SHAPE OPTIMAL DESIGN ON AN ENGINEERING WORKSTATION

C. Fleury[1] and D. Liefooghe[2]

Mechanical, Aerospace and Nuclear
Engineering Department
University of California, Los Angeles
Los Angeles, California 90024

Abstract

Shape optimal design of two-dimensional structures discretized in finite elements is investigated with emphasis on creating an interactive Computer Aided Design capability. The paper will first provide a short description of the approach followed to create an appropriate geometric model, involving a relatively small number of design variables. Next the important question of how to perform the sensitivity analysis is discussed. A general procedure is proposed, that takes advantage of the parametric modeling concepts used to create the design model. The sensitivity analysis can then be readily accomplished by a semi-analytical approach, i.e., a finite difference approximation of the stiffness matrix is employed at the element level. The numerical solution to the shape optimal design problem is obtained by resorting to an efficient optimizer based on the convex linearization method.

A two-dimensional pre- and post-processing module is described, which is aimed at performing interactive shape optimal design on an engineering workstation. This module exhibits some innovative visualization techniques that should highly facilitate the task of the designer. Examples of application to classical shape optimal design problems are offered to illustrate the various functions of the new interactive optimization system. Although the current capability is restricted to simple 2-D structures, it is based upon very general concepts that could be readily implemented in general purpose finite element systems such as MSC/NASTRAN.

[1] Associate Professor
[2] Graduate Student

Paper presented at the MSC/NASTRAN Users' Conference, Universal City, California, March 20-21, 1986.

INTRODUCTION

Shape optimization techniques are concerned with improving some characteristics of a structure by modifying its boundaries in an optimal way. Most often such a shape optimal design problem consists of minimizing an objective function subject to constraints insuring the feasibility of the structural design. Mathematically the optimization problem considered in this paper can be written in the following form:

$$\text{minimize } f(x) \tag{1}$$

subject to the constraints :

$$c_j(x) \leqq \bar{c}_j \qquad j=1,m \tag{2}$$

$$\underline{x}_i \leqq x_i \leqq \bar{x}_i \qquad i=1,n \tag{3}$$

The objective function (1) is a non-linear function of the design variables x_i. It usually represents a structural characteristic to be minimized (e.g. weight). The non-linear inequalities (2) are the behavior constraints, which impose limitations on structural response quantities (e.g. stresses and displacements). The design variables must also be bounded by the side constraints (3), where $\underline{x}_i$ and $\bar{x}_i$ are respectively the lower and upper limits that reflect fabricational and analysis validity considerations.

Care must be exercised in the selection of the design variables x_i. The coordinates of the boundary nodes of the finite element model can be used as design variables (a common practice in early work on shape optimization, e.g. Ref.[1])). This choice exhibits however many severe drawbacks. The set of design variables is very large and the cost and difficulty of the minimization process increase. It has a tendency to generate unrealistic designs due to the independent node movement and additional constraints avoiding such designs are difficult to cope with. Moreover an automatic mesh generator is necessary to maintain the mesh integrity throughout the optimization process. The solution is to avoid a one-to-one correspondence between the finite element model and the design variables.

One of the ways to achieve this goal is to use the "design element" concept. In this approach the structure is decomposed into a few subregions of simple geometry. These subregions are described in a compact way by using a limited number of control nodes (or master nodes). Each region consists of several finite elements. During the optimization process the geometry of conveniently selected subregions is allowed to change: these regions are called design elements. The movements of the corresponding control nodes are the design variables.

The concept was initially introduced in Ref.[2] where two-dimensional isoparametric finite element interpolation functions were used to describe the design element boundary. Recently blending functions commonly used in computer graphics for interactive generation of curves and surfaces (Bezier, B-splines) have been proposed to describe the boundaries [3,4]. This is also the approach followed in the present paper. The shape variables are thus the positions of the master nodes which control two families of curves, whose cartesian product defines the design element.

This formulation lends itself well to shape optimal design problems. The blending functions provide a large flexibility for the geometric description. With the B-spline formulation, boundary regularity requirements are automatically taken into account. In addition a few design elements are generally sufficient to fully describe the regions that are modified during the optimization process. The optimization problem has therefore a limited amount of design variables, and it provides an elegant way to perform the sensitivity analysis on the basis of finite difference techniques, as described further in this paper. However it should be noted that an analytical formulation of the sensitivity derivatives can also be established [4].

Finally it is relatively straightforward to generate a suitable finite element mesh and to maintain its integrity throughout the optimization process. Indeed the design element can be mathematically defined as the cartesian product of two families of curves. This provides an analytical interpolation scheme, which permits determining the coordinates of any point (finite element node) inside the design element or on its boundaries. A regular mesh is initially constructed in the curvilinear coordinate system of the design element. Next coordinate transformations are applied to obtain the mesh in the real design element (see Fig. 1).

This representation, wherein the FEM mesh can be directly derived from the coordinates of the control nodes, leads to the distinction between a design model and an analysis model. The design model consists of the small number of design elements, with their geometry determined by the control nodes, and the fixed subregions. By entering a relatively small number of design elements, it is possible to create a compact design model that describes well the structure to be optimized. The analysis model is the finite element model, characterized by the node coordinates of the mesh, the types and material properties of the elements, the applied loads and boundary conditions, etc... The analysis model can directly be derived from the design model at any stage of the iterative optimization process, because of the adopted internal parametric representation. Therefore the concept of design model is one level above that of analysis model. Ultimately, when the discretization and analysis schemes will become sufficiently reliable, it can be expected that only the design model will remain accessible to the user, most of the analysis model being completely transparent.

The foregoing optimization concepts have been implemented in a finite element system made up of two parts. The first module is used in batch mode to perform the structural analysis and its associated sensitivity analysis. It corresponds to an essential ingredient of the research effort described in this paper ,i.e., to demonstrate the validity of a finite difference approach for generating the shape sensitivity results. The second module is an interactive optimum design system that uses the sensitivity coefficients produced by the first module. This module contains innovative graphics display capabilities that should considerably facilitate the task of a designer willing to optimize a structural shape. Substantial research and development efforts have been devoted to devising this new interactive optimizer.

OPTIMIZATION METHOD

In this study the finite element method is used to analyze the behavior of a structure i.e. weight, displacements and stresses. In order to improve a structural design the first derivatives of these quantities with respect to the design variables must be evaluated: this is the purpose of the sensitivity analysis. These derivatives are subsequently used by an optimizer to find a better design. Both the sensitivity analysis method and the optimizer used in this investigation will be described.

Sensitivity analysis

The derivatives relevant for the shape optimal design problem are computed in this work by a finite difference scheme embedded into a finite element solver. The current implementation is restricted to two-dimensional elastic structures in plane stress or plane strain, modeled with isoparametric eight-node elements. However it is important to emphasize that the same concepts can be employed to deal with more complicated design problems (e.g. solid models) or more sophisticated finite elements (e.g. three-dimensional structure involving plates and shells). In fact the approach presented below is largely independent of the types of finite elements used in the analysis model, and therefore it could be readily implemented in a large scale general purpose FEM system such as MSC/NASTRAN [5], or even in a Personal Computer package such as MSC/PAL [6].

This section shows how the sensitivity analysis can be incorporated into any FEM program with a very little amount of modifications, and it explains in detail how the weight, displacement and stress derivatives are computed. In a displacement finite element approach, the structural analysis consists of solving the equilibrium equations:

$$[K]\{u\} = \{F\} \tag{4}$$

Differentiating this set of equations with respect to the design variable x_i, gives:

$$[K]\left\{\frac{\partial u}{\partial x_i}\right\} = \left\{\frac{\partial F}{\partial x_i}\right\} - \left[\frac{\partial K}{\partial x_i}\right] \{u\} \tag{5}$$

This equation means that the displacement derivatives can be obtained by solving the original set of equations with another loading $\{\tilde{g}_i\}$, the so-called pseudo-load:

$$\{\tilde{g}_i\} = \left\{\frac{\partial F}{\partial x_i}\right\} - \left[\frac{\partial K}{\partial x_i}\right] \{u\} \tag{6}$$

The number of additional pseudo-load vectors is n*c, where n is the number of design variables and c, the number of applied loading cases. The element stresses are expressed in terms of the displacements by the equation:

$$\{\sigma\} = [T]\{u\} \tag{7}$$

so that the stress derivatives become:

$$\left\{\frac{\partial \sigma}{\partial x_i}\right\} = [T]\left\{\frac{\partial u}{\partial x_i}\right\} + \left[\frac{\partial T}{\partial x_i}\right]\{u\} \qquad (8)$$

The total weight of the structure is calculated by adding the contributions of each isoparametric element. The individual element weight is itself obtained through a conventional numerical integration scheme:

$$W = \sum W_{el} = \sum \sum_{i}^{r} \rho w_i \det(J(\xi_i)) \qquad (9)$$

where r denotes the number of Gauss points.

The finite difference approach is introduced by perturbating each design variable x_i by a small amount dx_i and regenerating the mesh for the modified structure. This means that each element stiffness matrix must be computed again as many times as the number of design variables. The total weight $\tilde{W}$ of the perturbated structure can now be recalculated using Eq. (9).

The derivative of the total weight can then be approximated:

$$\frac{\partial W}{\partial x_i} = \frac{\tilde{W}(x_i+dx_i)-W(x_i)}{dx_i} \qquad (10)$$

After assembling the regenerated element stiffness matrices, the global stiffness matrix of the perturbated structure can be obtained. The pseudo-loads can then be computed from Eq. (6). Approximating the derivative of the global stiffness matrix with respect to the design variable x_i by its finite difference derivative, it comes:

$$\{\tilde{g}_i\} = - \frac{[\tilde{K}(x_i+dx_i)] - [K(x_i)]}{dx_i}\{u\} \qquad (11)$$

or,

$$\{\tilde{g}_i\} = \frac{[K(x_i)]\{u\} - [\tilde{K}(x_i+dx_i)]\{u\}}{dx_i} \qquad (12)$$

The pseudo-loads are thus calculated as the difference between two vectors which result from the multiplication of the original displacement solution $\{u\}$ with respectively the original assembled stiffness matrix and the assembled stiffness matrix of the perturbated structure.

