XL(1),BUFR(1)),(NL(1),BUFR(901))
      EQUIVALENCE (LSYMA,LSYM(1)),
     1 (LSYMB,LSYM(2)),(LSYMC,LSYM(3))
      EQUIVALENCE (UN(1),XD(7)),(UNA,UN(1)),
     1 (UNB,UN(2)),(UNC,UN(3))
      EQUIVALENCE (IBUF(1),BUFF(1))
      EQUIVALENCE (XIH(1),XI(3)),(DERH(1),DER(5))
      DATA VAC/1.0,1.0,1.0/
      DATA ILIB/1,1,9,21/
      DATA LIB/1,2, 2,3, 3,4, 4,1,
     1 1,2,5, 2,3,6, 3,4,7, 4,1,8,
     2 1,2,5,6, 2,3,7,8, 3,4,9,10, 4,1,11,12/
      DATA LINK/2,1,2,1/
      DATA IPQP/2,3,4,1/
      DATA IPQN/4,1,2,3/
      DATA IXLS/1,1,9,25/
      DATA XLS/1.0,1.0, -1.0,1.0, -1.0,-1.0, 1.0,-1.0,
     1          1.0,1.0, -1.0,1.0, -1.0,-1.0, 1.0,-1.0,
     2 0.0,1.0, -1.0,0.0, 0.0,-1.0, 1.0,0.0,
     3 1.0,1.0, -1.0,1.0, -1.0,-1.0, 1.0,-1.0,
     4 0.3333333333,1.0, -0.3333333333,1.0,
     5 -1.0,0.3333333333, -1.0,-0.3333333333,
     6 -0.3333333333,-1.0, 0.3333333333,-1.0,
     7 1.0,-0.3333333333, 1.0,0.3333333333/
      DATA GOF/0.5773502692, -0.5773502692,
     1 0.7745966692, 0.0, -0.7745966692,
     2 0.8611363116, 0.3399810436, -0.3399810436, -0.8611363116,
     3 0.9061798459, 0.5384693101, 0.0,
     4 -0.5384693101, -0.9061798459,
     5 0.9324695142, 0.6612093865, 0.2386191861,
     6 -0.2386191861, -0.6612093865, -0.9324695142/
```

```
      DATA GWT/1.0, 1.0,
     1 0.5555555555, 0.8888888889, 0.5555555555,
     2 0.3478548451, 0.6521451549, 0.6521451549, 0.3478548451,
     3 0.2369268851, 0.4786286705, 0.5688888889,
     4 0.4786286705, 0.2369268851,
     5 0.1713244924, 0.3607615730, 0.4679139346,
     6 0.4679139346, 0.3607615730, 0.1713244924/
      DATA IGOF/1,1,3,6,10,15/
      DATA LTS/20/
      DATA LUM/17/
      DATA LUO,MAN/6,100/
      DATA ISIXC,ISIXT/37777B,40000B/
      DATA NLM/7777B/
      DATA YAI/1.0,1.0, -1.0,1.0, -1.0,-1.0, 1.0,-1.0,
     1         1.0,1.0/
      DATA LUC/15/
      DATA LUF/16/
      LUGI=7*NOMO*NOMO
      JOKER=I.UM
      NCCT=3*NCC
      IAN=0
C     SCALE FACTORS AND PRECISION OF INTEGRATION
      RECH=1.0/ECH
      RYMO=1.0/YMO
      RAT=ECH/YMO
      NONH=NONOC+1
      ELT=2.0*PINO
      PIN=(ELT-1.0)*0.25**ELT
      NONON=9*NONOC
      LBUF=NBLM*LBLM
      JA=1
      KMAT=1
      DO 1 IA=1,NSUB
C     COEFFICIENTS/EQN
      JB=3*IEQL(IA)+NCC
      MOTS(IA)=JB
C     EQNS/SOL BLOCK = TRIPLES/FOR BLOCK
      JC=LBUF/JB
      JC=MINU(JC,LTS)
      MMAT(IA)=JC
C     WORDS USED/SOL BLOCK
      JD=JB*JC
      JMAT(IA)=JD
C     DIVERSE INCREMENTS
      JE=3*JB
      JF=JB+1
      JG=3*JC
C     POINTER TO EQNS FOR SUBREGION IA
      LUMD(IA)=KMAT
      JH=1
      DO 2 IB=1,NSUB
C     POINTER TO EQUATION COEFFICIENTS FOR TRIPLES
C     BELONGING TO SUBREGION IB, IF ANY
      IDEB(IB)=JH
      JI=INK(JA)
      JA=JA+1
      GOTO (2,3,4,5),JI
```

```
C       SKIP DISPLACEMENT TRIPLES ON INTERFACE
    3   JH=JH+3*JEQL(IB)
        GOTO 2
C       TRACTION TRIPLES ON INTERFACE
    4   JH=JH+3*LEQL(IB)
        GOTO 2
C       ALL TRIPLES ASSOCIATED WITH SUBREGION IB
    5   JH=JH+3*IEQH(IB)
    2   CONTINUE
C       POINTERS TO DIAGONAL AND SECOND MEMBERS
        JDEB=IDEB(IA)
        KDEB=JH
        CALL ITBUF (1,LUR,BUFR(1),BUFR(LENR),LUO,IERR)
        JH=ISUB(IA)
        JI=NSQ(JH)
        JJ=NSL(JH)
C       TEMPORARY FILE OF GIVEN FUNCTIONS AND DISCONTINUITIES
        JK=(JJ-1)/LUAN+1
        DO 6 IB=1,JK
        CALL ITBUF (1,LUA,BUFB(1),BUFB(LUAL),LUO,IERR)
        CALL ITBUF (0,LUB,BUFB(1),BUFB(LUAL),LUO,IERR)
    6   CONTINUE
        REWIND LUB
C       TEMPORARY FILE OF MOST COMMONLY USED INTEGRATION POINTS
        JK=MUG(IA)
        DO 97 IB=1,JK
        CALL ITBUF (1,LUG,BUFC(1),BUFC(LUGL),LUO,IERR)
        CALL ITBUF (0,LUC,BUFC(1),BUFC(LUGL),LUO,IERR)
   97   CONTINUE
        REWIND LUC
        IF (NSUB.EQ.1) GOTO 98
C       TEMPORARY FILE OF FACTORS FOR ELIMINATION OF SURPLUS TRACTIONS
        JK=MUE(IA)
        DO 99 IB=1,JK
        CALL ITBUF (1,LUE,BUFF(1),BUFF(LUEL),LUO,IERR)
        CALL ITBUF (0,LUF,BUFF(1),BUFF(LUEL),LUO,IERR)
   99   CONTINUE
        REWIND LUF
   98   CONTINUE
C       ELASTIC CONSTANTS AND SYMMETRIES THIS SUBREGION
        CALL ITMKE (JH,XA,XA,XA,XA,XA,IERR)
        CALL ITNCO (KSYM(JH),NLT,KSYMB)
        DO 7 IB=1,3
        IF (KSYMB(IB).EQ.0B) GOTO 8
C       REFLECT W R TO COORDINATE IB
        LSYM(IB)=2
        GOTO 7
    8   LSYM(IB)=1
    7   CONTINUE
        LUGJ=LUGI*LSYMA*LSYMB*LSYMC
        NHI=JEQD(IA)-1
        JK=(IEQH(IA)-1)/JC+1
        JL=1
        DO 9 IB=1,JK
C       FIRST AND LAST TRIPLES THIS BLOCK
        NLO=NHI+1
        NHI=NHI+JC
```

```
      NHI=MINO(NHI,LEQF(IA))
      JM=NHI-NLO+1
C     FOR DIRECTION COSINES FIRST ARGUMENT OF KERNELS
      JN=1
      JO=NONOD
      DO 92 IC=1,JJ
      CALL ITUNV (NL,NLT,JN,JO)
      DO 93 ID=1,NONOD
      JP=AND(NLT(ID),NLM)
      IF (JP.EQ.0) GOTO 94
      IF (JP.LT.NLO.OR.JP.GT.NHI) GOTO 93
      JQ=JP-NLO+1
      IF (ID.GT.NONOC) GOTO 95
      JDCO(JQ)=JP
      GOTO 93
C     INTERFACE TRACTION
   95 JR=ID-NONOC
      JDCO(JQ)=AND(NLT(JR),NLM)
   93 CONTINUE
   94 JN=JN+NONOD
      JO=JO+NONOD
   92 CONTINUE
      JN=JE*JM
      CALL ITZER (A,JN)
      JO=9*JM
      CALL ITZER (TS,JO)
      JO=JI
      JP=1
      JQ=NONOD
      JR=LUAN+1
      KA=LUGL+1
      KB=LUEN+1
      JMT=3*JM
      DO 10 IC=1,JJ
      CALL ITUNS (NS,JS,JO)
C     ORIENTATION OF NORMAL
      FAC=1.0
      IF (AND(JS,ISIXT).EQ.0B) GOTO 11
      FAC=-FAC
   11 JS=AND(JS,ISIXC)
      CALL ITPIK (X,KEL,XA,JS)
      CALL ITUNV (NL,NLT,JP,JQ)
      INT=1
      JT=1
      JV=1
      DO 12 ID=1,NONOD
      JW=NLT(ID)
      NLU(ID)=JW
      JW=AND(JW,NLM)
      NLT(ID)=JW
      IF (JW.EQ.0) GOTO 13
C     POSITION IN EQUATIONS
      DO 14 IE=1,NSUB
      IF (JW.GT.LEQF(IE)) GOTO 14
C     TRIPLE BELONGS TO SUBREGION IE
      LA=IDEB(IE)
      IF (LA.EQ.1.AND.IE.LT.IA) GOTO 15
```

```
      IAQ(ID)=LA+3*(JW-JEQD(IE))
      GOTO 16
C     ONLY INTERFACE TRACTION TRIPLES STORED SUBREGION IE
   15 IAQ(ID)=LA+3*(JW-LEQD(IE))
      GOTO 16
   14 CONTINUE
   16 CONTINUE
      CALL ITUNS (NCO,LA,JW)
      CALL ITNCH (LA,NCHT(JT),NCAT(JT),NCBT(JT))
      JT=JT+3
      IF (ID.GT.NONOC) GOTO 12
C     DIRECTION COSINES
      CALL ITDCO (JW,LA,IDCO,DCO,UT)
      CALL ITTRS (UT,DCA(JV),IERR)
      CALL ITSWO (DCA(JV+1),DCA(JV+3))
      CALL ITSWO (DCA(JV+2),DCA(JV+6))
      CALL ITSWO (DCA(JV+5),DCA(JV+7))
      JV=JV+9
   12 CONTINUE
C     SEGMENT IS ON INTERFACE
      INT=2
   13 CONTINUE
      JT=INT*NONOC
C     FOR PLACING OF DIAGONAL TERMS
      DO 19 ID=1,JT
      JV=NLT(ID)
      IF (JV.LT.NLO.OR.JV.GT.NHI) GOTO 19
C     ID-TH TRIPLE IS IN THIS BLOCK
      JW=JV-NLO+1
      LA=JE*(JW-1)
      IF (ID.GT.NONOC) GOTO 20
      KDCO(JW)=LA+IAQ(ID)
      GOTO 19
C     INTERFACE TRACTIONS
   20 LB=ID-NONOC
      KDCO(JW)=LA+IAQ(LB)
   19 CONTINUE
      IF (JR.LE.LUAN) GOTO 21
C     FILL BUFFER OF GIVEN FUNCTIONS AND DISCONTINUITIES
      CALL ITBUF (1,LUB,BUFB(1),BUFB(LUAL),LUO,IERR)
      JR=1
      JU=1
   21 CONTINUE
      IF (KA.LE.LUGL) GOTO 100
C     FILL BUFFER OF INTEGRATION POINTS
      CALL ITBUF (1,LUC,BUFC(1),BUFC(LUGL),LUO,IERR)
      KA=1
  100 CONTINUE
      JV=JL
      JW=1
      JZ=1
      LA=KDEB
      DO 22 ID=NLO,NHI
      FACA=FAC
      CALL ITSET (VAC,VACA,3)
C     CONTRIBUTIONS TO MATRIX AND SECOND MEMBER THIS TRIPLE
      CALL ITZER (UVL,NONON)
```

