Calculation of the Nodal Forces in the 20-node Isoparametric Three-Dimensional Solid Element by the SHAPE Computer Program

H. Gorenflo, O. Jandl

IPP 4/167
IPP III/43 March 1978

MAX-PLANCK-INSTITUT FÜR PLASMAPHYSIK

8046 GARCHING BEI MÜNCHEN

MAX-PLANCK-INSTITUT FÜR PLASMAPHYSIK

GARCHING BEI MÜNCHEN

Calculation of the Nodal Forces in the 20-node Isoparametric Three-Dimensional Solid Element by the SHAPE Computer Program

H. Gorenflo, O. Jandl

IPP 4/167
IPP III/43 March 1978

Die nachstehende Arbeit wurde im Rahmen des Vertrages zwischen dem Max-Planck-Institut für Plasmaphysik und der Europäischen Atomgemeinschaft über die Zusammenarbeit auf dem Gebiete der Plasmaphysik durchgeführt.

IPP 4/167
IPP III/43

Calculation of the Nodal Forces in the 20-node Isoparametric Three-Dimensional Solid Element by the SHAPE Computer Program.

H. Gorenflo
O. Jandl

March 1978

Abstract

To calculate mechanical stress and displacement by the finite element method, the forces acting on the nodes of the volume grid must be known. This report describes the calculation of these nodal forces for the 20-node isoparametric solid element. The nodal forces are calculated from the force density values by using the shape functions, which describe the displacement field within an element. This method is the most accurate one for calculating the forces on the nodes of an element.

The program is based on the node numbering used in the finite element programs SOLID SAP IV and SOLID SAP V. A detailed description is given for the special case of toroidal D-coils. The magnetic force densities for this application are calculated with the computer program HEDO2 [7].

TABLE OF CONTENTS

1. Introduction

The calculation of mechanical stress and strain in three-dimensional structures by means of finite element programs calls for some additional routines. Figure 1 shows a scheme for calculating the stress and strain on a coil with the SAP program [1]. This starts with a magnetic field program for computing the force densities (j x B) in the winding region of the coil. An important additional routine (often referred to as a coupling program because of its function) is the SHAPE program, with which the force distribution on the nodes can be computed on the basis of the force densities. The force distribution is obtained by means of shape functions [6]. To achieve a high degree of accuracy, the calculations are based on the 20-node isoparametric solid element (Fig. 2a). The computation of the forces is demonstrated in Sec. 3 in the case of a D-coil. The numbering of the nodes per element and hence the computation of the nodal forces are based on the FE program, SOLID SAPIV or SOLID SAPV. With minor modifications it is possible to adapt the SHAPE program to any FE program.

2. Description of program structure

2.1 Isoparametric element

As describes in [2], a solid body is divided into isoparametric elements (Fig. 2) with 20 nodes. The structure is determined by the coordinates (X_ℓ, Y_ℓ, Z_ℓ) of the nodes. The isoparametric element is described by dimensionless intrinsic coordinates (ξ_ℓ, η_ℓ, ζ_ℓ) [3 , 4]. These (ξ_ℓ, η_ℓ, ζ_ℓ) coordinates are listed in Table 1. The element described by these coordinates (hexahedron) may be regarded as the mapping of an element with spatially curved bounds with (x, y, z) coordinates onto the (ξ, η,ζ) coordinates, where the shape functions (2) are used as transformation functions (1). The following relations are valid:

$$X = \sum_{\ell=1}^{20} N_\ell\ (\xi, \eta, \zeta)\ X_\ell,$$

$$Y = \sum_{\ell=1}^{20} N_\ell\ (\xi, \eta, \zeta)\ Y_\ell, \qquad (1)$$

$$Z = \sum_{\ell=1}^{20} N_\ell\ (\xi, \eta, \zeta)\ Z_\ell.$$

The shape functions N_ℓ are [4]

$$N_\ell(\xi,\eta,\zeta) = \frac{1}{8}(1+\xi\xi_\ell)\ (1+\eta\eta_\ell)\ (1+\zeta\zeta_\ell)\ (\xi\xi_\ell + \eta\eta_\ell + \zeta\zeta_\ell - 2) \quad \text{for } \ell = 1(1)8,$$

$$N_\ell(\xi,\eta,\zeta) = \frac{1}{4}(1-\xi^2)\ (1+\eta\eta_\ell)\ (1+\zeta\zeta_\ell) \quad \text{for } \ell = 9(2)15,$$

$$N_\ell(\xi,\eta,\zeta) = \frac{1}{4}(1-\eta^2)\ (1+\xi\xi_\ell)\ (1+\zeta\zeta_\ell) \quad \text{for } \ell = 10(2)16, \qquad (2)$$

$$N_\ell(\xi,\eta,\zeta) = \frac{1}{4}(1-\zeta^2)\ (1+\xi\xi_\ell)\ (1+\eta\eta_\ell) \quad \text{for } \ell = 17(1)20.$$

The values ξ_ℓ, η_ℓ and ζ_ℓ are the node coordinates of a unit element according to Table 1, and the values (X_ℓ, Y_ℓ, Z_ℓ) are the coordinates of the real structure.

Node	Coordinates		
ℓ	ζ_ℓ	η_ℓ	ξ_ℓ
1	1	1	1
2	1	1	-1
3	1	-1	-1
4	1	-1	1
5	-1	1	1
6	-1	1	-1
7	-1	-1	-1
8	-1	-1	1
9	1	1	0
10	1	0	-1
11	1	-1	0
12	1	0	1

ℓ	ζ_ℓ	η_ℓ	ξ_ℓ
13	-1	1	0
14	-1	0	-1
15	-1	-1	0
16	-1	0	1
17	0	1	1
18	0	1	-1
19	0	-1	-1
20	0	-1	1

Table 1 (ξ, η, ζ) coordinates of the unit element in Fig. 2a

2.2 Computation of the nodal forces

The mechanical load on a structure is given in the form of a force density $\vec{f}(x, Y, Z) = (f_x, f_y, f_z)$, (i.e. a force per unit volume). For every isoparametric element it is possible to compute the total force exerted at the centre of an element according to

$$\vec{F} = \int_{vol} \vec{f}\,(X, Y, Z)\; d\,(vol) \qquad (3)$$

To perform the integration in the (ξ, η, ζ) space, the Jacobian matrix J [5] has to be calculated according to (6) since the transformation of the volume element from the (X, Y, Z) space to the (ξ, η, ζ) space is given by

$$dx\;dy\;dz = \det\,[J]\;d\xi\;\;d\eta\;\;d\zeta\;. \qquad (4)$$

The integral (3) is then written in the form [4, 5]

$$\vec{F} = \int\int\int \vec{f}\;(\xi,\,\eta,\,\zeta)\;\det\,[J]\;d\xi\;\;d\eta\;\;d\zeta\;. \qquad (5)$$

The arguments of $\vec{f}$ are transformed to (X, Y, Z) coordinates according to the formulae (1).

2.2.1 Jacobian matrix

For the 20-node element the Jacobian matrix is given for each node (ξ, η, ζ) as follows [4]:

$$[J]= \begin{bmatrix} N_{1,\xi}, N_{\ell,\xi}, \ldots\ldots\ldots, N_{20,\xi} \\ N_{1,\eta}, N_{\ell,\eta}, \ldots\ldots\ldots, N_{20,\eta} \\ N_{1,\zeta} N_{\ell,\xi}, \ldots\ldots\ldots N_{20,\zeta} \end{bmatrix} \begin{bmatrix} X_1 & Y_1 & Z_1 \\ \vdots & \vdots & \vdots \\ X_\ell & Y_\ell & Z_\ell \\ \vdots & \vdots & \vdots \\ X_{20} & Y_{20} & Z_{20} \end{bmatrix} , \quad (6)$$

where X_ℓ, Y_ℓ, Z_ℓ with $\ell = 1(1)20$ are the nodes of an element in the (X, Y, Z) space. The formulae for the partial derivatives of the shape functions N_ℓ (2) are

$$N_{\ell,\xi} = \frac{\partial N_\ell}{\partial \xi} = \frac{\xi_\ell}{8}(1+\eta\eta_\ell)\ (1+\zeta\zeta_\ell)\ (2\xi\xi_\ell+\eta\eta_\ell+\zeta\zeta_\ell-1) \quad \text{for } \ell = 1(1)8,$$

$$N_{\ell,\xi} = \frac{\partial N_\ell}{\partial \xi} = -\frac{\xi}{2}\ (1+\eta\eta_\ell)\ (1+\zeta\zeta_\ell) \quad \text{for } \ell = 9(2)15,$$

$$N_{\ell,\xi} = \frac{\partial N_\ell}{\partial \xi} = \frac{\xi_\ell}{4}\ (1-\eta^2)\ (1+\zeta\zeta_\ell) \quad \text{for } \ell = 10(2)16,$$

$$N_{\ell,\xi} = \frac{\partial N_\ell}{\partial \xi} = \frac{\xi_\ell}{4}\ (1+\eta\eta_\ell)\ (1-\zeta^2) \quad \text{for } \ell = 17(1)20,$$

$$N_{\ell,\eta} = \frac{\partial N_\ell}{\partial \eta} = \frac{\eta\ell}{8}\ (1-\xi\xi_\ell)\ (1+\zeta\zeta_\ell)\ (\xi\xi_\ell+2\eta\eta_\ell+\zeta\zeta_\ell-1) \quad \text{for } \ell = 1(1)8,$$

$$N_{\ell,\eta} = \frac{\partial N_\ell}{\partial \eta} = \frac{\eta\ell}{4}\ (1-\xi^2)\ (1+\zeta\zeta_\ell) \quad \text{for } \ell = 9(2)15,$$

$$N_{\ell,\eta} = \frac{\partial N_\ell}{\partial \eta} = -\ \frac{\eta}{2}\ (1+\xi\xi_\ell)\ (1+\zeta\zeta_\ell) \quad \text{for } \ell = 10(2)16,$$

$$N_{\ell,\eta} = \frac{\partial N_\ell}{\partial \eta} = \frac{\eta\ell}{4}\ (1+\xi\xi_\ell)\ (1-\zeta^2) \quad \text{for } \ell = 17(1)20,$$

$$N_{\ell,\zeta} = \frac{\partial N_\ell}{\partial \zeta} = \frac{\zeta_\ell}{8}\ (1+\xi\xi_\ell)\ (1+\eta\eta_\ell)\ (\xi\xi_\ell+\eta\eta_\ell+2\zeta\zeta_\ell-1) \quad \text{for } \ell = 1(1)8,$$

$$N_{\ell,\zeta} = \frac{\partial N_\ell}{\partial \zeta} = \frac{\zeta_\ell}{4} (1-\xi^2)(1+\eta\eta_\ell) \qquad \text{for } \ell = 9(2)15,$$

$$N_{\ell,\zeta} = \frac{\partial N_\ell}{\partial \zeta} = \frac{\zeta_\ell}{4} (1+\xi\xi_\ell)(1-\eta^2) \qquad \text{for } \ell = 10(2)16,$$

$$N_{\ell,\zeta} = \frac{\partial N_\ell}{\partial \zeta} = -\frac{\zeta}{2} (1+\xi\xi_\ell)(1+\eta\eta_\ell) \qquad \text{for } \ell = 1(1)20.$$

2.2.2 Integration according to Gauss

Numerical integration of eq. (5) is performed by the Gauss method [3, 5, 6], yielding the following formula (a detailed description of the numerical problems is given in [6]):

$$\vec{F} = \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} \vec{f}(\xi,\eta,\zeta) \det[J]\, d\xi\, d\eta\, d\zeta = \sum_{i=1}^{\ell}\sum_{j=1}^{m}\sum_{h=1}^{n}$$

$$\vec{f}(\xi_i, \eta_j, \zeta_k)\, W_i\, W_j\, W_k \det[J] \qquad (8)$$

where (ξ_i, η_j, ζ_k) are the mesh points for the integration and W_i, W_j, W_k are the weightings by which the function values $\vec{f}(\xi_i, \eta_j, \zeta_k)$ are multiplied. In the numerical calculation $\ell = m = n = 3$ was found to afford sufficient accuracy. As three mesh points are used in each of the directions ξ, η and ζ (the number of mesh points could be chosen different in each direction), the mesh points are

$$\xi_1 = \eta_1 = \zeta_1 = -0.7745966\ldots,$$

$$\xi_2 = \eta_2 = \zeta_2 = 0 \qquad , \qquad (9)$$

$$\xi_3 = \eta_3 = \zeta_3 = 0.7745966\ldots \quad .$$

The respective weightings W_i are

$$W_1 = \frac{5}{9},$$

$$W_2 = \frac{8}{9},$$

$$W_3 = \frac{5}{9}.$$

This yields for each element a total of 27 mesh points (all combinations of ξ_i, η_j and ζ_k with i=1,2,3; j=1,2,3 and k=1,2,3).

2.2.3 Division of the total force with shape functions

Section 2.2.2 described the computation of the total force $\vec{F}$ of a finite element according to (8). In the SAP program the individual force components F_x, F_y, F_z are required for each node of a structure. The division of the forces of a finite element is done by means of the shape functions (2). Here the relation

$$\sum_{\ell=1}^{20} N_\ell \ (\xi_o, \eta_o, \zeta_o) = 1 \tag{10}$$

if of use (eq. (10)) can easily be checked by substituting the shape functions (2)). Extending the formula (8) with (10) yields [6]

$$\vec{F} = \sum_{i=1}^{3} \sum_{j=1}^{3} \sum_{k=1}^{3} \sum_{\ell=1}^{20} W_i \ W_j \ W_k \ \vec{f} \ (\xi_i, \eta_j, \zeta_k) \ \det [J]$$

$$N_\ell (\xi_i, \eta_j, \zeta_k), \tag{11}$$

where the calculation of $\vec{F}$ does not change. By omitting the summation over in eq. (11) it is possible to compute the nodal forces $\vec{F}$ according to

$$\vec{F}_\ell = \sum_{i=1}^{3} \sum_{j=1}^{3} \sum_{k=1}^{3} \vec{f} \ (\xi_i, \eta_j, \zeta_k) \ W_i W_j W_k \ \det [J]$$

$$N_\ell \ (\xi_i, \eta_j, \zeta_k) \tag{12}$$

with ℓ = 1(1)20.

As a structure generally consists of several elements, it is possible for up to 8 elements to have a common node (Fig. 3). The total force exerted at a node is composed of the sum of all individual forces exerted on the node from the adjacent elements.

