Automatic mesh generation processor for 3-D parabolic boundary discretization

C. BURGOS, M. DOBLARÉ, E. ALARCÓN

INTRODUCTION

In recent years Boundary Element techniques (BE) have gained increasing momentum in comparison with Finite Element (FE) ones. The main reason is probably the reduction in dimensionality of the domain to be discretized as well as the ability to model bodies with large volume-to-surface ratios.

Although the discretization with BE is simpler than with FE it is, nevertheless, cumbersome, and requires tedious checking, when three-dimensional bodies have to be managed.

Thus a computer program has been developed which takes into account the experience gained by previous FE researchers in areas like blended interpolation, and that can be used as a pre-processor for three-dimensional BE codes.

This program is merely part of a concerted effort intended to produce an effective computational tool; other parts of which have been described elsewhere.[1]

The basis of the program is the concept of isoparametric macro-elements that, after subdivision according to certain rules, are adjusted to the actual geometry through the use of transfinite ('blended') interpolation as defined by Birkhoff *et al.*,[2] Gordon[3] or Gordon *et al.*[4]

A familiar method uses the linear combination

$$Bf = I_\xi f + I_\eta f - I_\xi I_\eta f \tag{1}$$

called the blended spline approximate to f, where

$$I_\xi f(\xi, \eta) = \sum_{i=1}^{n} f(\xi_i, \eta)\, p_i(\xi)$$

$$I_\eta f(\xi, \eta) = \sum_{i=0}^{n} f(\xi, \eta_i)\, p_j(\eta)$$

$$I_\xi I_\eta f = \sum_{0<i<n} f(\xi_i, \eta_i)\, p_i(\xi)\, p_j(\eta) \tag{2}$$

that is, I_ξ, I_η are blended interpolates to f; $I_\xi I_\eta$ is the usual tensor product interpolate and p represents the interpolating polynomials.

This idea is complemented by the definition of special rules of subdivision and by a graphic output as an echo check of the data.

It is interesting to point out that due to the particular features of BE as well as of the program PECET it has not been necessary to introduce a re-numbering scheme for the final mesh.

GENERATION BASIS

As usual, the boundary of the domain is sub-divided into several macro-elements S_i such that the whole boundary S can be mapped by them

$$S = \cup S_i$$

$$S_i \cap S_j = L_{ij} \tag{3}$$

L_{ij} is the border between elements i and j. Although it is not strictly necessary, we have decided to sub-divide the line L_{ij} into a form compatible with both i and j elements.

The choice of the macro-elements is dependent on the ingenuity of the user, and is generally dictated by the body shape and by the element catalogue of the program. In general they are quadrilaterals that can degenerate into triangles.

The lines can be sub-divided either by imposing the compatibility as data in the input, by working with each element or by sub-dividing the lines and then using it as a basis for meshing the element. The first option presents problems of repeating work and the second needs more storage. Due to the modularity of the master program PECET and to the segmentation process utilized by it, it was decided to use the second alternative.

Lines typology and sub-division algorithm

As indicated by Fig. 1 four types of lines can be used

Straight lines (A, B)
Parabolic lines (A, B, C)
Circular arcs (A, B, O, ν)
Circumferences (A, O, ν).

The direction $A \circlearrowleft B$ is related to ν by

$$\nu = \frac{OA \wedge OB}{|OA \wedge OB|} \tag{4}$$

In addition to the previous data it is necessary to specify the number of sub-divisions in every line as well as the kind of length variation inside them. If ξ is the required natural co-ordinate it was found to be useful to use the following relation which, with only the parameter R, provides a rich family of densification possibilities.

$$\xi(J, R) = \bar{\xi} + \frac{R}{2}(\bar{\xi}^2 - 1) \tag{5}$$

where

$$\bar{\xi} = \frac{2(J-1)}{N} - 1, \qquad J = 1, 2, \ldots, N+1 \tag{6}$$

N being the number of desired subdivisions.

