

## A THREE-DIMENSIONAL B.I.E.M. PROGRAM

M. Doblaré & E. Alarcón

Polytechnic University. Madrid. Spain.

### ABSTRACT

The program PECET (Boundary Element Program in Three-Dimensional Elasticity) is presented in this paper.

This program, written in FORTRAN V and implemented on a UNIVAC 1100, has more than 10,000 sentences and 96 routines and has a lot of capabilities which will be explained in more detail.

The object of the program is the analysis of 3-D piecewise heterogeneous elastic domains, using a subregionalization process and 3-D parabolic isoparametric boundary elements.

The program uses special data base management which will be described below, and the modularity followed to write it gives a great flexibility to the package.

The Method of Analysis includes an adaptive integration process, an original treatment of boundary conditions, a complete treatment of body forces, the utilization of a Modified Conjugate Gradient Method of solution and an original process of storage which makes it possible to save a lot of memory.

### BRIEF DESCRIPTION OF THE PROGRAM

The main idea developed in the program is the modularity. The program is composed of a number of independent subroutines, which are controlled by the main modulus which leads all the process of execu-

tion. There are many advantages, among which it is important to notice:

- the superposition of the moduli, by a segmentation process
- flexibility during the implementation and correction.

The main modulus is composed of the main program and a master routine with the name GENIO, which opens the files, defines the principal and auxiliary memory sizes and calls the different moduli of the process.

This segment always remains in memory with auxiliary routines, but the other segments are introduced one by one in order to save core storage. These segments are:

- input and definition of the data
- preparation and treatment of these data
- integration and formation of the matrix
- resolution of the system
- output of the results.

There are also some general routines mainly to manage the data in a special way.

#### DATA BASE MANAGEMENT

Basically, the data structure uses two concepts:

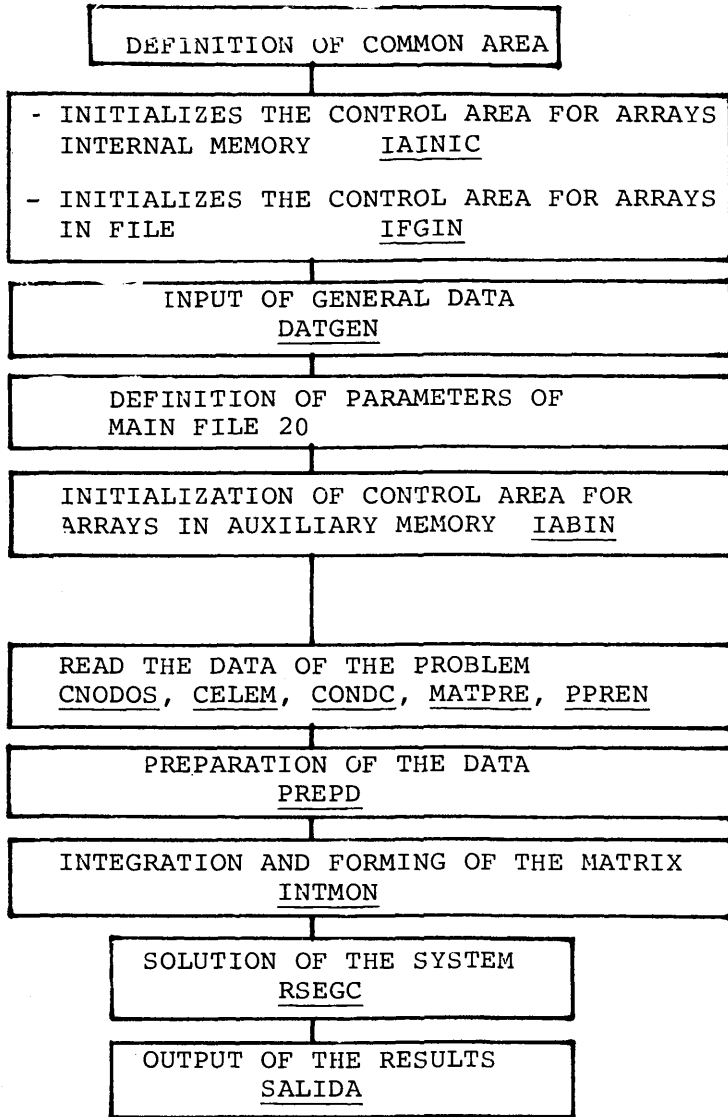
- dynamic storage in the internal memory
- use of the auxiliary memory by the multiple buffer area method (M.B.A.M.).

Using the first of these concepts, it is possible to get optimum management of the memory. To get more flexibility, a package of subroutines has been developed; they help to create, erase a store and recreate the array you want in a simple manner, and at the point of the program where it is necessary.

The second concept is also very important, and it is used for the treatment of big arrays which have to be stored in auxiliary memory. It is possible to use these arrays in exactly the same way as those stored in internal memory.

This is done by another set of subroutines which creates, erases and looks for the required element of any array as if it were a normal one.

There are two different main processes to manage the data base: the use of a buffer area for all the



data stored in auxiliary memory and the pagination process.

The first one has some advantages when the program uses the array in a sequential form, and the second one when this access is done in a random manner. The first procedure requires very simple management.