3. Computation of the colume forces for a D-coil

3.1 Computation of the magnetic forces of a D-coil

The magnetic forces of a D coil are calculated with the HEDO2 program (see description of program [7]) as force components per unit length. With HEDO2 it is possible to treat arbitrarily shaped plain coil assemblies with constant rectangular winding cross-sections. It is only the mechanical stress in a D-shaped coil forming part of a toroidal coil assembly that is calculated. In principle, the method of calculating the force distribution that is described in the following is not limited to magnetic force densities for coils. It is possible to take arbitrary solid structures into account. The D-coil shown in Fig. 4 is approximated with straight solid elements. The periphery is subdivided into n_3 sections, the thickness D (Y direction) into n_1 sections, and the width B into n_2 sections. The contour of the coil is uniquely determined by the coordinates of the points $P_{1,m} = (X_{1,m}, Z_{1,m})$ (inner coil periphery), $P_{c,m} = (X_{c,m}, Z_{c,m})$ (central coil periphery) and $P_{2,m} = (X_{2,m}, Z_{2,m})$ (outer coil periphery) with $m = 1(1)n_3+1$, and by B and D their subdivisions n_2 and n_1. For each element in Fig. 4c the forces $\vec{g} = (g_x, g_y, g_z)$ at the centre are computed per unit length. The volume forces $\vec{h} = (h_x, h_y, h_z)$ are obtained with the following formula:

$$\vec{h}_{LX,LY,LN} = \vec{g}_{LX,LY,LN} \cdot D \cdot B/(n_1 \cdot n_2)$$

with $LX = 1(1)n_2$, $LY = 1(1)n_1$, and $LN = 1(1)n_3$. (13)

3.2 Computation of the spline coefficients with the HEDO2IN program

The HEDO2IN program prepares the output data of HEDO2 for SHAPE. As the volume forces are given at certain points only (main points of the volume elements), an interpolation method has to be used to obtain the values at the mesh points (ξ_i, η_j, ζ_k) which are used in the Gauss integration (12) to calculate the volume integral (8). To interpolate over the cross-section of the coil, one uses the two-dimensional spline function

$$S(u,v) = \sum_{i=1}^{4} \sum_{j=1}^{4} C_{i,j} \cdot (U-U_o)^{i-1} \cdot (V-Vo)^{j-1}, \qquad (14)$$

where the values $C_{i,j}$ are the two-dimensional spline coefficients. As shown in Fig. 5, the volume forces $\vec{h}_{LX,LY,LN}$ (the notation being simplified by omitting the subscripts LX, LY) at the centre of each element are given by eq. (13). It is of advantage for the spline interpolation if the volume forces are referred to the planes E_m (E_m is the plane through the points $P_{1,m}$ and $P_{2,m}$ with $m = 1(1)n_3+1$). Unlike the planes defined by the "main points" of the elements (e.g. the planes E_s in Fig. 5), the planes defined by the intersections of the straight lines are of equal cross-sectional area. (The difference is due to approximating the arcs by straight lines.) The cross-sectional area of the plane E_m is shown in Fig. 5b, where the regions with equal spline coefficients (21) are hatched in the interpolation. The volume forces are interpolated to the planes E_m according to the formulae (15 - 17). The interpolation is linear:

$$\vec{f}_m = (S_m \vec{h}_{m-1} + S_{m-1} \vec{h}_m)/(S_m + S_{m-1}) \text{ for } m = 2(1)n_3 \qquad (15)$$

or $m = 2(1)2 \cdot n_3$; see eq. (17) below

with

$$S_{m-1} = \sqrt{(X_{C,m}-X_{C,m-1})^2 + (Z_{C,m}-Z_{C,m-1})^2}$$

and

$$S_m = \sqrt{(X_{c,m+1}-X_{c,m})^2 + (Z_{c,m+1}-Z_{c,m})^2}$$

($X_{c,m}$ and $Z_{c,m}$ are the coordinates of the points $P_{c,m}$, which are on the geometric centre line of the coil). The values for the plane with Z = O are quadratically interpolated from the two adjacent segments since the volume forces $h_x(-Z)= h_x(Z)$ and $h_y(-Z) = h_y(Z)$ in the normal case are linear functions to the plane Z = O. The component h_z is given by h_z (Z = O) = O.

$$\vec{f}_{(x,y),1} = (T_2^{\,2} \cdot \vec{h}_{(x,y),1} - T_1^{\,2}\, \vec{h}_{(x,y),2})/(T_2^{\,2}-T_1^{\,2})$$

with $T_1 = S_1/2$ and $T_2 = S_1 + S_2/2$ (16)

$$\vec{f}_{(x,y),n_3+1} = (T_2^{\,2} \cdot \vec{h}_{(x,y),n_3} - T_1^{\,2}\, \vec{h}_{(x,y),n_3-1})/T_2^{\,2}-T_1^{\,2})$$

with

$$T_1 = S_{n_3}/2 \text{ and } T_2 = Sn_3 + S_{n-1}/2$$

($\vec{f}_{(x,y)}$ are the force components f_x and f_y).

In the special case of asymmetric forces with respect to Z = O it holds that

$$\vec{f}_1 = \vec{f}_{2n_3+1} = (\vec{h}_1 + \vec{h}_{2n_3})/2 \qquad (17)$$

This case is given when a coil assembly asymmetric with respect to the plane Z = O was computed in HEDO2 [7]. Here the forces are calculated not only for Z $\geq$ O but also for Z $\leq$ O.

The spline coefficients are calculated with the IBCICU program [8] for the mesh points (see Fig. 5a) U_{LX} with LX = 1(1)n_2-1 and V_{LY} with LY = 1(1)n_1-1 for each plane E_m with m = 1(1)L, where L = n_3 + 1 for force distributions symmetric with respect to Z = 0 or $L = 2 \cdot n_3$ for asymmetric force distributions.

3.3 Interpolation of the volume forces (VKRFT subroutine)

It must be possible to calculate the volume forces $\vec{f}(x,y,z)$ for every point $P(X_o,Y_o,Z_o)$ in the winding region of the coil. This is because of the Gauss integration and, in addition, the subdivision is, as a rule, not identical with that for the force density calculation, which is performed in the (ξ, η, ζ) space. For this purpose one needs the $\vec{f}(\xi_i, \eta_j, \zeta_k)$ for the 27 mesh points. The corresponding values for the mesh points known in the (ξ, η, ζ) space are calculated according to (18):

$$X_i = \sum_{\ell=1}^{20} N_\ell \; (\xi_i, \eta_j, \zeta_k) \; X_\ell,$$

$$Y_j = \sum_{\ell=1}^{20} N_\ell \; (\xi_i, \eta_j, \zeta_k) \; Y_\ell, \tag{18}$$

$$Z_k = \sum_{\ell=1}^{20} N_\ell \; (\xi_i, \eta_j, \zeta_k) \; Z_\ell,$$

and also the volume forces relating to these points. As described in Sec. 3.2, the spline coefficients for the planes E_m between $P_{1,m}$ and $P_{2,m}$ (m = 1(1)L) are calculated (Fig. 6). For the interpolation of the volume forces $\vec{f}_p$ to the point $P_o(X_o,Y_o,Z_o)$ first the adjacent planes E_{m-1} and E_m are determined. For this purpose the equations of the straight lines (19) $g_{1,m}$, $g_{2,m}$, $g_{3,m}$ and $g_{4,m}$ (Fig. 6b) are formed from the points $P_{1,m-1}$, $P_{1,m}$, $P_{2,m-1}$ and $P_{2,m}$ and the coordinates X_o and Z_o of point P_o are inserted in the equations of the straight lines:

$$g_{1,m} \equiv (X-X_{1,m-1})\ (Z_{1,m-1}-Z_{2,m-1})+(S_{2,m-1}-X_{1,m-1})\ (Z-Z_{1,m-2}),$$

$$g_{2,m} \equiv (X-X_{1,m})\ (Z_{1,m}-Z_{2,m})+(X_{2,m}-X_{1,m})\ (Z-Z_{1,m}),$$

$$g_{3,m} \equiv (X-X_{1,m-1})(Z_{1,m-1}-Z_{1,m})+(X_{1,m}-X_{1,m-1})(Z-Z_{1,m-1}) \qquad (19)$$

$$g_{4,m} \equiv (X-X_{2,m-1})\ (Z_{2,m-1}-Z_{2,m})+(X_{2,m}-X_{2,m-1})(Z-Z_{2,m-1})$$

for m = 2(1) L.

The function values of $g_{1,m}\ (X_o,Z_o)$ and $g_{2,m}(X_o,Z_o)$ are then of opposite sign if the point $P_o(X_o,Y_o,Z_o)$ is located between the two straight lines. This is equally valid for $g_{3,m(Xo,Zo)}$ and $g_{4,m}(X_o,Z_o)$ and leads to the condition that the point $P_o(X_o,Y_o,Z_o)$ is located in the region which is bounded by the four straight lines when

$g_{1,m}(X_o,Z_o) \cdot g_{2,m}(X_o,Z_o) \leq 0$ and $g_{3,m}(X_o,Z_o) \cdot g_{4,m}(X_o,Z_o) \leq 0$. The value U required for the spline interpolation (see Fig. 5c) is obtained by drawing a parallel to $g_{3,m}$ or $g_{4,m}$ through the point $P_o(X_o,Y_o,Z_o)$ and determining the intersection with $g_{1,m}$ or $g_{2,m}$, respectively. As can be seen in Fig. 6b, it is possible to calculate U from d_3 and d_4. To calculate d_3 and d_4, one uses the formula (20) for the distance d of a point (X,Z) from a straight line given by two points (X_1,Z_1) and (X_2,Z_2):

$$d = \frac{(Z-Z_1)\ (X_1-X_2)\ +\ (X-X_2)\ (Z_1-Z_2)}{(X_1-X_2)^2\ +\ (Z_1-Z_2)^2} \qquad (20)$$

With eq. (20) the values $d_3 = \dfrac{-g_{3,m}(X_o,Z_o)}{S_3}$ and

$$d_4 = \frac{g_{4,m}(X_o,Z_o)}{S_4}$$ can be calculated with

$$S_3 = \sqrt{(X_{1,m-1}-X_{1,m})^2 + (Z_{1,m-1}-Z_{1,m})^2}$$

and

$$S_4 = \sqrt{(X_{2,m-1}-X_{3,m})^2 + (Z_{2,m-1}-Z_{2,m})^2} .$$

One then obtains for U

$$U = \frac{B \cdot d_3}{d_3+d_4} = \frac{B \cdot S_4 \cdot g_{3,m}}{S_4 \cdot g_{3,m} - S_3 \cdot g_{4,m}} . \tag{21}$$

For the planes E_m and E_{m-1} U is equal because the coil segment (Fig. 6b) is symmetric to the axis $\overline{AC}$. This is because in the HEDO2 program the contour of the coil is approximated with arcs. As can be seen in cross-section $\overline{GH}$ (Fig. 6c), the point $P_o(X_o,Y_o,Z_o)$ is imaged onto the point $\bar{P}$ (U,V) in the plane E_m. The mesh point U_{LX}, V_{LY} for point $\bar{P}$ is determined for the spline interpolation. The volume force $\vec{f}_{\bar{P},m}$ for the plane E_m is calculated with the 2-dimensional spline function:

$$\vec{f}_{\bar{P},m} = \sum_{i=1}^{4} \sum_{j=1}^{4} C_{i,j,LX,LY,m} (U-U_{LX})^{i-1} - (V-V_{LY})^{j-1} \tag{22}$$

where $C_{i,j,LX,LY,m}$ are the spline coefficients for the plane E_m. In the same way $\vec{f}_{\bar{P},m-1}$ is calculated for the plane E_{m-1}. The values $\vec{f}_{\bar{P},m}$ and $f_{\bar{P},m-1}$ obtained are linearly interpolated to the point $P_o(X_o,Y_o,Z_o)$ according to

$$\vec{f}_{Po} = \frac{\vec{f}_{\bar{P},m} \cdot d_1 + \vec{f}_{\bar{P},m-1} \cdot d_2}{d_1 + d_2} = \frac{\vec{f}_{\bar{P},m} \cdot g_{1,m} - \vec{f}_{\bar{P},m-1} \cdot g_{2,m}}{g_{1,m} - g_{2,m}} \tag{23}$$

where the quantities $d_1 = \frac{g_{1,m}}{B}$ and $d_2 = - \frac{g_{2,m}}{B}$ (Fig. 6b) are replaced by the values $g_{1,m}(X_o, Z_o)$ and $g_{2,m}(X_o, Z_o)$ (d_1 and d_2 are obtained according to eq. (20)).

ACKNOWLEDGEMENT

The authors are grateful to Dr. Söll for many helpful suggestions during the preparation of this report.

FIGURE CAPTIONS

Fig. 1 Calculation scheme

Fig. 2 Node numbering of a three-dimensional solid element

a) Basic shape of a finite element (20 nodes)

b) Representation in Cartesian coordinates

Fig. 3 Subdivision of a structure into finite Elements

• nodes connected to one element

o " " " two elements

□ " " " four "

■ " " " eight "

Fig. 4 Scheme of a D coil subdivided for the force calculations with HEDO2. Output locations for the magnetic forces are shown by dots.

a) Segmentation of the coil along the circumference

b) " of the winding

c) Volume element ΔV with the force densities g_x, g_y and g_z.

Fig. 5 Scheme of a D coil with the subdivision for the calculation of the spline coefficients

a) Segmentation of the coil along the circumference

b) View $\overline{GH}$

+ At this points spline coefficients are computed. Shaded areas have the same spline functions.

• Points without spline coefficients.

Fig. 6 Scheme of a D coil showing the subdivision for interpolation the volume forces $\overset{*}{f}(X,Y,Z)$ at point $P_o(X_o,Y_o,Z_o)$

a) Segmentation of the coil along the circumference

b) Interpolation in a segment of the coil

c) The spline coefficients to (U_{LX}, V_{LY}) are valid for the shaded area.