The displacement derivatives can now be obtained by solving the equilibrium equations with these pseudo-loads as additional loading cases:

$$\left\{\frac{\partial u}{\partial x_i}\right\} = [K]^{-1}\{\tilde{g}_i\} \qquad (13)$$

Any particular component can be selected out by the equations:

$$d = \langle b\rangle \{u\} \qquad (14)$$

$$\left\{\frac{\partial d}{\partial x_i}\right\} = \langle b\rangle \left\{\frac{\partial u}{\partial x_i}\right\} \qquad (15)$$

where $\langle b\rangle$ denotes a unit vector.

Finally the element stress derivatives have to be computed. The stresses for the perturbated structure are given by:

$$\{\tilde{\sigma}\} = [\tilde{T}]\{\tilde{u}\} \qquad (16)$$

However, as the displacement gradients are calculated by the pseudo-loads technique, the equilibrium equations are not solved for the perturbated structure and thus $\{\tilde{u}\}$ are not available. This displacement solution is approximated by:

$$\{\tilde{u}\} = \{u\} + \left\{\frac{\partial u}{\partial x_i}\right\} dx_i = \{u\} + \{\Delta u\} \qquad (17)$$

Likewise the stress matrix $[\tilde{T}]$ is approximated as:

$$[\tilde{T}] = [T] + [\Delta T] \qquad (18)$$

so that Eq. (16) becomes:

$$\begin{aligned}\{\tilde{\sigma}\} &= ([T]+[\Delta T])(\{u\}+\{\Delta u\}) \\ &= [T]\{u\} + [T]\{\Delta u\} + [\Delta T]\{u\} + [\Delta T]\{\Delta u\} \qquad (19)\end{aligned}$$

The last term is one order of magnitude smaller than the other terms and can therefore be neglected. Using Eq. (7) the first term can be identified as representing the stresses of the original structure. Hence:

$$\{\tilde{\sigma}\} = \{\sigma\} + [T]\{\Delta u\} + [\Delta T]\{u\} \qquad (20)$$

These stresses are used to get an approximation to the stress derivatives:

$$\left\{\frac{\partial \sigma}{\partial x_i}\right\} = \frac{\{\tilde{\sigma}\} - \{\sigma\}}{dx_i} = [T]\left\{\frac{\Delta u}{\partial x_i}\right\} + \frac{[\Delta T]}{dx_i}\{u\} \qquad (21)$$

This is a finite difference equivalent of Eq. (8). The importance of the above derivation is however that since the finite element program used does not employ a stress matrix [T], Eq. (8) cannot be used explicitly. Rather the stresses in the FEM program are computed for each Gauss integration point inside the element, without calculating an average stress for the whole element. This is done in an element subroutine, which uses geometry information on the element (shape functions) and the displacement solution. This corresponds to $\{\sigma\}=[T]\{u\}$, since the stress matrix incorporates this geometric information.

According to the above derivation we can now proceed as follows. Using the original mesh and the displacement solution of the original structure $\{u\}$, the stresses are evaluated at the Gauss points. With Eq. (19) in mind, this procedure can be repeated using the perturbated geometry and the approximated displacement solution for this modified structure. The first will give the exact values for the Gauss point stresses in the original structure, while the latter provides a good approximation for the values of the Gauss point stresses in the perturbated structure.

For simplicity the element stress will be taken as the stress at the center Gauss point. Using the two sets of element stresses, a finite difference approximation to the element stress derivative can be obtained as follows:

$$\{\frac{\partial \sigma}{\partial x_i}\} = \frac{\{\tilde{\sigma}\} - \{\sigma\}}{dx_i} \qquad (22)$$

Any particular component can be selected out by the equations:

$$\sigma = \langle b \rangle \{\sigma\} \qquad (23)$$

$$\frac{\partial \sigma}{\partial x_i} = \langle b \rangle \{\frac{\partial \sigma}{\partial x_i}\} \qquad (24)$$

The foregoing developments provides thus a means to compute the derivatives of the structural weight, of the displacements, and of the stresses by finite differencing. The steps to be followed are schematized in the block diagram of Fig. 2.

Optimizer

The CONLIN optimizer [7,8] was used to solve the optimization problem. It is a specially well suited optimizer for structural optimization, based on a convex approximation scheme.

The convex linearization method, and the associated CONLIN optimizer, exhibits many interesting features and it is applicable to a broad class of structural optimization problems. The method employs mixed design variables (either direct or reciprocal) in order to get first order, conservative approximations to the objective function and to the constraints. In this approach the primary optimization problem is replaced with a sequence of explicit subproblems having a simple algebraic structure. Each subproblem is convex and separable, and therefore it can be readily solved by using a dual method formulation.

The dual problem consists of maximizing a quasi-unconstrained dual function depending only on the Lagrangian multipliers associated with the behavior constraints. These multipliers are in fact the dual variables subject to simple non-negativity constraints. The CONLIN algorithm can be viewed as a generalization of well established approaches to pure sizing structural optimization problems (no change in geometry), namely "approximation concepts" and "optimality criteria" techniques [9]. As such it is capable of addressing a broader class of problems with considerable facility of use. The method has recently demonstrated its ability to solve efficiently shape optimal design problems involving two-dimensional structures in plane strain or plane stress [3,4] as well as trusses of variable configuration [10].

At each successive iteration point, CONLIN only requires evaluation of the objective and constraint functions and their first derivatives with respect to the design variables. These quantities are provided by the finite element analysis system equipped with sensitivity analysis features. The CONLIN optimizer will then select by itself an appropriate approximation scheme on the basis of the sign of the derivatives. CONLIN usually generates a nearly optimal design within less than 10 iterations, most often by producing a sequence of steadily improving feasible designs. Finally it should be emphasized that the CONLIN method is simple enough to lead to a relatively small computer code, well organized to avoid high còre requirement.

The initial optimization problem is thus transformed in CONLIN into a sequence of explicit subproblems, which are solved in the dual space. The efficiency of this dual formulation is due to the fact that the dimensionality of the dual space is relatively low and depends on the number of active constraints at each design iteration. CONLIN advantages, which make it especially well suited to the undertaken study include:

- it does not demand a high level of accuracy for the sensitivity analysis results, because it is based on conservative approximation concepts, allowing them to be obtained from finite difference techniques;
- it has an inherent tendency to produce a sequence of steadily improving feasible designs and usually generates the optimal design within less than 10 FEM analyses;
- each CONLIN iteration is accomplished very rapidly, even for relatively large scale problems; this is an important feature within an interactive environment.

INTERACTIVE OPTIMUM DESIGN SYSTEM

As reliable optimization algorithms, such as CONLIN, are now available to solve shape optimal design problems, substantial development efforts must be devoted to implement an appropriate user interface. The ultimate goal is to devise suitable tools for a designer to effectively incorporate optimization concepts into the real design cycle. Indeed an important aspect of finite element analysis and optimization capabilities should be their ability to really help the user in accelerating the design process. The interactive engineering design system of the future should be able to produce extensive graphical outputs, displaying in an expressive way meaningful results.

These graphics display capabilities should be organized to be easy to use interactively. Industrial applications namely reveal that it is not generally possible to optimize a structure in one single computer job. Frequently it is necessary to improve the structural model, to modify the set of retained behavior constraints (because some constraints, initially not critical, become critical), or to relax some constraints if their maximum values are too severe and no feasible solution can be found. Sometimes it is needed to redefine the design model, or to introduce additional geometric constraints such as tangency requirements. It is also quite interesting to display intermediate analysis results and stop or correct the optimization procedure. This leads to a splitting of the tasks corresponding to the flow chart in Fig. 3.

The finite element analysis of a structure and the sensitivity analysis are time consuming tasks, which could be typically performed during the night as batch job. The task of redesigning on the basis of these results is typically an interactive job performed during the day. The designer has then the choice to accept the new design and send it back to the finite element optimization code for a new iteration, or to intervene in the optimization process and make some of the data adjustments previously mentioned. Note however that the optimization computer code would always be able to perform several successive optimization iterations in order, for instance, to conclude an optimization process.

This scheme requires thus an interactive engineering design system, capable of displaying all the relevant information concerning the modified design. This interactive system can be viewed as a post-processing module, called when the design optimization process has been interrupted after a finite element analysis is completed (including the sensitivity analysis). The system should also contain pre-processing capabilities to allow the user to define the optimization data and make data adjustments as the optimization process continues.

At the present stage of the development efforts, a computer program has been developed, largely functioning as a post-processor. Is is envisaged that the pre-processing functions will be implemented in the near future. The system contains most of the conventional graphics displays usually found in pre- and post-processing programs:

- representation of finite element mesh, applied loads and boundary conditions (i.e. characteristics of analysis model);

- undeformed and deformed geometry plots (i.e. displacement analysis results);

- color-coding of elements based on specific stress components (i.e. stress recovery results).

In addition innovative graphics capabilities were implemented in connection with the new optimization concepts:

- visualization of the design model with the design variable locations, together with their lower and upper limits;

- visualization of the analysis model (finite element mesh);

- the CONLIN optimizer can be supplied with the sensitivity data produced by the finite element optimization code; the optimizer results can be immediately verified with the design model plot or analysis model plot of the modified geometry.

Note that this interactive optimization feature could be implemented because the CONLIN method is simple enough to lead to a relatively small, well organized computer code.

The foregoing functions allow the user to see the new shape generated after each iteration and they represent therefore a very valuable tool to verify the validity of the results produced by the optimizer. In addition, the user can interact with the system, by modifying the positions of some control nodes and examining the effect on the design model, as well as on the analysis model. This is exactly the way the batch program uses to calculate the sensitivities through a finite difference scheme.

Other graphic display functions include:

- evolution plots, representing the values of the objective function, design variables and constraints in terms of the number of iterations;

- visualization of slices in the design space: the user selects significant design variables and the program plots the corresponding 2-D design space (contours of objective function, constraint surfaces defining the feasible domain, location of optimum design, etc...); these graphics displays are made on the basis of the convex linearization scheme, which constitutes an excellent explicit approximation.

It is also possible to produce simultaneously a plot of the design model and a plot of the design subspace. The user selects on the design model the design variables to form the design subspace. He has the opportunity of highlighting a behavior constraint both in the design model plot and in the design space plot. He can have the lower and upper limits on the design variables displayed both in the two plots.