Because in our case the data set is used in a random way and because the management has to be developed by the programmer and has to be as simple as possible, the best solution is a combination of these two different processes.

This method was named "The Multiple Buffer Area Method" and it consists of the assignment of several buffer areas in the internal memory for each array to be treated while the real storage is done in a direct access file with records of the same size as those defined for the buffer areas. The program uses two different buffer areas for each array.

The information for use of this system is organized by a General Control Area for the whole system and a Particular Control Area for each array, containing the name of the array, the size, the position in the total buffer area, number of accesses to the first and second buffer area, etc.

With this treatment, and for this problem in particular, management of the data base is obtained with maximum simplicity for the user.

#### DESCRIPTION OF THE PROGRAM

The program is divided into two different sections: one of them accessible to the user and composed of the main program and the master routine and a second one not accessible, divided into several principle subroutines or moduli which use other levels of routines.

The Control program defines the COMMON area, the computer characteristics and the storage files and calls the master routine GENIO, finishing with the closure of the files.

The master subroutine leads all the processes and a flow-chart can be seen in the next figure:

A complete description of the program and users manual can be seen in [7].

### ADVANTAGES

The main advantage of this package over others can be summarized as follows:

- 1) A much simpler management of the data, especially for the user.
- 2) A great flexibility because of the modularity and simplicity of the connections among the segments.
- 3) The adaptability to different computers, the specific features of which are few and are grouped at the beginning of the program.
- 4) A detailed list of errors which makes it possible to find the cause of an error quickly (see [7]).
- 5) An easy understanding of the program, because of the amount of notes and the short length of the routines.
- 6) A complete treatment of the boundary conditions.
- 7) The treatment of body forces (temperature, self-weight, centrifugal forces, points loads).
- 8) Offers an approximate solution for the interface nodes with more unknowns than equations.
- 9) An effective modified version of Conjugate Gradient which allows an important saving of memory.

The programs also have all the advantages of previous programs of this kind (Lachat and Watson [16]), such as adaptive integration, storage of the matrix, subregionalization, etc.

### METHOD OF ANALYSIS

As has been stated previously, the program is based on the use of the B.I.E.M. in 3-D Elasticity. The base equation of this method is the well-known Somigliana identity

$$\delta_{ki} u_i(x) = \int_{\delta\Omega} U_{ik}(x,y) t_i(y) ds_y - \int_{\delta\Omega} T_{ik}(x,y) u_i ds_y + \int_{\Omega} U_{ik}(x,y) X_i(y) ds_y \quad (1)$$

in which  $C_{ik}$  is given by (see Hartmann [14])

$$C_{ik} = \delta_{ik} \quad \text{for internal points}$$

$$C_{ik} = 0 \quad \forall i,k \quad \text{for external points}$$

$$C_{ik} = \int_{S_\varepsilon} \frac{-1}{8\pi(1-\nu)} \begin{bmatrix} (1-2\nu)+3 \cos^2 \phi \sin^2 \theta & 3 \cos \phi \sin \phi \sin^2 \theta & 3 \cos \phi \sin \phi \cos \theta \\ & (1-2\nu)+3 \sin^2 \phi \sin^2 \theta & 3 \sin \phi \sin \phi \cos \theta \\ & & (1-2\nu)+\cos^2 \theta \end{bmatrix}$$

$\sin \theta d\phi d\theta$  for boundary points,

$S_\varepsilon$  being the surface of the intersection volume between the domain and a sphere of radius  $\varepsilon$  centered at the point  $x$ .

The Kernels  $U_{ik}$  and  $T_{ik}$  correspond to the displacement and stress vectors in a boundary point due to a unit point load acting in a point  $x$  of the infinite 3-D space or, in other words, the resulting Kernels of the fundamental solution of Navier's equation. The expressions of these Kernels are [3]

$$U_{ik} = \frac{1}{16(1-\nu)Gr} [(3-4\nu) \delta_{ik} + r_i r_k] \quad (3)$$

$$T_{ik} = - \frac{1}{8(1-\nu)r^2} \{ r_i n_j [(1-2\nu) \delta_{ik} + 3r_i r_k] + (1-2\nu) (r_i n_k - n_i r_k) \}$$

in which  $r$  is  $r = ||x-y||$  and  $y$  is any point of the boundary (domain) in the first and second (third) integrals. The vectors  $\underline{u}$ ,  $\underline{t}$ , are the displacement and stress vectors in a boundary point in the real problem and  $\underline{X}$  the known body forces vector in every point of the domain.

This equation makes it possible to solve the

problem with the single necessity of discretizing the boundary if the volume integral of the body forces is easily solvable (as usually happens). Once the boundary unknowns are computed, it is possible to calculate the displacements at any interior point using the equation (1) with  $C_{ik} = \delta_{ik}$ .