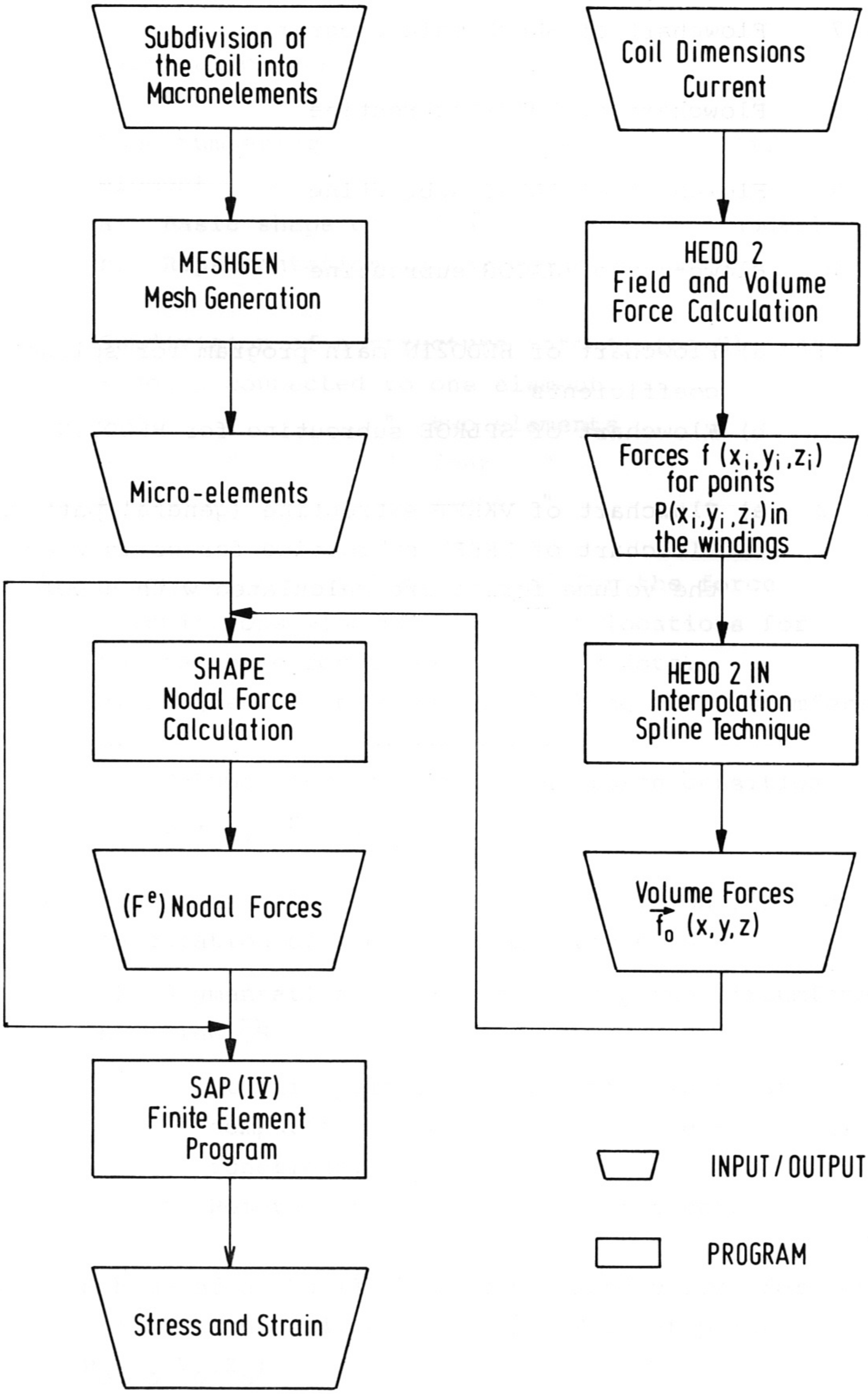
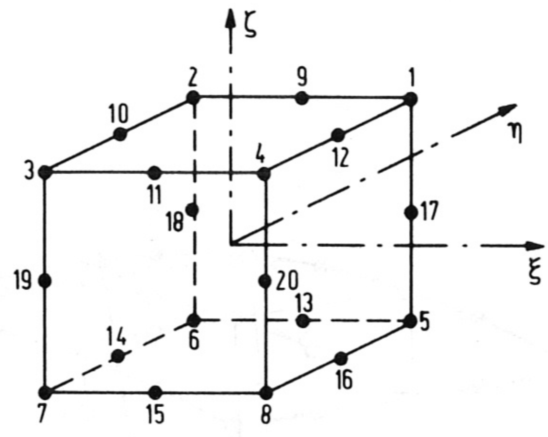
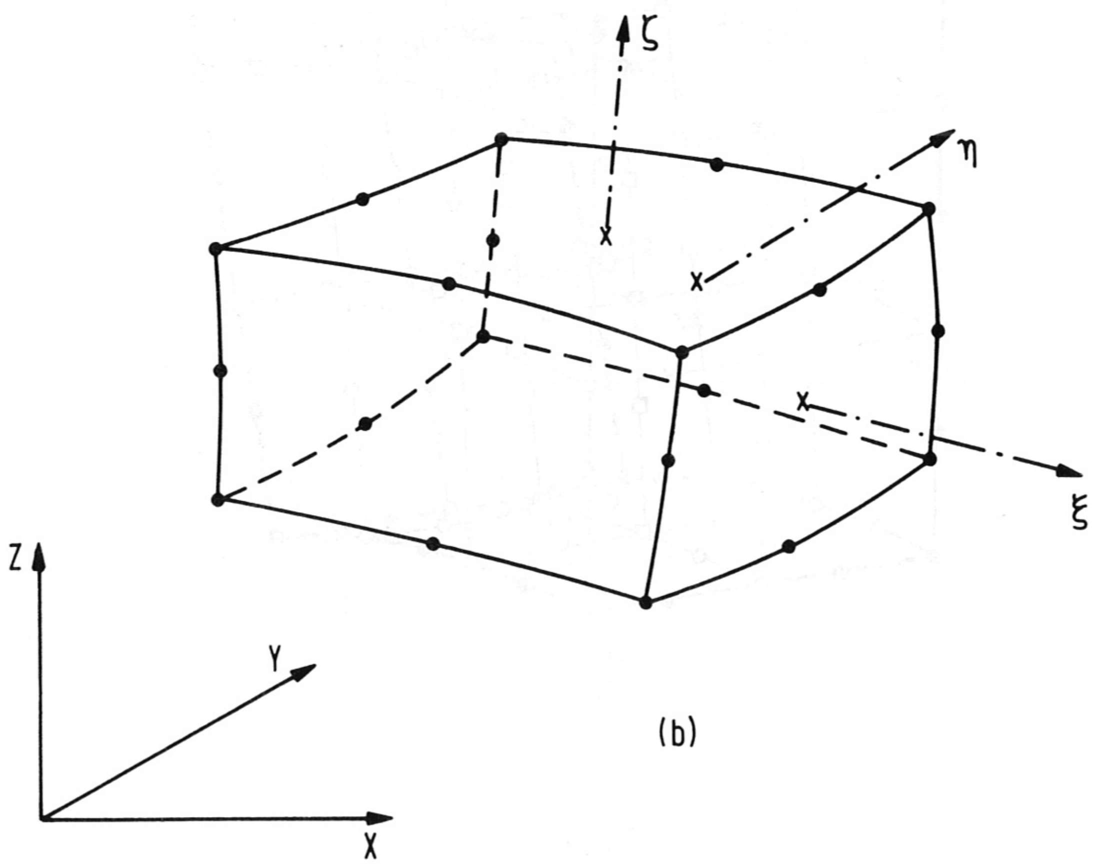


Fig. 1 Calculation scheme



(a)



(b)

Fig. 2

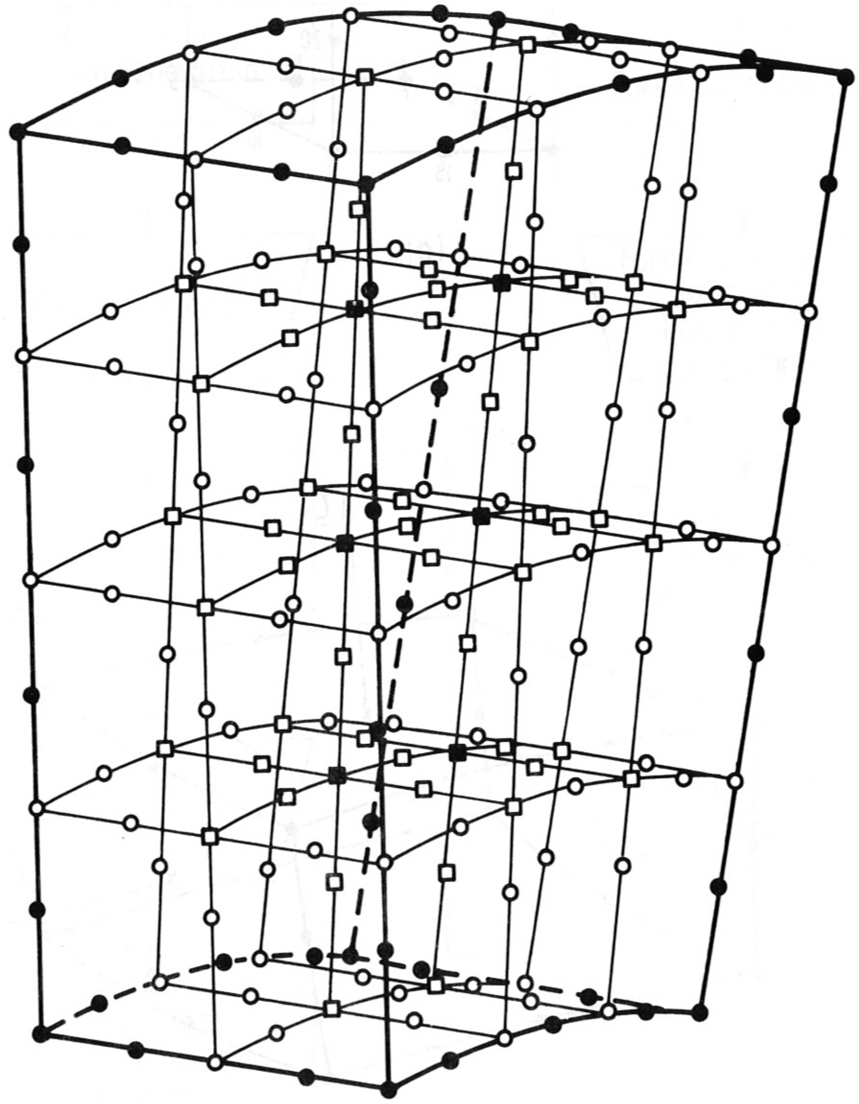


Fig. 3

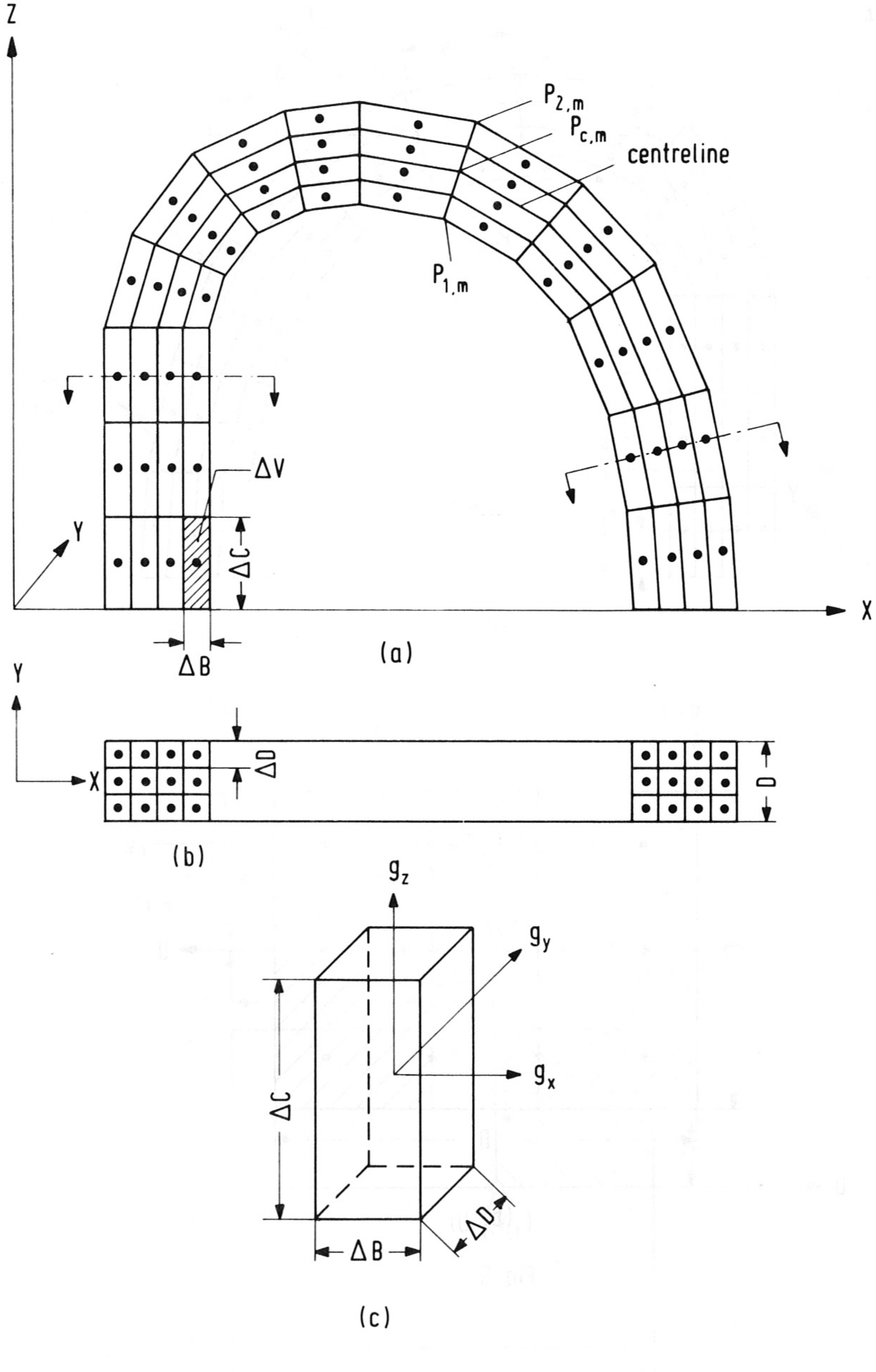
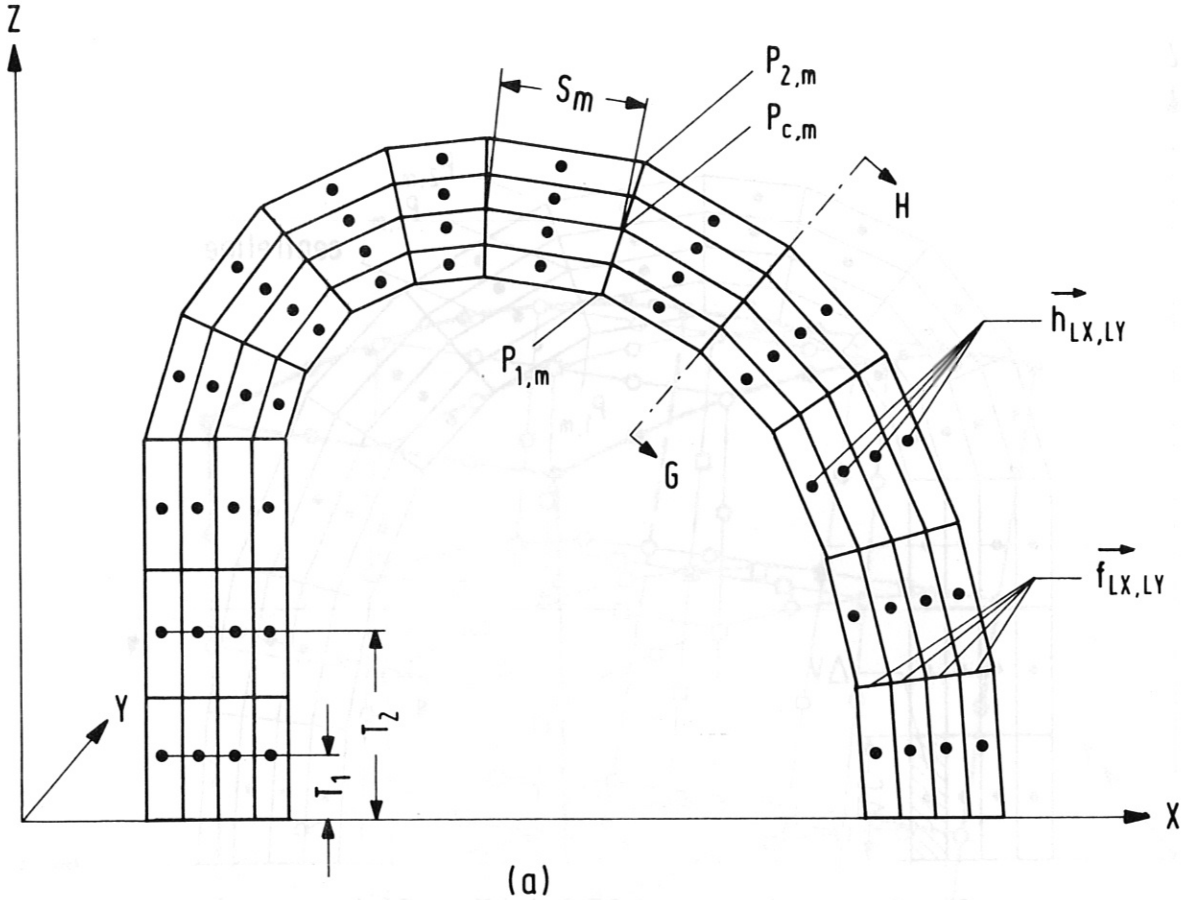
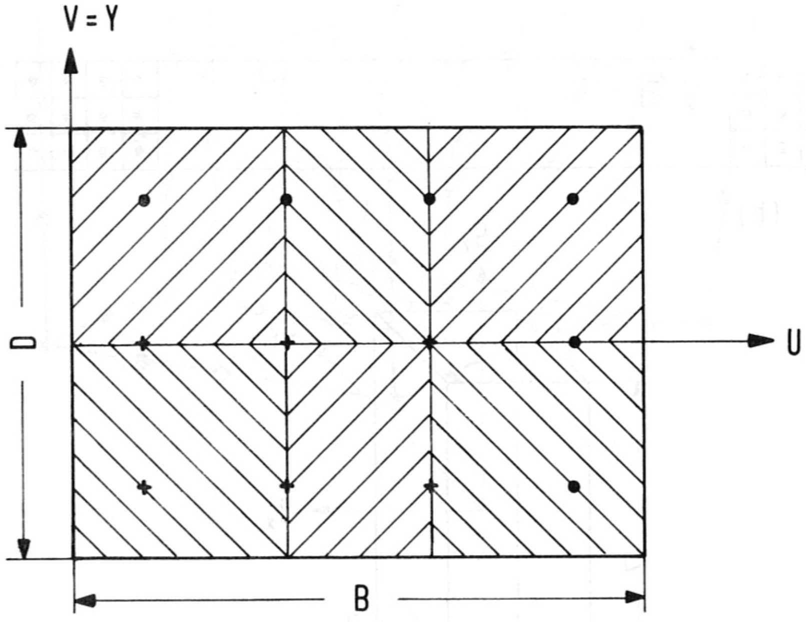