```
      CALL ITZER (TVL,NONON)
      LB=1
C     LOOK FOR MATCH OF DEGREE OF FREEDOM NUMBER
      LE=JDCO(JZ)
      DO 23 IE=1,NONOC
      IF (NLT(IE).NE.LE) GOTO 23
C     SINGULARITY
      LC=IE
      LD=1
      GOTO 24
   23 CONTINUE
      LC=0
      LD=0
   24 CONTINUE
      KD=KA
C     SYMMETRY W R TO X1
      DO 25 IE=1,LSYMA
      FACB=FACA
      CALL ITSET (VACA,VACB,3)
C     SYMMETRY W R TO X2
      DO 26 IF=1,LSYMB
      FACC=FACB
      CALL ITSET (VACB,VACC,3)
C     SYMMETRY W R TO X3
      DO 27 IG=1,LSYMC
C     CONTRIBUTIONS THIS ORTHANT
      CALL ITZER (UTL,NONON)
      CALL ITZER (TTL,NONON)
C     REFLECTED NODAL COORDINATES
      DO 28 IH=1,3
      ELT=VACC(IH)
      LE=IH
      DO 29 II=1,8
      XB(LE)=XA(LE)*ELT
      LE=LE+3
   29 CONTINUE
   28 CONTINUE
      IF (LB.EQ.1.OR.LC.EQ.0) GOTO 30
C     CHECK DISTANCE BETWEEN TRIPLE AND REFLECTED TRIPLE
      LE=JV
      DO 31 IH=1,3
      ELT=XL(LE)*(1.0-VACC(IH))
      LE=LE+1
      IF (ELT.LT.1.0E-5) GOTO 31.
      LD=0
      GOTO 30
   31 CONTINUE
C     SINGULARITY
      LD=1
   30 CONTINUE
      IF (LD.NE.0) GOTO 32
C     NO SINGULARITY
      CALL ITSSA (1,PIN,XL(JV),XB,NR,NI,IERR)
      IF (IERR.EQ.0) GOTO 33
  998 IAN=IAN+1
      IF (IAN.GT.MAN) GOTO 999
      LE=JV+2
```

```
      CALL ITUNS (JEL,LF,JS)
      WRITE (LUO,200) LF,(XL(IO),IO=JV,LE)
  200 FORMAT (   33H ERREUR POUR LA MATRICE, ELEMENT ,I3,
     1 15H COORDONNEES   ,3F14.4)
      GOTO 34
   33 CONTINUE
C     SUBSEGMENTS EACH WAY
      LE=NR(1)
      LF=NR(2)
      KE=0
      IF (LE.GT.1.OR.LF.GT.1) GOTO 101
      IF (NI(1).NE.NOMO.OR.NI(2).NE.NOMO) GOTO 101
C     STANDARD INTEGRATION
      KE=1
      KF=KD
  101 CONTINUE
C     JACOBIAN FOR TRANSFORM SUBSEGMENT - SEGMENT
      XYA=1.0/LE
      XYB=1.0/LF
      AREA=XYA*XYB
      XIA=2.0*XYA
      XIB=2.0*XYB
C     INTRINSIC COORDINATE X1 OF SUBSEGMENT ORIGIN
      XOA=1.0-XYA
      LG=1
      DO 35 IH=1,LE
C     INTRINSIC COORDINATE X2 OF SUBSEGMENT ORIGIN
      XOB=1.0-XYB
      DO 36 II=1,LF
C     ORDER OF INTEGRATION THIS SUBSEGMENT
      LH=NI(LG)
      LI=NI(LG+1)
      LJ=IGOF(LH)
      LK=IGOF(LI)
      DO 37 IJ=1,LH
C     X1 OF INTEGRATION POINT
      XI(1)=XOA+XYA*GOF(LJ)
      GWA=AREA*GWT(LJ)
      LL=LK
      DO 38 IK=1,LI
C     X2 OF INTEGRATION POINT
      XI(2)=XOB+XYB*GOF(LL)
      IF (KE.EQ.0) GOTO 102
C     STANDARD INTEGRATION
      XC(1)=BUFC(KF)
      KF=KF+1
      XC(2)=BUFC(KF)
      KF=KF+1
      XC(3)=BUFC(KF)
      KF=KF+1
      UNA=BUFC(KF)
      KF=KF+1
      UNB=BUFC(KF)
      KF=KF+1
      UNC=BUFC(KF)
      KF=KF+1
      ELT=BUFC(KF)
```

```
      KF=KF+1
      GOTO 103
  102 CONTINUE
      CALL ITGEO (XB,XI,XC,XD,ELT,IERR)
      IF (IERR.NE.0) GOTO 998
C     ORIENTATION OF NORMAL
      UNA=UNA*FACC
      UNB=UNB*FACC
      UNC=UNC*FACC
  103 CONTINUE
      CALL ITMKE (0,XL(JV),XC,UN,UT,TT,IERR)
      IF (IERR.NE.0) GOTO 998
C     JACOBIAN*GAUSS WEIGHT
      GWB=GWA*ELT*GWT(LL)
      LM=1
      DO 39 IL=1,NONOC
      CALL ITSHA (NONO,IL,XI,FUN)
C     MULTIPLY BY SHAPE FUNCTION THIS TRIPLE
      FACT=GWB*FUN
      DO 40 IM=1,9
      UTL(LM)=UTL(LM)+FACT*UT(IM)
      TTL(LM)=TTL(LM)+FACT*TT(IM)
      LM=LM+1
   40 CONTINUE
   39 CONTINUE
      LL=LL+1
   38 CONTINUE
      LJ=LJ+1
   37 CONTINUE
      XOB=XOB-XIB
      LG=LG+2
   36 CONTINUE
      XOA=XOA-XIA
   35 CONTINUE
      GOTO 41
C     SINGULARITY
   32 CONTINUE
      LE=ILIB(NONO)
      JDEG=0
      KDEG=0
      LF=LE
      DO 131 IH=1,4
      LG=LIB(LF)
      LH=LIB(LF+1)
      IF (NLT(LG).NE.NLT(LH)) GOTO 133
C     DEGENERATE SIDE
      JDEG=IH
      IF (NLT(LG).NE.JDCO(JZ)) GOTO 133
C     KERNEL AND COORDINATE SINGULARITIES COINCIDE
      KDEG=1
      LI=2*(IH-1)+1
      CALL ITSET (YAI(LI),XAI(5),4)
  133 LF=LF+NONO
  131 CONTINUE
      IF (KDEG.NE.0) GOTO 132
      LF=IXLS(NONO)+2*(LC-1)
      XAI(5)=XLS(LF)
```

```
      XAI(6)=XLS(LF+1)
      XAI(7)=XAI(5)
      XAI(8)=XAI(6)
  132 CONTINUE
      DO 42 IH=1,4
C     CHECK FOR DEGENERATE TRIANGLE BASE
      IF (IH.EQ.JDEG) GOTO 43
C     CHECK FIRST ARGUMENT NOT ON BASE
      LF=LE
      DO 44 II=1,NONO
      LG=LIB(LF)
      IF (NLT(LG).EQ.JDCO(JZ)) GOTO 43
      LF=LF+1
   44 CONTINUE
      CALL ITSSB (PIN,XL(JV),XB,XAI(5),IH,NR,NI,IERR)
      IF (IERR.NE.0) GOTO 998
      LF=2*(IH-1)+1
      CALL ITSET (YAI(LF),XAI,4)
      LXAI=1
C     SEE IF DEGENERATE SIDE ADJOINS TRIANGLE BASE
      IF (IPQN(IH).NE.JDEG) GOTO 139
      ETA=1.0
      LF=8
      GOTO 140
  139 CONTINUE
      IF (IPQP(IH).NE.JDEG) GOTO 141
      ETA=-1.0
      LF=10
  140 LXAI=2
C     INTRINSIC COORDINATES FOR SECOND LINEAR TRANSFORM
      CALL ITSET (XAI,XAI(9),8)
      LG=LF+LINK(IH)
      XAI(LG)=-XAI(LG)
  141 LF=2*LXAI
      LG=4*LXAI
C     SUBSEGMENTS EACH WAY
      LH=NR(1)
      LI=NR(2)
C     JACOBIAN FOR TRANSFORM SUBSEGMENT - TRIANGLE COORDINATES
      XYA=1.0/LH
      XYB=1.0/LI
      AREA=XYA*XYB
      XIA=2.0*XYA
      XIB=2.0*XYB
C     FIRST TRIANGLE COORDINATE OF SUBSEGMENT ORIGIN
      XOA=1.0-XYA
      LJ=1
      DO 46 II=1,LH
C     ORDER OF INTEGRATION THIS COLUMN OF SUBSEGMENTS
      LK=NI(LJ)
      LL=NI(LJ+1)
      LM=IGOF(LK)
      LN=IGOF(LL)
C     SECOND TRIANGLE COORDINATE OF SUBSEGMENT ORIGIN
      XOB=1.0-XYB
      DO 47 IJ=1,LI
      LO=LM
```

```
      DO 48 IK=1,LK
C     FIRST TRIANGLE COORDINATE OF INTEGRATION POINT
      YI(1)=XOA+XYA*GOF(LO)
      GWA=AREA*GWT(LO)
      LP=LN
      DO 49 IL=1,LL
C     SECOND TRIANGLE COORDINATE OF INTEGRATION POINT
      YI(2)=XOB+XYB*GOF(LP)
C     INTRINSIC COORDINATES AND JACOBIAN FOR TRANSFORM
C     TRIANGLE COORDINATES TO INTRINSIC COORDINATES
      CALL ITZER (XI,LF)
      CALL ITZER (DER,LG)
      LQ=1
      DO 136 IM=1,4
      CALL ITSHB (2,IM,YI,ELTA,DERO)
      LR=1
      LS=1
      LT=LQ
      DO 137 IN=1,LXAI
      LU=LT
      DO 138 IO=1,2
      ELTB=XAI(LU)
      XI(LS)=XI(LS)+ELTA*ELTB
      DER(LR)=DER(LR)+DERO(1)*ELTB
      LR=LR+1
      DER(LR)=DER(LR)+DERO(2)*ELTB
      LR=LR+1
      LS=LS+1
      LU=LU+1
  138 CONTINUE
      LT=LT+8
  137 CONTINUE
      LQ=LQ+2
  136 CONTINUE
      IF (LXAI.EQ.1) GOTO 142
C     NONLINEAR TRANSFORM
      ELT=ETA*YI(1)
      ELTA=2.0-ELT-YI(2)
      ELTB=(YI(2)-ELT)/ELTA
C     WEIGHT FUNCTIONS FOR LINEAR TRANSFORMS
      ELTC=0.5*(1.0+ELTB)
      ELTD=ELTC-ELTB
      ELTB=ELTA*ELTA
      DERO(1)=ETA*(YI(2)-1.0)/ELTB
      DERO(2)=(1.0-ELT)/ELTB
      LQ=1
      DO 143 IM=1,2
      ELTA=XI(IM)
      ELTB=XIH(IM)
      ELTE=ELTA-ELTB
C     INTRINSIC COORDINATE
      XI(IM)=ELTA*ELTC+ELTB*ELTD
      DO 144 IN=1,2
C     JACOBIAN BY PRODUCT FORMULA
      DER(LQ)=ELTE*DERO(IN)
     1 +DER(LQ)*ELTC+DERH(LQ)*ELTD
      LQ=LQ+1
```