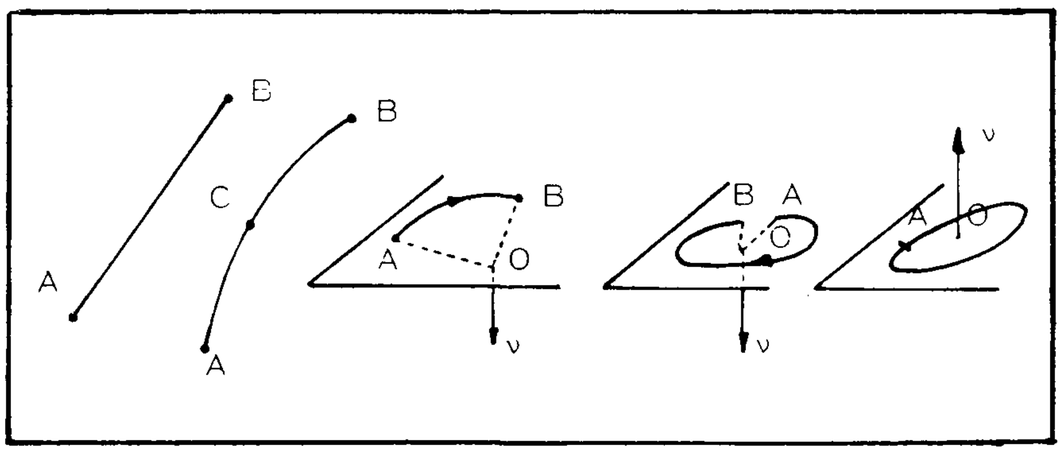


Figure 1. Line possibilities

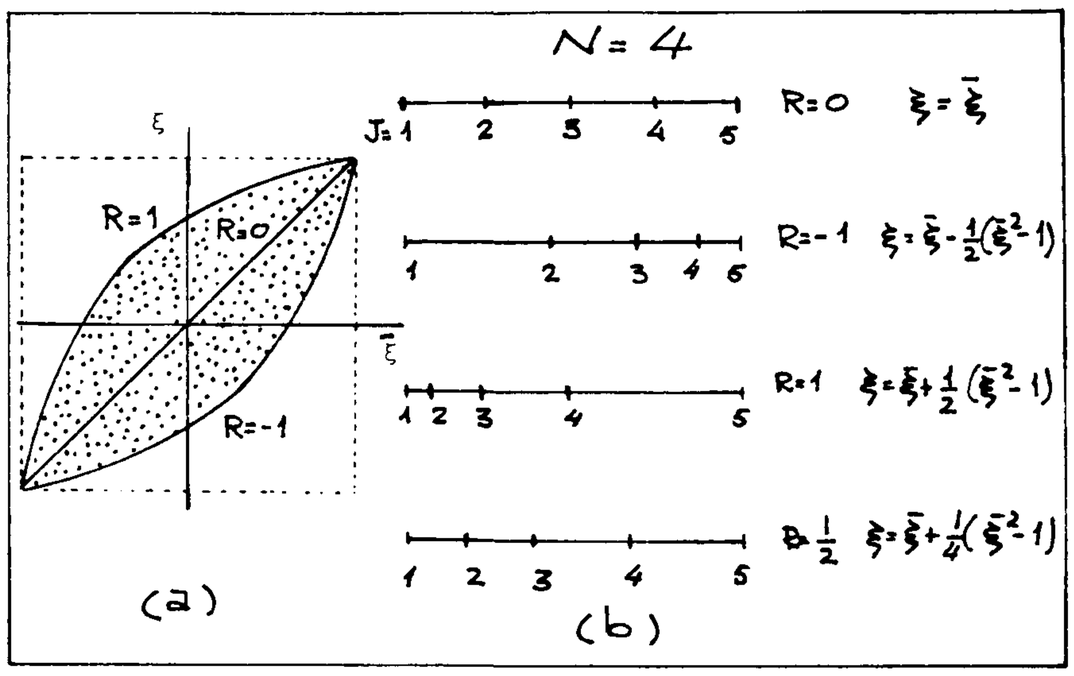


Figure 2. Sub-division algorithm

Fig. 2(a) indicates the trends of the segmentation; where R is positive the discretization concentrates around the initial node and coalesces near the final one for negative values. In Fig. 2(b) we can see the results for the case $N = 4$.

This sub-division is intended to model only the extreme points in every element. The third node, necessary for a parabolic interpolation, is subsequently introduced midway between the previous ones.

After sub-division, a reverse transformation produces the final Cartesian co-ordinates

$$\mathbf{x} = \frac{1-\xi}{2}\mathbf{x}_A + \frac{1+\xi}{2}\mathbf{x}_B \qquad \text{for straight lines (a)}$$

$$\mathbf{x} = \tfrac{1}{2}\xi(1-\xi)\,\mathbf{x}_A + (1-\xi^2)\,\mathbf{x}_C + \tfrac{1}{2}\xi(1+\xi)\,\mathbf{x}_B$$

for parabolic lines (b) (7)

Circular arcs and circumferences are interpolated as in equation 7(a) producing cylindrical co-ordinates that are then transformed to Cartesian ones.

Nodes are numbered sequentially from line to line after a previous check to avoid repetition in common nodes.

MACRO-ELEMENTS TYPOLOGY

The macro-element catalogue is ordered after three possibilities as follows:

I Plane elements: Interpolation is made in Cartesian co-ordinates
Non-plane elements: Interpolation is made in natural co-ordinates

II Triangular elements
Quadrilateral elements

III Intrinsic co-ordinates: Cylindrical, Spherical, Cartesian

The first class is motivated in order to eliminate problems like that presented in Fig. 3(a) where a regular sub-division in the mean line can be less convenient than a Cartesian one.

Class II is self-explanatory and finally class III attempts a better approximation of the geometry in cases which are quite usual in practical applications. A reason for doing this will be shown below.

MACRO-ELEMENT SUB-DIVISION

The basic case is a square equally sub-divided into two opposite sides, for instance $L_2 = L_4$ (Fig. 4.2).

To obtain the segmentation in horizontal rows, in number $L_2 - 1$, we define three schemes:

(1) Each segment in line j is related to another *one* in row $j+1$ (Fig. 4(b)).
(2) Each segment in line j is related to *two* in $j+1$ (Fig. 4(c)).
(3) Each segment in line j is related to n ($n2$) ones in $j+1$ (Fig. 4(d)).

If $L_1 \neq L_3$, for instance, $L_1 < L_3$ there are three possibilities

(a) $2L_3 < L_1$

(b) $2L_3 > L_1$ and $\dfrac{L_3}{L_1} < 2^{L_2} + 1$

(c) $\dfrac{L_3}{L} > 2^{L_2} + 1$

Case (a) is solved combining (1) and (2). Starting from row 3, with L_3 segments, one progresses to L_1 with procedure (2) until the row where the number L_1 is reached and then one follows with rule (1) (Fig. 5(a)). Case (b)