The stress tensor in the domain can be computed by applying Lamé's operator to equation (1) at point  $x$ . The result is given by [18].

$$\begin{aligned} \sigma_{ij}(x) = & \int_{\delta\Omega} D_{ijk}(x,y) t_k(y) ds_y - \\ & - \int_{\delta\Omega} S_{ijk}(x,y) u_k(y) ds_y - \\ & + \int_{\Omega} D_{ijk}(x,y) X_k(y) d\Omega_y \end{aligned} \quad (4)$$

where  $D_{ijk}$  and  $S_{ijk}$  are new tensors whose expressions are

$$\begin{aligned} D_{ijk} = & \frac{1}{8\pi(1-\nu)r^2} [(1-2\nu)(\delta_{ik}r_{,j} + \delta_{jk}r_{,i} - \delta_{ij}r_{,k}) + \\ & + 3r_{,i}r_{,j}r_{,k}] \end{aligned} \quad (5)$$

$$\begin{aligned} S_{ijk} = & \frac{G}{4\pi(1-\nu)r^3} \{ 3 [(1-2\nu)\delta_{ij}r_{,k} + \\ & + \nu(\delta_{ik}r_{,j} + \delta_{jk}r_{,i}) - 5r_{,i}r_{,j}r_{,k}] r_{,i}n_i + \\ & + 3\nu(n_i r_{,j}r_{,k} + n_j r_{,i}r_{,k}) + \\ & + (1-2\nu)(3n_k r_{,i}r_{,j} + n_j \delta_{ik} + n_i \delta_{jk}) - \\ & - (1-4\nu)n_k \delta_{ij} \} \end{aligned}$$

To finish the initial formulation we can say that the domain integral of the body forces can be, in the usual cases, reduced to a boundary integral as can be seen in Doblaré [7].

#### NUMERICAL APPROXIMATION

The discretization of the equation (1) for boundary points is done following the concepts of the F.E.M. using boundary elements with a certain geometry and approximating the unknowns in each of these elements, using the usual shape functions for plane finite elements.

The present program uses quadrilateral and triangular parabolic isoparametric elements, with eight and six nodes respectively (Fig. 1).

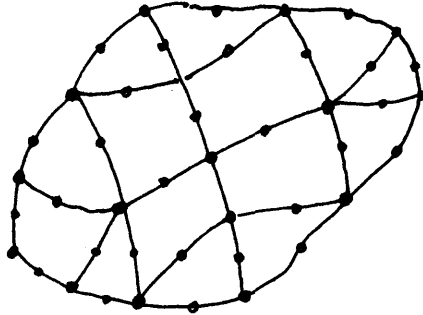


Fig. 1

Also, in order to include in the formulation piecewise heterogeneous bodies, a subregionalization process has been used, expressing the Somigliana equation in each subregion and adding the equilibrium and compatibility equations in each of the interfaces

$$\begin{aligned}
 (\delta_{ik} - C_{ik}) u_i + \int_{\delta\Omega(r)} T_{ik} u_i ds_y = \int_{\delta\Omega(r)} U_{ik} t_i ds_y + \\
 \int_{\delta\Omega(r)} E_{ik} x_i ds_y + \int_{\Omega(r)} F_k ds_y \quad (6)
 \end{aligned}$$

$$\begin{aligned}
 u_i^{(r)}(x) &= u_i^{(s)}(x) \\
 t_i^{(r)}(x) &= -t_i^{(s)}(x) \\
 x \in \delta\Omega(r) \cap \delta\Omega(s) \quad (7)
 \end{aligned}$$

Discretizing each subregion  $r$  with  $p(r)$  elements, each of them with  $n$  nodes, we can write equation (6) for each subregion as

$$\begin{aligned}
 \sum_{b=1}^{p(r)} \sum_{c=1}^n B_{ikbc}^\alpha u_i(d^{(r)}(b,c)) = \sum_{b=1}^{p(r)} \sum_{c=1}^n [A_{ikbc}^\alpha t_i(d^{(r)}(b,c))] + \sum_{b=1}^{p(r)} P_{kb}^\alpha \quad (8)
 \end{aligned}$$

$$b \in [1, 2, \dots, p^{(r)}], \quad c \in [1, 2, \dots, n], \quad d \in [1, 2, \dots, q^{(r)}]$$



$$\begin{aligned}
B_{ik_{bc}}^\alpha &= \int_{\delta\Omega_b} U_{ik} [x^\alpha, y(\xi, \eta)] N^C(\xi, \eta) J(\xi, \eta) d\xi d\eta \quad \text{for } \alpha = d(r) \text{ (b,c)} \\
B_{ik_{bc}}^\alpha &= \int_{\delta\Omega_b} U_{ik} [x^\alpha, y(\xi, \eta)] N^C(\xi, \eta) J(\xi, \eta) d\xi d\eta + [\delta_{ik} - C_{ik}(\alpha)] \\
&\quad \alpha = d(r) \text{ (b,c)} \\
A_{ik_{bc}}^\alpha &= \int_{\delta\Omega_b} T_{ik} [x^\alpha, y(\xi, \eta)] N^C(\xi, \eta) J(\xi, \eta) d\xi d\eta \\
P_{k_b}^\alpha &= \int_{\delta\Omega_b} [E_{ik} [x^\alpha, y(\xi, \eta)] X_i(\xi, \eta) + F_k [x^\alpha, y(\xi, \eta)]] \\
&\quad J(\xi, \eta) d\xi d\eta \quad (9)
\end{aligned}$$

or in a more compact form

$$\underline{B}^{(r)} \underline{u}^{(r)} = \underline{A}^{(r)} \underline{t}^{(r)} + \underline{P}^{(r)} \quad (10)$$