Fig. 4



(a)



(b)

Fig. 5

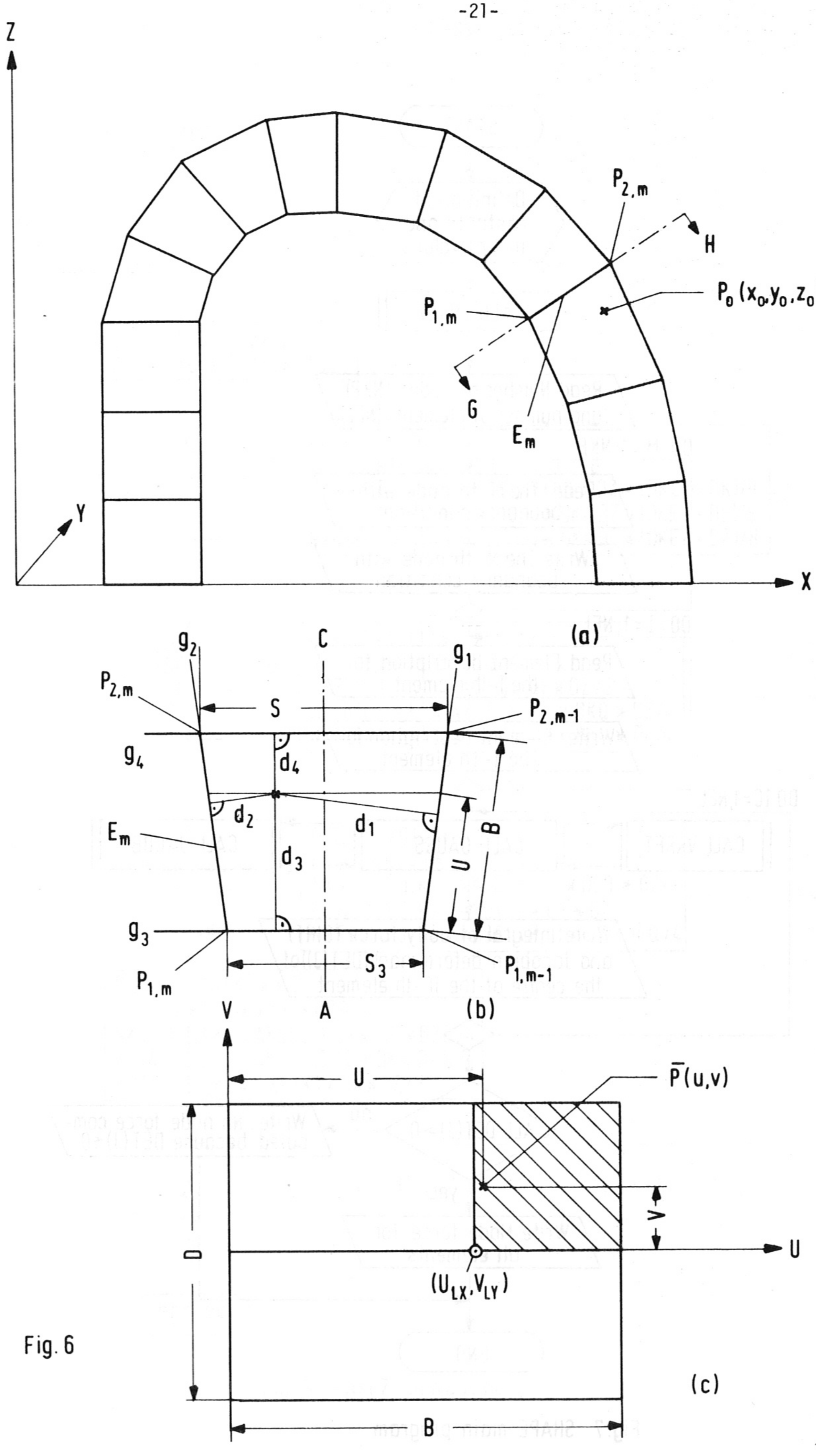

Z
Y
X
$P_{2,m}$
H
$P_0(x_0, y_0, z_0)$
$P_{1,m}$
G
E_m
(a)
g_2
C
g_1
$P_{2,m}$
S
$P_{2,m-1}$
g_4
d_4
d_2
d_1
E_m
d_3
U
B
g_3
$P_{1,m}$
S_3
$P_{1,m-1}$
V
A
(b)
U
$\bar{P}(u,v)$
D
V
U
(U_{LX}, V_{LY})
(c)
B