APPLICATIONS

The purpose of this section it to demonstrate the effectiveness of the concepts presented in this paper. As previously mentioned these concepts are quite general and could be readily introduced into a large scale FEM system such as MSC/NASTRAN for the analysis and sensitivity analysis, and MSC/GRASP for the creation of the geometric design model. However the current implementation at UCLA is restricted to two-dimensional structures in plane stress or plane strain modeled with isoparametric eight-node elements. The computer program is oriented toward the use on a Personal Computer (IBM PC-AT type), and as such it might be a worthwhile addition to the capabilities of a FEM analysis package such as MSC/PAL.

The examples given below are not meant to represent practical applications. Rather they have been devised to illustrate the various functions of our interactive shape optimization system.

Quarter plate under central force

The first example is concerned with the structure defined in Fig. 4a. It corresponds to a quarter of a plate under a central acting force. The goal is to determine the shape of the side BC which minimizes the weight of the structure, with an upper bound on the displacement at node A. The optimization process was started from an initially feasible design and from an initially infeasible design. The detailed results are reported below.

The design model of the quarter plate is made up of a subregion around the side BC with changing geometry (design element) and a subregion containing the rest of the structure with fixed geometry. The side BC, which represents the moving boundary of the design element, is described by a B-spline of order 4 with 6 control nodes. The design element boundary in the other direction is represented by a B-spline of order 2 with 2 control nodes (linear). The design element is defined as the cartesian product of these two families of curves and has thus 12 control nodes. As illustrated in Fig.4b these control nodes belong to one of three different categories: fixed nodes, moving nodes and internal nodes.

The unknowns of the problem are the positions of the 4 moving control nodes. These positions are determined by the distances from their respective fixed reference poles. Hence these 4 distances are the design variables of the optimization process. Making use of the problem symmetry only 2 decision variables remain . The only design constraint is the displacement of the node A in the direction of the applied load. For practical reasons, this constraint can be replaced by an equivalent constraint on either the horizontal or the vertical displacement of node A, which are equal because of symmetry. In addition side constraints are specified for the decision variables, preventing unreasonable large displacements of the moving control nodes.

This leads to the following problem statement:

$$\begin{aligned} &\min \quad W\ (x_1,x_2) \\ \text{s.t.} \quad & u_a\ (x_1,x_2) \leqq \bar{u} \\ & \underline{x}_1 \leqq x_1 \leqq \bar{x}_1 \\ & \underline{x}_2 \leqq x_2 \leqq \bar{x}_2 \end{aligned}$$

Fig. 4c shows the displacements obtained from a finite element analysis of the original structure (deformed geometry plot). The initial weight is equal to 3.4865 kg; the horizontal and vertical displacement of node A are both equal to 0.16039 mm. By adopting a displacement limit $\bar{u}$ = 0.16200 mm, the initial structure corresponds to a feasible starting point. Before calling the CONLIN optimizer a sensitivity analysis must be performed in order to evaluate the first derivatives of the displacement u with respect to the design variables x_1 and x_2.

The design space corresponding to the first subproblem created by CONLIN is displayed in Fig. 4d. The two axes represent the design variables x_1 and x_2. In this specific case the CONLIN optimizer generates a linear approximation of the structural weight, while the displacement constraint is linearized with respect to the reciprocals of the design variables. Hence the contours of the weight objective function are parallel straight lines while the constraint surface is drawn as a curve. Clearly the optimum lies at the point where this curve is tangent to a constant weight line. This optimum point corresponds to a new design of the structure, i.e. the positions of the control points have been changed by the optimizer, leading to an improved shape of the side BC (see Fig. 4e). From the modified design model, the finite element mesh is updated, the structure is reanalyzed, and the optimizer called again in order to still improve the design. This process is repeated until convergence is achieved to an optimal shape.

The iteration history of the optimization process is summarized in Fig. 5a. One can observe the quick convergence of the CONLIN optimizer. The optimal shape is displayed in Fig. 5b.

A second optimization run was undertaken with a slightly different specification. The displacement limit was reduced to $\bar{u}$ = 0.15950 mm, making the initial structure with $u_a^x = u_a^y$ = 0.16039 mm an infeasible starting point. The iteration history produced by the CONLIN optimizer is summarized in Fig. 5c and the optimal shape is displayed in Fig. 5d.

Optimization of a hole in a biaxial stress field

The second problem is concerned with the minimization of stress concentrations in a structural component. This example has been used as key test-case to evaluate the CONLIN optimizer when supplied with finite difference gradients, because an analytical solution is available.

The plate with hole displayed in Fig. 6a is loaded with a combined tension in two perpendicular directions. For this initial configuration, stress concentrations will occur at the boundary of the hole. The problem is to determine the shape of the hole for which the tangential stresses around the hole are uniform. It can be proven that the shape of the hole with this property is elliptic, with the ratio k of the axes being equal to the ratio of the applied tensile stresses [11]. The value of the tangential stress σ_t is given by:

$$\sigma_t = (1 + k)\sigma_1 \qquad (25)$$

$$k = \frac{b}{a} = \frac{\sigma_2}{\sigma_1} \qquad (26)$$

The problem can be formulated under the following equivalent form: minimize the weight of the plate while constraining the maximum value of the tangential stress. Since the tangential stress is not available in the current version of the program, the Von Mises stress was used instead. It is believed that the Von Mises stress around the hole is close in value to the tangential stress since the radial stress is zero on a free boundary.

What follows is a description of how one would solve this problem using an interactive engineering design system, such as the one described in this paper.

Square plate with a circular hole

The first step which is required is the model description, using the design element concept. Since only the shape of the hole is to be changed, an adequate representation of the structure is to use one design element around the hole with one changing border, and one subregion containing the rest of the plate with fixed geometry. As the design elements are defined by their boundary curves, it is now up to the user to specify which type of curves he wants to use and to locate the governing points. In this example the design element boundaries are 2 periodic B-splines of order 13 defined by 16 poles. So there are totally 32 control nodes, the 16 determining the inner contour are moving and the 16 poles shaping the outer contour are fixed. The design model as displayed by the program is shown in Fig. 6b.

This model desciption leads to a shape optimization problem, with the 16 distances between the moving control nodes and their respective fixed reference poles as design variables. Employing the double symmetry of the problem, this number can be reduced to only 5 design variables. The design model display of Fig. 6b allows a clear visualization of these design variables as well as their associated side constraints.

The next step is to generate the analysis model from the design model description. As explained before, a mesh of isoparametric quadrilateral elements can be generated automatically inside each design element; the mesh inside the fixed subregion must be defined by the user. The interactive design system can display the mesh, as well as other basic information such as numberings, loads, support points, control nodes (Fig. 6c). The user can then anticipate what will be done during the sensitivity analysis, by moving those control nodes which are design variables and examine the effect on the finite element model, that is the mesh deformation. The design constraints can be specified in terms of the analysis model. Using the mesh symmetry, the Von Mises stress in 12 of the 24 elements bordering the hole is constrained. Note that the mesh is not double symmetric, the reason being that in the case of periodic B-splines there is no simple way to impose the coordinates of the point corresponding to the initial value of the parameter describing the curve.

At this point the user is ready to submit the created data to the finite element optimization code for the finite element analysis and the sensitivity analysis of the initial design (one iteration). Upon completion of this job, the analysis results (displacements and stresses) can be visualized by the interactive system. From the Von Mises plot (Fig. 6d) one can observe the important stress concentration associated with the initial design, exceeding the imposed stress limit (260 N/mm^2), resulting in an infeasible design.

Using the interactive shape optimization capability of the system, supplied with the calculated sensitivity results, the new shape can be computed and immediately verified with the design model plot and the analysis model plot (Fig. 6e). This is a critical point in the optimization procedure. If the modified shape is acceptable, the optimization process should be continued and the new analysis model can be submitted to the FEM-sensitivity program for another iteration. If the new shape is undesirable, the user can intervene through the system in the optimization by modifying constraints or parameters associated with the CONLIN optimizer. In particular for this case, since it was seen that the initial design is seriously infeasible, no useful solution for the optimization was found unless the relaxation capability of CONLIN was activated. This requires the user to add acceptable increments to the constraint bounds and provides some influence on the CONLIN results. For this purpose, it is also very useful to be able to display slices of the design space with the interactive system. As can be seen from Fig. 6f, showing the design space for the design variables 2 and 4, the feasible domain is empty; with relaxation the design space is opened and a feasible region exists.

For this example with relaxation applied during the first optimization stage, the new shape is acceptable and additional iterations could be done. At each iteration, similar interactive handling can be done, though the user is mostly concerned with the display of the successive shapes. As the number of iterations increased, it became interesting to view the progress of the optimization process with the evolution plots of the objective function, design variables and constraints. From these plots it became apparent that values of side constraints on some design variables had to be adjusted to avoid them from becoming active, i.e., design variable 5 was close to its lower bound.

The shape obtained after 7 iterations is shown in Fig. 7a. It was decided to stop the optimization process since satisfactory convergence was obtained to the optimal elliptic shape with axis ratio 0.5 . As shown in Fig. 7b, the user can also have both the initial and the optimal shape displayed simultaneously to assess the boundary changes. The stress plot (Fig. 7c) reveals the constancy of the Von Mises stresses at the edge of this optimized hole: the constant stress lines run nearly parallel to the edge of the hole. This observation can also be drawn from the results summarized in Table 1. The user can ask for a graphical representation of the iteration history of all quantities of interest with the evolution plots. Fig. 7d shows respectively the weight and design variable 5 in terms of the number of iterations; the user can assure himself with this graphical capability of proper convergence. From the weight evolution plot it can be seen that only at the first iteration relaxation had to be applied: it resulted in a feasible design, as are all the subsequent designs. This can be verified in the design space plot corresponding to the final design (Fig. 7e), which shows a non-empty feasible region.

Square plate with an elliptic hole

In order to further validate the interactive optimization module, a more demanding design problem was devised. By adopting the previously obtained optimal design (horizontal ellipse) as starting design and reversing the loading (see Fig. 8a), a vertically elongated ellipse should be generated as the optimal shape. As can be anticipated from the initial material distribution, this initial design is still more infeasible than the previous one. At the first iteration, the CONLIN optimizer using relaxation found a feasible design, one with a very small hole (Fig. 8b) and thus with a considerable higher weight (peak in weight evolution plot). From this feasible design, the optimization is proceeded, minimizing the weight and converging to the appropriate optimal shape (Fig. 8c). A similar interactive procedure was used to reach this solution. The evolution plots for the weight and the design variable 5 summarize some of the results (Fig. 8d).