```
  144 CONTINUE
  143 CONTINUE
  142 CONTINUE
      ELTB=DER(1)*DER(4)-DER(2)*DER(3)
      CALL ITGEO (XB,XI,XC,XD,ELT,IERR)
      IF (IERR.NE.0) GOTO 998
C     ORIENTATION OF NORMAL
      UNA=UNA*FACC
      UNB=UNB*FACC
      UNC=UNC*FACC
      CALL ITMKE (0,XL(JV),XC,UN,UT,TT,IERR)
      IF (IERR.NE.0) GOTO 998
C     JACOBIAN*GAUSS WEIGHT
      GWB=GWA*ELT*GWT(LP)*ELTB
      LQ=1
      DO 50 IM=1,NONOC
      CALL ITSHA (NONO,IM,XI,FUN)
C     MULTIPLY BY SHAPE FUNCTION THIS TRIPLE
      FACT=GWB*FUN
      IF (NLT(IM).EQ.JDCO(JZ)) GOTO 51
      DO 52 IN=1,9
      UTL(LQ)=UTL(LQ)+FACT*UT(IN)
      TTL(LQ)=TTL(LQ)+FACT*TT(IN)
      LQ=LQ+1
   52 CONTINUE
      GOTO 50
C     OMIT SINGULAR COMPONENT
   51 DO 53 IN=1,9
      UTL(LQ)=UTL(LQ)+FACT*UT(IN)
      LQ=LQ+1
   53 CONTINUE
   50 CONTINUE
      LP=LP+1
   49 CONTINUE
      LO=LO+1
   48 CONTINUE
      XOB=XOB-XIB
   47 CONTINUE
      XOA=XOA-XIA
      LJ=LJ+2
   46 CONTINUE
   43 LE=LE+NUNO
   42 CONTINUE
C     FOR LEADING DIAGONAL SUBMATRIX, KERNEL T
   41 CONTINUE
      LE=JW
      DO 54 IH=1,9
      SUM=0.0
      LF=IH
      DO 55 II=1,NONOC
      SUM=SUM+TTL(LF)
      LF=LF+9
   55 CONTINUE
      TS(LE)=TS(LE)+SUM
      LE=LE+1
   54 CONTINUE
C     SYMMETRY TRANSFORM
```

```
      DO 56 IH=1,3
      ELT=VACC(IH)
      LE=IH
      DO 57 II=1,NONOE
      UVL(LE)=UVL(LE)+UTL(LE)*ELT
      TVL(LE)=TVL(LE)+TTL(LE)*ELT
      LE=LE+3
   57 CONTINUE
   56 CONTINUE
   34 LB=LB+1
      KD=KD+LUGI
      FACC=-FACC
      VACC(3)=-VACC(3)
   27 CONTINUE
      FACB=-FACB
      VACB(2)=-VACB(2)
   26 CONTINUE
      FACA=-FACA
      VACA(1)=-VACA(1)
   25 CONTINUE
C     DISTRIBUTE TO EQUATIONS
      LB=JU
      LC=JU+NONOE
      LD=1
      LE=1
      LF=NONOC+1
      DO 58 IE=1,NONOC
C     TRANSFORM TO LOCAL AXES OF SECOND ARGUMENT
      LG=LE
      LH=1
      DO 59 IF=1,3
      CALL ITTRA (UVL(LG),DCA(LE),UT(LH),1)
      CALL ITTRA (TVL(LG),DCA(LE),TT(LH),1)
      LG=LG+3
      LH=LH+3
   59 CONTINUE
      IF (NLT(IE).NE.ID.AND.NLT(LF).NE.ID) GOTO 60
C     GIVEN DISPLACEMENT OR DISPLACEMENT DISCONTINUITY
C     NOTE - NLT(LF)=0 FOR SURFACE SEGMENT
      LG=LB
      LH=NCCT*(ID-NLO)+1
      IF (INT.EQ.2) GOTO 126
      LI=LC
      DO 127 IF=1,NCC
      LJ=LG
      LK=LI
      LL=LD
      DO 128 IG=1,3
      IF (NCBT(LL).EQ.0B) GOTO 129
      LM=LJ
      GOTO 130
  129 LM=LK
  130 DD(LH)=BUFB(LM)
      LH=LH+1
      LJ=LJ+1
      LK=LK+1
      LL=LL+1
```

```
128 CONTINUE
    LG=LG+NONOF
    LI=LI+NONOF
127 CONTINUE
    GOTO 60
126 CONTINUE
    DO 61 IF=1,NCC
    CALL ITSET (BUFB(LG),DD(LH),3)
    LG=LG+NONOF
    LH=LH+3
 61 CONTINUE
 60 CONTINUE
    LG=1
    LH=IAQ(IE)
    LI=LA
    IF (INT.EQ.2) GOTO 62
C   SEGMENT ON SURFACE
    CALL ITNLG (NLU(IE),LJ,NLV)
    DO 63 IF=1,3
    LJ=LB
    LK=LC
    LL=LD
    LM=LH
    DO 64 IG=1,3
    IF (NCBT(LL).NE.0B) GOTO 65
C   DISPLACEMENT UNKNOWN
    ELTA=TT(LG)
    ELTB=ELTA*RECH
    ELT=UT(LG)*RYMO
    GOTO 66
C   TRACTION UNKNOWN
 65 ELTA=-UT(LG)
    ELTB=ELTA*RYMO
    ELT=-TT(LG)*RECH
    IF (NLV(IG).EQ.0B) GOTO 67
 66 CONTINUE
    IF (NCAT(LL).EQ.0B.AND.NCHT(LL).EQ.0B) GOTO 67
C   INCREMENT MATRIX COEFFICIENT
    A(LM)=A(LM)+ELTA
C   TRANSFORM GIVEN FUNCTION AND DISCONTINUITY
C   OR TWO GIVEN FUNCTIONS INTO SECOND MEMBER
 67 LN=LJ
    LO=LK
    LP=LI
    DO 68 IH=1,NCC
    A(LP)=A(LP)+ELT*BUFB(LN)-ELTB*BUFB(LO)
    LN=LN+NONOF
    LO=LO+NONOF
    LP=LP+1
 68 CONTINUE
    LG=LG+1
    LJ=LJ+1
    LK=LK+1
    LL=LL+1
    LM=LM+1
 64 CONTINUE
    LH=LH+JB
```

```
      LI=LI+JB
   63 CONTINUE
      GOTO 69
C     SEGMENT ON INTERFACE
   62 CONTINUE
      LJ=IAQ(LF)
      DO 70 IF=1,3
      LK=LB
      LL=LC
      LM=LD
      LN=LH
      LO=LJ
      DO 71 IG=1,3
C     MATRIX COEFFICIENT FOR TRACTION UNKNOWN
      ELTA=-UT(LG)
      ELTB=ELTA*RYMO
      LP=LM+NONOE
      IF (NCAT(LP).EQ.0B) GOTO 72
      LP=NLT(LF)
      IF (LP.GE.JEQD(IA).AND.LP.LE.LEQF(IA)) GOTO 17
C     UNKNOWN DOES NOT BELONG TO SUBREGION IA
      ELTA=-ELTA
   17 CONTINUE
      A(LO)=A(LO)+ELTA
C     MATRIX COEFFICIENT FOR DISPLACEMENT UNKNOWN, IF ANY
   72 CONTINUE
      ELTC=TT(LG)
      ELTD=ELTC*RECH
      IF (NCAT(LM).EQ.0B) GOTO 73
      A(LN)=A(LN)+ELTC
C     TRANSFORM DISCONTINUITIES OR DISCONTINUITY AND GIVEN FUNCTION
C     INTO SECOND MEMBER - MULTIPLIERS SAME BOTH CASES
   73 LP=LK
      LQ=LL
      LR=LI
      DO 74 IH=1,NCC
      A(LR)=A(LR)-ELTB*BUFB(LQ)-ELTD*BUFB(LP)
      LP=LP+NONOF
      LQ=LQ+NONOF
      LR=LR+1
   74 CONTINUE
      LG=LG+1
      LK=LK+1
      LL=LL+1
      LM=LM+1
      LN=LN+1
      LO=LO+1
   71 CONTINUE
      LH=LH+JB
      LI=LI+JB
      LJ=LJ+JB
   70 CONTINUE
   69 LB=LB+3
      LC=LC+3
      LD=LD+3
      LE=LE+9
      LF=LF+1
```

```
   58 CONTINUE
C     INCREMENT POINTERS TO MATRIX
      DO 75 IE=1,JT
      IAQ(IE)=IAQ(IE)+JE
   75 CONTINUE
      JV=JV+3
      JW=JW+9
      JZ=JZ+1
      LA=LA+JE
   22 CONTINUE
      IF (NSUB.EQ.1.OR.INT.EQ.2) GOTO 104
C     ELIMINATION OF SURPLUS TRACTIONS
      JV=1
      DO 105 ID=1,NONOC
      JW=NLU(ID)
      CALL ITNLG (JW,LA,NLV)
      JW=0
      DO 106 IE=1,3
      IF (NLV(IE).EQ.0B.OR.NCHT(JV).EQ.0B) GOTO 107
      JW=JW+1
      ISU(JW)=IE-1+IAQ(ID)-JN
  107 JV=JV+1
  106 CONTINUE
      IF (JW.EQ.0) GOTO 105
C     ELIMINATIONS THIS TRIPLE
      LA=0
  108 CONTINUE
      DO 109 IE=1,JW
      IF (KB.LE.LUEN) GOTO 110
C     BUFFER OF SUBSTITUTION DATA
      CALL ITBUF (1,LUF,BUFF(1),BUFF(LUEL),LUO,IERR)
      KB=1
      KC=1
  110 CONTINUE
      IF (IE.GT.1) GOTO 111
C     NUMBER OF INTERFACE SEGMENTS
      LB=IBUF(KC)
      LC=KC+3
      LD=1
      DO 112 IF=1,NONH
      LE=IBUF(LC)
C     POSITION IN EQUATIONS
      DO 113 IG=1,NSUB
      IF (LE.GT.LEQF(IG)) GOTO 113
C     TRIPLE BELONGS TO SUBREGION IG
      LF=IDEB(IG)
      IF (LF.EQ.1.AND.IG.LT.IA) GOTO 114
      JAQ(IF)=LF+3*(LE-JEQD(IG))
      GOTO 115
  114 JAQ(IF)=LF+3*(LE-LEQD(IG))
      GOTO 115
  113 CONTINUE
  115 CALL ITUNS (NCO,LF,LE)
      CALL ITNCO (LF,NCAT(LD),NCBT)
      LC=LC+1
      LD=LD+3
  112 CONTINUE
```

```
C       FACTOR FOR SUBSTITUTION IN TERMS OF TRACTION
        ELT=1.0
        IF (LE.GE.JEQD(IA).AND.LE.LE.LEQF(IA)) GOTO 111
C       TRACTION TRIPLE DOES NOT BELONG TO SUBREGION IA
        ELT=-ELT
  111   LC=KC+LUEJ
        LD=ISU(IE)
        LE=KDEB
        CALL ITSEI (JAQ,KAQ,NONH)
        DO 116 IF=1,JMT
        ELTA=A(LD)
        LF=1
        LG=LC
C       SUBSTITUTION IN TERMS OF DISPLACEMENT
        ELTB=ELTA*RAT
        DO 117 IG=1,NONOC
        LH=KAQ(IG)
        DO 118 IH=1,3
        IF (NCAT(LF).EQ.0B) GOTO 119
        A(LH)=A(LH)+ELTB*BUFF(LG)
  119   LF=LF+1
        LG=LG+1
        LH=LH+1
  118   CONTINUE
  117   CONTINUE
C       SUBSTITUTION IN TERMS OF TRACTION
        ELTB=ELTA*ELT
        LH=KAQ(NONH)
        DO 120 IG=1,3
        IF (NCAT(LF).EQ.0B) GOTO 121
        A(LH)=A(LH)+ELTB*BUFF(LG)
  121   LF=LF+1
        LG=LG+1
        LH=LH+1
  120   CONTINUE
C       CONTRIBUTIONS TO SECOND MEMBERS
        ELTB=ELTA*RYMO
        LH=LE
        DO 122 IG=1,NCC
        A(LH)=A(LH)-ELTB*BUFF(LG)
        LG=LG+1
        LH=LH+1
  122   CONTINUE
C       INCREMENT POINTERS
        DO 123 IG=1,NONH
        KAQ(IG)=KAQ(IG)+JB
  123   CONTINUE
        LD=LD+JB
        LE=LE+JB
  116   CONTINUE
        KB=KB+1
        KC=KC+LUEI
  109   CONTINUE
        LA=LA+1
        IF (LA.LT.LB) GOTO 108
C       ZEROISE ELIMINATED COLUMNS
        DO 124 IE=1,JW
```