Fig. 6

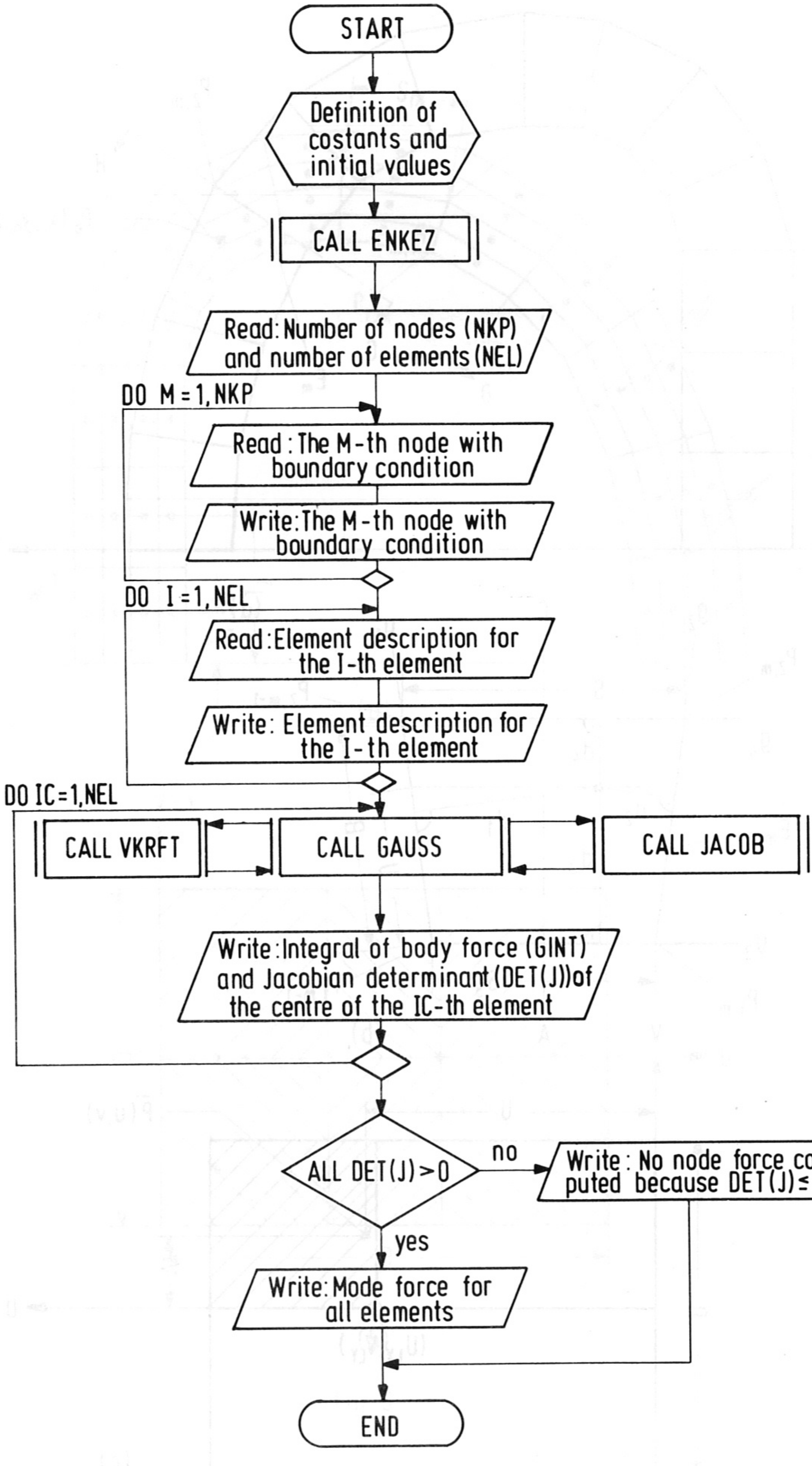


Fig. 7 SHAPE main program

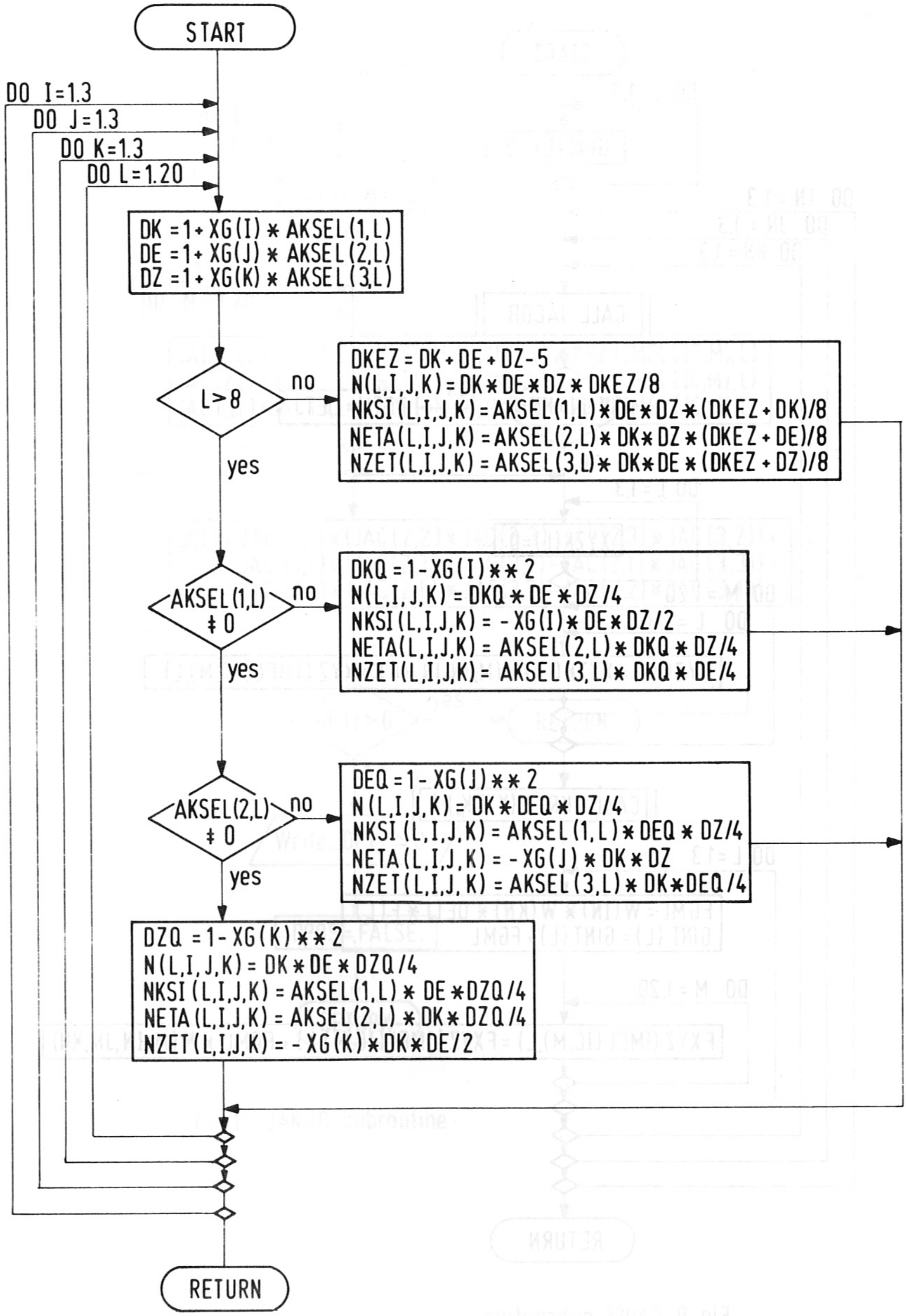


Fig. 8 ENKEZ subroutine

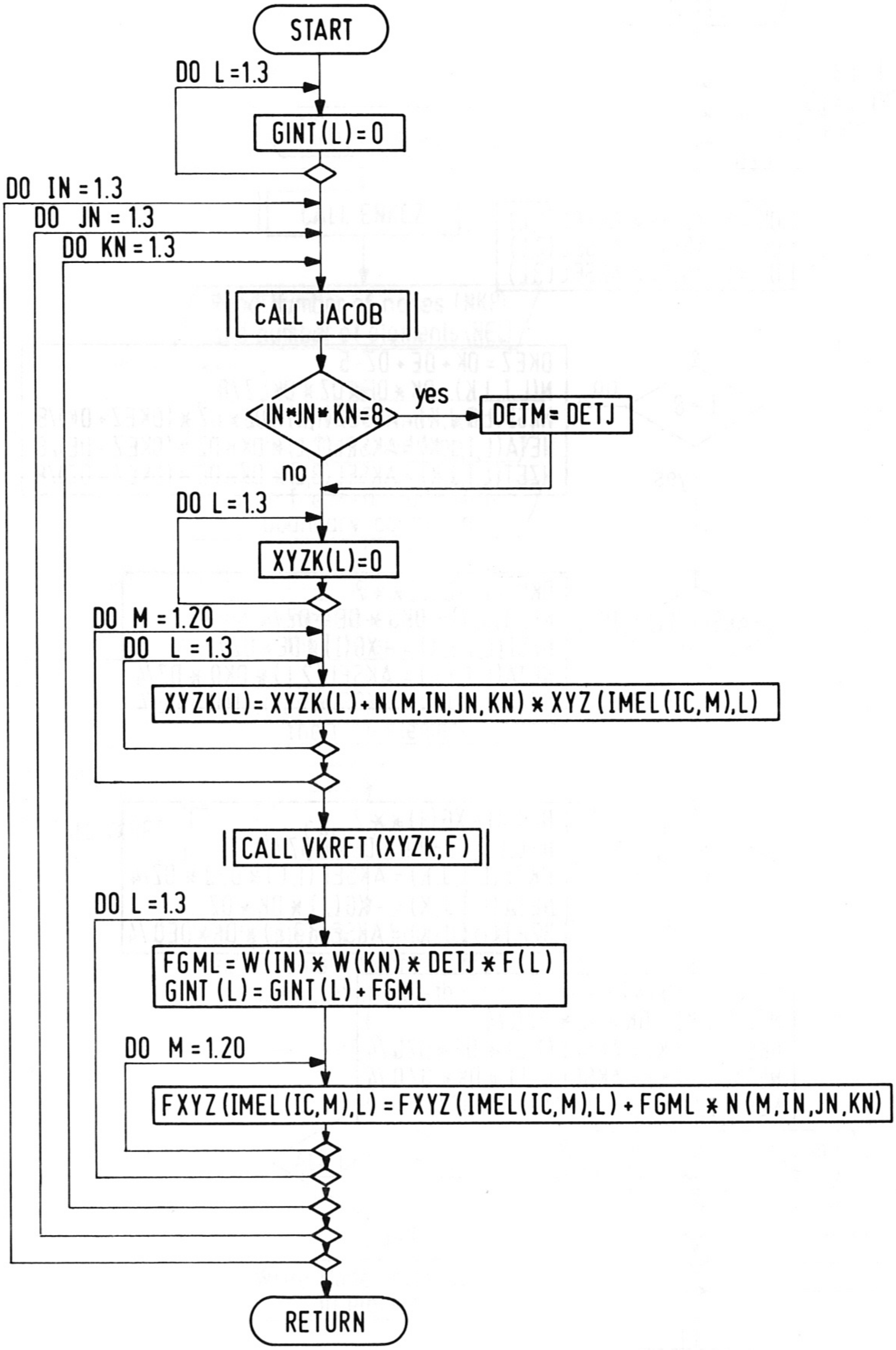


Fig. 9 GAUSS subroutine

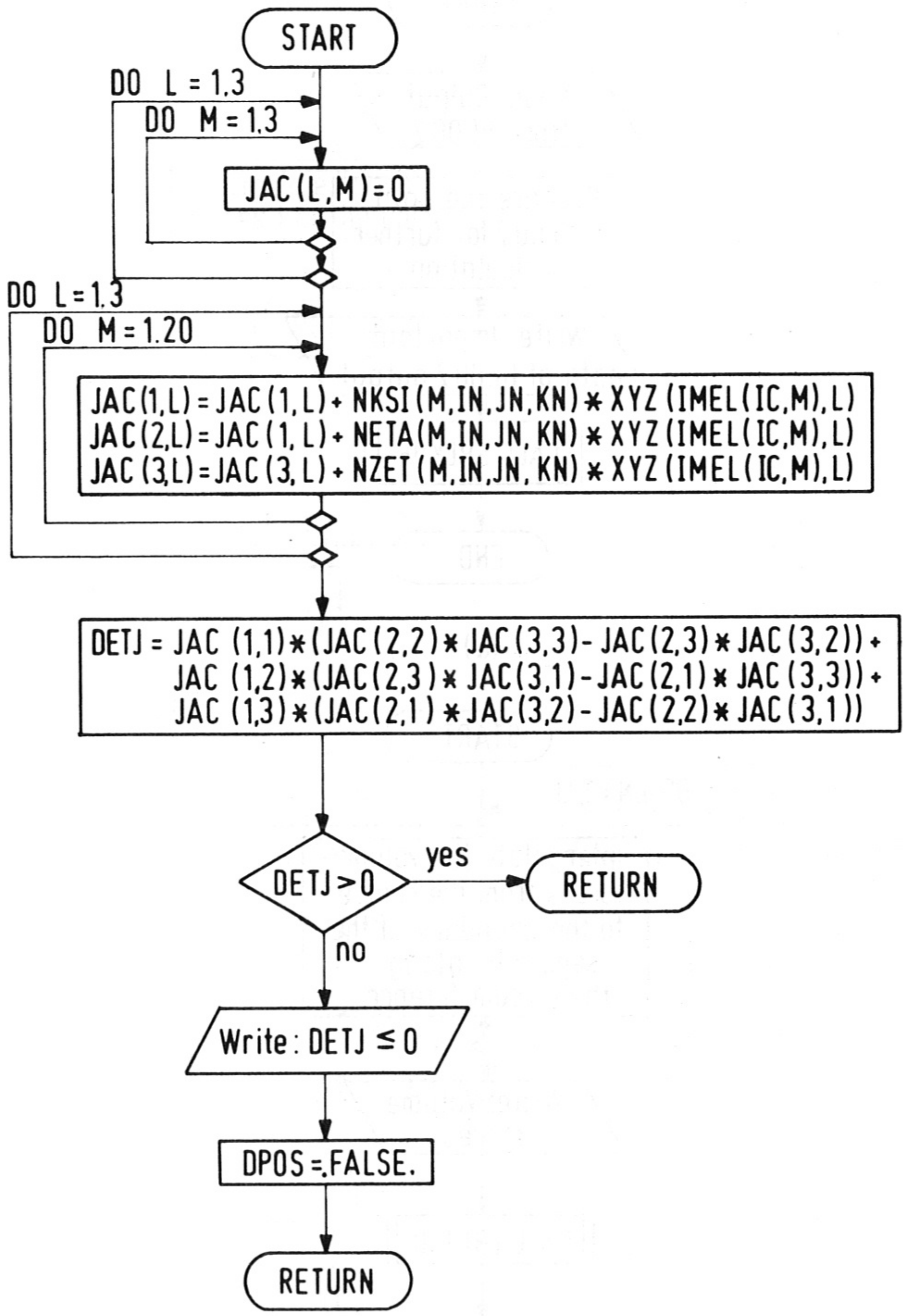
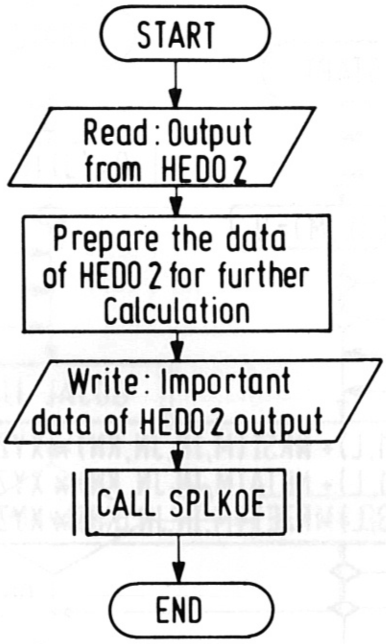
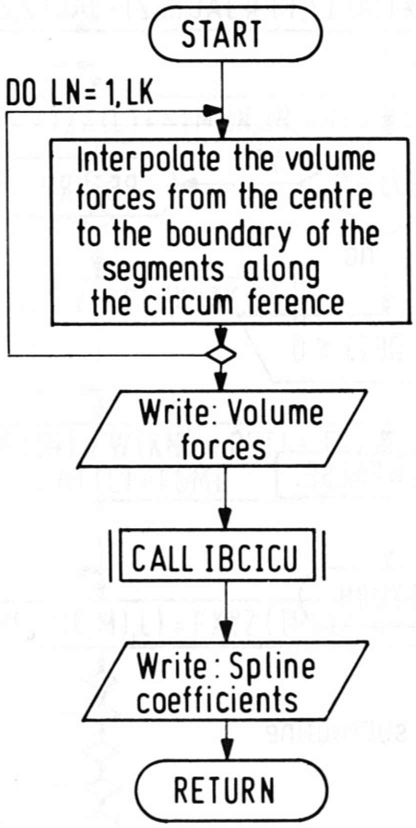


Fig.10 JAKOB subroutine

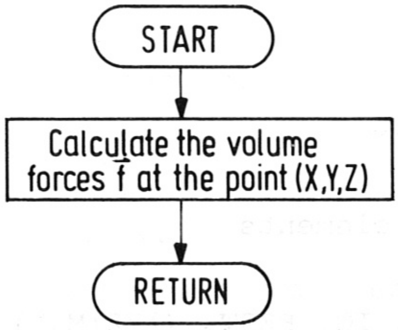


(a)

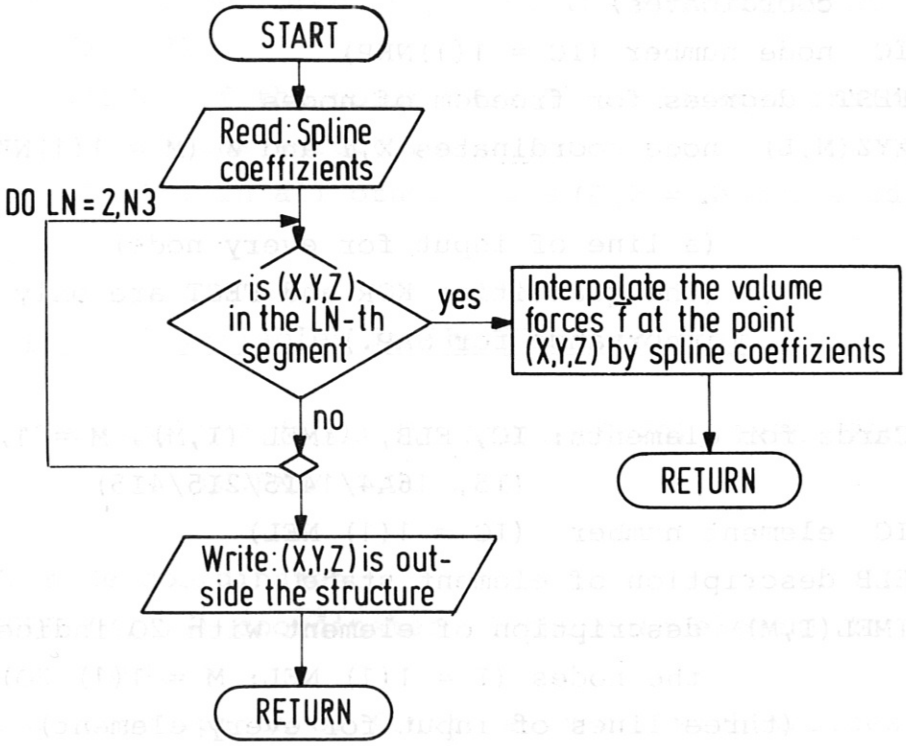


(b)

Fig. 11 (a) HEDO 2 IN program
(b) SPLKOE subroutine



(a)



(b)

Fig. 12 VKRFT subroutine

4. Appendix

4.1 Input for SHAPE

1) Control card: NKP, NEL (10I5)

NKP number of nodes

NEL number of finite elements

2) Cards for nodes: KOR, IC, FEST, (XYZ(M,L), L=1,3)
(A1, I4, 6I5, 3F10.5)

KOR (C = cylindrical coordinates, blank = Cartesian coordinates)

IC node number (IC = 1(1)NKP)

FEST degrees for freedom of nodes

XYZ(M,L) node coordinates X,Y and Z (M = 1(1)NKP; L = 1,3)

(a line of input for every node)

(The quantities KOR and FEST are only of importance for SAP.)