Quarter plate model

Finally a different, more economical design model was investigated. Using the double symmetry of the problem only one quarter of the plate need in fact to be studied. A different design model had to be created for this purpose. The region that can be modified was described by a 36 control nodes design element. It is defined as the cartesian product of a B-spline of order 7 with 9 control nodes and a B-spline of order 4 with 4 control nodes. The displacements of the 9 moving control nodes, describing the hole boundary, along the meridian directions play the role of design variables (Fig. 9a). It was believed that by using control node pairs at the ends of the design element moving border, control over the tangency of the curve could be enhanced, when both pairs move together over the same distances. As a result, these pole pairs correspond to only one design variable and the problem has thus only 7 decision variables.

The analysis model was generated and the design constraints specified. Constraining the Von Mises stress in the six elements at the hole boundary with additional side constraints on the design variables, results in a much smaller optimization problem. The optimization was done interactively but was stopped after 7 iterations since the generated shapes were undesirable. As one can observe from Fig. 9b, the overall shape is elliptic. However at the ends of the curve describing the hole boundary, the tangents are not horizontal or vertical. This would mean that for the full plate solution, we would have a discontinuity of the tangent at these points. Note that the full plate model, using periodic B-splines, does not have this tangency problem: the continuity requirements are automatically satified. The trade-off is however that the full plate has to be modeled and analyzed, increasing drastically the CPU time both for analysis and gradient calculation.

This example, though not fully successful, shows the usefulness of an interactive design system, to follow closely the optimization process and act appropriately. Other geometric models might be investigated to see if better results can be obtained.

CONCLUSIONS

This paper had two main objectives. The first one was to discuss a new approach to shape optimal design of elastic structures discretized by the finite element method. The second goal was to describe an interactive optimization system that is intended to become a powerful Computer Aided Design tool.

The approach followed can be summarized as follows. First the behavior of the structure is analyzed by using the finite element method. Subsequently a sensitivity analysis is performed to evaluate the first derivatives of the structural response quantities: these derivatives are obtained through a finite difference approach. They are then used by the CONLIN optimizer, which selects an improved design. A reanalysis of the modified design is next performed after updating the finite element mesh. This iterative process is repeated until convergence to an acceptable optimum design has been achieved, which usually requires less than 10 FEM analyses.

Although conceptually simple, the finite difference approach presented in this paper reveals very promising because of its generality and ease of implementation. Analytical derivatives for the shape optimal design problem can also be obtained, but in a very complex way, which is dependent on the type of finite element used. Introducing this analytical method in general purpose finite element packages, containing a vast library of element types, would require a huge development effort. The finite difference approach proposed herein is on the contrary highly general, in that the scheme is valid for any type of element, and can be implemented relatively easily.

The generality of the finite difference method is not limited to the choice of finite elements. The method is also general with regard to the optimization problem statement. Without too much complexity, the sensitivity analysis scheme described in this paper can be introduced in optimal shape design under frequency constraints, structures under thermal loading or three-dimensional structures. Future research should be done to extend the application of the method to those areas.

It should be recognized, however, that coupling geometric modeling concepts and finite differencing suffers from some limitations. A drawback is that the method is computationally expensive. In the present stage of the study no special efforts were spent to increase the efficiency of the algorithm in this respect, as the primary concern was to check the validity of the approach. Further work is therefore planned to analyze how the CPU time is distributed over the different steps and to increase the computational efficiency.

Another disadvantage is that the derivatives are of course not exact. However the CONLIN optimizer does not seem very sensitive to this potential lack of accuracy. Future work will be devoted to investigating the accuracy of the derivatives, i.e. how it may be improved by selecting a suitable step size in the finite difference scheme [12].

On the other hand the two dimensional pre-and post processor module previously described, represents a first step in the development effort to implement shape optimization concepts into the real design cycle. It is the interactive component in the logical division of the optimization task between a finite element optimization code, producing the analysis results in batch mode (including sensitivities), and an interactive design system, helping the designer to control the optimization process.

At the present stage the interactive module is mainly functioning as a post-processor. It is capable of interactive shape optimization by calling the CONLIN optimizer, and it exhibits some innovative visualization techniques that seem to constitute appropriate tools to facilitate the designer¶s task. Future work should be directed toward the development of pre-processing modules, allowing an efficient introduction of the optimization data. Additional study can also be accomplished to devise a well organized data base for all the information involved in the shape optimization process.

ACKNOWLEDGEMENT

This research was jointly supported by the MacNeal-Schwendler Corporation and the University of California under the MICRO program (Proposal No. 85-204).

REFERENCES

[1] Zienkiewicz, O.C. and Campbell, J.S., "Shape optimalization and Sequential Linear Programming", in Optimum Structural Design (Ed. R.H. Gallagher and O.C. Zienkiewicz), Wiley, New York, 1973, pp.109-126.

[2] Imam, H., "Three Dimensional Shape Optimization", International Journal for Numerical Methods in Engineering, Vol. 18, pp. 661-673, 1982.

[3] Braibant, V. and Fleury, C., "Shape Optimal Design using B-Splines", Computer Methods in Applied Mechanics and Engineering, Vol. 44, pp. 247-267, 1984.

[4] Braibant, V. and Fleury, C., "Shape Optimal Design - A Performing C.A.D. Oriented Formulation", Proc. AIAA/ASME/ASCE/AHS 25th Structures, Structural Dynamics and Materials Conference, Palm Springs, California, 1984.

[5] MSC/NASTRAN User's and Application Manuals, The MacNeal-Schwendler Corporation, 1984.

[6] MSC/pal Reference and Application Manual, The MacNeal-Schwendler Corporation, 1984.

[7] Fleury C. and Braibant, V., "Structural Optimization - A New Dual Method Using Mixed Variables", International Journal for Numerical Methods in Engineering, to appear, 1986.

[8] Fleury, C., "Shape Optimal Design by the Convex Linearization Method", in The Optimum Shape: Automated Structural Design" (J. BENNETT and M. BOTKIN, eds.), 1986, to appear.

[9] Fleury, C., "Reconciliation of Mathematical Programming and Optimality Criteria Approaches to Structural Optimization", Chapter 10 of Foundations of Structural Optimization : A Unified Approach (A.J. MORRIS, ed.), John Wiley and Sons, 1982, pp. 363-404.

[10] Kuritz, S., "Configuration Optimization of Trusses using Convex Linearization Techniques", Master Thesis, UCLA, 1986.

[11] Muskhelishvili, N.I., "Some basic Problems of the Mathematical Theory of Elasticity", Noordhoff, Groningen - Holland, pp.320-372, 1953

[12] Iott, J., Haftka, R.T., Adelman, H.M., "Selecting Step Sizes in Sensitivity Analysis by Finite Differences", NASA TM-86382, NASA Langley Research Center, Hampton, Virginia, August 1985

LIST OF FIGURES

	Iteration number k							
	0	1	2	3	4	5	6	7
total weight [kg]	13.09	13.68	13.53	13.34	13.26	13.22	13.19	13.16
element stresses [N/mm^2]								
1	312	263	253	255	257	258	259	258
2	226	246	249	257	259	259	259	258
3	113	206	231	248	253	255	257	259
4	32	156	196	219	230	236	242	249
5	65	177	212	233	242	246	250	255
6	170	230	243	255	258	259	259	259
7	273	257	252	256	258	259	259	257
8	347	265	253	255	256	258	259	258
9	398	262	255	256	257	258	259	259
10	412	257	257	258	258	258	258	258
11	408	259	256	257	257	258	258	258
12	377	264	254	255	256	258	259	259

Imposed stress limit : 260 N/mm^2

Table 1 CONLIN iteration history for plate with hole problem (starting from circular hole)