```
      LA=ISU(IE)
      DO 125 IF=1,JMT
      A(LA)=0.0
      LA=LA+JB
  125 CONTINUE
  124 CONTINUE
  105 CONTINUE
  104 CONTINUE
      KA=KA+LUGJ
      JO=JO+1
      JP=JP+NUNOD
      JQ=JQ+NONOD
      JR=JR+1
      JU=JU+NONC
   10 CONTINUE
      REWIND LUB
      REWIND LUC
      REWIND LUF
C     DIAGONAL CONTRIBUTION (USE OF SOMIGLIANA IDENTITY)
C     TRANSFORM W R TO FIRST ARGUMENT
      JO=1
      JP=1
      JQ=1
      JR=JDEB
      JS=KDEB
      JT=NLO
      DO 76 IC=1,JM
C     FIXITY, DIRECTION COSINES
      CALL ITUNS (NCO,JV,JT)
      CALL ITNCO (JV,NCAT,NCBT)
      JU=JDCO(IC)
      CALL ITUNS (NCO,JV,JU)
      CALL ITDCO (JU,JV,IDCO,DCO,UT)
      CALL ITTRS (UT,DCA,IERR)
      CALL ITSWO (DCA(2),DCA(4))
      CALL ITSWO (DCA(3),DCA(7))
      CALL ITSWO (DCA(6),DCA(8))
C     TRANSFORM CONTRIBUTION W R TO SECOND ARGUMENT
      JV=1
      DO 79 ID=1,3
      CALL ITTRA (TS(JO),DCA,TT(JV),1)
      JO=JO+3
      JV=JV+3
   79 CONTINUE
      JV=1
      JW=KDCO(IC)
      LA=JS
      DO 84 ID=1,3
      LB=JP
      LC=JW
      DO 85 IE=1,3
      ELTA=TT(JV)
      IF (NCBT(IE).NE.0B.AND.JU.EQ.JT) GOTO 87
      IF (NCAT(IE).EQ.0B) GOTO 87
C     DECREMENT MATRIX
      A(LC)=A(LC)-ELTA
C     TRANSFORM GIVEN DISPLACEMENT OR DISCONTINUITY INTO SECOND MEMBER
```

```
C       MULTIPLIER SAME BOTH CASES
   87 ELTB=ELTA*RECH
      LD=LB
      LE=LA
      DO 88 IF=1,NCC
      A(LE)=A(LE)+ELTB*DD(LD)
      LD=LD+3
      LE=LE+1
   88 CONTINUE
      JV=JV+1
      LB=LB+1
      LC=LC+1
   85 CONTINUE
      JW=JW+JB
      LA=LA+JB
   84 CONTINUE
C     TRANSFORM EQUATIONS TO LOCAL AXES OF FIRST ARGUMENT
      JU=JQ
      DO 89 ID=1,JB
      CALL ITTRA (A(JU),DCA,A(JU),JB)
      JU=JU+1
   89 CONTINUE
C     ELIMINATE UNKNOWNS ZERO BY SYMMETRY
      JU=JQ
      DO 90 ID=1,3
      IF (NCAT(ID).NE.0B) GOTO 91
      CALL ITZER (A(JU),JB)
      A(JR)=1.0
   91 JR=JR+JF
      JU=JU+JB
   90 CONTINUE
      JP=JP+NCCT
      JQ=JQ+JE
      JS=JS+JE
      JT=JT+1
   76 CONTINUE
C     NUMBER OF SOL BLOCKS TO BE WRITTEN
      JO=(JN-1)/JD+1
      CALL ITMAT (LUS,LUM,LBLM,NBLM,JO,KMAT,JD)
      JL=JL+JG
      JDEB=JDEB+JG
    9 CONTINUE
    1 CONTINUE
      IF (IAN.NE.0) GOTO 997
      REWIND LUR
      REWIND LUA
      REWIND LUG
      REWIND LUE
      REWIND LUS
      RETURN
  999 WRITE (LUO,201) MAN
  201 FORMAT (     9H PLUS DE ,I3,
     1 24H ERREURS POUR LA MATRICE)
  997 STOP
      END
```

```
      SUBROUTINE ITSOL
C     BLOCK SOLUTION OF BANDED SYSTEM OF EQUATIONS
      DIMENSION FI(2410),FO(2410)
      DIMENSION IRQ(100),IRL(100),B(6000),C(4500),
     1 IEH(10),JED(10),JEL(10),LED(10),LEL(10),
     2 MOT(10),IAQ(11),IAL(11),MOTT(10),LUMF(10),
     3 LENB(10),LENC(10)
      COMMON/FIXE/IDENT(8),JDENT(40),IFLAG(10),NOPAR(15),
     1 NSYM,ISYM(3),NXB,NEB,NSUB,JSUB(70),
     2 INK(100),IEQH(10),IEQL(10),JEQD(10),JEQF(10),JEQL(10),
     3 KEQD(10),KEQF(10),KEQL(10),LEQD(10),LEQF(10),LEQL(10),
     4 NLIB,LUA,LUB,LUR,LBLM,NBLM,LENR,
     5 ECH,YMO,LCO,NCC,LUD(14),KGRAV(60),
     6 LUAL,LUAN,LUS,PIN,LUM,LUMD(10),
     7 MOTS(10),MMAT(10),JMAT(10)
      COMMON/CHANGE/A(19280),AO(2410)
      EQUIVALENCE (ICUE,AO(2410)),
     1 (IRQ(1),A(1)),(IRL(1),A(101)),
     2 (B(1),A(201)),(C(1),A(6201))
      DATA EPS,EPSA/1.0E-6,1.0E-50/
      DATA NBLO/8/
      DATA LUO/6/
      DATA LUCO/16/
      NSUBH=NSUB+1
      LBUF=NBLM*LBLM
C     PARAMETERS OF SYSTEM IN TERMS OF COEFFICIENTS AND EQUATION NUMRER
      DO 1 IA=1,NSUB
      IEH(IA)=3*IEQH(IA)
      JED(IA)=3*(JEQD(IA)-1)+1
      JEL(IA)=3*JEQL(IA)
      LED(IA)=3*(LEQD(IA)-1)+1
      LEL(IA)=3*LEQL(IA)
C     INSTANTANEOUS LENGTH OF EQUATION
      MOT(IA)=MOTS(IA)
    1 CONTINUE
      JA=1
      JB=1
      DO 2 IA=1,NSUB
C     FORMAT OF EQUATIONS SUBREGION IA
      JC=JA
      JD=1
      DO 3 IB=IA,NSUB
      JE=INK(JC)
      GOTO (4,5,4,6),JE
    4 IAL(JD)=0
      GOTO 7
    5 IAL(JD)=JEL(IB)
      GOTO 7
    6 IAL(JD)=IEH(IB)
    7 JC=JC+1
      JD=JD+1
    3 CONTINUE
C     SECOND MEMBERS
      IAL(JD)=NCC
      IAI=LUMD(IA)
      IAO=IAI
      JC=IEH(IA)
```

```
      JD=0
      JE=1
    8 CONTINUE
C     READ SLOW BLOCK
      JF=MIN0(NBLO*MMAT(IA),JC-JD)
      JG=JF-1
C.    NUMBER OF SOL BLOCKS TO READ
      JH=JG/MMAT(IA)+1
      JI=JMAT(IA)
      JJ=1
      DO 9 IB=1,JH
      CALL ITDRW (1,LUM,A(JJ),LBLM,NBLM,IAI)
      JJ=JJ+JI
    9 CONTINUE
      JH=MOT(IA)
      JI=JH+1
      JJ=1
      IF (JG.EQ.0) GOTO 10
C     REDUCE SLOW BLOCK
      JK=JH
      DO 11 IB=1,JG
      ELT=A(JJ)
      IF (ABS(ELT).LT.EPS) GOTO 999
C     ELIMINATE AN UNKNOWN
      ELT=1.0/ELT
      JL=JJ+JH
      JM=JJ+1
      DO 12 IC=IB,JG
      ELTA=A(JL)
      IF (ABS(ELTA).LT.EPSA) GOTO 13
C     MODIFY AN EQUATION
      ELTB=ELTA*ELT
      JN=JL+1
      DO 14 ID=JM,JK
      A(JN)=A(JN)-ELTB*A(ID)
      JN=JN+1
   14 CONTINUE
   13 JL=JL+JH
   12 CONTINUE
      JJ=JJ+JI
      JK=JK+JH
   11 CONTINUE
   10 CONTINUE
      IF (ABS(A(JJ)).LT.EPS) GOTO 999
      JJ=1
      JK=JH
      DO 15 IB=1,JF
      JL=JE+JK
      IF (JL.LE.LBUF) GOTO 16
C     EMPTY OUTPUT BUFFER
      ICUE=JE-JK-1
      CALL ITDRW (0,LUM,AO,LBLM,NBLM,IAO)
      JE=1
   16 CALL ITSET (A(JJ),AO(JE),JK)
      JJ=JJ+JI
      JE=JE+JK
      JK=JK-1
```

```
   15 CONTINUE
C     REDUCTION OF REST OF SYSTEM
      JD=JD+JF
      JJ=JA
      DO 17 IB=IA,NSUB
C.    NUMBER OF EQUATIONS TO BE REDUCED
      IF (IB.GT.IA) GOTO 18
      JK=JC-JD
      IF (JK.FQ.0) GOTO 19
      GOTO 20
   18 JK=IEH(IB)
   20 CONTINUE
C     UNKNOWNS ELIMINATED FROM SLOW BLOCK BUT ABSENT FROM FAST BLOCK
      JL=0
      JM=INK(JJ)
      GOTO (19,19,21,22),JM
   21 JL=MAX0(JL,LED(IA)-JB)
      IF (JL.GE.JF) GOTO 19
C     POINTER TO SLOW BLOCK
   22 JM=JI*JL+1
      JN=JL+1
C     POINTERS TO COEFFICIENTS TO BE MODIFIED IN FAST BLOCK
      JO=IAL(1)-JL+1
      JP=2
      IF (IA.EQ.NSUB) GOTO 23
      JQ=JJ+1
      JR=IA+1
      DO 24 IC=JR,NSUB
      IAQ(JP)=JO
      JS=INK(JQ)
      GOTO (25,26,25,27),JS
   26 JO=JO+JEL(IC)
      GOTO 25
   27 JO=JO+IEH(IC)
   25 JP=JP+1
      JQ=JQ+1
   24 CONTINUE
   23 IAQ(JP)=JO
      IFI=MAX0(IAI,LUMD(IB))
      IFO=IFI
      JO=0
      JP=1
      JQ=MOT(IB)
      JR=JQ-JF+JL
      JS=IAQ(2)-2
   28 CONTINUE
C     READ FAST BLOCK
      CALL ITDRW (1,LUM,FI,LBLM,NBLM,IFI)
      JT=MIN0(MMAT(IB),JK-JO)
      JU=JM
      JV=1
      JW=JS
      DO 29 IC=JN,JF
C     ELIMINATE AN UNKNOWN
      ELT=1.0/A(JU)
      JX=0
      JY=JV
```