3) Cards for elements: IC, ELB, (IMEL (I,M), M = 1,20)
(I5, 16A4/14I5/2I5/4I5)

IC element number (IC = 1(1) NEL)

ELB description of element state

IMEL(I,M) description of element with 20 indices of the nodes (I = 1(1) NEL; M = 1(1) 20)

(three lines of input for every element)

(The quantity ELB is only of importance for SAP.)

4) In calculating coils according to Sec. 3 the spline coefficients computed with HEDO2IN are entered at logical unit number 10.

(Input for HEDO2IN: Results from HEDO2 at logical unit number 9.

Output from HEDO2IN: Spline coefficients calculated with IMSL-subroutine IBCICU).

In calculating other problems, the subroutine VKRFT (XZY, F, NORMAL) must be changed according to the problem.

Meaning of the variables:

XYZ(1) = X-coordinate
XYZ(2) = Y-coordinate
XYZ(3) = Z-coordinate
F(1) = F_X-volume force component in X-direction at (X,Y,Z)
F(2) = F_Y-volume force component in Y-direction at (X,Y,Z)
F(3) = F_Z-volume force component in Z-direction at (X,Y,Z)
NORMAL = .TRUE. if the point (X,Y,Z) is located inside the structure
.FALSE. in all other cases (see chapter 4.3d).

4.2 Important variables in the SHAPE program

a) COMMON /AA/ XYZ (2000,3), AKSEL (3,20), IMEL (250,20), IC,FXYZ (2000,3)

XYZ node coordinates
AKSEL (ξ, η, ζ) coordinates of the unit element in Table 1
IMEL description of finite elements with indices of the 20 nodes
IC index of the element
FXYZ components of the nodal forces

b) COMMON /GAUSP/ XG(3), N(20,3,3,3), NKSI (20,3,3,3), NETA (20,3,3,3), NZET (20,3,3,3), W (3), IN, JN, KN, GINT(3), DETJ, DETM, DPOS

XG(3) mesh points for Gauss integration (-0.7745966, 0,0.7745966)
N shape functions for Gauss integration

NKSJ= $\frac{\partial N}{\partial \xi}$ partial derivatives for Gauss integration

NETA = $\frac{\partial N}{\partial \eta}$ partial derivatives for Gauss integration

NZET = $\frac{\partial N}{\partial \eta}$ " " " " "

W weightings " " " $(\frac{5}{9}, \frac{8}{9}, \frac{5}{9})$

IN, JN, KN indices for Jacobian determinant

GINT = $\vec{F}$ integral of the volume forces for the IC element

DETJ Jacobian determinant for ξ = XG(IN), η = XG(JN), ζ = XG(KN) for the IC element

DETM Jacobian determinant for $\xi = \eta = \zeta = 0$ for the IC element

DPOS is .TRUE. when all DETJ > 0, otherwise .FALSE.

4.3 Output SHAPE

1) Control output of the node coordinates, specifying degrees of freedom

2) Control output of the description of elements (20 indices of the nodes for each element)

3) In computations of coils according to Sec. 3 it is the contour of the coil that is given as output

4) Forces F_x, F_y and F_z at the centre of each finite element and its Jacobian determinant for $\xi = \eta = \zeta = 0$.

5) Node forces FX, FY, FZ computed for each element according to shape functions

6) Additional output of the nodal forces to the logic unit number 7 (standard punch unit) in card image format. This output is for the input in the SAP program.

The following statements in the output should be noted:

a) *** ERROR IN THE NUMBERING ***

This call indicates that either the nodes or the finite elements in the input are not consecutively numbered.

b) *** NO NODE FORCE COMPUTED BECAUSE JABOCIAN DET < = 0***

After this statement the program is terminated without output of nodal forces.

c) *** ELEMENT NO. .... JACOBIAN-DET (KSJ=..., ETA=..., ZETA=...) = ...***

This statement indicates the element in which a Jacobian determinant $\leq$ 0 occurred. The computation is continued for checking purposes, but without computing nodal forces.

d) *** XYZ = ... IS OUTSIDE THE STRUCTURE ***

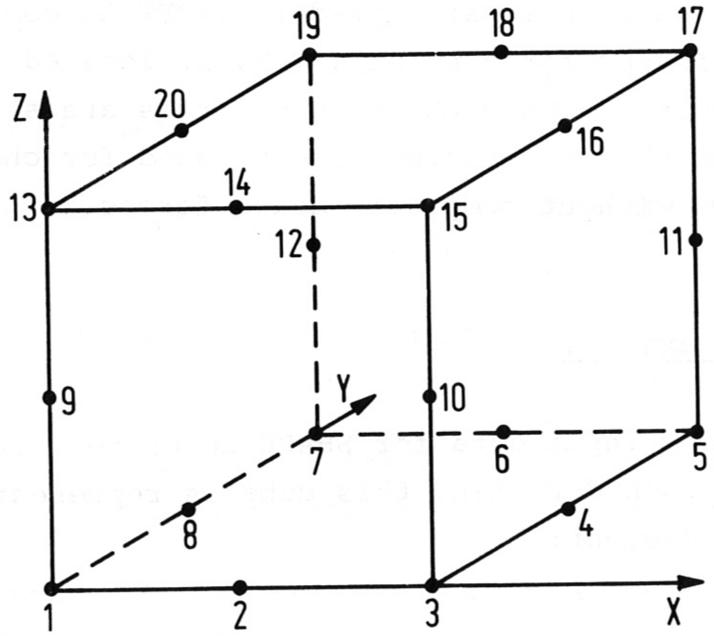
In this case it is attempted in VKRFT to compute the volume forces for a point (X, Y, Z) located outside the structure. Here the volume forces are set equal to zero and the computation is continued for checking purposes without computing nodal forces.

4.4 Input example

The following input data for SHAPE describe a cube with 2 units in each directin; this cube is represented by one finite element:

```
node number                                x-coord.     y-coord.      z-coord.
20   1
 1   0    0    0    0    0    0   0.0000000   0.0000000    0.00000000
 2   0    0    0    0    0    0   1.0000000   0.0000000    0.00000000
 3   0    0    0    0    0    0   2.0000000   0.0000000    0.00000000
 4   0    0    0    0    0    0   2.0000000   1.0000000    0.00000000
 5   0    0    0    0    0    0   2.0000000   2.0000000    0.00000000
 6   0    0    0    0    0    0   1.0000000   2.0000000    0.00000000
 7   0    0    0    0    0    0   0.0000000   2.0000000    0.00000000
 8   0    0    0    0    0    0   0.0000000   1.0000000    0.00000000
 9   0    0    0    0    0    0   0.0000000   0.0000000    1.00000000
10   0    0    0    0    0    0   2.0000000   0.0000000    1.00000000
11   0    0    0    0    0    0   2.0000000   2.0000000    1.00000000
12   0    0    0    0    0    0   0.0000000   2.0000000    1.00000000
13   0    0    0    0    0    0   0.0000000   0.0000000    2.00000000
14   0    0    0    0    0    0   1.0000000   0.0000000    2.00000000
15   0    0    0    0    0    0   2.0000000   0.0000000    2.00000000
16   0    0    0    0    0    0   2.0000000   1.0000000    2.00000000
17   0    0    0    0    0    0   2.0000000   2.0000000    2.00000000
18   0    0    0    0    0    0   1.0000000   2.0000000    2.00000000
19   0    0    0    0    0    0   0.0000000   2.0000000    2.00000000
20   0    0    0    0    0    0   0.0000000   1.0000000    2.00000000
 1             1         1
17   19   13   15   5    7    1      3   18     20   14      16      6   8
 2    4
11   12    9   10
```

The last four lines represent the element description according to the numbering sequence of SAP IV