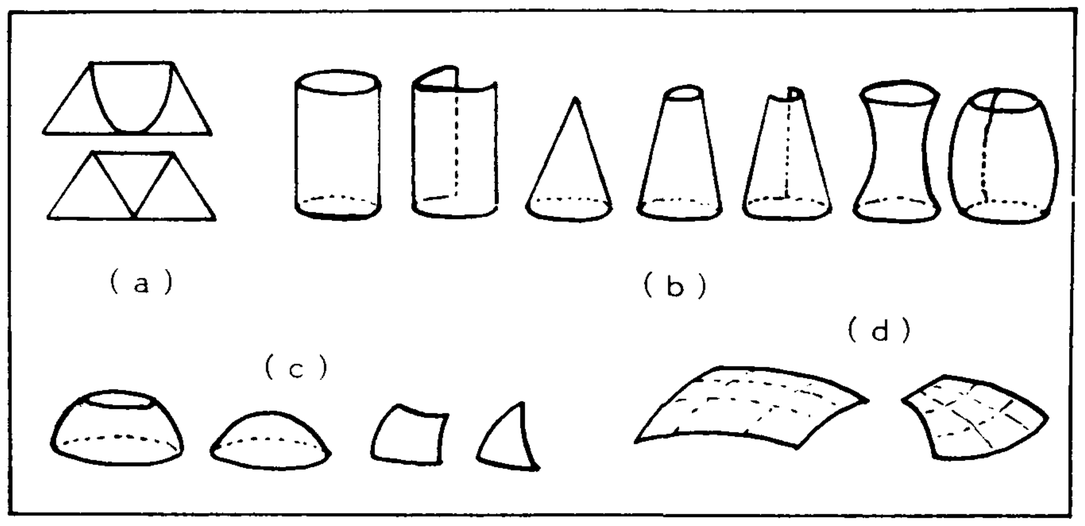


Figure 3. Macro-elements

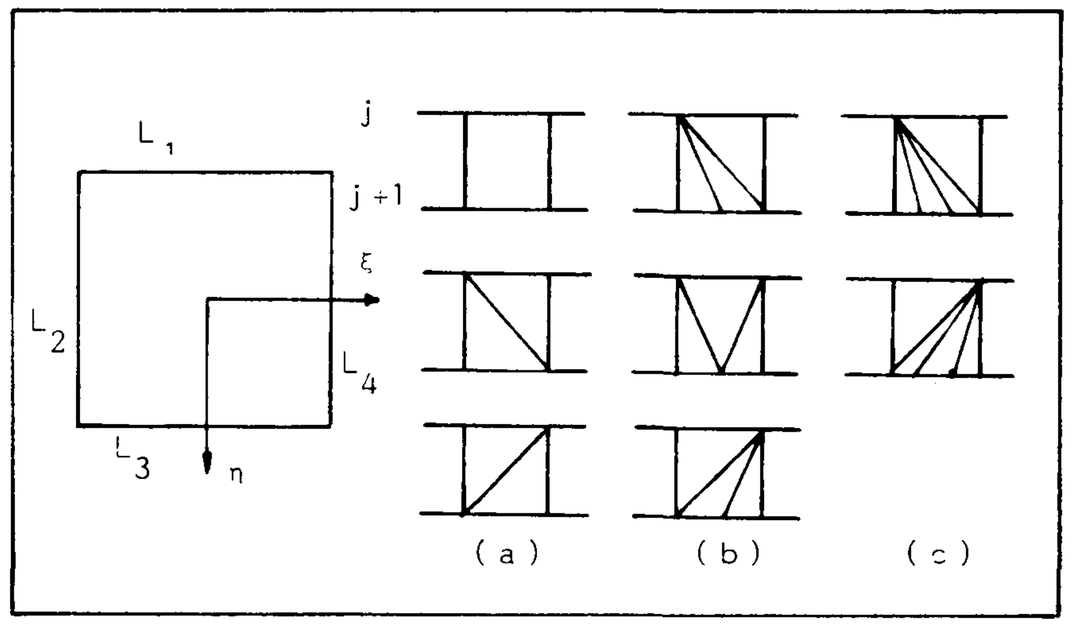


Figure 4. Macro-element subdivision

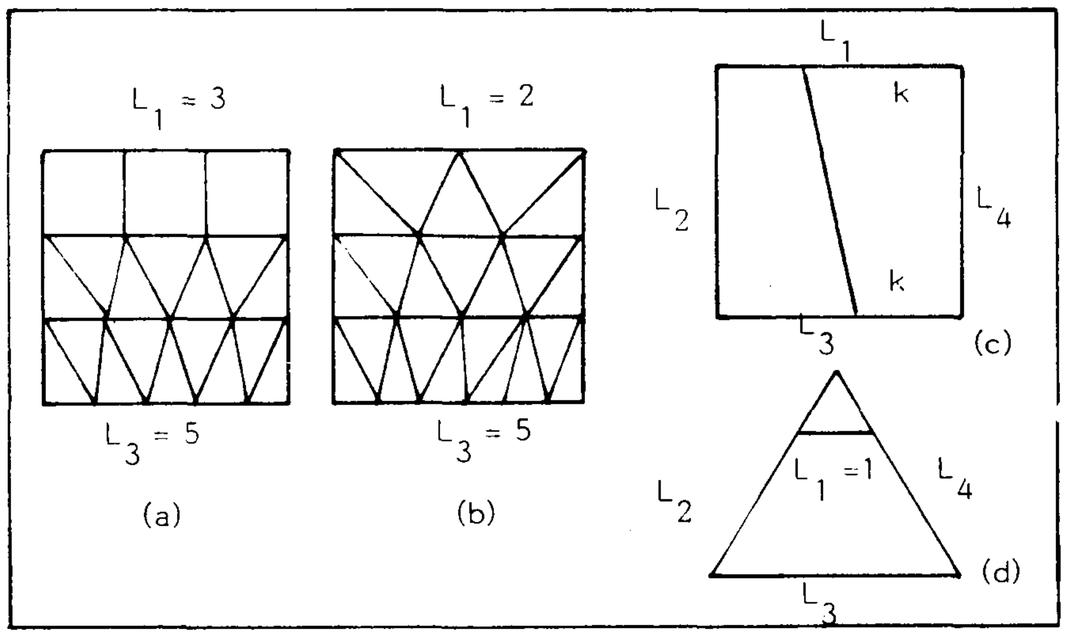


Figure 5. Sequential subdivision