```
      DO 30 ID=1,JT
      ELTA=FI(JY)
      IF (ABS(ELTA).LT.EPSA) GOTO 31
C     MODIFY AN EQUATION
      ELTB=ELTA*ELT
      JZ=JU+1
      LA=1
      DO 32 IE=IA,NSUBH
      IF (LA.GT.1) GOTO 33
C     NUMBER OF COEFFICIENTS TO MODIFY FOR SUBREGION IA IS VARIABLE
      LB=JW
      LC=JY+1
      GOTO 34
   33 LB=IAL(LA)
      LC=JX+IAQ(LA)
   34 CONTINUE
      IF (LB.EQ.0) GOTO 35
      LD=JZ+LB-1
      DO 36 IF=JZ,LD
      FI(LC)=FI(LC)-ELTB*A(IF)
      LC=LC+1
   36 CONTINUE
      JZ=JZ+LB
   35 LA=LA+1
   32 CONTINUE
   31 JX=JX+JQ
      JY=JY+JQ
   30 CONTINUE
      JU=JU+JI
      JV=JV+1
      JW=JW-1
   29 CONTINUE
C     TRANSFER TO OUTPUT BUFFER
      JU=JR-1
      DO 37 IC=1,JT
      JW=JP+JU
      IF (JW.LE.LBUF) GOTO 38
C     EMPTY OUTPUT BUFFER
      CALL ITDRW (0,LUM,FO,LBLM,NBLM,IFO)
      JP=1
   38 CALL ITSET (FI(JV),FO(JP),JR)
      JV=JV+JQ
      JP=JP+JR
   37 CONTINUE
      JO=JO+JT
      IF (JO.LT.JK) GOTO 28
C     WRITE LAST FAST BLOCK THIS SUBREGION
      CALL ITDRW (0,LUM,FO,LBLM,NBLM,IFO)
C     NEW EQUATION AND FILE FORMAT THIS SUBREGION
      MOT(IB)=JR
      MMAT(IB)=LBUF/JR
      JMAT(IB)=JR*MMAT(IB)
   19 JJ=JJ+NSUB
   17 CONTINUE
      JB=JB+JF
      IF (JD.EQ.JC) GOTO 39
C     MODIFY FORMAT OF EQUATIONS SUBREGION IA
```

```
      IAL(1)=IAL(1)-JF
      GOTO 8
C     POSITION AND LENGTH LAST REDUCED EQUATION SUBREGION IA
   39 MOTT(IA)=JH-JG
      LUMF(IA)=IAO
      ICUE=JE-MOTT(IA)
C     WRITE LAST BLOCK OF REDUCED EQUATIONS SUBREGION IA
      CALL ITDRW (0,LUM,AO,LBLM,NBLM,IAO)
      JA=JA+NSUBH
    2 CONTINUE
C     STRUCTURE OF SOLUTION FILES
      JA=1
      DO 40 IA=1,NSUB
      JB=1
      DO 41 IB=1,NSUB
C     POINTER TO UNKNOWNS BELONGING TO SUBREGION IB
      IRQ(JA)=JB
      JC=INK(JA)
      GOTO (42,43,44,45),JC
   42 JD=0
      GOTO 46
   43 JD=JEL(IB)
      GOTO 46
   44 JD=LEL(IB)
      GOTO 46
   45 JD=IEH(IB)
   46 IRL(JA)=JD
      JB=JB+JD
      JA=JA+1
   41 CONTINUE
      MOT(IA)=MOTS(IA)-NCC
      LENB(IA)=NCC*MOT(IA)
      LENC(IA)=NCC*IEH(IA)
   40 CONTINUE
C     BACKWARD PASS
      LUC=LUCO
      JA=NSUB
      JB=JA*JA
      JC=JB
C     POSITION OF LAST EQUATION, ALREADY IN CORE
      JD=ICUE
      DO 47 IA=1,NSUB
      IF (IA.GT.1) GOTO 48
      JF=IEH(NSUB)
C     SOLVE LAST EQUATION
      ELT=1.0/AO(JD)
      JG=JD+1
      JH=JF
      DO 49 IB=1,NCC
      B(JH)=AO(JG)*ELT
      JG=JG+1
      JH=JH+JF
   49 CONTINUE
      JE=NCC+2
      JD=JD-JE
      JG=JF-1
      IAI=LUMF(NSUB)-NBLM
```

```
      GOTO 50
C     INTERACTION WITH HIGHER NUMBERED SUBREGIONS
   48 CONTINUE
      JG=IRQ(JC)-1
      JF=MOT(JA)-JG
      JH=NSUB
      JI=JB
      JJ=IA-1
      DO 51 IB=1,JJ
      JK=LENC(JH)
      CALL ITBUF (1,LUC,C(1),C(JK),LUO,IERR)
      CALL ITBUF (0,LUB,C(1),C(JK),LUO,IERR)
      JL=IRL(JI)
      IF (JL.EQ.0) GOTO 52
C     RETAIN SOLUTIONS SUBREGION JH REQUIRED FOR SUBREGION JA
      JM=IEH(JH)
      JN=1
      JO=IRQ(JI)-JG
      DO 53 IC=1,NCC
      CALL ITSET (C(JN),B(JO),JL)
      JN=JN+JM
      JO=JO+JF
   53 CONTINUE
   52 JH=JH-1
      JI=JI-1
   51 CONTINUE
      REWIND LUC
      JG=IEH(JA)
      JE=MOTT(JA)
      IAI=LUMF(JA)
   50 CONTINUE
      JH=JG
      DO 54 IB=1,JG
      IF (JD.GT.0) GOTO 55
C     REFILL EQUATION BUFFER
      JI=IAI
      CALL ITDRW (1,LUM,AO,LBLM,NBLM,JI)
      IAI=IAI-NBLM
      JD=ICUE
   55 CONTINUE
C     CALCULATE ONE UNKNOWN
      ELT=1.0/AO(JD)
      JI=JH
      JJ=JH+1
      JK=JF
      JL=JD+JE-NCC
      DO 56 IC=1,NCC
C     SUM OVER UNKNOWNS ALREADY CALCULATED
      SUM=0.0
      JM=JD+1
      DO 57 ID=JJ,JK
      SUM=SUM+AO(JM)*B(ID)
      JM=JM+1
   57 CONTINUE
      B(JI)=(AO(JL)-SUM)*ELT
      JI=JI+JF
      JJ=JJ+JF
```

```
      JK=JK+JF
      JL=JL+1
   56 CONTINUE
      JH=JH-1
      JE=JE+1
      JD=JD-JE
   54 CONTINUE
C     COMPRESS AND WRITE RESULTS
      JG=IEH(JA)
      JH=1
      JI=1
      DO 58 IB=1,NCC
      CALL ITSET (B(JH),C(JI),JG)
      JH=JH+JF
      JI=JI+JG
   58 CONTINUE
      JG=LENC(JA)
      CALL ITBUF (0,LUB,C(1),C(JG),LUO,IERR)
      REWIND LUB
C     SWOP LOGICAL UNIT NUMBERS
      JG=LUB
      LUB=LUC
      LUC=JG
      JA=JA-1
      JB=JB-NSUB
      JC=JC-NSUBH
   47 CONTINUE
C     ASSEMBLE RESULTS FOR ITRES
      JA=NSUB
      DO 59 IA=1,NSUB
      JB=JA
      JC=NSUB
      JD=MOT(IA)
      DO 60 IB=1,NSUB
C     READ RESULTS SUBREGION JC
      JE=LENC(JC)
      CALL ITBUF (1,LUC,C(1),C(JE),LUO,IERR)
      JE=IRL(JB)
      IF (JE.EQ.0) GOTO 61
C     TRANSFER DATA TO FILE LUB
      JF=IRQ(JB)
      JG=JF
      IF (INK(JB).EQ.3) GOTO 62
      JH=1
      GOTO 63
   62 JH=LED(JC)-JED(JC)+1
   63 JI=IEH(JC)
      DO 64 IC=1,NCC
      CALL ITSET (C(JH),B(JG),JE)
      JG=JG+JD
      JH=JH+JI
   64 CONTINUE
C     CHECK WHETHER ANY MORE OF LUC NEED BE READ
      IF (JF.EQ.1) GOTO 65
   61 JB=JB-1
      JC=JC-1
   60 CONTINUE
```

```
   65 REWIND LUC
C     WRITE DATA FOR SUBREGION IA
      JD=LENB(IA)
      CALL ITBUF (0,LUB,B(1),B(JD),LUO,IERR)
      JA=JA+NSUB
   59 CONTINUE
      REWIND LUB
      RETURN
  999 WRITE (LUO,200)
  200 FORMAT (
     1 47H UN COEFFICIENT DIAGONAL DE LA MATRICE EST ZERO)
      STOP
      END
```

```
      SUBROUTINE ITSSB (PIN,XA,XB,XAI,LIN,NR,NI,IERR)
C     INTEGRATION SCHEME FOR SINGULAR CASE - XAI ARE
C     INTRINSIC COORDINATES OF FIRST ARGUMENT XA
      DIMENSION XA(3),XB(24),XAI(2),NR(2),NI(20),
     1 LINK(4),XLIN(4),IP(2),DEO(4),
     2 XI(2),XC(3),XD(9),XR(3),DEX(2),DEY(2),DEZ(2)
      DATA MIT,STEP,EPS/10,0.6,0.01/
      DATA MSS/20/
      DATA MST/100/
      DATA LINK/2,1,2,1/
      DATA XLIN/1.0,-1.0,-1.0,1.0/
      DATA IP/2,1/
      DATA DEO/1.0,1.0,-1.0,-1.0/
      DATA NOMX,NOMN/6,2/
      DATA NOMO/3/
      RNOMN=2*NOMN+1
      NOMXD=2*NOMX
      ASM=PIN/(NOMXD+1)
C     CONSTANT COORDINATE ON LINE
      JA=LINK(LIN)
      XI(JA)=XLIN(LIN)
C     VARIABLE COORDINATE
      JB=IP(JA)
      XI(JB)=0.0
C     ITERATE FOR MINIMUM DISTANCE TO LINE FROM XA
      CALL ITGEO (XB,XI,XC,XD,ELT,JERR)
      DO 1 IA=1,3
      XR(IA)=XA(IA)-XC(IA)
    1 CONTINUE
      CALL ITIPR (XR,XR,SUM)
      RO=SQRT(SUM)
      JC=3*(JB-1)+1
      DO 2 IA=1,MIT
      CALL ITNOR (XR,XR,IERR)
      IF (IERR.NE.0) GOTO 3
C     COSINE ANGLE BETWEEN XD(JC),XR * LENGTH XD(JC)
C     AND DISTANCE MOVED**2
      GR=0.0
      SUM=0.0
      JD=JC
      DO 4 IB=1,3
      ELT=XD(JD)
      GR=GR+ELT*XR(IB)
      SUM=SUM+ELT*ELT
      JD=JD+1
    4 CONTINUE
      IF (SUM.GE.1.0E-20) GOTO 5
C     SINGULARITY
      IERR=1
      GOTO 3
C     XI CHANGE
    5 GR=RO*GR/SUM
      ELT=ABS(GR)
      IF (ELT.LE.STEP) GOTO 6
C     STEP TOO LARGE FOR SAFETY
      GR=GR*STEP/ELT
    6 CONTINUE
```

```
C       ADJUST INTRINSIC COORDINATE
        ELT=XI(JB)+GR
        ELT=AMAX1(ELT,-1.0)
        XI(JB)=AMIN1(ELT,1.0)
        CALL ITGEO (XB,XI,XC,XD,ELT,JERR)
        DO 7 IB=1,3
        XR(IB)=XA(IB)-XC(IB)
      7 CONTINUE
        CALL ITIPR (XR,XR,SUM)
        RN=SQRT(SUM)
C       COMPARE CHANGE IN DISTANCE WITH DISTANCE
        ELT=RO-RN
        IF (ELT.LT.EPS*RN) GOTO 8
        RO=RN
      2 CONTINUE
C       DERIVATIVES FOR CALCULATION OF ASPECT RATIOS
C       W R TO INTRINSIC COORDINATES OF SEGMENT
      8 JC=1
        DO 9 IA=1,2
        CALL ITIPR (XD(JC),XD(JC),SUM)
        DEX(IA)=0.5*SQRT(SUM)
        JC=JC+3
      9 CONTINUE
        DET=DEU(LIN)
C       W R TO FIRST INTRINSIC COORDINATE OF TRIANGLE
        DEY(1)=DEX(JB)
C       SECOND
        SUM=0.0
        DO 10 IA=1,2
        ELT=0.5*(XI(IA)-XAI(IA))*DEX(IA)
        SUM=SUM+ELT*ELT
     10 CONTINUE
        DEY(2)=SQRT(SUM)
C       FIND HOW MANY SUBSEGMENTS EACH WAY
        DO 11 IA=1,2
        DER=DEY(IA)
        RADN=RN
        DO 12 IB=1,MSS
C       ASPECT RATIO OF SUBSEGMENT
        ASP=DER/RADN
        IF (ASP**NOMXD.GE.ASM) GOTO 13
C       TAKE IB SUBSEGMENTS
        JC=IB
        GOTO 14
     13 RADN=RADN+RN
     12 CONTINUE
        JC=MSS
     14 NR(IA)=JC
     11 CONTINUE
        JC=NR(1)*NR(2)
        IF (JC.LE.MST) GOTO 26
C       REDUCE NUMBER OF SUBSEGMENTS
        ELT=JC
        ELTA=MST/ELT
        ELTB=SQRT(ELTA)
        NR(1)=NR(1)*ELTB
        NR(2)=MST/NR(1)
```