#### COMPUTATION OF THE INTEGRATION CONSTANTS

The computation of the elements of each of the matrices A, B, P is now needed; these elements can be expressed as

$$\begin{aligned}
A_{ik_{bc}}^\alpha &= \int_{\delta\Omega_b} U_{ik} N^C ds_y \\
B_{ik_{bc}}^\alpha &= \int_{\delta\Omega_b} T_{ik} N^C ds_y \\
P_{k_b}^\alpha &= \int_{\delta\Omega_b} E_{ik} X_i ds_y + \int_{\delta\Omega_b} F_k ds_y
\end{aligned} \quad (11)$$

These integrals can not be calculated analytically in an easy way because of the complexity of the Jacobian of the parabolic elements. For this reason a numerical approach has been followed.

However the singularity of the kernels  $U_{ik}$  and  $T_{ik}$  (tends to  $\infty$  when  $r$  tends to zero) implies that this integration scheme needs a careful use for the integration over the elements near the collocation point in order to reduce the integration error.

An adaptive process has been used to solve these difficulties following Lachat [16], with a standard Gaussian process, but with a previous refinement of the element dividing each of them into a certain number of subelements and choosing the number of integration points for each of these subelements according to an error bound previously defined in function of some variables, e.g. the distance between the element and the singular point, the size, and the geometry of the element, etc.

When the nodal point belongs to the integration element, a different approach is used in order to place a great number of integration points near the singular one. The element is divided into two or three triangles and each of them is considered as a degenerated quadrilateral subelement. The disposition of integration points is shown in Fig. 2.

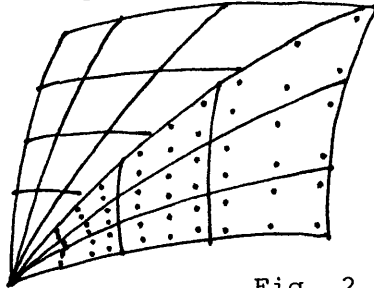


Fig. 2

### THE LINEAR EQUATION SYSTEM

Once the equilibrium and compatibility equations have been introduced for the interface nodes, the complete linear system is reduced to a problem with

$\sum_{r=1}^R p(r) \times 3$  equations where  $R$  is the total number of equations.

However, the number of unknowns can be greater than the number of equations. This number of un-

knowns is  $\sum_{i=1}^I 6 p(i) + \sum_{c=1}^R 3 p(c)$  where  $I$  is the number of interfaces and  $p(i)$  and  $p(c)$  are the number of nodes of each interface or subregion external boundary of the problem.

Two different problems arise during the formation of the linear system:

- a) To reduce the number of unknowns at interface nodes.
- b) To get a matrix structure adapted to the posterior solving process; we have chosen a structure which allows a systematic process of storing, an appropriate situation of the zeros in order to eliminate them during the storing process and finally an adequate disposition for the solving process.

The first point is obtained by using a new interpolation function for the stress vector in the element to which the node with more unknowns than equations belongs. Each of these functions, with a lower order than the usual ones, takes the value 1 in each node and 0 in the rest, except for the special node. After this, it is possible to extrapolate the stress vector, as Fig. 3 shows, obtaining expressions (12) and (13).

$$t_i = a_{1_i} + a_{2_i} \xi + a_{3_i} \eta + a_{4_i} \xi^2 + a_{5_i} \eta^2 + a_{6_i} \xi \eta + a_{7_i} \xi^2 \eta \quad (12)$$

$$t_1 = t_4 + t_2 - t_3 \quad (13)$$

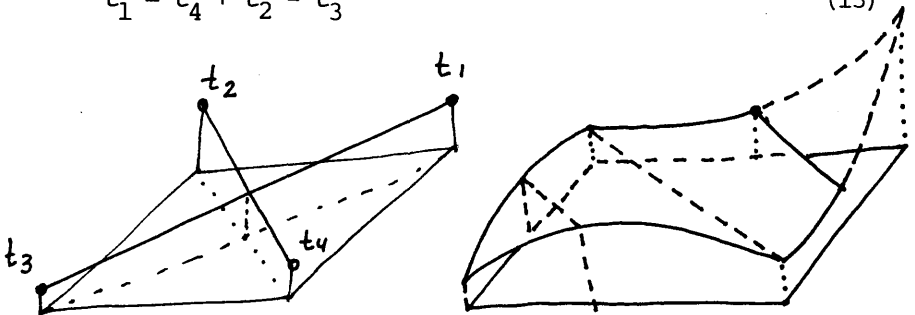


Fig. 3

The other point is solved, following Lachat, ordering the subregions as in Fig. 4, obtaining a great number of grouped zeros which are not stored and saving a lot of memory.