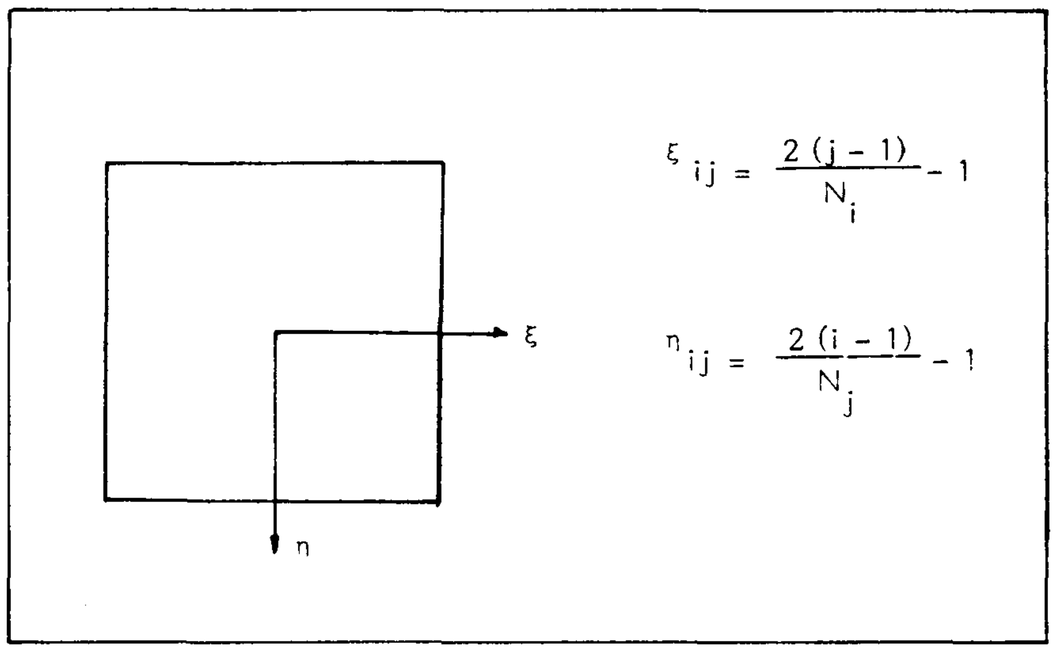


Figure 6. Natural coordinates

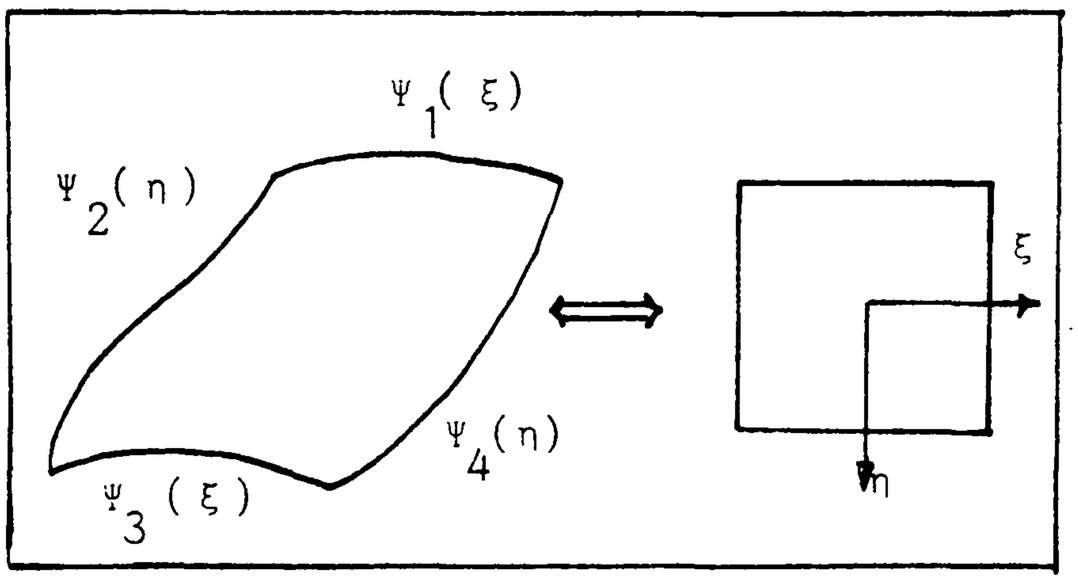


Figure 7. From natural to global coordinates

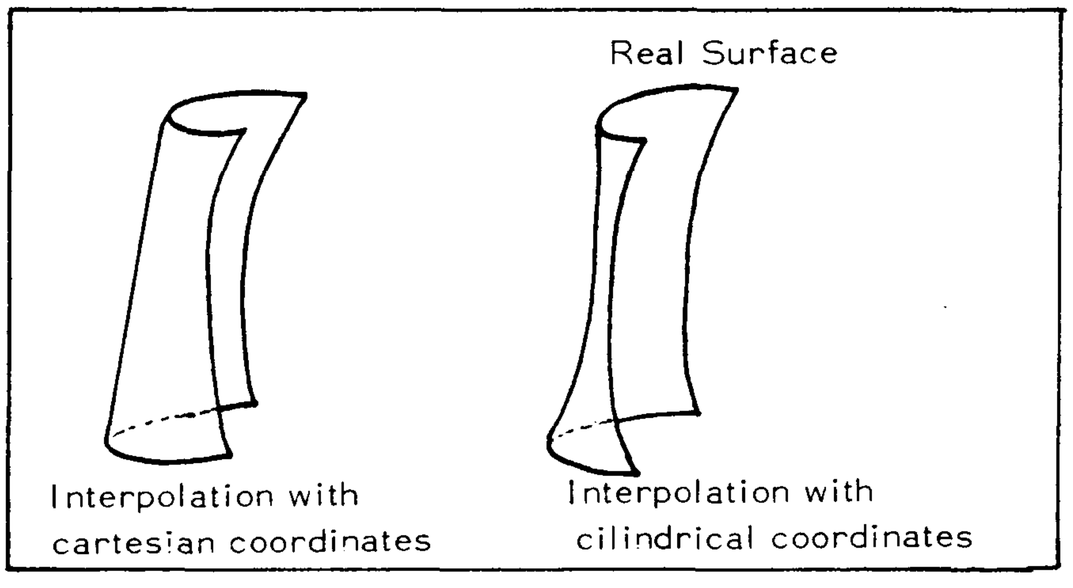


Figure 8.