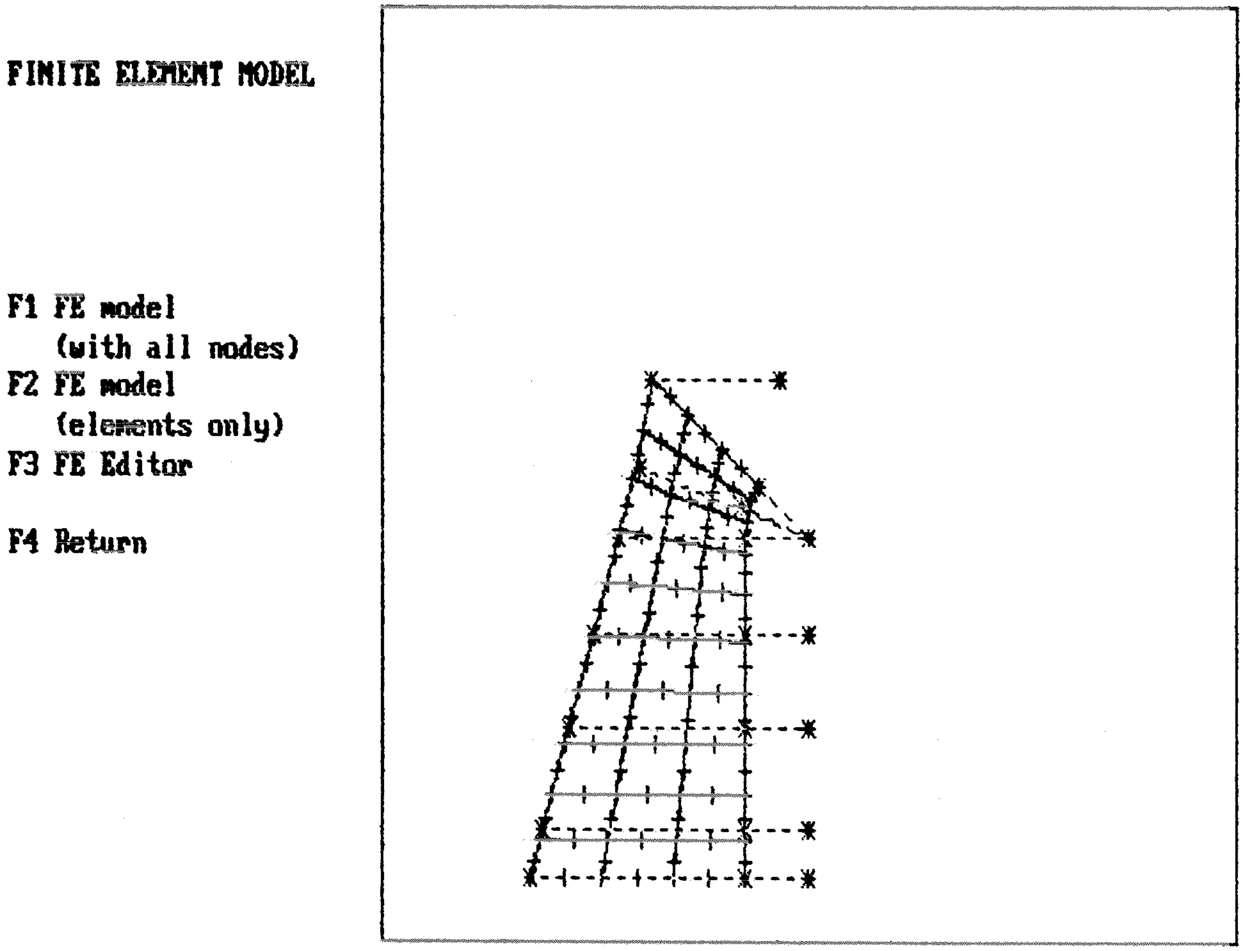
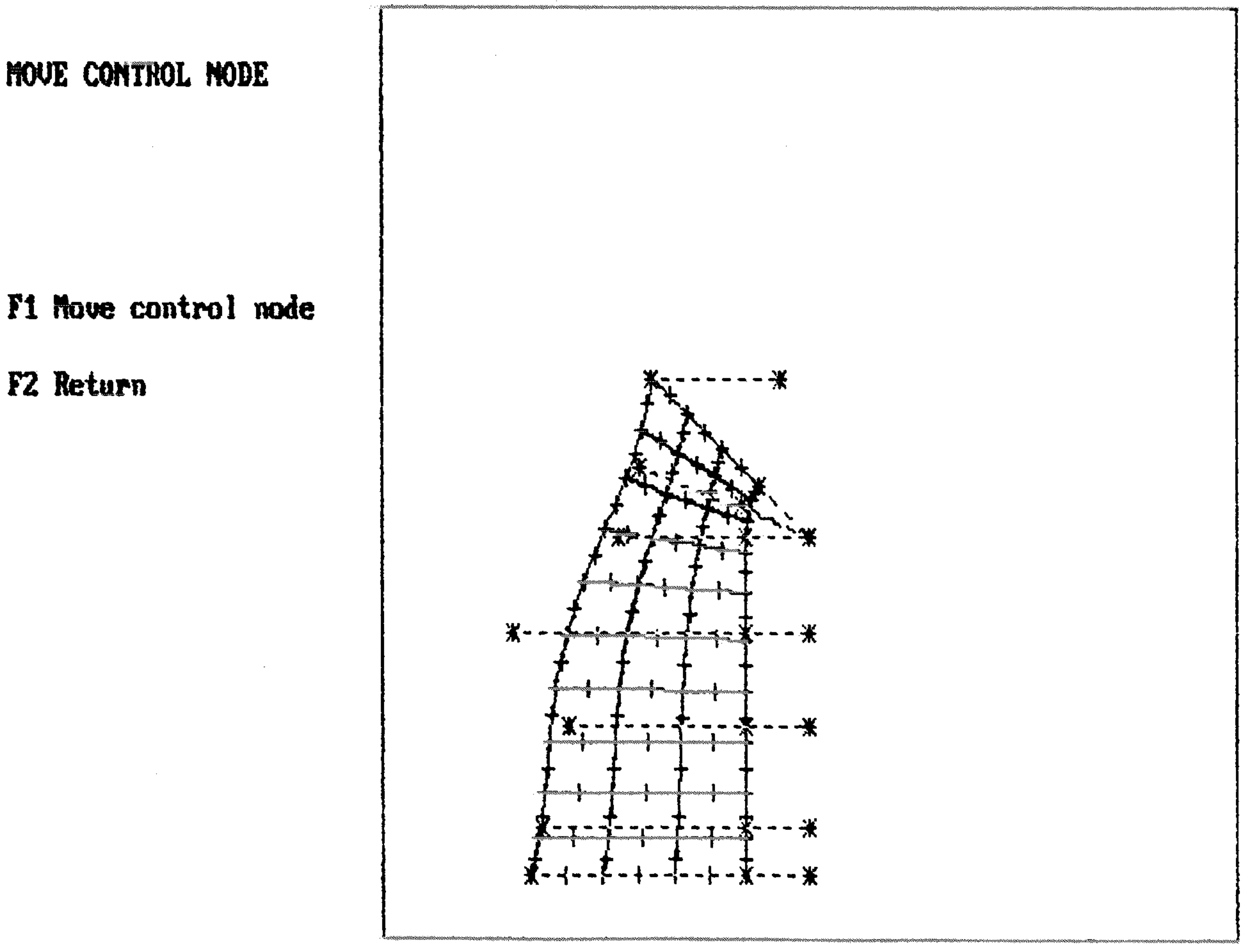

FINITE ELEMENT MODEL
F1 FE model
(with all nodes)
F2 FE model
(elements only)
F3 FE Editor
F4 Return

Figure 1


MOVE CONTROL NODE
F1 Move control node
F2 Return

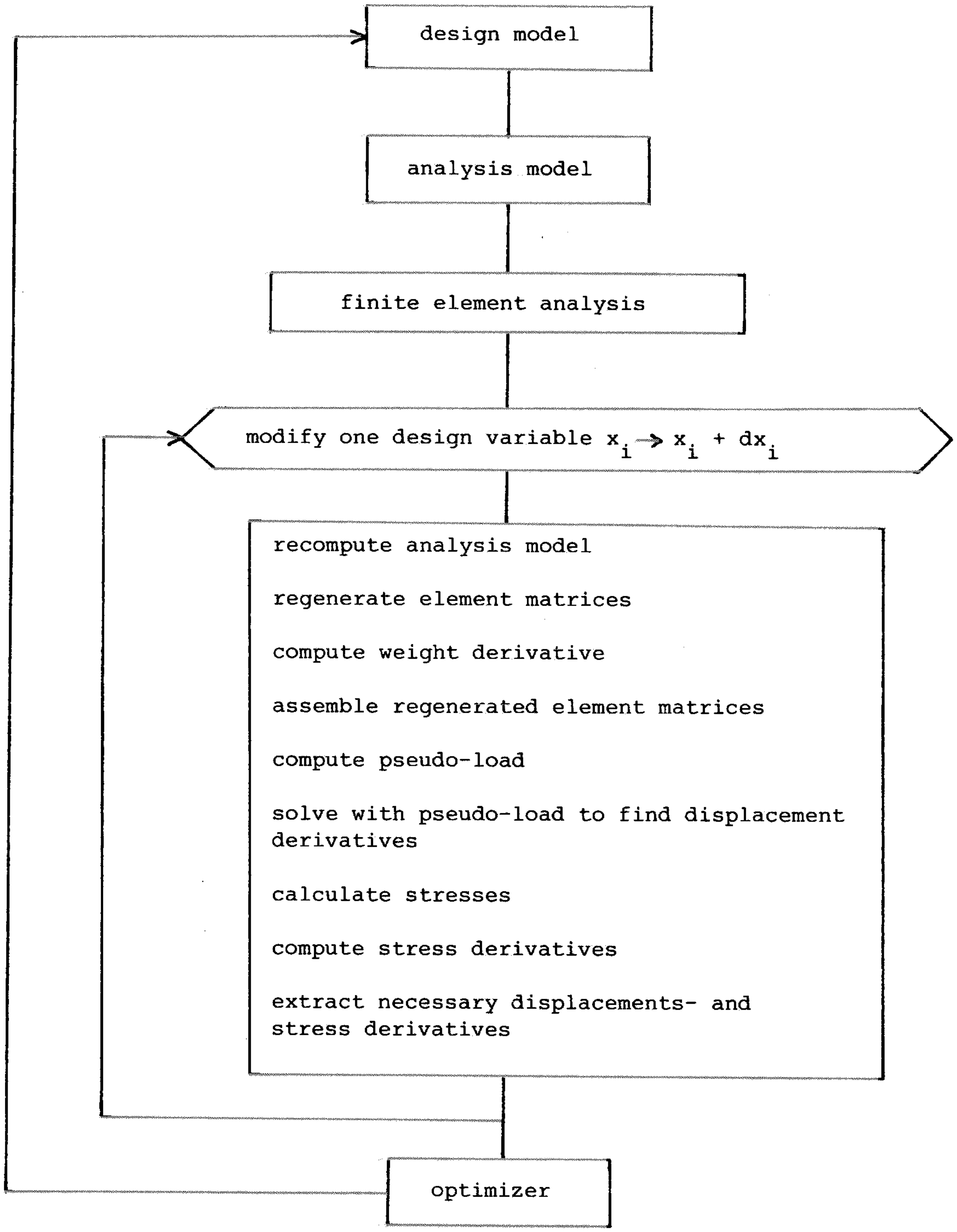
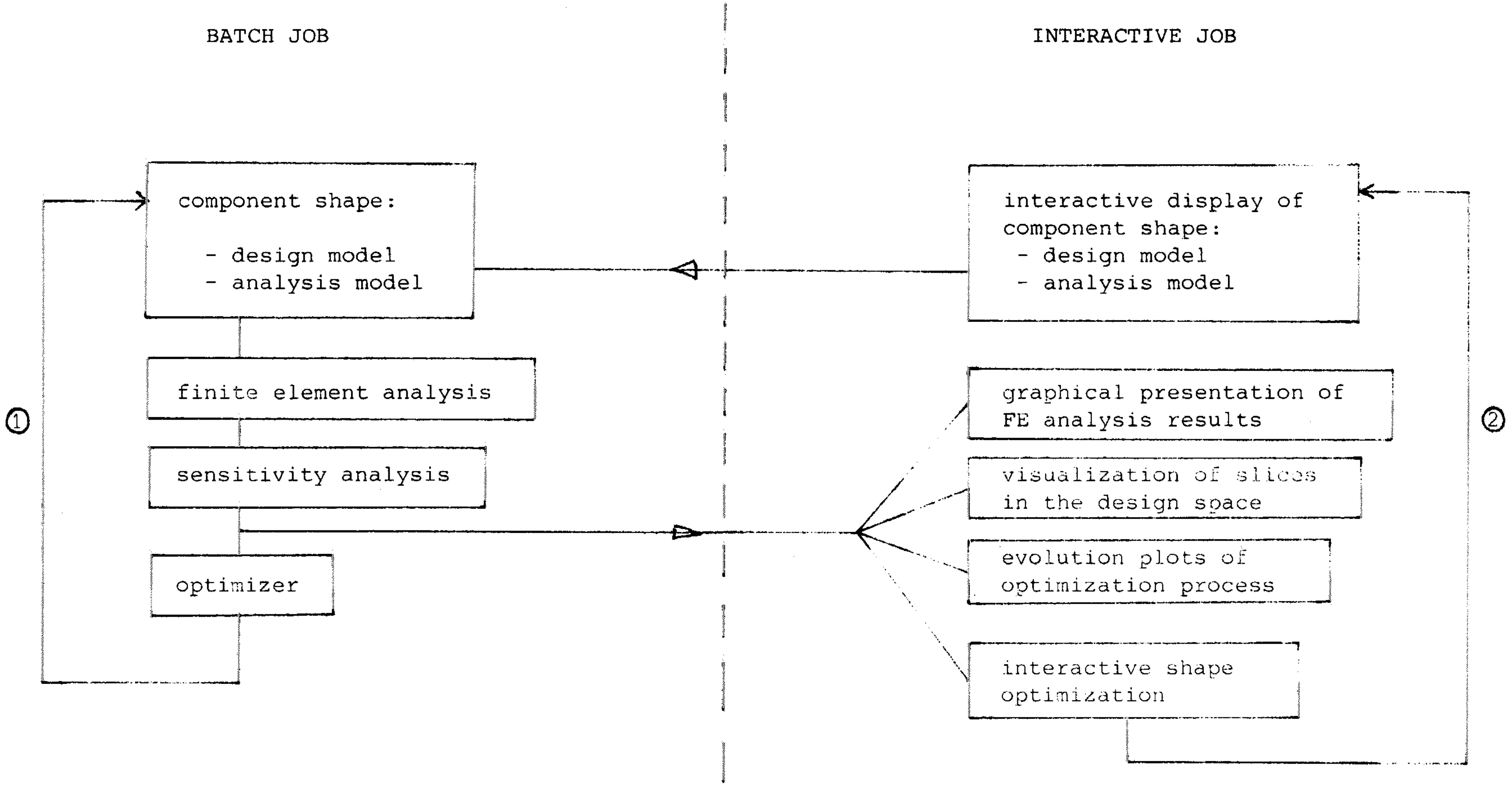