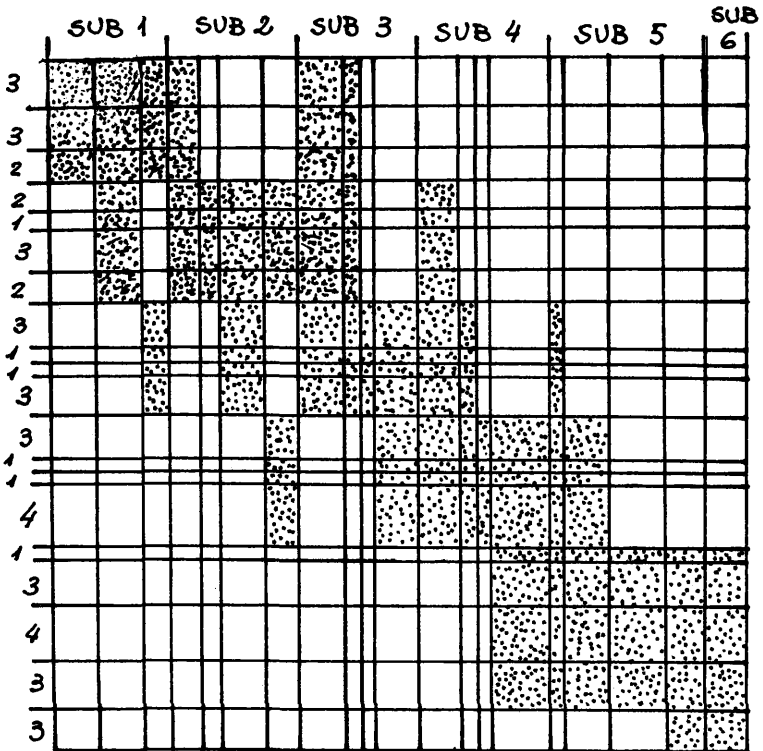


Fig. 4

BOUNDARY CONDITIONS

This is one of the topics less studied in other similar programs.

The main problem now is that there is an excess of unknowns and it is necessary to eliminate them.

The present program uses additional relations to solve this problem.

For example, in the two cases of Fig. 5, at node P, there are 6 unknown stresses and only 3 inte

gral equations. This is solved using relations between the stresses in each of the faces, such as Cauchy's relations.

In order to make the process systematic, four different conditions and 29 different cases in a simple node, solving each of them in a different way, making it possible to group all of them in 4 different processes:

- the normal case (only 3 unknowns in one element)
- there is no need to use additional relations but they are necessary for storing information
- use of Cauchy's relations
- the unknowns are the principal stresses, and the principal directions are approximated in some way.

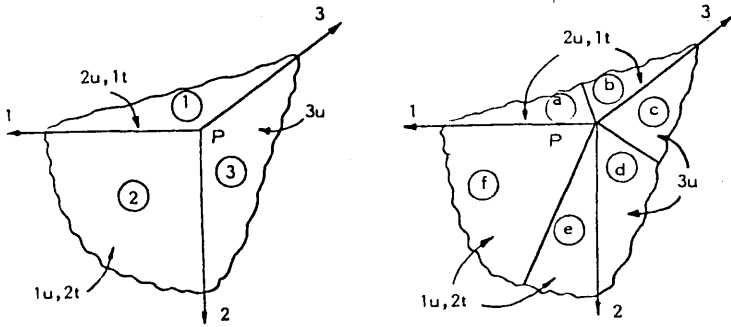


Fig. 5

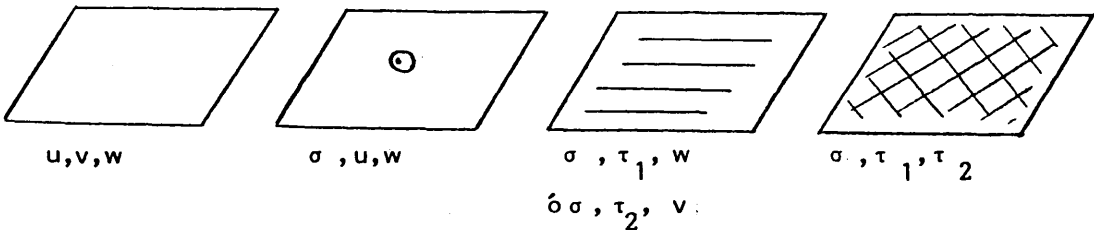


Fig. 6

## RESOLUTION OF THE EQUATION SYSTEM

This program is written for solving large elastic 3-D problems with a great number of equations in a fully populated and nonsymmetric matrix. However, it is possible to get a lot of zeros in this matrix using the subregionalization process.

It is possible to save a great deal of memory by not storing these zeros. However, it is not possible to do so if we use an elimination scheme for solving the system. Also, and in order to optimize the storing process, this is done column by column and it would be necessary to use a special routine to use an elimination process.

For these and other reasons, we have chosen an iterative solution method, such as the Conjugate Gradient Method, which has been modified in order to optimize the specific problem we have at hand and use an approximate initial vector for a rapid convergence.

### E X A M P L E S

=====

#### SPHERE SUBJECT TO INTERNAL PRESSURE

This example has been solved and compared with the theoretical solution given in [13].

The problem consists of a sphere with internal radius of 30 and external radius of 110 subject to a pressure of 1,000 with a Young's Modulus of 2.5 and Poisson's ratio of 0.25.

The discretization of 18 elements and 36 nodes is shown in Fig. 7 and, in order to prove the M.B.A.M. method, only a memory of 6K was used in the program for the data.