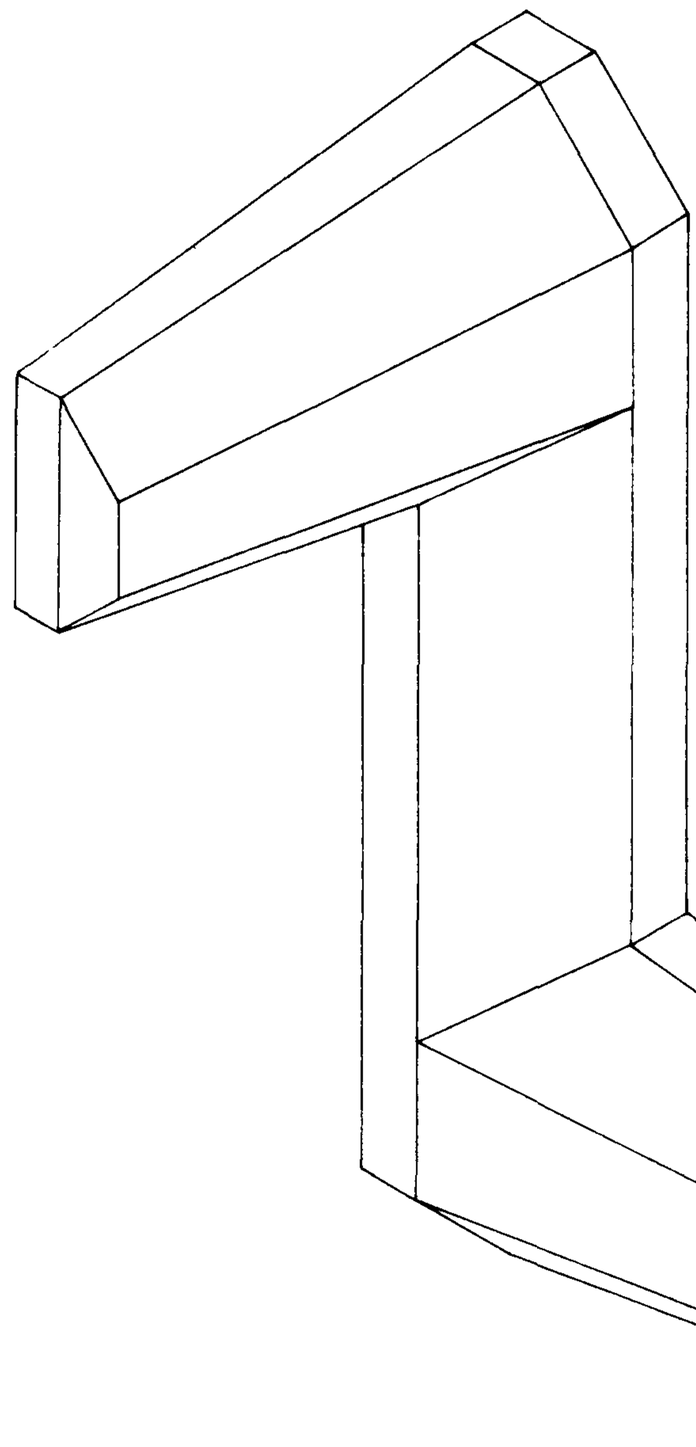


Figure 9. Dolos's Macro-elements

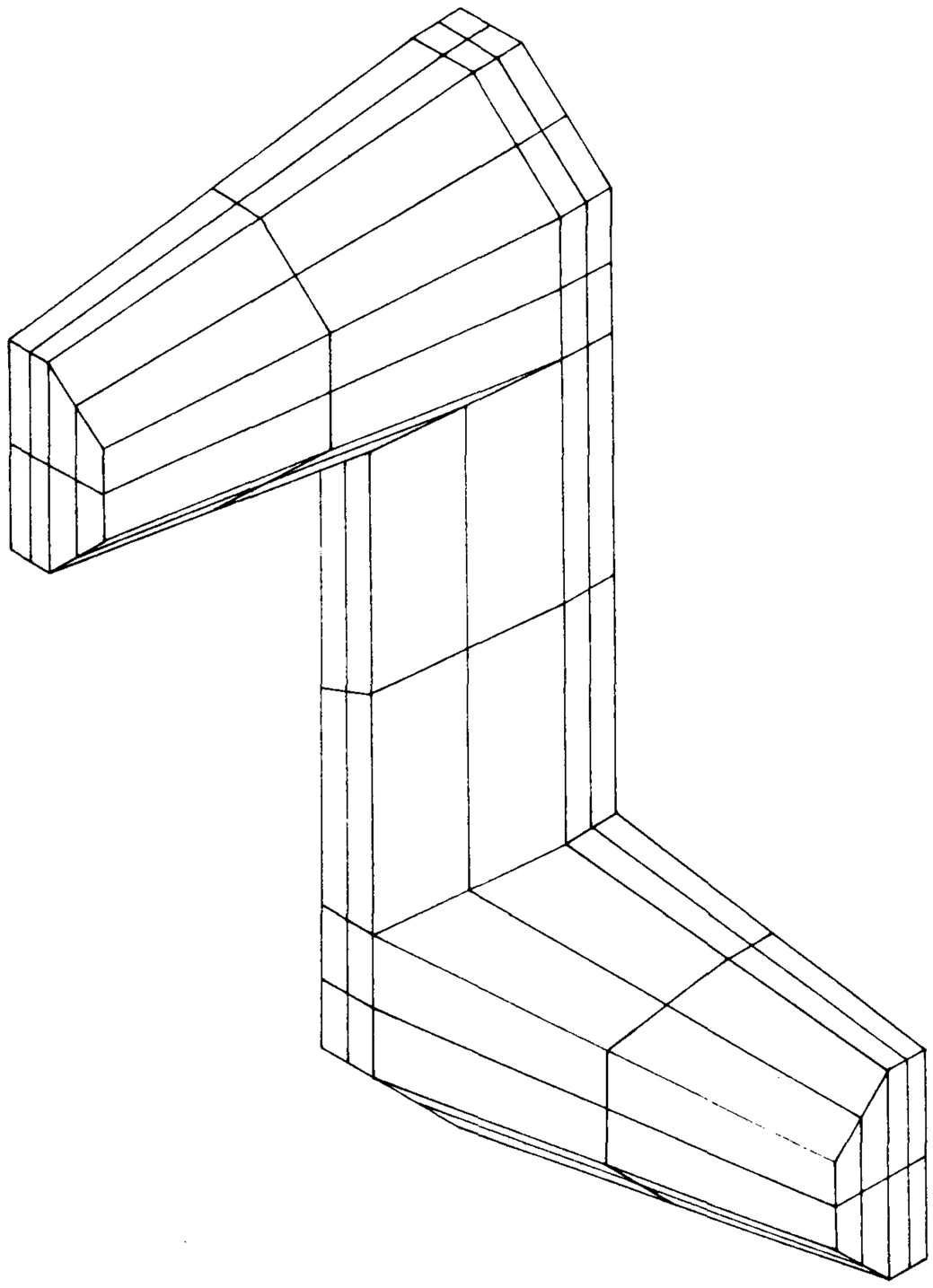


Figure 10. Dolos's final discretization

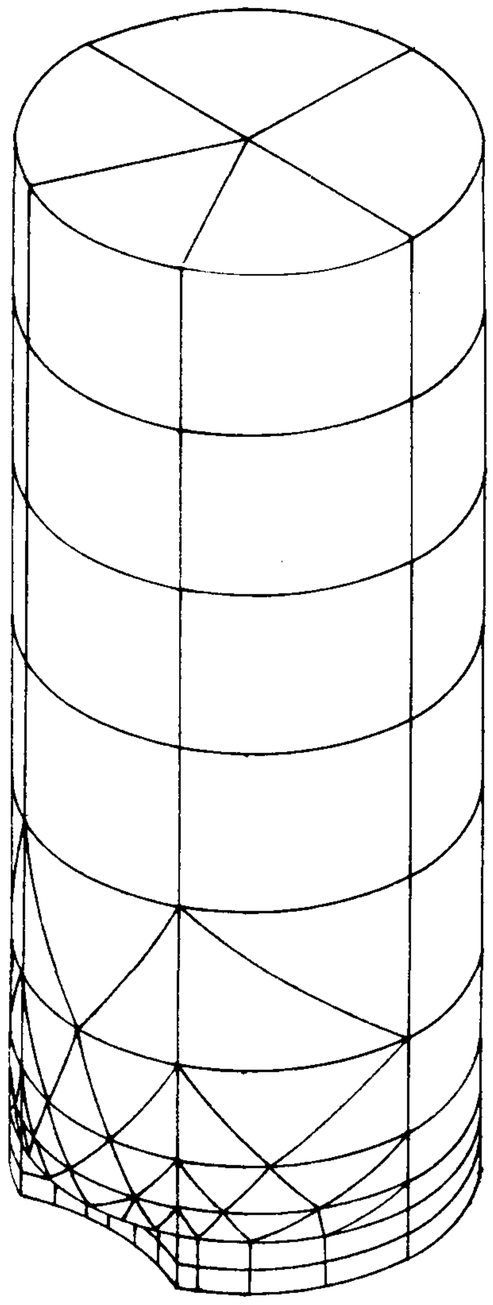


Figure 11. Notched cylinder

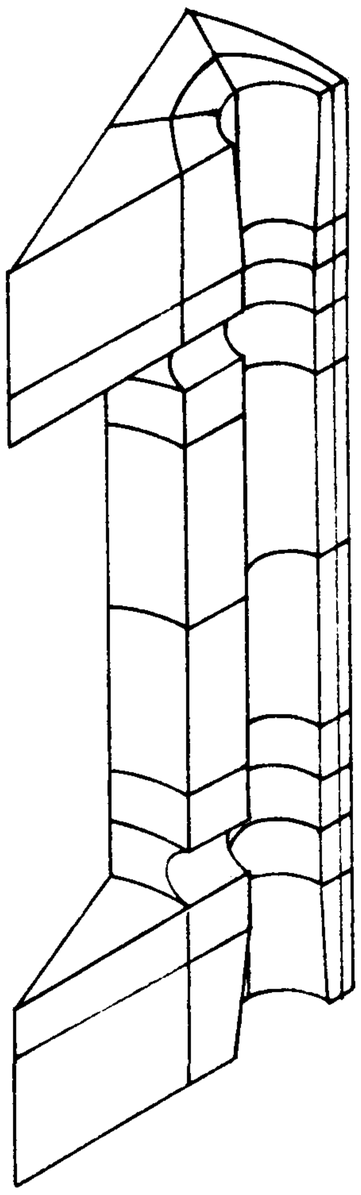


Figure 12. Segment of a vessel

is as before but now there will be several rules (2) inside each row. The first is situated in side 4 and the other one distributed as

$$p_k = \left(\frac{N_i}{n}\right) k, \qquad k = 1, 2, \ldots, n-1 \tag{8}$$

where n is the number of rules type 2 between rows i and iHi; N is the number of segments in row i; p_k is the position of rule (2) k as in Fig. 5(b).