```
   26 CONTINUE
      DO 27 IA=1,2
      JC=NR(IA)
      DEY(IA)=DEY(IA)/JC
   27 CONTINUE
      JC=NR(1)
C     INTRINSIC COORDINATE OF TRIANGLE IS DESCENDING
C     FUNCTION OF SUBSEGMENT NUMBER
      MR=0.5*JC*(1.0-DET*XI(JB))+1.0
      MR=MIN0(MR,JC)
C     CHOOSE ORDER OF FORMULA FOR EACH COLUMN OF SUBSEGMENTS
      DETA=1.0/JC
      DETB=0.5/NR(2)
      YAI=2.0*DETA*DET
      YA=DET-YAI
      JD=1
      DO 15 IA=1,JC
      IF (IA.EQ.MR) GOTO 16
C     VARIABLE INTRINSIC COORDINATE
      XI(JB)=YA
      CALL ITGEO (XB,XI,XC,XD,ELT,JERR)
      SUM=0.0
      DO 17 IB=1,3
      ELT=XA(IB)-XC(IB)
      SUM=SUM+ELT*ELT
   17 CONTINUE
      IF (SUM.GE.1.0E-20) GOTO 18
  999 IERR=1
      GOTO 3
C     DERIVATIVES W R TO INTRINSIC COORDINATES OF SEGMENT
   18 JE=1
      DO 19 IB=1,2
      CALL ITIPR (XD(JE),XD(JE),SUMA)
      DEX(IB)=0.5*SQRT(SUMA)
      JE=JE+3
   19 CONTINUE
C     DERIVATIVES W R TO INTRINSIC COORDINATES OF SUBSEGMENT
      DEZ(1)=DETA*DEX(JB)
      SUMA=0.0
      DO 20 IB=1,2
      ELT=DETB*(XI(IB)-XAI(IB))*DEX(IB)
      SUMA=SUMA+ELT*ELT
   20 CONTINUE
      DEZ(2)=SQRT(SUMA)
      YA=YA-YAI
      GOTO 21
C     CRITICAL SUBSEGMENT
   16 CONTINUE
      IF (RN.LT.1.0E-10) GOTO 999
      SUM=RN*RN
      DEZ(1)=DEY(1)
      DEZ(2)=DEY(2)
   21 CONTINUE
      DO 22 IB=1,2
      ELT=DEZ(IB)
C     ASPECT RATIO**2
      ASP=ELT*ELT/SUM
```

```
      ELTA=RNOMN
      ELTB=ASP**NOMN
      DO 23 IC=NOMN,NOMX
      IF (ELTA*ELTB.GT.PIN) GOTO 24
      JE=IC
      IF (NR(IB).GT.1) GOTO 25
      JE=MAX0(JE,NOMO)
      GOTO 25
   24 ELTA=ELTA+2.0
      ELTB=ELTB*ASP
   23 CONTINUE
      JE=NOMX
   25 NI(JD)=JE
      JD=JD+1
   22 CONTINUE
   15 CONTINUE
    3 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ITMKE (JA,XA,XB,UN,UT,TT,IERR)
C     CALCULATES KERNELS OF THE INTEGRAL EQUATION
      DIMENSION XA(3),XB(3),UN(3),UT(9),TT(9)
      COMMON/FIXE/IDENT(8),JDENT(40),ITEST,JOHN,IFLAG(8),NOPAR(15),
     1 NSYM,ISYM(3),NXB,NEB,NSUB,JSUB(10),YM(10),PR(10),
     2 DEN(250),NLIB,LUA,LUB,LUR,LBLM,NBLM,LENR,
     3 ECH,YMO
      DATA HPI/25.13274123/
      IF (JA.EQ.0) GOTO 1
C     ELASTIC CONSTANTS FOR SUBREGION JA
      COA=PR(JA)
      COB=(1.0-COA)*HPI
      COC=1.0-2.0*COA
C     SCALE FACTOR FOR KERNEL U
      UNA=YMO/ECH
      EUB=UNA*(1.0+COA)/(COB*YM(JA))
      EUA=EUB*(3.0-4.0*COA)
C     FOR KERNEL T
      ETA=COC/COB
      ETB=3.0/COB
      GOTO 2
C     CALCULATE KERNELS
    1 CONTINUE
      COA=XA(1)-XB(1)
      COB=XA(2)-XB(2)
      COC=XA(3)-XB(3)
      SUM=COA*COA+COB*COB+COC*COC
      IF (SUM.GT.1.0E-20) GOTO 3
      IERR=1
      GOTO 2
    3 CONTINUE
      RADS=1.0/SUM
C     RECIPROCAL OF DISTANCE
      RAD=SQRT(RADS)
      COA=COA*RAD
      COB=COB*RAD
      COC=COC*RAD
C     PRODUCTS OF DIRECTION COSINES
      COAA=COA*COA
      COBB=COB*COB
      COCC=COC*COC
      COAB=COA*COB
      COBC=COB*COC
      COCA=COC*COA
      UNA=UN(1)
      UNB=UN(2)
      UNC=UN(3)
      UNCO=UNA*COA+UNB*COB+UNC*COC
C     SCALAR MULTIPLIERS
      EUAR=EUA*RAD
      EUBR=EUB*RAD
      ETAR=ETA*RADS
      ETARN=ETAR*UNCO
      ETBRN=ETB*RADS*UNCO
C     KERNEL U
      UT(1)=EUAR+EUBR*COAA
      UT(4)=EUBR*COAB
```

```
      UT(5)=EUAR+EUBR*COBB
      UT(7)=EUBR*COCA
      UT(8)=EUBR*COBC
      UT(9)=EUAR+EUBR*COCC
      UT(2)=UT(4)
      UT(3)=UT(7)
      UT(6)=UT(8)
C     SYMMETRIC COMPONENT OF KERNEL T
      TT(1)=ETARN+ETBRN*COAA
      TAB=ETBRN*COAB
      TT(5)=ETARN+ETBRN*COBB
      TCA=ETBRN*COCA
      TBC=ETBRN*COBC
      TT(9)=ETARN+ETBRN*COCC
C     ANTISYMMETRIC COMPONENT
      AAB=(UNA*COB-UNB*COA)*ETAR
      ABC=(UNB*COC-UNC*COB)*ETAR
      ACA=(UNC*COA-UNA*COC)*ETAR
C     OFFDIAGONAL T
      TT(2)=TAB+AAB
      TT(3)=TCA-ACA
      TT(4)=TAB-AAB
      TT(6)=TBC+ABC
      TT(7)=TCA+ACA
      TT(8)=TBC-ABC
      IERR=0
    2 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ITMAT (LUS,LUM,LBLM,NBLM,JA,KMAT,JMAT)
C     WRITES JA SOL BLOCKS OF EQUATIONS TO SEQUENTIAL AND RANDOM
C     ACCESS FILES
      COMMON/CHANGE/X(13900),A(7230)
      DATA LUO/6/
      JB=1
      JC=JMAT
      DO 1 IA=1,JA
      JD=JB
      DO 2 IB=1,NBLM
      CALL WRITMS (LUM,A(JD),LBLM,KMAT)
      KMAT=KMAT+1
      JD=JD+LBLM
    2 CONTINUE
C     FOR CALCULATION OF RESIDUES
      CALL ITBUF (0,LUS,A(JB),A(JC),LUO,IERR)
      JB=JB+JMAT
      JC=JC+JMAT
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ITSHA (NONO,JA,XI,FUN)
C     CALCULATES SHAPE FUNCTION JA AT POINT XI
      DIMENSION XI(2),IP(2),IXLS(4),XLS(48),
     1 ICOP(4),ICO(12),XIA(2),XIS(2)
      DATA IP/2,1/
      DATA ICOP/1,1,1,5/
      DATA ICO/2,1,2,1,2,2,1,1,2,2,1,1/
      DATA IXLS/1,1,9,25/
      DATA XLS/1.0,1.0, -1.0,1.0, -1.0,-1.0, 1.0,-1.0,
     1         1.0,1.0, -1.0,1.0, -1.0,-1.0, 1.0,-1.0,
     2 0.0,1.0, -1.0,0.0, 0.0,-1.0, 1.0,0.0,
     3 1.0,1.0, -1.0,1.0, -1.0,-1.0, 1.0,-1.0,
     4 0.3333333333,1.0, -0.3333333333,1.0,
     5 -1.0,0.3333333333, -1.0,-0.3333333333,
     6 -0.3333333333,-1.0, 0.3333333333,-1.0,
     7 1.0,-0.3333333333, 1.0,0.3333333333/
      JB=2*(JA-1)+IXLS(NONO)
      DO 1 IA=1,2
      ELT=XLS(JB)
      XIA(IA)=ELT
      XIS(IA)=ELT*(XI(IA)+ELT)
      JB=JB+1
    1 CONTINUE
      IF (JA.GT.4) GOTO 2
C     CORNER FUNCTION
      PROD=XIS(1)*XIS(2)
      GOTO (4,4,5,6),NONO
C     LINEAR
    4 FUN=0.25*PROD
      GOTO 7
C     PARABOLIC
    5 FUN=0.25*PROD*(XIA(1)*XI(1)+XIA(2)*XI(2)-1.0)
      GOTO 7
C     CUBIC
    6 FUN=0.03125*PROD*(9.0*(XI(1)*XI(1)+XI(2)*XI(2))-10.0)
      GOTO 7
C     EDGE FUNCTION
    2 CONTINUE
      JB=ICOP(NONO)+JA-5
      JC=ICO(JB)
      JD=IP(JC)
      SUM=XI(JD)
      ELT=XIS(JC)*(1.0-SUM*SUM)
      IF (NONO.EQ.4) GOTO 8
C     PARABOLIC
      FUN=0.5*ELT
      GOTO 7
C     CUBIC
    8 FUN=0.28125*ELT*(1.0+9.0*XIA(JD)*SUM)
    7 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ITSHB (NONO,JA,XI,FUN,DER)
C     CALCULATES SHAPE FUNCTION AND DERIVATIVES
      DIMENSION XI(2),DER(2),
     1 IP(2),IXLS(4),XLS(48),ICOP(4),ICO(12),
     2 XIA(2),XIS(2)
      DATA IP/2,1/
      DATA ICOP/1,1,1,5/
      DATA ICO/2,1,2,1,2,2,1,1,2,2,1,1/
      DATA IXLS/1,1,9,25/
      DATA XLS/1.0,1.0, -1.0,1.0, -1.0,-1.0, 1.0,-1.0,
     1         1.0,1.0, -1.0,1.0, -1.0,-1.0, 1.0,-1.0,
     2 0.0,1.0, -1.0,0.0, 0.0,-1.0, 1.0,0.0,
     3 1.0,1.0, -1.0,1.0, -1.0,-1.0, 1.0,-1.0,
     4 0.3333333333,1.0, -0.3333333333,1.0,
     5 -1.0,0.3333333333, -1.0,-0.3333333333,
     6 -0.3333333333,-1.0, 0.3333333333,-1.0,
     7 1.0,-0.3333333333, 1.0,0.3333333333/
      JB=2*(JA-1)+IXLS(NONO)
      DO 1 IA=1,2
      ELT=XLS(JB)
      XIA(IA)=ELT
      XIS(IA)=ELT*(XI(IA)+ELT)
      JB=JB+1
    1 CONTINUE
      IF (JA.GT.4) GOTO 2
C     CORNER FUNCTION
      PROD=XIS(1)*XIS(2)
      GOTO (4,4,5,6),NONO
C     LINEAR
    4 FUN=0.25*PROD
      DER(1)=0.25*XIA(1)*XIS(2)
      DER(2)=0.25*XIA(2)*XIS(1)
      GOTO 7
C     PARABOLIC
    5 SUM=XIA(1)*XI(1)+XIA(2)*XI(2)-1.0
      FUN=0.25*PROD*SUM
      DER(1)=0.25*XIA(1)*(PROD+XIS(2)*SUM)
      DER(2)=0.25*XIA(2)*(PROD+XIS(1)*SUM)
      GOTO 7
C     CUBIC
    6 ELT=9.0*(XI(1)*XI(1)+XI(2)*XI(2))-10.0
      FUN=0.03125*PROD*ELT
      PROD=18.0*PROD
      DER(1)=0.03125*(PROD*XI(1)+XIA(1)*XIS(2)*ELT)
      DER(2)=0.03125*(PROD*XI(2)+XIA(2)*XIS(1)*ELT)
      GOTO 7
C     EDGE FUNCTION
    2 CONTINUE
      JB=ICOP(NONO)+JA-5
      JC=ICO(JB)
      JD=IP(JC)
      ELT=XI(JD)
      ELTA=1.0-ELT*ELT
      ELTB=XIS(JC)*ELTA
      IF (NONO.EQ.4) GOTO 8
C     PARABOLIC
      FUN=0.5*ELTB
```