Note numbers for input example

4.5 List of the Program SHAPE

```
C     CALCULATION OF NODAL FORCES ACCORDING TO SHAPE FUNCTION
C     NECESSARY SUBROUTINES :
C        ENKEZ   : CALCULATION OF SHAPE FUNCTIONS AND THEIR DERIVATIVES
C                  AT 27 MESH POINTS
C        GAUSS   : INTEGRATION OF FORCES ACCORDING TO GAUSS BY USING 27
C                  MESH POINTS
C        JACOB   : CALCULATION OF JACOBIAN DETERMINANT FOR FINITE ELEMENTS
C        VKRFT(XYZ,F,NORMAL) : VOLUME FORCES FUNCTION
C              THE ARGUMENTS ARE :
C     XYZ(1) = X - COORDINATE    AND F(1) = VOLUME FORCE IN X - DIRECTION
C     XYZ(2) = Y - COORDINATE    AND F(2) = VOLUME FORCE IN Y - DIRECTION
C     XYZ(3) = Z - COORDINATE    AND F(3) = VOLUME FORCE IN Z - DIRECTION
C     NORMAL = .TRUE. IF THE POINT (X,Y,Z) IS INSIDE THE STRUCTURE
C            = .FALSE. IN ALL OTHER CASES
C     INPUT :
C     1. LINE :   NKP,NEL                                        (10I5)
C                 NKP = NUMBER OF NODES
C                 NEL = NUMBER OF FINITE ELEMENTS
C     2. LINE :   KOR,IC,FEST,(XYZ(M,L),L=1,3)        (A1,I4,6I5,3F10.5)
C                 KOR = (C=CYLINDRICAL   ,BLANK=CARTESIAN COORDINATES)
C                 IC   = NODE NUMBER   (IC=1(1)NKP)
C                 FEST(I) = DEGREES OF FREEDOM OF NODES       (I=1(1)6)
C                 XYZ(M,L)= X,Y,Z - NODE  COORDINATES (M=1,NKP)
C     3. LINE :   IC,ELB,(IMEL(I,M),M=1,20)         (I5,16A4/14I5/2I5/4I5)
C                 IC   = ELEMENT NUMBER
C                 ELB(M)   = DESCRIPTION OF ELEMENT STATE
C                 IMEL(I,M)= DESCRIPTION OF ELEMENT WITH 20 INDICES OF
C                            THE NODES (I=1(1)NEL ; M=1(1)20)
C     OUTPUT :
C     1.         NODE COORDINATES WITH DEGREE OF FREEDOM
C     2.         ELEMENT DESCRIPTION (20 NODE INDICES FOR EACH ELEMENT)
C     3.         FORCES FX,FY,FZ AT CENTRE OF EACH ELEMENT AND ITS
C                JACOBIAN DETERMINANT FOR KSI=ETA=ZETA=0
C     4.         NODAL FORCES FX,FY,FZ ACCORDING TO SHAPE FUNCTIONS
C     5.         ADDITIONAL OUTPUT OF NODAL FORCES TO
C                 G.FT07F001 DD DCB=(RECFM=FB,LRECL=80,BLKSIZE=2240)
      COMMON /AA/XYZ(2000,3),AKSEL(3,20),IMEL(250,20),IC,FXYZ(2000,3)
      REAL*4  N,NKSI,NETA,NZET,ELB(16),SUM(3)/3*0.0/
      COMMON /GAUSSP/ XG(3),N(20,3,3,3),NKSI(20,3,3,3),NETA(20,3,3,3),
     >        NZET(20,3,3,3),W(3),IN,JN,KN,GINT(3),DETJ,DETM,DPOS
      INTEGER   FEST(6)
      LOGICAL DPOS
      CALL ENKEZ
      READ(5,102) NKP,NEL
      WRITE(6,204) NKP
      DO 1 M=1,NKP
      READ(5,100) KOR, IC,FEST,(XYZ(M,L),L=1,3)
      IF(IC.NE.M) WRITE(6,205)
    1 WRITE(6,200) KOR,IC,FEST,(XYZ(M,L),L=1,3)
      WRITE(6,206) NEL
      DO 2 I =1,NEL
      READ(5,101)IC,ELB,(IMEL(I ,M),M=1,20)
      IF(IC.NE.I) WRITE(6,205)
    2 WRITE(6,203)  I,IC,ELB,(IMEL(I ,M),M=1,20)
      DO 3 IC=1,NEL
      CALL GAUSS
      DO 5 L=1,3
```

```
    5 SUM(L)=SUM(L)+GINT(L)
      IF(IC.EQ.1) WRITE(6,202)
    3 WRITE(6,201) IC,GINT,DETM
      WRITE(6,208) SUM
      IF(DPOS)  GOTO  4
      WRITE(6,301)
      WRITE(7,301)
      STOP
    4 WRITE(6,207)
      DO 6 IC=1,NKP
      WRITE(6,302) IC,(FXYZ(IC,L),L=1,3)
    6 WRITE(7,302) IC,(FXYZ(IC,L),L=1,3)
      STOP
  100 FORMAT(A1,I4,6I5,3F10.5)
  101 FORMAT(I5,16A4/ 14I5/2I5/4I5)
  102 FORMAT(10I5)
  200 FORMAT(1XA1,I4,6I5,1P3E13.5)
  201 FORMAT(1XI6,1P4E13.5)
  202 FORMAT('1ELEMENT'16X'FORCES'17X'JACOBIAN DET.'/4X'NO.'5X'FX'11X
     >        'FY'11X'FZ'10X,'(0,0,0)'//)
  203 FORMAT(I5,' :'/10XI5,16A4/10X20I5)
  204 FORMAT('1  NKP ='I6,'   NODES'/'0  NO.'7X'DEGREES OF FREEDOM'15X'X'
     >        12X'Y'12X'Z'//)
  205 FORMAT('0**** ERROR IN THE NUMBERING ***')
  206 FORMAT('1ELEMENT DESCRIPTION  NEL ='I6/' 1. LINE : ELEMENT NUMBER
     >AND ELEMENT STATE'/' 2. LINE : DESCRIPTION OF ELEMENT BY 20 NODE I
     >NDICES'/'0  NO.'//)
  207 FORMAT('1NODAL FORCES ACCORDING TO SHAPE FUNCTIONS'/
     >    '0  NO. LOAD     FX'8X'FY'8X'FZ'//)
  208 FORMAT('0SUM :  '1P3E13.5)
  301 FORMAT('1*** NO NODAL FORCES COMPUTED BECAUSE JACOBIAN DET. <= 0 *
     >**')
  302 FORMAT(I5,4X'1'1P3E10.3)
      END
C   CALCULATION OF SHAPE FUNCTIONS N AND THEIR PARTIAL DERIVATIVES
C   NKSI,NETA,NZET AT 27 MESH POINTS FOR GAUSS INTEGRATION
      SUBROUTINE ENKEZ
      COMMON /GAUSSP/ XG(3),N(20,3,3,3),NKSI(20,3,3,3),NETA(20,3,3,3),
     >                NZET(20,3,3,3),W(3),IN,JN,KN,GINT(3),DETJ,DETM,DPOS
      COMMON /AA/XYZ(2000,3),AKSEL(3,20),IMEL(250,20),IC,FXYZ(2000,3)
      REAL*4  N,NKSI,NETA,NZET
      LOGICAL DPOS
      DO 1 I=1,3
      DO 1 J=1,3
      DO 1 K=1,3
      DO 1 L=1,20
      DK=1.0+XG(I)*AKSEL(1,L)
      DE=1.0+XG(J)*AKSEL(2,L)
      DZ=1.0+XG(K)*AKSEL(3,L)
      IF(L.GT.8)  GOTO  2
      DKEZ=DK+DE+DZ-5.0
      N(L,I,J,K)=0.125*DK*DE*DZ*DKEZ
      NKSI(L,I,J,K)=0.125*AKSEL(1,L)*DE*DZ*(DKEZ+DK)
      NETA(L,I,J,K)=0.125*AKSEL(2,L)*DK*DZ*(DKEZ+DE)
      NZET(L,I,J,K)=0.125*AKSEL(3,L)*DE*DK*(DKEZ+DZ)
      GOTO  1
    2 IF(AKSEL(1,L).NE.0)  GOTO   3
```

```
      DKQ=1.0-XG(I)**2
      N(L,I,J,K)=0.25*DKQ*DE*DZ
      NKSI(L,I,J,K)=-0.5*XG(I)*DE*DZ
      NETA(L,I,J,K)=0.25*AKSEL(2,L)*DKQ*DZ
      NZET(L,I,J,K)=0.25*AKSEL(3,L)*DKQ*DE
      GOTO  1
    3 IF(AKSEL(2,L).NE.0)   GOTO  4
      DEQ=1.0-XG(J)**2
      N(L,I,J,K)=0.25*DK*DEQ*DZ
      NKSI(L,I,J,K)=0.25*AKSEL(1,L)*DEQ*DZ
      NETA(L,I,J,K)=-0.5*XG(J)*DK*DZ
      NZET(L,I,J,K)=0.25*AKSEL(3,L)*DK*DEQ
      GOTO  1
    4 DZQ=1.0-XG(K)**2
      N(L,I,J,K)=0.25*DK*DE*DZQ
      NKSI(L,I,J,K)=0.25*AKSEL(1,L)*DE*DZQ
      NETA(L,I,J,K)=0.25*AKSEL(2,L)*DK*DZQ
      NZET(L,I,J,K)=-0.5*XG(K)*DK*DE
    1 CONTINUE
      RETURN
      END
C     INTEGRATION ACCORDING TO GAUSS WITH 27 MESH POINTS
      SUBROUTINE GAUSS
      COMMON /GAUSSP/ XG(3),N(20,3,3,3),NKSI(20,3,3,3),NETA(20,3,3,3),
     >       NZET(20,3,3,3),W(3),IN,JN,KN,GINT(3),DETJ,DETM,DPOS
      COMMON /AA/XYZ(2000,3),AKSEL(3,20),IMEL(250,20),IC,FXYZ(2000,3)
      REAL*4  N,NKSI,NETA,NZET,F(3),XYZK(3)
      LOGICAL DPOS
      DO 7 L=1,3
    7 GINT(L)=0.0
      DO 1 IN=1,3
      DO 1 JN=1,3
      DO 1 KN=1,3
      CALL  JACOB
      IF(IN*JN*KN.EQ.8) DETM=DETJ
      DO 6 L=1,3
    6 XYZK(L)=0.0
      DO 3 M=1,20
      DO 5 L=1,3
    5 XYZK(L)=XYZK(L)+N(M,IN,JN,KN)*XYZ(IMEL(IC,M),L)
    3 CONTINUE
      CALL  VKRFT(XYZK,F,DPOS)
      DO 4 L=1,3
      FGML=W(IN)*W(JN)*W(KN)*DETJ*F(L)
      GINT(L)=GINT(L)+FGML
      DO 4 M=1,20
      FXYZ(IMEL(IC,M),L)=FXYZ(IMEL(IC,M),L)+FGML*N(M,IN,JN,KN)
    4 CONTINUE
    1 CONTINUE
      RETURN
      END
C     CALCULATION OF JACOBIAN DETERMINANT FOR FINITE ELEMENTS
      SUBROUTINE JACOB
      COMMON /GAUSSP/ XG(3),N(20,3,3,3),NKSI(20,3,3,3),NETA(20,3,3,3),
     >       NZET(20,3,3,3),W(3),IN,JN,KN,GINT(3),DETJ,DETM,DPOS
      COMMON /AA/XYZ(2000,3),AKSEL(3,20),IMEL(250,20),IC,FXYZ(2000,3)
      REAL*4  N,NKSI,NETA,NZET,JAC(3,3)
```

```
      LOGICAL DPOS
      DO 2 L=1,3
      DO 2 M=1,3
    2 JAC(L,M)=0.0
      DO 1 L=1,3
      DO 1 M=1,20
      JAC(1,L)=JAC(1,L)+NKSI(M,IN,JN,KN)*XYZ(IMEL(IC,M),L)
      JAC(2,L)=JAC(2,L)+NETA(M,IN,JN,KN)*XYZ(IMEL(IC,M),L)
      JAC(3,L)=JAC(3,L)+NZET(M,IN,JN,KN)*XYZ(IMEL(IC,M),L)
    1 CONTINUE
      DETJ=JAC(1,1)*(JAC(2,2)*JAC(3,3)-JAC(2,3)*JAC(3,2))+
     >      JAC(1,2)*(JAC(2,3)*JAC(3,1)-JAC(2,1)*JAC(3,3))+
     >      JAC(1,3)*(JAC(2,1)*JAC(3,2)-JAC(2,2)*JAC(3,1))
      IF(DETJ.GT.0.0)  RETURN
      WRITE(6,200)  IC,XG(IN),XG(JN),XG(KN),DETJ
      DPOS=.FALSE.
      RETURN
  200 FORMAT(' *** ELEMENT NO.'I5,' JACOBIAN DET(KSI='F6.3,',ETA='F6.3,
     >  ',ZETA='F6.3,') ='1PE13.5,' ****')
      END
C     INITIAL VALUES AND CONSTANTS
      BLOCK DATA
      COMMON /GAUSSP/ XG(3),N(20,3,3,3),NKSI(20,3,3,3),NETA(20,3,3,3),
     >           NZET(20,3,3,3),W(3),IN,JN,KN,GINT(3),DETJ,DETM,DPOS
      COMMON /AA/XYZ(2000,3),AKSEL(3,20),IMEL(250,20),IC,FXYZ(2000,3)
      LOGICAL DPOS
      DATA XG/-.7749967,.0,.7749967/,W/.5555556,.8888889,.5555556/
      DATA GINT/3*0./,FXYZ/6000*0./,DPOS/.TRUE./
      DATA AKSEL/3*1.,-1.,2*1.,2*-1.,2*1.,-1.,3*1.,2*-1.,1.,4*-1.,1.,
     >2*-1.,0.,2*1.,-1.,0.,1.,0.,-1.,2*1.,0.,1.,0.,1.,2*-1.,0.,-1.,.0,
     >     2*-1.,1.,0.,-1.,2*1.,0.,-1.,1.,0.,2*-1.,0.,1.,-1.,0./
      END
```

```
C    THIS PROGRAM READS THE OUTPUT FROM HED02 ON G.FT09F001 AND
C    CALCULATES THE SPLINE COEFFICIENTS FOR INTERPOLATION OF VOLUME
C    FORCES IN PROGRAM SHAPE .
      COMMON /PNTS/PYY(8),LK,N1,N2,N3A,B,D,DELB,DELD,PX(2,61),PZ(2,61)
      COMMON /01/FGS(3,60,16,8)
      COMMON /MIT/ PXM(61),PZM(61),DELX(61),DELZ(61)
      INTEGER*2       NA(400),IZU(200),IZU2(200),IZU3(201),IZU4(200),
     >                IZU5(200),IZU6(200),KRM(200)
      COMMON /GNRL/ PXMB(3000),PZMB(3000),DELXB(3000),DELZB(3000),IKKT,
     >                LKS(200)
      COMMON /GLOB/ BB(200),DB(200),GB(200),WB(200),NRB(200),NYB(200)
C    READING THE OUTPUT FROM HED02
      READ (9) IKKT,KKT
      READ  (9)   NSP,IMIT,IZU3(KKT)
      DO 8 K=1,NSP
    8 READ  (9)   BB(K),DB(K),WB(K),GB(K),NRB(K),NYB(K)
      WRITE  (6,100)
      WRITE  (6,200)(K,BB(K),DB(K),NRB(K),NYB(K),WB(K),GB(K),K=1,NSP)
      DO 9 I=1,NSP
    9 READ  (9) IZU(I),IZU2(I)
      DO 6 I=1,IKKT
    6 READ  (9) IZU3(I),IZU4(I)
      DO 7 I=1,IMIT
    7 READ  (9) IZU5(I),IZU6(I),KRM(I),LKS(I)
      DO 1 K=1,IMIT
      KR=KRM(K)
      DO 3 J=1,KR
    3 READ (9)
      IZA=IZU3(K)+1
      IZE=IZU3(K+1)
      WRITE  (6,600)K,(IZU2(I),I=IZA,IZE)
    1 CONTINUE
      DO 2 K=1,NSP
    2 READ (9)
      DO 5 I=1,IKKT
      KK=IZU4(I)
      LK=LKS(KK)
      KKM=IZU6(KK)
      DO 5 J=1,LK
      IKK=J+KKM
    5 READ (9) PXMB(IKK),PZMB(IKK),DELXB(IKK),DELZB(IKK)
      READ (9)
      READ (9) IKR,N11,N3A,N1,N2,N3
  107 DO 18 I=1,N1
      DO 11 J=1,N2
      DO 13 IL=1,N3
   13 READ (9) DUM,PYY(I),(DUM,L=1,7),(FGS(L,IL,J,I),L=1,3)
   11 CONTINUE
      READ (9)
   18 CONTINUE
C    COMPLETE INTPUT DATA IF NECESSARY
      LK=LKS(IKR)
      IF(N1.EQ.N11)  GOTO 20
      N1P=N1+1
      DO 21 I=N1P,N11
      PYY(I)=-PYY(N11-I+1)
      DO 21 J=1,N2
```

```
      DO 21 IL=1,N3
      FGS(1,IL,J,I)=FGS(1,IL,J,N11-I+1)
      FGS(2,IL,J,I)=-FGS(2,IL,J,N11-I+1)
      FGS(3,IL,J,I)=FGS(3,IL,J,N11-I+1)
   21 CONTINUE
   20 N1=N11
      B=BB(IKR)
      D=DB(IKR)
      DELB=B/N2
      DELD=D/N1
      KKM=IZU6(IKR)
      DO 22 J=1,LK
      L=KKM+J
      PXM(J)=PXMB(L)
      PZM(J)=PZMB(L)
      DELX(J)=DELXB(L)*DELB
      DELZ(J)=DELZB(L)*DELB
      IF(N3.LT.LK) GOTO 22
      PXM(2*LK-J)=PXM(J)
      PZM(2*LK-J)=-PZM(J)
      DELX(2*LK-J)=DELX(J)
      DELZ(2*LK-J)=-DELZ(J)
   22 CONTINUE
      IF(N3.GT.LK) LK=2*LK-1
      DO 33 LN=1,LK
      DX=0.5*N2*DELX(LN)
      DZ=0.5*N2*DELZ(LN)
      PX(1,LN)=PXM(LN)-DX
      PX(2,LN)=PXM(LN)+DX
      PZ(1,LN)=PZM(LN)-DZ
      PZ(2,LN)=PZM(LN)+DZ
   33 CONTINUE
      WRITE(6,202)(I,PXM(I),PZM(I),DELX(I),DELZ(I),(PX(J,I),PZ(J,I),
     >     J=1,2),I=1,LK)
      WRITE(6,401) IKR
      CALL SPLKOE
      STOP
  100 FORMAT('1COIL DATA',/,1X,'---------',/'ORADIAL THICKNESS : WINDING
     > HEIGTH'/' AXIAL THICKNESS  : WINDING WIDTH'/
     >       'OCOIL ',4X,'RADIAL     ',5X,'AXIAL    '10X,'SUBDIVISION    ',
     > 9X'TURNS'9X'CURRENT'/' NO.'2(5X'THICKNESS')7X'RADIAL'6X'AXIAL'/
     >       11X,'B (M)',9X,'D (M)'13X,'NR',9X,'NY',13X,'W',10X,
     >       'I (A)',/)
  200 FORMAT(1X,I3,1X,1P2E14.5,9X,I3,8X,I3,5X,1P2E14.5)
  202 FORMAT(//T17,'CENTRE LINE'T75,'BOUNDARY LINE'/T65,'INSIDE'20X'OUTS
     >IDE'/5X'I'8X'X'12X'Z'9X'DELX'9X'DELZ'2(12X'X'12X'Z')/
     >       (I6,1P8E13.5))
  600 FORMAT('OSUBDIVISION OF THE',I3,1X'. GEOMETRIC CENTRE LINE BY CIRC
     >ULAR ARCHES FOR THE FOLLOWING COILS :'/(1X20(I3,' ,')/) )
  401 FORMAT('OCALCULATION OF THE SPLINE COEFFICIENTS FOR THE'I4,'. COIL
     > .'    )
      END
C     CALCULATION OF THE 2-DIMENSIONAL SPLINE COEFFICIENTS FOR THE FORCE
C     DISTRIBUTION OF A COIL COMPUTED WITH HEDO2 AT THE PLANES FOR
C     LN = 1 (1) N3A .
C     CX : SPLINE COEFFICIENTS FOR THE VOLUME FORCES IN X DIRECTION FOR
C          THE LN-TH PLAIN .
```

```
C     CY : SPLINE COEFFICIENTS FOR THE VOLUME FORCES IN Y DIRECTION FOR
C          THE LN-TH PLAIN .
C     CZ : SPLINE COEFFICIENTS FOR THE VOLUME FORCES IN Z DIRECTION FOR
C          THE LN-TH PLAIN .
C     ADDITIONAL OUTPUT OF THE SPLINE COEFFICIENTS ON G.FT10F001  .
      SUBROUTINE SPLKOE
      COMMON /KOEF/CX(4,4,16,8),CY(4,4,16,8),CZ(4,4,16,8)
      COMMON /KRAFT/VKX(16,8),VKY(16,8),VKZ(16,8)
      COMMON /PNTS/Y(8),LK,N1,N2,N3,B,D,DELB,DELD,PX(2,61),PZ(2,61)
      COMMON /01/FGS(3,60,16,8)
      COMMON /MIT/ PXM(61),PZM(61),DELX(61),DELZ(61)
      NAMELIST /IER/LN,LX,LY,IRX,IRY,IRZ
      REAL*4 X(16),WK(80)/80*0.0/
      FL=DELB*DELD
      DO 1 I=1,N2
    1 X(I)=0.5*DELB+(I-1)*DELB
      WRITE(10) N1,N2,LK,B,D,DELB,DELD,(X(LX),LX=1,N2),(Y(LY),LY=1,N1)
     >          ,((PX(I,LN),I=1,2),LN=1,LK),((PZ(I,LN),I=1,2),LN=1,LK)
      NX1=N2-1
      NY1=N1-1
      DO 2 LN=1,LK
      IF(LN.NE.1.AND.LN.NE.LK)  GOTO  7
      IF(PXM(1).EQ.PXM(LK))   GOTO  10
      K =1-2*(LN/LK)
      S1=SQRT((PXM(LN+K)-PXM(LN))**2+(PZM(LN+K)-PZM(LN))**2)/2.0
      S2=2.0*S1+SQRT((PXM(LN+2*K)-PXM(LN+K))**2+(PZM(LN+2*K)-PZM(LN+K))
     >   **2)/2.0
      L1=MIN0(LN,N3)
      DO 8  LX=1,N2
      DO 8  LY=1,N1
      VKX(LX,LY)=(S2**2*FGS(1,L1,LX,LY)-S1**2*FGS(1,L1+K,LX,LY))/(S2**2-
     >      S1**2)/FL
      VKY(LX,LY)=(S2**2*FGS(2,L1,LX,LY)-S1**2*FGS(2,L1+K,LX,LY))/(S2**2-
     >      S1**2)/FL
      VKZ(LX,LY)=0.0
    8 CONTINUE
      GOTO 9
   10 DO 11 LX=1,N2
      DO 11 LY=1,N1
      VKX(LX,LY)=(FGS(1,1,LX,LY)+FGS(1,N3,LX,LY))/(2.0*FL)
      VKY(LX,LY)=(FGS(2,1,LX,LY)+FGS(2,N3,LX,LY))/(2.0*FL)
   11 VKZ(LX,LY)=(FGS(3,1,LX,LY)+FGS(3,N3,LX,LY))/(2.0*FL)
      GOTO 9
    7 S1=SQRT((PXM(LN)-PXM(LN-1))**2+(PZM(LN)-PZM(LN-1))**2)/2.0
      S2=SQRT((PXM(LN)-PXM(LN+1))**2+(PZM(LN)-PZM(LN+1))**2)/2.0
      DO 4 LX=1,N2
      DO 4 LY=1,N1
      VKX(LX,LY)=(S2*FGS(1,LN-1,LX,LY)+S1*FGS(1,LN,LX,LY))/(S2+S1)/FL
      VKY(LX,LY)=(S2*FGS(2,LN-1,LX,LY)+S1*FGS(2,LN,LX,LY))/(S2+S1)/FL
      VKZ(LX,LY)=(S2*FGS(3,LN-1,LX,LY)+S1*FGS(3,LN,LX,LY))/(S2+S1)/FL
    4 CONTINUE
    9 WRITE(6,200)  LN,N2,N1,PX(1,LN),PZ(1,LN),PX(2,LN),PZ(2,LN),
     >              DELX(LN),DELZ(LN),(Y(LY),LY=1,N1)
      DO 6 LX=1,N2
      WRITE(6,201) X(LX),(VKX(LX,LY),LY=1,N1)
      WRITE(6,204) (VKY(LX,LY),LY=1,N1)
      WRITE(6,204) (VKZ(LX,LY),LY=1,N1)
```

```
    6 CONTINUE
      DO 5 LX=1,NX1
      DO 5 LY=1,NY1
      CALL IBCICU (VKX,16,X,N2,Y,N1,LX,LY,CX(1,1,LX,LY),WK,IRX)
      CALL IBCICU (VKY,16,X,N2,Y,N1,LX,LY,CY(1,1,LX,LY),WK,IRY)
      CALL IBCICU (VKZ,16,X,N2,Y,N1,LX,LY,CZ(1,1,LX,LY),WK,IRZ)
      IF(IRX+IRY+IRZ.NE.0)  WRITE(6,IER)
    5 CONTINUE
      WRITE(10) ((((CX(I,J,LX,LY),I=1,4),J=1,4),LX=1,NX1),LY=1,NY1),
     >          ((((CY(I,J,LX,LY),I=1,4),J=1,4),LX=1,NX1),LY=1,NY1),
     >          ((((CZ(I,J,LX,LY),I=1,4),J=1,4),LX=1,NX1),LY=1,NY1)
    2 CONTINUE
      RETURN
  200 FORMAT('1VOLUME FORCES FOR THE'I4,'. PLAIN'
     > 10X'NX ='I4,'    NY ='I4/' BETWEEN  ('1PE13.5,' ,'E13.5,' ) AND ('
     >   E13.5,' ,'E13.5,' ) DELX ='E13.5,' DELZ ='E13.5/
     >' 1. LINE FX(X,Y) IN (N/M**3)'/' 2. LINE FY(X,Y) IN (N/M**3)'/
     >' 3. LINE FZ(X,Y) IN (N/M**3)'/'0     X'6X'Y'0P10F12.4)
  201 FORMAT(/1X0PF10.4,2X1P10E12.4)
  204 FORMAT(13X1P10E12.4)
      END
```

```
C     INTERPOLATION OF VOLUME FORCES FOR D - COILS
      SUBROUTINE VKRFT(XYZ,F,NORMAL)
      COMMON /DF9/ ID9
      REAL*4  KX(2),KY(2),KZ(2),XYZ(3),F(3)
      COMMON /KOEF/ CX(4,4,15,7,2),CY(4,4,15,7,2),CZ(4,4,15,7,2)
      COMMON /VAR/XP(16),YP(8),N1,N2,N3,B,D,DELB,DELD,NX1,NY1,PX(2,61),
     >          PZ(2,61)
      LOGICAL   ANFANG/.TRUE./,NORMAL
      IF (ANFANG) CALL PDS
      ANFANG=.FALSE.
      X=XYZ(1)
      Y=XYZ(2)
      Z=XYZ(3)
      G1=(X-PX(1,1))*(PZ(1,1)-PZ(2,1))+(PX(2,1)-PX(1,1))*(Z-PZ(1,1))
      DO 1 LN=2,N3
      G2= (X-PX(1,LN))*(PZ(1,LN)-PZ(2,LN))+(PX(2,LN)-PX(1,LN))*
     >    (Z-PZ(1,LN))
      G3=(X-PX(1,LN-1))*(PZ(1,LN-1)-PZ(1,LN))+(PX(1,LN)-PX(1,LN-1))*
     >    (Z-PZ(1,LN-1))
      G4=(X-PX(2,LN-1))*(PZ(2,LN-1)-PZ(2,LN))+(PX(2,LN)-PX(2,LN-1))*
     >    (Z-PZ(2,LN-1))
      IF(G1*G2.LE.0.0.AND.G3*G4.LE.0.0) GOTO 3
      G1=G2
    1 CONTINUE
      WRITE(6,201) XYZ
      NORMAL=.FALSE.
      F(1)=0.0
      F(2)=0.0
      F(3)=0.0
    3 IF(.NOT.NORMAL)  RETURN
      S1=SQRT((PX(1,LN-1)-PX(1,LN))**2+(PZ(1,LN-1)-PZ(1,LN))**2)
      S2=SQRT((PX(2,LN-1)-PX(2,LN))**2+(PZ(2,LN-1)-PZ(2,LN))**2)
      DB=B*S2*G3/(S2*G3-S1*G4)
      FAKT=G1/(G1-G2)
      IX=(DB-DELB/2.0)/DELB+1.00001
      IY=(Y+(D-DELD)/2.0)/DELD+1.00001
      IX=MINO(NX1,MAXO(1,IX))
      IY=MINO(NY1,MAXO(1,IY))
      DX=DB-XP(IX)
      DY=Y-YP(IY)
      IF(ID9-1.EQ.LN)  GOTO  6
      ID9=LN-1
      FIND(9'ID9)
      DO 2 K=1,2
      READ(9'ID9) ((((CX(I,J,LX,LY,K),I=1,4),J=1,4),LX=1,NX1),LY=1,NY1),
     >            ((((CY(I,J,LX,LY,K),I=1,4),J=1,4),LX=1,NX1),LY=1,NY1),
     >            ((((CZ(I,J,LX,LY,K),I=1,4),J=1,4),LX=1,NX1),LY=1,NY1)
    2 CONTINUE
    6 DO 4 K=1,2
      KX(K)=0.0
      KY(K)=0.0
      KZ(K)=0.0
      DXI=1.0
      DO 4 I=1,4
      DYJ=1.0
      DO 5 J=1,4
      KX(K)= KX(K)+CX (I,J,IX,IY,K)*DXI*DYJ
```

```
      KY(K)= KY(K)+CY  (I,J,IX,IY,K)*DXI*DYJ
      KZ(K)= KZ(K)+CZ  (I,J,IX,IY,K)*DXI*DYJ
      DYJ=DY*DYJ
    5 CONTINUE
      DXI=DX*DXI
    4 CONTINUE
      F(1)=(KX(1)+(KX(2)-KX(1))*FAKT)
      F(2)=(KY(1)+(KY(2)-KY(1))*FAKT)
      F(3)=(KZ(1)+(KZ(2)-KZ(1))*FAKT)
      RETURN
  201 FORMAT(' ***XYZ ='1P3E13.5,'  IS OUTSIDE THE STRUCTURE ***')
      END
C    CREATING A PARTITIONED DATA SET (PDS) ON G.FT09F001 WITH THE
C    VOLUME FORCES FROM G.FT10F001 ,CALCULATED BY HEDO2IN
C    NECESSARY JCL CARDS :
C    //G.FT09F001 DD SPACE=(XXXXX,N3),DCB=(RECFM=VBS,BLKSIZE=XXXXX)
C     WITH XXXXX = 192*(N2-1)*(N1-1)+8
C    //G.FT10F001 DD (SPLINE-COEEFICIENTS CALCULATED WITH HEDO2IN)
C    REPLACE N3 AND XXXXX IN THE LINE : DEFINE FILE 9 (N3,XXXXX,L,ID9)
      SUBROUTINE PDS
      COMMON /DF9/ ID9
      COMMON /KOEF/ CX(4,4,15,7,2),CY(4,4,15,7,2),CZ(4,4,15,7,2)
      COMMON /VAR/XP(16),YP(8),N1,N2,N3,B,D,DELB,DELD,NX1,NY1,PX(2,61),
     >            PZ(2,61)
      DEFINE FILE 9(31,08648,L,ID9)
      ID9=1
      READ(10) N1,N2,N3,B,D,DELB,DELD,(XP(LX),LX=1,N2),(YP(LY),LY=1,N1),
     >        ((PX(I,LN),I=1,2),LN=1,N3),((PZ(I,LN),I=1,2),LN=1,N3)
      WRITE(6,200) N1,N2,N3,B,D,DELB,DELD
      WRITE(6,201) (XP(LX),LX=1,N2)
      WRITE(6,202) (YP(LY),LY=1,N1)
      WRITE(6,203) (LN,(PX(I,LN),PZ(I,LN),I=1,2),LN=1,N3)
      NX1=N2-1
      NY1=N1-1
      DO 2 LN=1,N3
      READ(10) ((((CX(I,J,LX,LY,1),I=1,4),J=1,4),LX=1,NX1),LY=1,NY1),
     >         ((((CY(I,J,LX,LY,1),I=1,4),J=1,4),LX=1,NX1),LY=1,NY1),
     >         ((((CZ(I,J,LX,LY,1),I=1,4),J=1,4),LX=1,NX1),LY=1,NY1)
      WRITE(9'ID9)((((CX(I,J,LX,LY,1),I=1,4),J=1,4),LX=1,NX1),LY=1,NY1),
     >            ((((CY(I,J,LX,LY,1),I=1,4),J=1,4),LX=1,NX1),LY=1,NY1),
     >            ((((CZ(I,J,LX,LY,1),I=1,4),J=1,4),LX=1,NX1),LY=1,NY1)
    2 CONTINUE
      ID9=-ID9
      RETURN
  200 FORMAT('1D - COIL DATA'/'0N1 ='I6,5X'N2 ='I6,5X'N3 =' I6/
     >      '0B ='1PE13.5,5X'D ='E13.5,5X'DELB ='E13.5,5X'DELD ='E13.5)
  201 FORMAT('0XP ='1P10E12.4/(5X10E12.4))
  202 FORMAT('0YP ='1P10E12.4/(5X10E12.4))
  203 FORMAT('0'25X'BOUNDARY LINE'/18X'INSIDE'20X'OUTSIDE'/
     >   '    LN'2(8X'X'12X'Z'4X) /(1XI4,1P4E13.5))
      END
```