design model
analysis model
finite element analysis
modify one design variable $x_i \rightarrow x_i + dx_i$
recompute analysis model
regenerate element matrices
compute weight derivative
assemble regenerated element matrices
compute pseudo-load
solve with pseudo-load to find displacement derivatives
calculate stresses
compute stress derivatives
extract necessary displacements- and stress derivatives
optimizer

Figure 2



① classical optimization loop

② interactive optimization loop

Figure 3

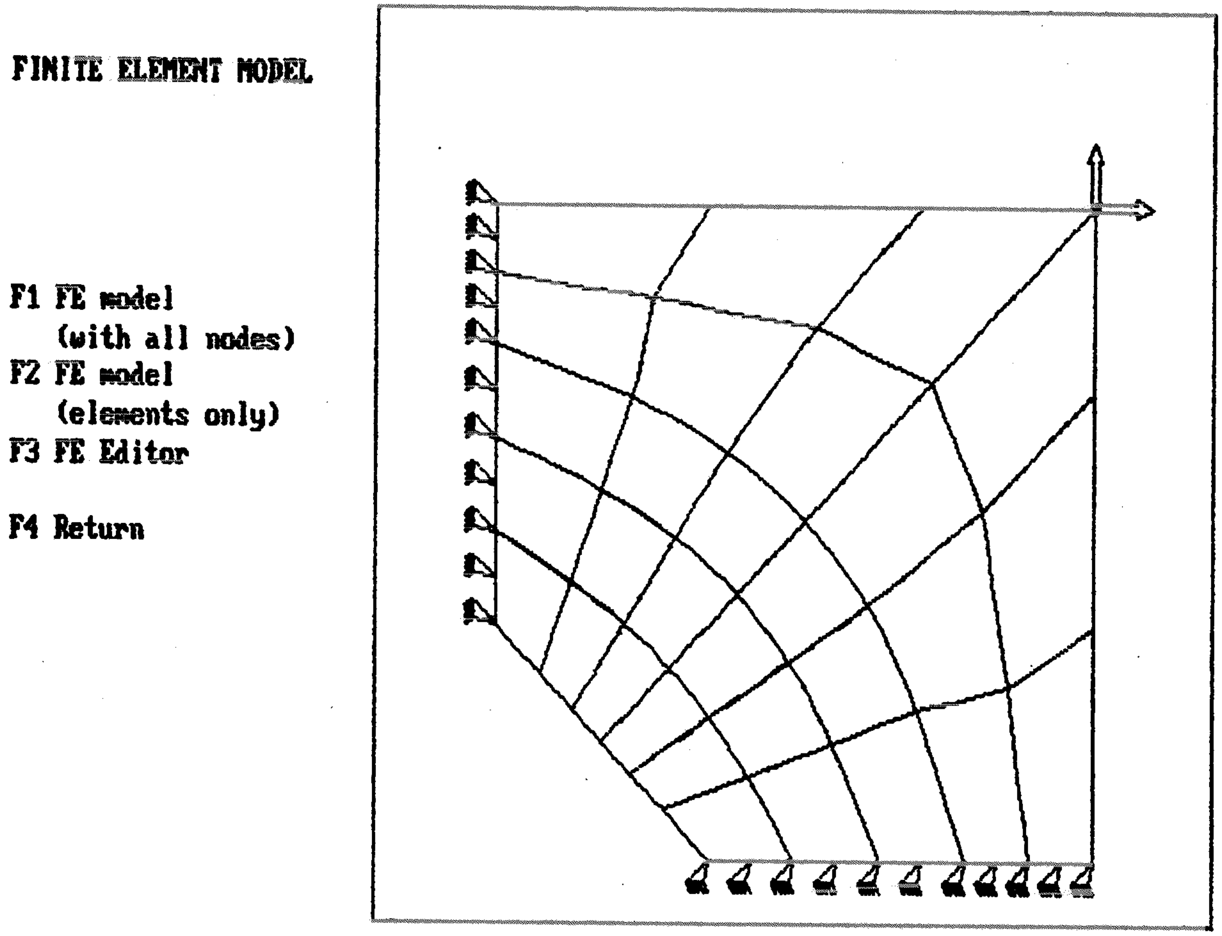

FINITE ELEMENT MODEL
F1 FE model
(with all nodes)
F2 FE model
(elements only)
F3 FE Editor
F4 Return

Fig.4a

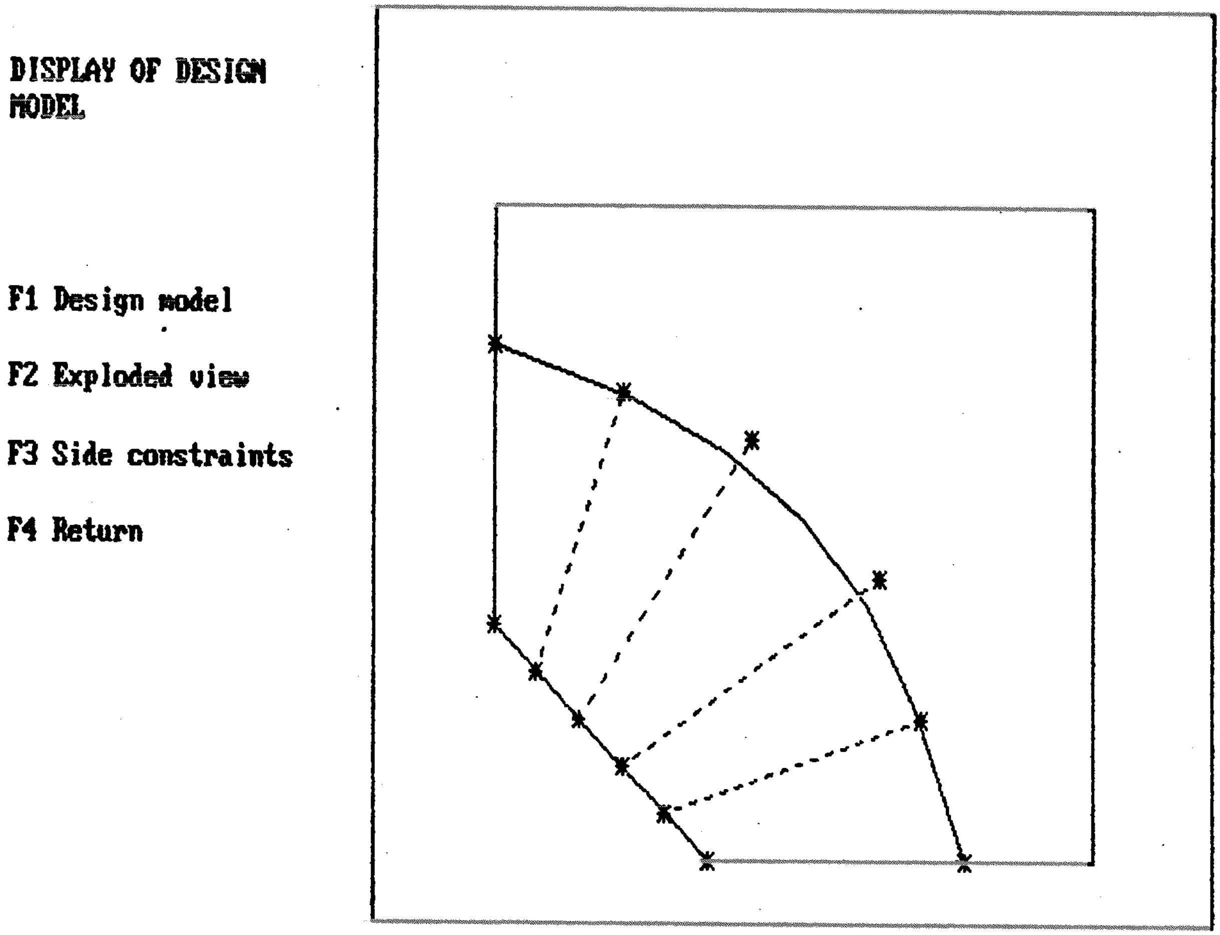

DISPLAY OF DESIGN
MODEL
F1 Design model
F2 Exploded view
F3 Side constraints
F4 Return