```
      DER(JC)=0.5*XIA(JC)*ELTA
      DER(JD)=-ELT*XIS(JC)
      GOTO 7
C     CUBIC
    8 ELTC=1.0+9.0*XIA(JD)*ELT
      FUN=0.28125*ELTB*ELTC
      DER(JC)=0.28125*XIA(JC)*ELTA*ELTC
      DER(JD)=0.28125*XIS(JC)*(9.0*ELTA*XIA(JD)-2.0*ELT*ELTC)
    7 CONTINUE
      RETURN
      END


      SUBROUTINE ITUNS (JA,JB,JC)
C     UNPACKS JCTH INTEGER OF JA
      DIMENSION JA(1),MOVE(4)
      DATA MOVE/0,-15,-30,-45/
      DATA NBYTP,NBYTN/2,-2/
      DATA IQU/77777B/
C     WORD-1
      JD=SHIFT(JC-1,NBYTN)
C     POSITION IN WORD
      JE=JC-SHIFT(JD,NBYTP)
      JF=SHIFT(JA(JD+1),MOVE(JE))
      JB=AND(JF,IQU)
      RETURN
      END


      SUBROUTINE ITNLG (JA,NL,NLV)
C     UNPACK NL
      DIMENSION NLV(3),NLVM(3)
      DATA NLM/7777B/
      DATA NLVM/10000B,20000B,40000B/
      NL=AND(JA,NLM)
C     FIXITIES
      DO 1 IA=1,3
      NLV(IA)=AND(JA,NLVM(IA))
    1 CONTINUE
      RETURN
      END


      SUBROUTINE ITNCO (JA,NCOA,NCOB)
C     UNPACKS COEFFICIENT OF NCO
      DIMENSION NCOA(3),NCOB(3),NCM(3),NCN(3)
      DATA NCM/10B,20B,40B/
      DATA NCN/1B,2B,4B/
      DO 1 IA=1,3
C     ELIMINATION OF EQUATIONS BY SYMMETRY
      NCOA(IA)=AND(JA,NCM(IA))
C     TRACTION UNKNOWNS
      NCOB(IA)=AND(JA,NCN(IA))
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ITBUF (JA,LUF,JB,JC,LUO,IERR)
      IF (JA.EQ.1.OR.JA.EQ.3) GOTO 1
      BUFFER OUT (LUF,0) (JB,JC)
      GOTO 2
    1 BUFFER IN (LUF,0) (JB,JC)
    2 CONTINUE
      IF (UNIT(LUF).GT.-0.5) GOTO 3
      IERR=0
      GOTO 4
C     TROUBLE
    3 CONTINUE
      IF (JA.GE.2) GOTO 5
      WRITE (LUO,100)
  100 FORMAT (/  26H ERREUR POUR BUFFER OUT/IN)
      STOP
    5 IERR=1
    4 CONTINUE
      RETURN
      END


      SUBROUTINE ITNOR (XA,XB,IERR)
C     PUTS NORMALISED XA IN XB
      DIMENSION XA(3),XB(3)
      SUM=0.0
      DO 1 IA=1,3
      ELT=XA(IA)
      SUM=SUM+ELT*ELT
    1 CONTINUE
      IF (SUM.GE.1.0E-10) GOTO 2
      IERR=1
      RETURN
    2 CONTINUE
      SUM=1.0/SQRT(SUM)
      DO 3 IA=1,3
      XB(IA)=XA(IA)*SUM
    3 CONTINUE
      IERR=0
      RETURN
      END


      SUBROUTINE ITVPR (XA,XB,XC)
C     PUTS VECTOR PRODUCT XA*XB IN XC
C     XC MAY NOT BE THE SAME AS XA OR XB
      DIMENSION XA(3),XB(3),XC(3),IP(3)
      DATA IP/2,3,1/
      DO 1 IA=1,3
      JA=IP(IA)
      JB=IP(JA)
      XC(IA)=XA(JA)*XB(JB)-XA(JB)*XB(JA)
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ITTRA (XA,DCO,XB,JA)
C     TRANSFORM XA INTO NEW COORDINATES, PUT RESULT IN XB
C     DCO ARE DIRECTION COSINES OF NEW AXES
      DIMENSION XA(1),DCO(9),XB(1),XC(3)
      JB=1
      DO 1 IA=1,3
      SUM=0.0
      JC=1
      DO 2 IB=1,3
      SUM=SUM+DCO(JB)*XA(JC)
      JB=JB+1
      JC=JC+JA
    2 CONTINUE
      XC(IA)=SUM
    1 CONTINUE
      JB=1
      DO 3 IA=1,3
      XB(JB)=XC(IA)
      JB=JB+JA
    3 CONTINUE
      RETURN
      END


      SUBROUTINE ITIPR (XA,XB,ELT)
C     PUTS INNER PRODUCT XA,XB IN ELT
      DIMENSION XA(3),XB(3)
      ELT=0.0
      DO 1 IA=1,3
      ELT=ELT+XA(IA)*XB(IA)
    1 CONTINUE
      RETURN
      END


      SUBROUTINE ITZER (XA,JA)
C     ZEROISE JA ELEMENTS OF XA
      DIMENSION XA(1)
      DO 1 IA=1,JA
      XA(IA)=0.0
    1 CONTINUE
      RETURN
      END


      SUBROUTINE ITSET (XA,XB,JA)
C     TRANSFER JA ELEMENTS OF XA TO XB
      DIMENSION XA(1),XB(1)
      DO 1 IA=1,JA
      XB(IA)=XA(IA)
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ITTRS (DCA,DCB,IERR)
C     INVERTS 3*3 MATRIX
      DIMENSION DCA(9),DCB(9)
      CALL ITVPR (DCA(4),DCA(7),DCB(1))
      CALL ITVPR (DCA(7),DCA(1),DCB(4))
      CALL ITVPR (DCA(1),DCA(4),DCB(7))
C     DETERMINANT
      CALL ITIPR (DCA,DCB,ELT)
      IF (ABS(ELT).GE.1.0E-10) GOTO 1
      IERR=1
      GOTO 2
    1 ELT=1.0/ELT
      DO 3 IA=1,9
      DCB(IA)=DCB(IA)*ELT
    3 CONTINUE
      ELT=DCB(2)
      DCB(2)=DCB(4)
      DCB(4)=ELT
      ELT=DCB(3)
      DCB(3)=DCB(7)
      DCB(7)=ELT
      ELT=DCB(6)
      DCB(6)=DCB(8)
      DCB(8)=ELT
      IERR=0
    2 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ITSEI (KA,KB,JA)
      DIMENSION KA(1),KB(1)
      DO 1 IA=1,JA
      KB(IA)=KA(IA)
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ITPIK (X,KEL,XA,JA)
C     PICK OUT COORDINATES OF NODES OF SEGMENT JA
C     KEL IS PACKED
      DIMENSION X(1),KEL(1),XA(24),KELT(8)
      JC=8*JA
      JB=JC-7
      CALL ITUNV (KEL,KELT,JB,JC)
      JB=1
      DO 1 IA=1,8
      JC=3*(KELT(IA)-1)+1
      CALL ITSET (X(JC),XA(JB),3)
      JB=JB+3
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ITGEO (XA,XI,XB,XC,AREA,IERR)
C     GIVEN CARTESIAN COORDINATES OF NODES AND INTRINSIC COORDINATES
C     OF POINT, FINDS CARTESIAN COORDINATES, TANGENT AND NORMAL
      DIMENSION XA(24),XI(2),XB(3),XC(9),XD(3),
     1 NO(8),NOD(8),FOR(8),FORD(16)
      DATA NO/1,2,4,3,6,8,5,7/
      DATA NOD/1,3,7,5,11,15,9,13/
C     INTRINSIC COORDINATES
      XIA=XI(1)
      XIB=XI(2)
      XIAS=1.0-XIA*XIA
      XIBS=1.0-XIB*XIB
C     INTRINSIC COORDINATE OF NODE, AND INTRINSIC COORDINATE PRODUCTS
      YA=1.0
      PRODA=XIA
      PRODB=XIB
      JA=1
      JB=5
      JC=7
      DO 1 IA=1,2
      SUMA=1.0+PRODA
      SUMB=1.0+PRODB
C     SHAPE FUNCTION FOR MIDSIDE NODE ON X1=0
      JD=NO(JB)
      FOR(JD)=0.5*SUMB*XIAS
C     DERIVATIVES W.R.TO X1,X2
      JD=NOD(JB)
      FORD(JD)=-SUMB*XIA
      FORD(JD+1)=0.5*YA*XIAS
C     SHAPE FUNCTION FOR MIDSIDE NODE ON X2=0
      JD=NO(JC)
      FOR(JD)=0.5*SUMA*XIBS
C     DERIVATIVES W.R.TO X1,X2
      JD=NOD(JC)
      FORD(JD)=0.5*YA*XIBS
      FORD(JD+1)=-SUMA*XIB
C     INTRINSIC COORDINATE X1, AND INTRINSIC COORDINATE PRODUCT
      YC=1.0
      PRODC=XIA
      DO 2 IB=1,2
      SUMC=1.0+PRODC
      SUM=PRODC+PRODB
C     SHAPE FUNCTION FOR CORNER NODE, X1 VARIES MORE RAPIDLY
      JD=NO(JA)
      FOR(JD)=0.25*SUMC*SUMB*(SUM-1.0)
C     DERIVATES W.R.TO X1,X2
      JD=NOD(JA)
      FORD(JD)=0.25*YC*SUMB*(SUM+PRODC)
      FORD(JD+1)=0.25*YA*SUMC*(SUM+PRODB)
      YC=-YC
      PRODC=-PRODC
      JA=JA+1
    2 CONTINUE
      YA=-YA
      PRODA=-PRODA
      PRODB=-PRODB
      JB=JB+1
```

```
      JC=JC+1
    1 CONTINUE
C     CARTESIAN COORDINATES AND TANGENTS TO COORDINATE LINES
      CALL ITZER (XB,3)
      CALL ITZER (XC,3)
      CALL ITZER (XD,3)
      JA=1
      JB=1
      DO 3 IA=1,8
      ELT=FOR(IA)
      ELTA=FORD(JA)
      ELTB=FORD(JA+1)
      DO 4 IB=1,3
      YA=XA(JB)
      XB(IB)=XB(IB)+ELT*YA
      XC(IB)=XC(IB)+ELTA*YA
      XD(IB)=XD(IB)+ELTB*YA
      JB=JB+1
    4 CONTINUE
      JA=JA+2
    3 CONTINUE
      CALL ITSET (XD,XC(4),3)
C     JACOBIAN AND NORMAL
      CALL ITVPR (XC,XC(4),XC(7))
      CALL ITIPR (XC(7),XC(7),SUM)
      AREA=SQRT(SUM)
      CALL ITNOR (XC(7),XC(7),IERR)
      RETURN
      END
```