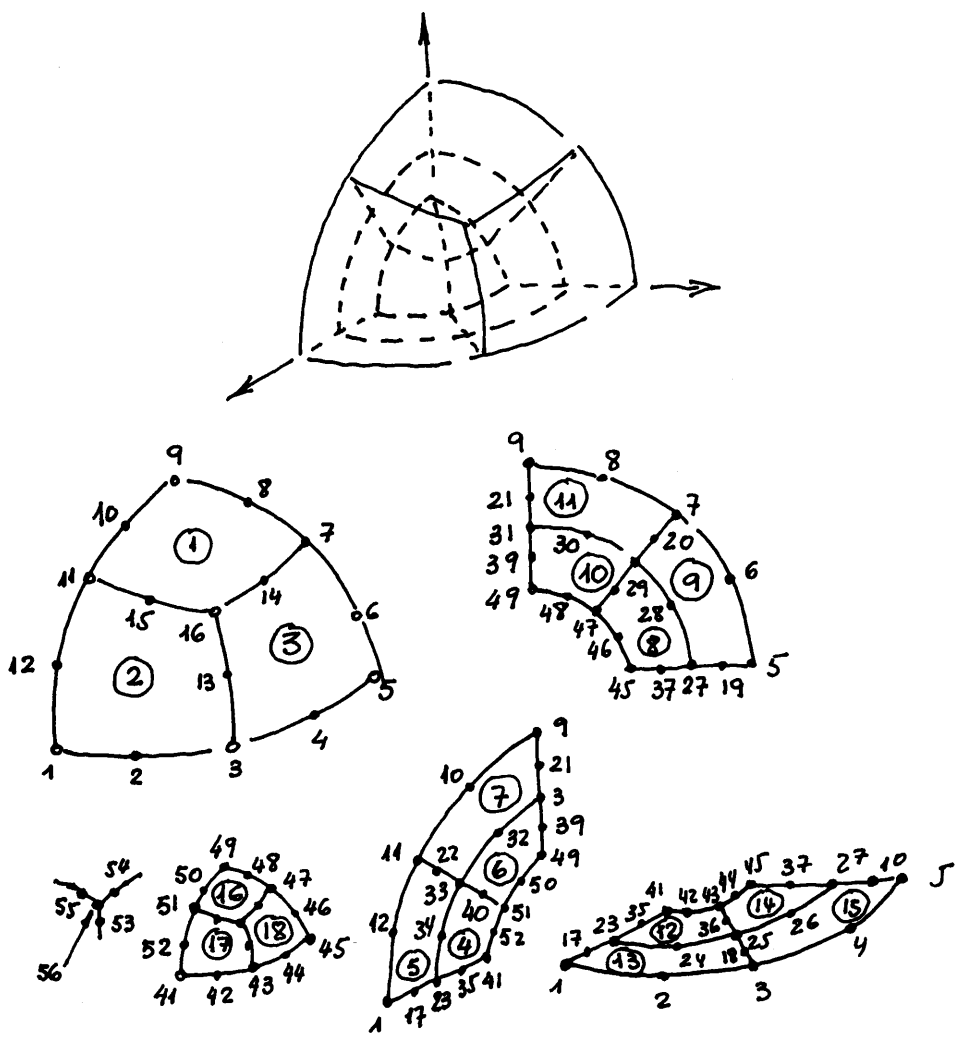


Fig. 7

The results are shown below:

NODE	VARIABLE	THEORY	PECET	ERROR %
1	$u_r$	6428.57	6401.	-0.43
17	$u_r$	7164.72	7109	-0.78
23	$u_r$	8492.06	8406.	-1.01
35	$u_r$	10928.57	10800.	-1.17
41	$u_r$	15714.29	15590.	-0.79
1	$\sigma_t$	214.29	207.5	-3.17
17	$\sigma_t$	249.48	250.9	0.57
23	$\sigma_t$	312.17	303.9	-2.65
35	$\sigma_t$	435.43	433.5	-0.44
41	$\sigma_t$	714.29	757.	5.98

The total time needed was 24'50".

It can be seen that the time was very high, which is logical because the amount of time of input-output is very important in this case and the access time of the computer used was also very high.

#### CUBE SUBJECT TO INTERNAL POINT LOADS

The problem consists of a simple cube supported at three different faces subjected to two points loads of 1000 acting in the interior of the body (see Fig. 8).

The deformation of the cube can be seen in Fig. 9.



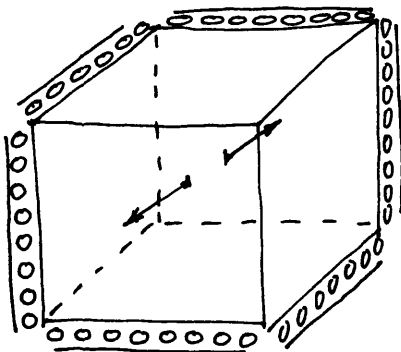


Fig. 8

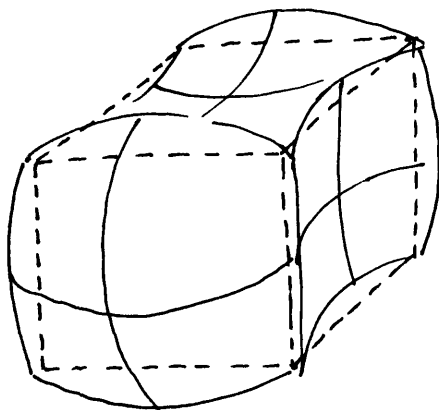


Fig. 9

### ANALYSIS OF A DOLOS

As an example of the treatment of boundary conditions, the analysis of a dolos has been developed (Fig. 10).