Fig.4b

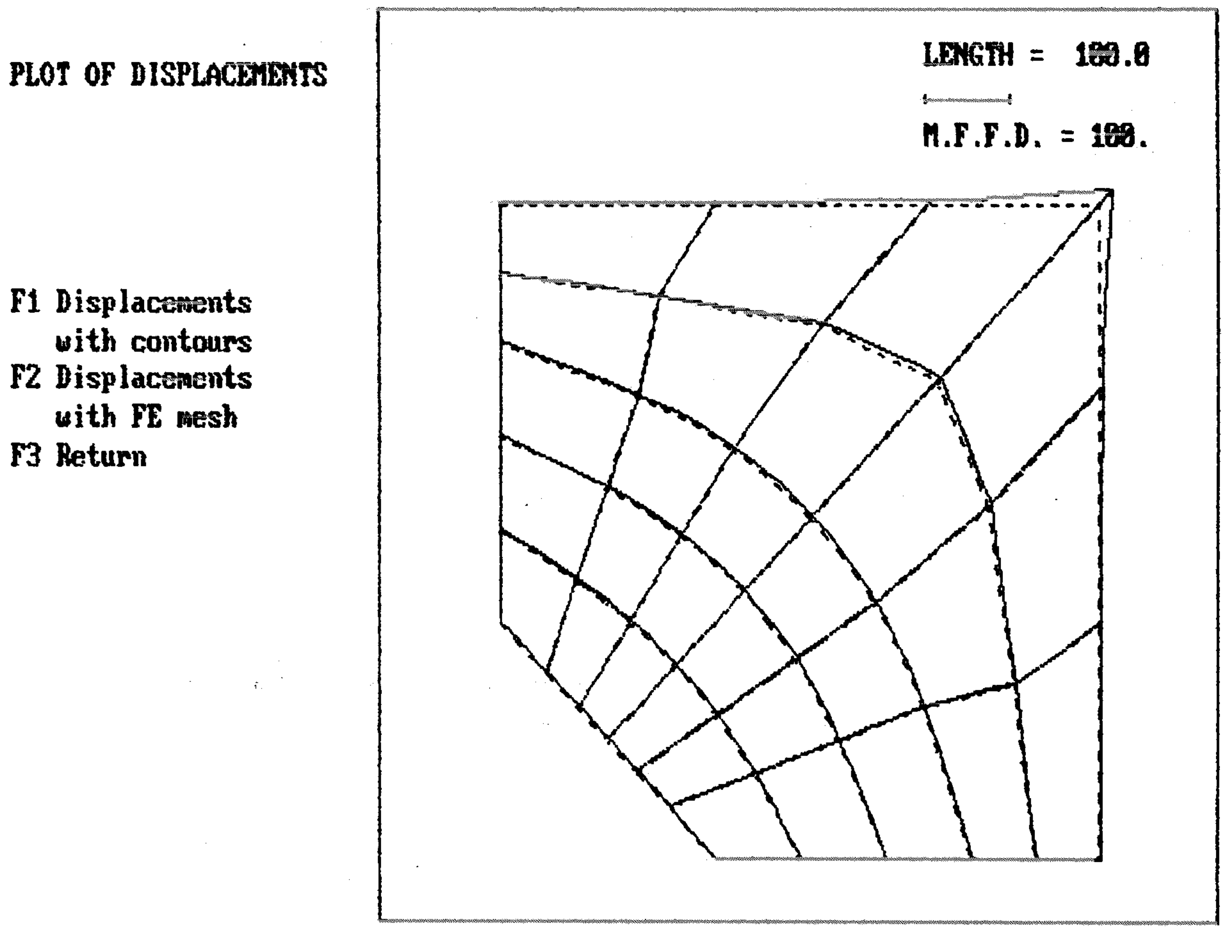

PLOT OF DISPLACEMENTS
F1 Displacements with contours
F2 Displacements with FE mesh
F3 Return
LENGTH = 100.0
M.F.F.D. = 100.

Figure 4c

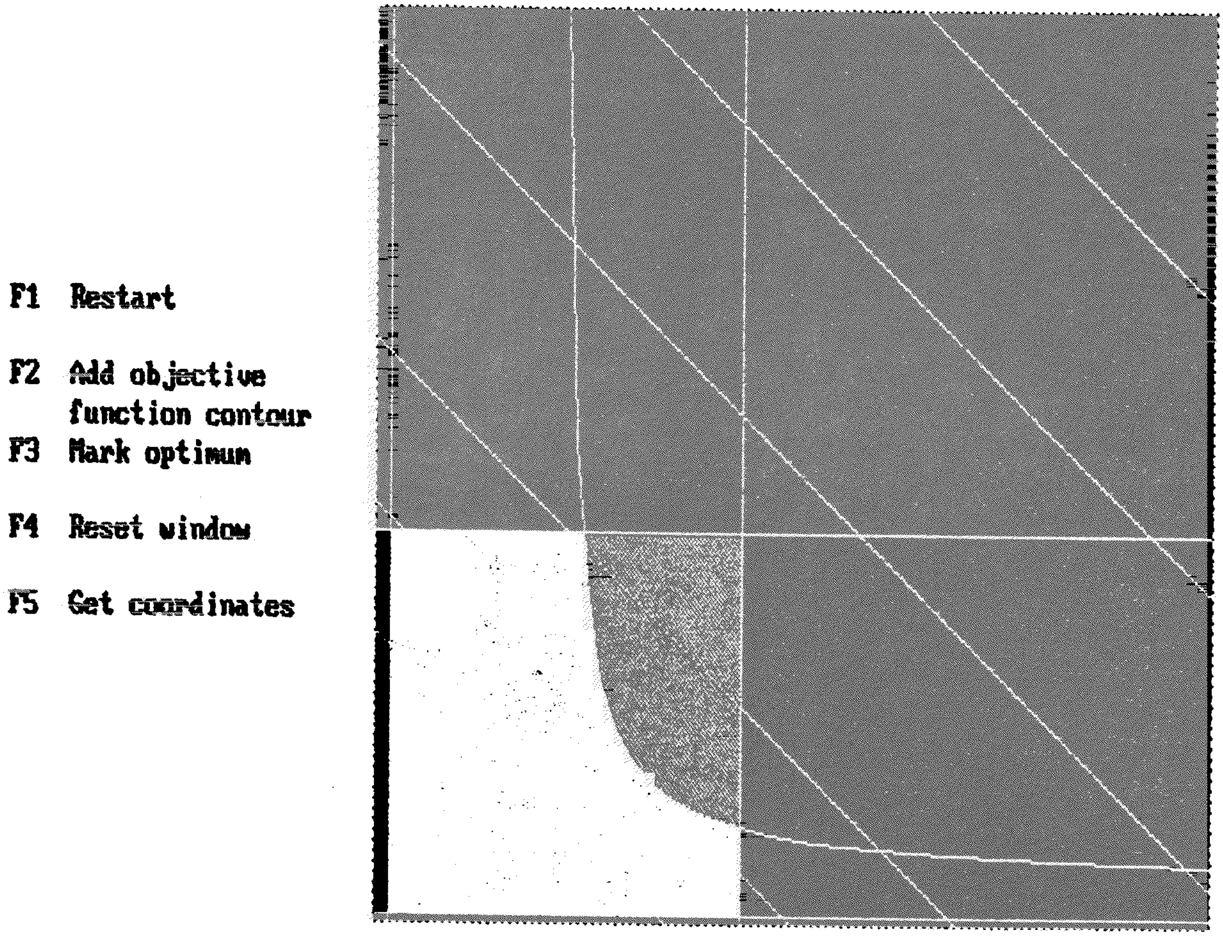

F1 Restart
F2 Add objective function contour
F3 Mark optimum
F4 Reset window
F5 Get coordinates