Finally case (c) is solved with rules of type 3 and connecting three or more divisions to each one when the available rules type (2) are exhausted.

When the basic square has a different number of segments on every side it is easy to reduce it to the previous case (Fig. 5(c)) and the same is true for the triangle, by dividing off the tip of the figure into a single element (Fig. 5(d)).

Once the number of sub-divisions in every row has been established the computation of the natural co-ordinates of nodes is straightforward. For instance the node in column j and row i is obtained in Fig. 6.

When the mesh is non-uniform we use the distribution of extreme points (J, R) while the central ones are introduced as indicated above.

Nodal co-ordinates

Once the natural co-ordinates have been computed they are transformed to cylindrical, spherical or Cartesian ones by the transfinite method, described at the beginning of the paper (Fig. 7). In this way for instance a circumference in the plane $z = c$, from points $\theta = 0$ to $\theta = \pi/2$ can be defined as

$$\psi(\xi) = \begin{pmatrix} r \\ \theta \\ z \end{pmatrix} = \begin{pmatrix} R \\ \dfrac{\pi}{4}(\xi + 1) \\ C \end{pmatrix} \tag{9}$$

The simpler linear interpolation produces[5]

$$\begin{aligned} \mathbf{x} = {} & \tfrac{1}{2}((1-\eta)\,\psi,(\xi) + (1+\eta)\,\psi_3(\xi) + (1-\xi)\,\psi_2(\eta) \\ & + (1+\xi)\,\psi_4(\eta)) - \tfrac{1}{4}((1-\eta)(1-\xi)\,\mathbf{x}(-1,-1) \\ & + (1-\eta)(1+\xi)\,\mathbf{x}(1,-1) + (1+\eta)(1-\xi) \\ & \times\ \mathbf{x}(-1,1) + (1+\xi)(1+\eta)\,\mathbf{x}(1,1)) \\ & -1 \leqslant \xi \leqslant 1 \\ & -1 \leqslant \eta \leqslant 1 \end{aligned} \tag{10}$$

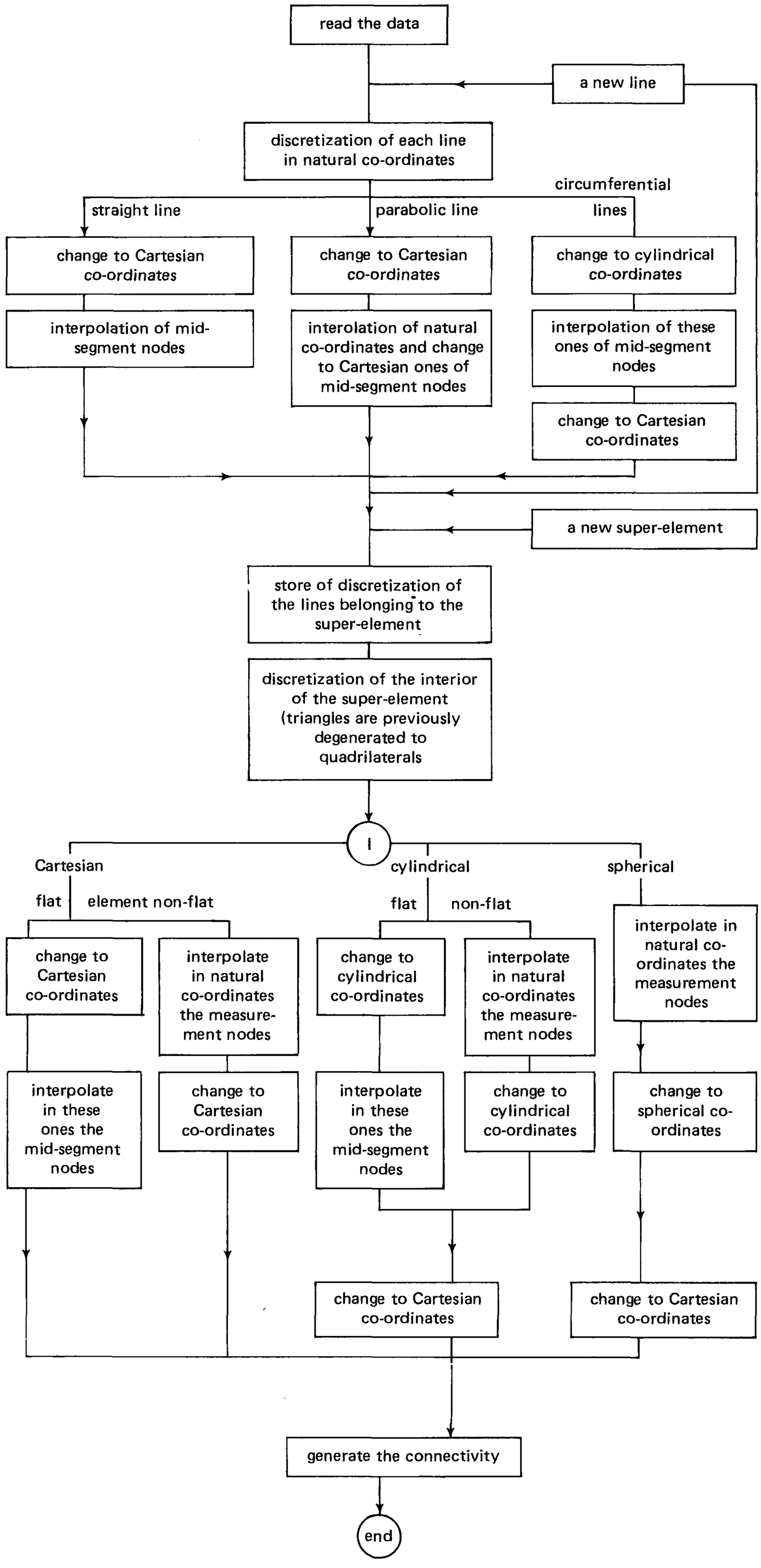
where the vector of co-ordinates $\mathbf{x}$ can be represented in either cylindrical, spherical or Cartesian co-ordinates. This blended interpolation allows a very accurate approximation of the geometry. The reason for using a different system of co-ordinates for macro-elements now becomes apparent: the interpolation among lines can de-figure basic features of the actual surface.