The discretization of a quarter of dolos is also shown in Fig. 11, while the results of the treatment of boundary conditions can be seen also, showing the information stored for some nodes in order to apply the special boundary conditions to each of them.

These conditions are: normal displacement restricted at the planes of symmetry and tangential stresses zero on them, and all the displacements are zero in elements 62 and 69 of the figure.

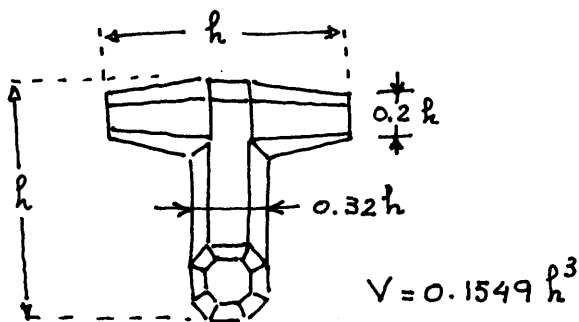


Fig. 10

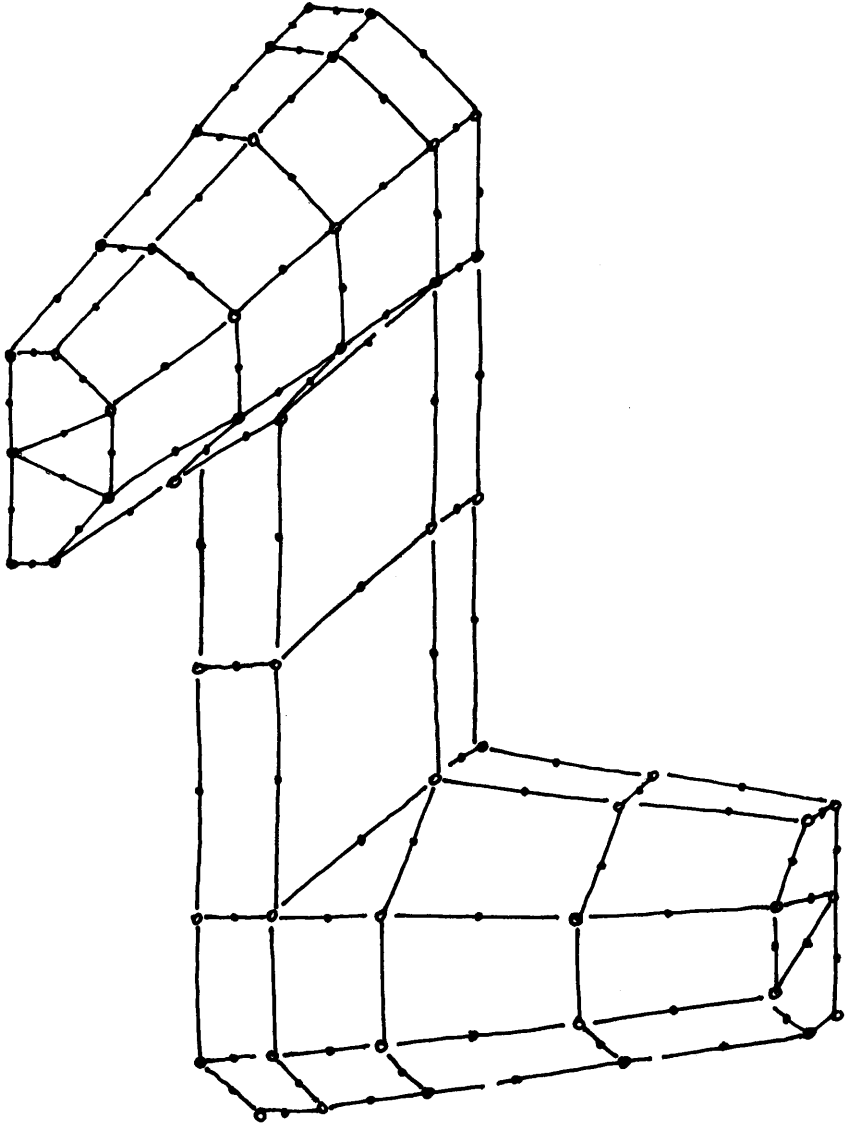
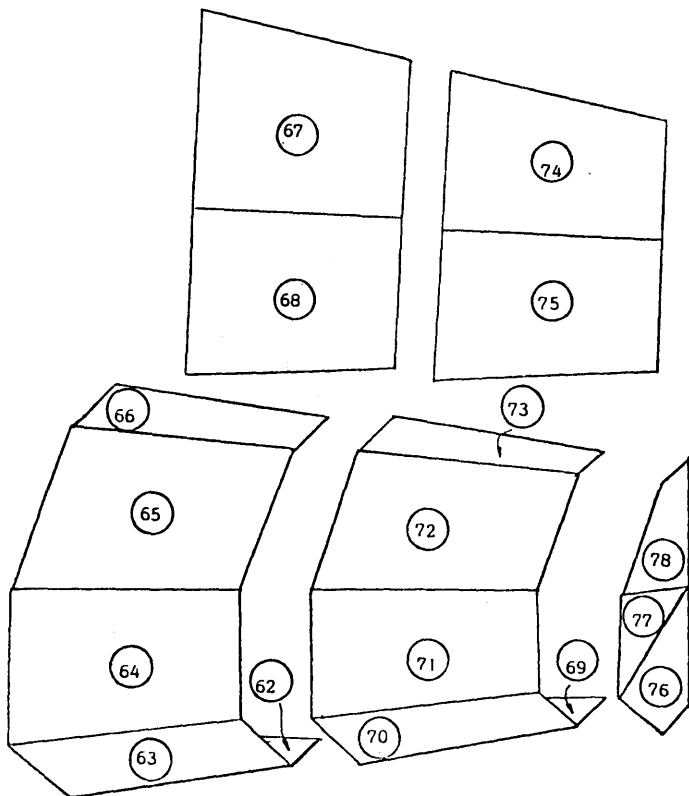