REFERENCES

[1] J.K. Bathe, E.L. Wilson, F.E. Peterson: "SAP V (A Structural Analysis Program for Static and Dynamic Response of Linear Systems"; University of California, Berkeley (1974)

[2] H. Gorenflo, O. Jandl: "Mesh Generation for the 20-node Isoparametric Solid Element by the Computer Program MESHGEN"; IPP Report 4/148 (1977)

[3] M. Söll, O. Jandl, H. Gorenflo: "Mechanical Stress Calculations for Toroidal Field Coils by the Finite Element Method". IPP Report 4/142 (1976)

[4] R.H. Gallagher: "Finite Element Analysis - Fundamentals". Printice-Hall, Inc., New Jersey (1975)

[5] R.D. Cock: "Concepts and Applications of Finite Element Analysis"; John Wiley & Sons, Inc. New York (1974)

[6] Pin Tong, John J. Rossettos: "Finite Element Method Basic Technique and Implementation"; MIT Press, Massachusetts (1977)

[7] P. Martin, H. Preis: "Program Description and Users' Manual for the HEDO2 Magnetic Field Computer Program"; IPP Report III/34 (1977)

[8] Source Code IBCICU (Bicubic Spline Two-dimensional Coefficient Calculator) from International Mathematical & Statistical Libraries Inc. (IMSL), Houston, Texas