Once the co-ordinates of every point have been defined the final transformation of Cartesian ones concludes the problem.

Program description

The program PECET has been written in FORTRAN V and implemented in a UNIVAC 1180. The number of lines is about 15000, 1500 of which correspond to the mesh generation processor, and about 2000 to the plotting output processor. The program is composed of about 150 routines, 50 of which correspond to both processors.

A detailed description of the program can be found in Doblaré[6] and the flow chart of the generation processor is shown below.


read the data
a new line
discretization of each line in natural co-ordinates
straight line
parabolic line
circumferential lines
change to Cartesian co-ordinates
change to Cartesian co-ordinates
change to cylindrical co-ordinates
interpolation of mid-segment nodes
interolation of natural co-ordinates and change to Cartesian ones of mid-segment nodes
interpolation of these ones of mid-segment nodes
change to Cartesian co-ordinates
a new super-element
store of discretization of the lines belonging to the super-element
discretization of the interior of the super-element (triangles are previously degenerated to quadrilaterals
I
Cartesian
cylindrical
spherical
flat
element non-flat
flat
non-flat
change to Cartesian co-ordinates
interpolate in natural co-ordinates the measure-ment nodes
change to cylindrical co-ordinates
interpolate in natural co-ordinates the measure-ment nodes
interpolate in natural co-ordinates the measurement nodes
interpolate in these ones the mid-segment nodes
change to Cartesian co-ordinates
interpolate in these ones the mid-segment nodes
change to cylindrical co-ordinates
change to spherical co-ordinates
change to Cartesian co-ordinates
change to Cartesian co-ordinates
generate the connectivity
end

RESULTS

The following figures present some examples, concerning different aspects of the program.

The first one represents the analysis of a Dolos, which was one of the examples presented in (2) without automatic generation of the mesh. The size of the data file necessary for an equivalent discretization, has been decreased in a 50%. Naturally for more refined discretizations the decreasing is much more important.

The second example corresponds to a cylinder with a circular crack showing the refinement of the mesh near the crack. Only macro-elements and lines were needed for the definition of these discretizations which were imposed by elements and nodes.

The third example, which can be seen in Fig. 11 represents a quarter of a nuclear reactor vessel with a complex geometry which need the use of parabolic macro-elements for an accurate definition.

CONCLUSIONS

An automatic Mesh Generation Preprocessor for BE Programs with a considerable of capabilities has been developed.

This program allows almost any kind of geometry and tipology to be defined with a small amount of external data, and with an important approximation of the boundary geometry.

Also the error checking possibility is very important for a fast comprobation of the results.

REFERENCES

1 Doblaré, M. and Alarcón, E. *A Three-Dimensional B.E.M. Program*, A Handbook of Finite Element Systems (Ed. by C. A. Brebbia), CML Publications, 1981

6 Doblaré, M. Formulación del Método de los Elemetnos de Contorno con Interpolación Parabólica, PhD Thesis, Pol. Univ. Madrid, 1981 (in Spanish)

2 Birkhoff, G. & Gordon, W. J. 'The draftsman's and related equations' *J. Approx. Theory*, 1968, **1**, 191–208

3 Gordon, W. J. Blending-function methods of bivariate and multivariate interpolation and approximation, *SIAM J. Num. Anal.* 1971, 8, 158

4 Gordon, W. J. and Hall, C. A. Construction of curvilinear coordinates systems and application to mesh generation, *Int. J. Num. Meth. Eng.* 1973, 7, 461

5 Gordon, W. J. and Hall, C. A. Transfinite elements methods blending-function interpolation over arbitrary curved element domains, *Num. Meth.* 1973, **21**, 109

6 Griffith, J. G. Drawing pictures of solid objects using a graph plotter, *Comp. J. Vol.* 1973. **16**, 35

7 Griffith, J. G. Eliminating hidden edges in line drawings, *Comp. Aided Design*, 1979, **11**, 71

8 Haber, R. *et al.* A general two-dimensional, graphical finite element preprocessot utilizing discrete transfinite mapping, *Int. Jour. for Numerical Methods in Eng.* 1981, **17**, 1015

9 Haber, R. *et al.* Discrete transfinite mappings for the description and meshing of three-dimensional surfaces using interactive computer graphs, *Int. Jour. for Numerical Methods in Eng.* 1982, **18**, 41

10 Imafuku, I. *et al.* A generalized automatic mesh generation scheme for finite element method, *Int. J. Num. Meth. Eng.* 1980, **15**, 713

11 Zienkiewicz, O. C. *The Finite Element Method*, McGraw-Hill, 1979