Fig. 11



## CONCLUSIONS AND AVAILABILITY

A 3-D boundary element program with parabolic elements has been developed with certain advantages over other previous programs.

This program solves only piecewise-heterogeneous elastic 3-D domain, but its flexibility makes it possible to add other different options in a simple way, such as plasticity (iterative process of solution is included), automatic generation of the mesh, or others which are currently being studied.

A complete description of the program (in Spanish) is available in reference [7], with the Theoretical Manual, The Programmer's Manual, the User's Manual and a description of each routine and arrays with all the details.

The owner of the program is Manuel Díaz del Río, Director de Planificación, RENFE (National Railways), Estación de Chamatrin, Madrid, Spain, from whom it may be ordered.

## REFERENCES

- 1.- ALARCON, E., MARTIN, A., PARIS, F. "Boundary Elements in Potential and Elasticity Theory" Computers and Structures, Vol.10, pp.351-362, 1973.
- 2.- BREBBIA, .A. "Boundary Elements Method for Engineers", Pentech Press, London, 1978.
- 3.- CRUSE, T.A. "Numerical Solutions in Three Dimensional Elastostatics", Int. y Sol. Struct. 5. 125874, 1969.
- 4.- CRUSE, T.A. "Application of the Boundary Integral Equation on Method to Three-Dimensional Stress Analysis", Computers and Structures, Vol. 3, pp.509-527, 1973.
- 5.- DATE, C.J. "An Introduction to Database Systems", Addison-Wesley, Reading, Massachussets, 1975.
- 6.- DAVIS, P.J. and RABINOWITZ, P. "Methods of Numerical Integration", Academic Press, 1975.
- 7.- DOBLARE, M. "Formulación tridimensional del método de los elementos de contorno con interpolación parabólica", Thesis, E.T.S.I.I., Madrid, 1981.
- 8.- DVORNIK, J. "Generalization of the C.G. Method Applied to Linear and Nonlinear Problems", J. Comp. Stu., 10, pp.217-223, 1979.
- 9.- ENGELS, H. "Numerical Quadrature and Cubature", Academic Press, 1980.
- 10.- FELIPPA, C.A. "Database Management in Scientific Computing I. General Description", Computers and Structures, Vol. 10, pp.53-61, 1979.
- 11.- FELIPPA, C.A. "Database Management in Scientific Computing II. Data Structures and Program Architecture", Computers and Structures, Vol.12, pp.131-145, 1980.
- 12.- FUNG, Y.C. "Foundations of Solids Mechanics", Prentice Hall, 1965.
- 13.- GAMBOLATI, G. "Fast Solution to Finite Element Flow Equations by Newton Iteration and Modified Conjugate Gradient Method", Stat. J. Num. Meth. Eng., 15, pp.661-675, 1980.
- 14.- HARTMANN, F. "Computing the C-Matrix in Non-Smooth Boundary Points". Second Int. Seminar in Recent Advances in B.E.M., pp.229-246, Southampton, March, 1980.

- 15.- KATZAN, H. "Computer Data Management and Data Base Technology", Van Nostrand Reinhold, New York, 1975.
- 16.- LACHAT, J.C. "A Further Development of the Boundary Integral Technique for Elastostatic", Ph.D. Thesis, University of Southampton, 1975.
- 17.- LACHAT, J.C. and WATSON, J.O. "Effective Numerical Treatment of Boundary Integral Equation: A Formulation for Three-Dimensional Elastostatic", Int. J. Num. Meth. in Eng., 10, pp.991-1005, 1976.
- 18.- PARIS, F. "El método de los elementos de contorno en la teoría del potencial y la elasticidad", Thesis, E.T.S.I.I., Madrid, 1979.
- 19.- STROUD, A.H. and SECREST, D. "Gaussian Quadrature Formulae", Prentice Hall, 1966.