```
      SUBROUTINE ITUNV (JA,JB,JC,JD)
C     UNPACKS INTEGERS JC-JD OF JA
      DIMENSION JA(1),JB(1),MOVE(4)
      DATA MOVE/0,-15,-30,-45/
      DATA NBYTP,NBYTN/2,-2/
      DATA IQU/77777B/
      DATA IBYTN,NBYT/-15,4/
C     WORD-1 AND POSITION IN WORD
      JE=SHIFT(JC-1,NBYTN)
      JF=JC-SHIFT(JE,NBYTP)
      JE=JE+1
      MOT=SHIFT(JA(JE),MOVE(JF))
      JG=JD-JC+1
      DO 1 IA=1,JG
      IF (JF.LE.NBYT) GOTO 2
C     NEXT WORD
      JE=JE+1
      MOT=JA(JE)
      JF=1
    2 CONTINUE
      JB(IA)=AND(MOT,IQU)
      JF=JF+1
C     LINE UP NEXT INTEGER
      MOT=SHIFT(MOT,IBYTN)
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ITMIN (XA,XB,RAD,XI,IERR)
C     FINDS MINIMUM DISTANCE FROM XA TO
C     SEGMENT WITH NODES XB
      DIMENSION XA(3),XB(24),XI(2),YI(2),
     1 XC(3),XD(9),XDH(3),XR(3),GR(2),XE(3)
      EQUIVALENCE (XDH(1),XD(4)),(GRA,GR(1)),(GRB,GR(2))
      DATA MIT,STEP,EPS/10,0.6,0.01/
      STEPS=STEP*STEP
C     BEGIN FROM CENTRE
      YI(1)=0.0
      YI(2)=0.0
      CALL ITGEO (XB,YI,XC,XD,ELT,JERR)
      DO 1 IA=1,3
      XR(IA)=XA(IA)-XC(IA)
    1 CONTINUE
      CALL ITIPR (XR,XR,SUM)
      RO=SQRT(SUM)
      DO 2 IA=1,MIT
      CALL ITNOR (XR,XR,IERR)
      IF (IERR.NE.0) GOTO 3
C     GRADIENT OF DISTANCE IN INTRINSIC COORDINATE DIRECTIONS
      JA=1
      DO 4 IB=1,2
      CALL ITIPR (XD(JA),XR,GR(IB))
      JA=JA+3
    4 CONTINUE
C     CARTESIAN COORDINATE CHANGE CORRESPONDING TO
C     INCREMENT (GRA,GRB) AND DISTANCE MOVED**2
      SUM=0.0
      DO 5 IB=1,3
      ELT=GRA*XD(IB)+GRB*XDH(IB)
      XE(IB)=ELT
      SUM=SUM+ELT*ELT
    5 CONTINUE
      IF (SUM.GE.1.0E-20) GOTO 6
C     EXTREMUM
      RN=RO
      GOTO 7
C     COSINE ANGLE BETWEEN XR,XE * LENGTH XE
    6 CALL ITIPR (XE,XR,ELT)
C     RATIO DISTANCE TO BE MOVED/LENGTH XE
      FACT=RO*ELT/SUM
      GRA=GRA*FACT
      GRB=GRB*FACT
      SUM=GRA*GRA+GRB*GRB
      IF (SUM.LE.STEPS) GOTO 8
C     STEP TOO LARGE FOR SAFETY
      FACT=STEP/SQRT(SUM)
      GRA=GRA*FACT
      GRB=GRB*FACT
    8 CONTINUE
C     ADJUST INTRINSIC COORDINATES
      DO 9 IB=1,2
      ELT=YI(IB)+GR(IB)
C     STAY INSIDE SEGMENT
      ELT=AMAX1(ELT,-1.0)
      YI(IB)=AMIN1(ELT,1.0)
```

```
    9 CONTINUE
      CALL ITGEO (XB,YI,XC,XD,ELT,JERR)
      DO 10 IB=1,3
      XR(IB)=XA(IB)-XC(IB)
   10 CONTINUE
      CALL ITIPR (XR,XR,SUM)
      RN=SQRT(SUM)
C     COMPARE CHANGE IN DISTANCE WITH DISTANCE
    7 ELT=RO-RN
      IF (ELT.LT.EPS*RN) GOTO 11
      RO=RN
    2 CONTINUE
   11 RAD=RN
      XI(1)=YI(1)
      XI(2)=YI(2)
    3 CONTINUE
      RETURN
      END


      SUBROUTINE ITNCH (JA,NCOH,NCOA,NCOB)
C     UNPACKS NINE BITS OF NCO
      DIMENSION NCOH(3),NCOA(3),NCOB(3),
     1 NCH(3),NCM(3),NCN(3)
      DATA NCH/100B,200B,400B/
      DATA NCM/10B,20B,40B/
      DATA NCN/1B,2B,4B/
      DO 1 IA=1,3
C     SUBSTITUTION FOR REDUNDANT TRACTIONS
      NCOH(IA)=AND(JA,NCH(IA))
C     SEE ITNCO
      NCOA(IA)=AND(JA,NCM(IA))
      NCOB(IA)=AND(JA,NCN(IA))
    1 CONTINUE
      RETURN
      END


      SUBROUTINE ITDCO (JA,JB,IDCO,DCO,DCA)
C     PICK OUT DIRECTION COSINES FOR TRIPLE JA
C     JB IS VALUE OF NCO
      DIMENSION IDCO(1),DCO(1),DCA(9),DEL(9)
      DATA DEL/1.0,0.0,0.0, 0.0,1.0,0.0, 0.0,0.0,1.0/
      JC=AND(JB,7B)
      IF (JC.EQ.0B) GOTO 1
      CALL ITUNS (IDCO,JC,JA)
      JD=9*(JC-1)+1
      CALL ITSET (DCO(JD),DCA,9)
      GOTO 2
    1 CALL ITSET (DEL,DCA,9)
    2 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ITSSA (MODE,PIN,XA,XB,NR,NI,IERR)
C     DETERMINES THE INTEGRATION SCHEME FOR CASE IN WHICH
C     FIRST ARGUMENT OF KERNELS IS NOT IN SEGMENT
C     MODE=1 FOR ITFOR, MODE=2 FOR ITINT
      DIMENSION XA(3),XB(24),NR(2),NI(200),ZERO(2),
     1 XI(2),YI(2),XC(3),XD(9),MR(2),DE(2)
      DATA ZERO/0.0,0.0/
      DATA MSS/20/
      DATA MST/100/.
      DATA NOMX,NOMN/6,2/
      DATA NOMO/3/
      RNOMN=2*NOMN+1
      NOMXD=2*NOMX
      IF (MODE.EQ.1) GOTO 18
      RNOMO=NOMN+1
      RNOMI=1.0
      ASM=PIN/((NOMXD+1)*(NOMX+1))
      GOTO 19
   18 RNOMO=1.0
      RNOMI=0.0
      ASM=PIN/(NOMXD+1)
   19 CONTINUE
C     MINIMUM DISTANCE
      CALL ITMIN (XA,XB,RAD,XI,IERR)
      IF (IERR.NE.0) GOTO 1
C     DATA FOR CALCULATING ASPECT RATIO
      CALL ITGEO (XB,ZERO,XC,XD,ELT,JERR)
      JA=1
      DO 2 IA=1,2
C     DERIVATIVE FOR ASPECT RATIO
      CALL ITIPR (XD(JA),XD(JA),SUM)
      DER=0.5*SQRT(SUM)
C     FIND HOW MANY SUBSEGMENTS THIS DIRECTION
      RADN=RAD
      DO 3 IB=1,MSS
C     ASPECT RATIO OF SUBSEGMENT
      ASP=DER/RADN
      IF (ASP**NOMXD.GE.ASM) GOTO 4
C     CRITERION CAN BE MET WITH IB SUBSEGMENTS
      JB=IB
      GOTO 5
    4 RADN=RADN+RAD
    3 CONTINUE
      JB=MSS
    5 NR(IA)=JB
      JA=JA+3
    2 CONTINUE
      JA=NR(1)*NR(2)
      IF (JA.LE.MST) GOTO 20
C     REDUCE NUMBER OF SUBSEGMENTS
      ELT=JA
      ELTA=MST/ELT
      ELTB=SQRT(ELTA)
      NR(1)=NR(1)*ELTB
      NR(2)=MST/NR(1)
   20 CONTINUE
      DO 21 IA=1,2
```

```
      JB=NR(IA)
C     INTRINSIC COORDINATE TAKEN TO BE DESCENDING
C     FUNCTION OF SUBSEGMENT NUMBER
      JC=0.5*JB*(1.0-XI(IA))+1.0
      MR(IA)=MIN0(JC,JB)
C     DERIVATIVE FOR ASPECT RATIO OF A SUBSEGMENT
      DE(IA)=DER/JB
   21 CONTINUE
C     CHOOSE ORDER OF FORMULA FOR EACH SUBSEGMENT
      JA=NR(1)
      JB=NR(2)
      JC=MR(1)
      JD=MR(2)
      JE=1
      YAI=2.0/JA
      YBI=2.0/JB
      YA=1.0-YAI
      DO 6 IA=1,JA
C     FIRST INTRINSIC COORDINATE OF NEAREST POINT
      IF (IA.EQ.JC) GOTO 7
      YI(1)=YA
      YA=YA-YAI
      GOTO 8
    7 YI(1)=XI(1)
    8 YB=1.0-YBI
      DO 9 IB=1,JB
C     SECOND INTRINSIC COORDINATE OF NEAREST POINT
      IF (IB.EQ.JD) GOTO 10
      YI(2)=YB
      YB=YB-YBI
      GOTO 11
   10 YI(2)=XI(2)
   11 CONTINUE
C     CALCULATE ASPECT RATIO**2 OF SEGMENT
      CALL ITGEO (XB,YI,XC,XD,ELT,JERR)
      SUM=0.0
      DO 12 IC=1,3
      ELT=XA(IC)-XC(IC)
      SUM=SUM+ELT*ELT
   12 CONTINUE
      IF (SUM.GE.1.0E-20) GOTO 13
      IERR=1
      GOTO 1
   13 CONTINUE
      DO 14 IC=1,2
      ELT=DE(IC)
C     ASPECT RATIO**2
      ASP=ELT*ELT/SUM
      ELTA=RNOMN
      ELTB=RNOMO
      ELTC=ASP**NOMN
      DO 15 ID=NOMN,NOMX
      IF (ELTA*ELTB*ELTC.GT.PIN) GOTO 16
C     TAKE ID INTEGRATION POINTS IN DIRECTION IC
      JF=ID
      IF (NR(IC).GT.1) GOTO 17
      JF=MAX0(JF,NOMO)
```

```
      GOTO 17
   16 ELTA=ELTA+2.0
      ELTB=ELTB+RNOMI
      ELTC=ELTC*ASP
   15 CONTINUE
      JF=NOMX
   17 NI(JE)=JF
      JE=JE+1
   14 CONTINUE
    9 CONTINUE
    6 CONTINUE
    1 CONTINUE
      RETURN
      END

      SUBROUTINE ITSWO (JA,JB)
C     SWOP JA AND JB
      JC=JB
      JB=JA
      JA=JC
      RETURN
      END

      SUBROUTINE ITDRW (MODE,LUM,A,LBLM,NBLM,IAO)
C     READS OR WRITES ONE SOL BLOCK OF EQUATIONS
      DIMENSION A(1)
      JA=1
      DO 1 IA=1,NBLM
      IF (MODE     .NE.0) GOTO 2
      CALL WRITMS (LUM,A(JA),LBLM,IAO,1)
      GOTO 3
    2 CALL READMS (LUM,A(JA),LBLM,IAO)
    3 IAO=IAO+1
      JA=JA+LBLM
    1 CONTINUE
      RETURN
      END
```