Fig.4d

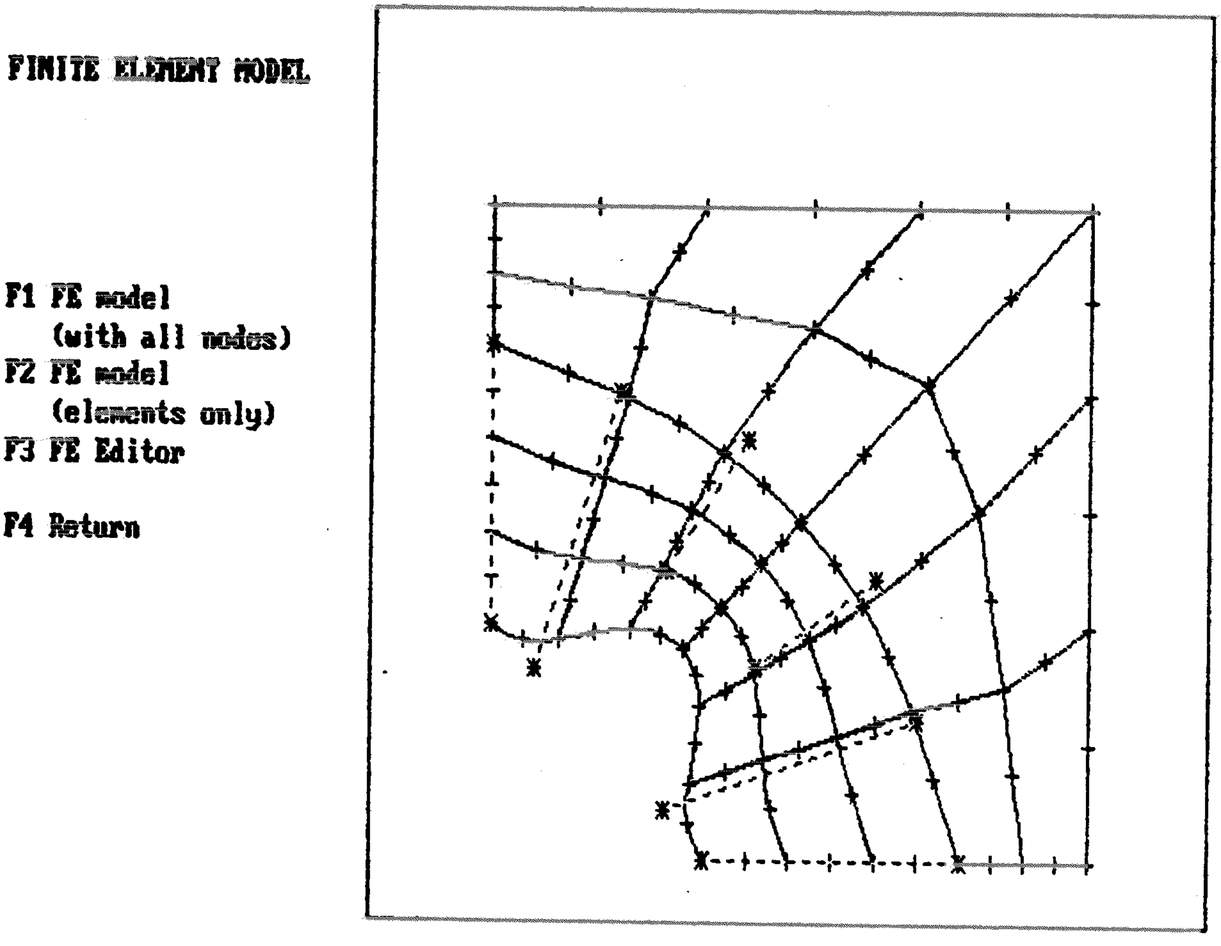

FINITE ELEMENT MODEL
F1 FE model (with all nodes)
F2 FE model (elements only)
F3 FE Editor
F4 Return

Fig.4e

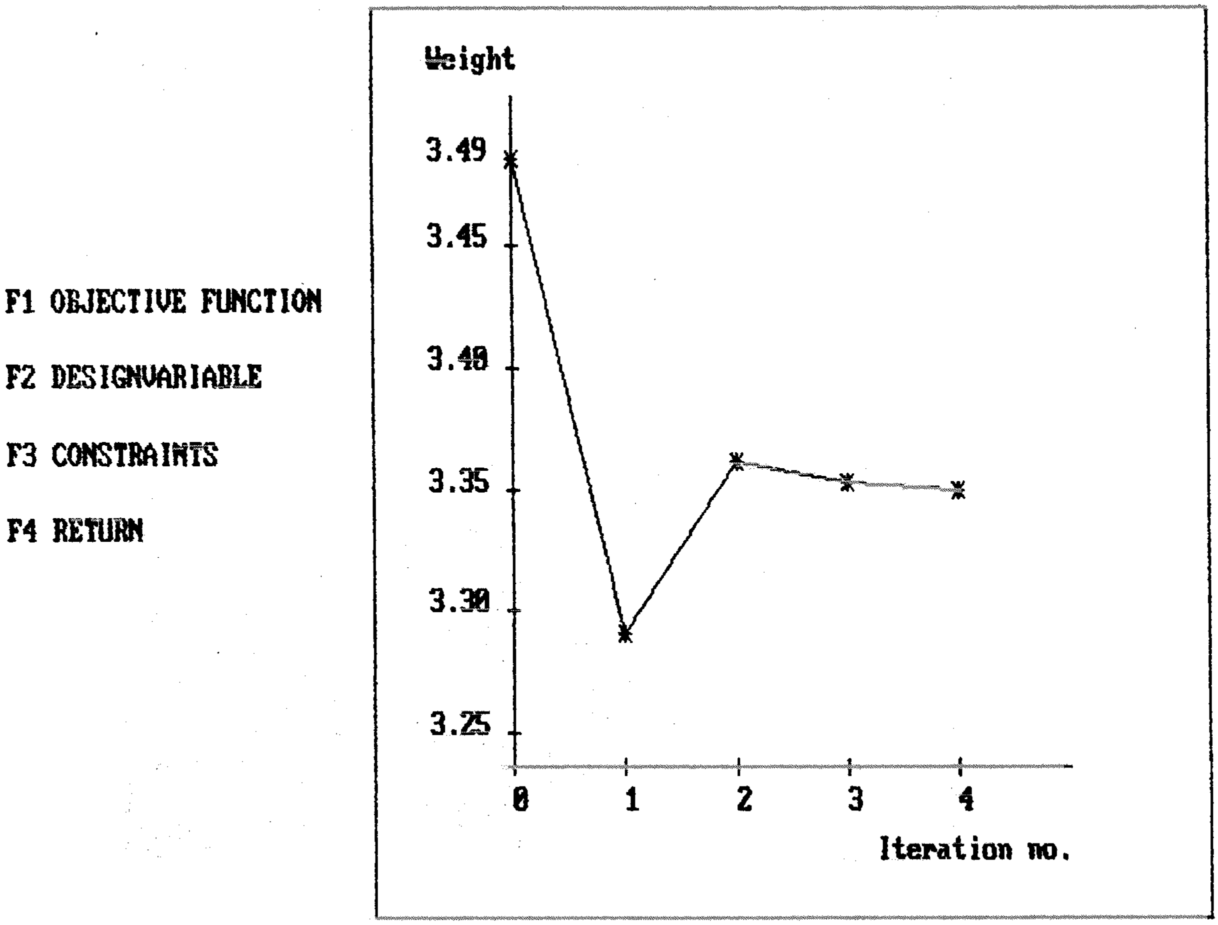
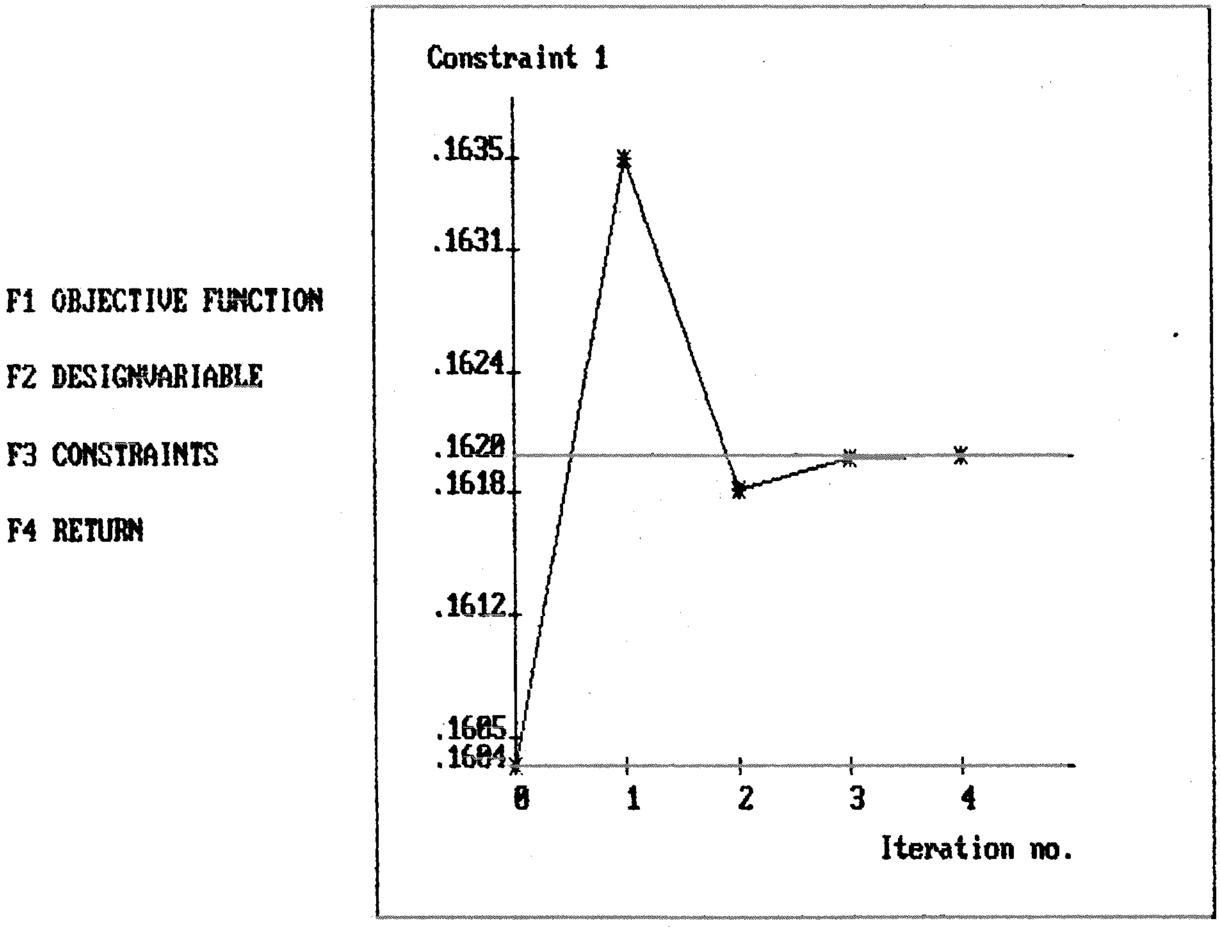

F1 OBJECTIVE FUNCTION
F2 DESIGNVARIABLE
F3 CONSTRAINTS
F4 RETURN
Weight
3.49
3.45
3.40
3.35
3.30
3.25
0
1
2
3
4
Iteration no.

Figure 5a


F1 OBJECTIVE FUNCTION
F2 DESIGNVARIABLE
F3 CONSTRAINTS
F4 RETURN
Constraint 1
.1635
.1631
.1624
.1620
.1618
.1612
.1605
.1604
0
1
2
3
4
Iteration no.

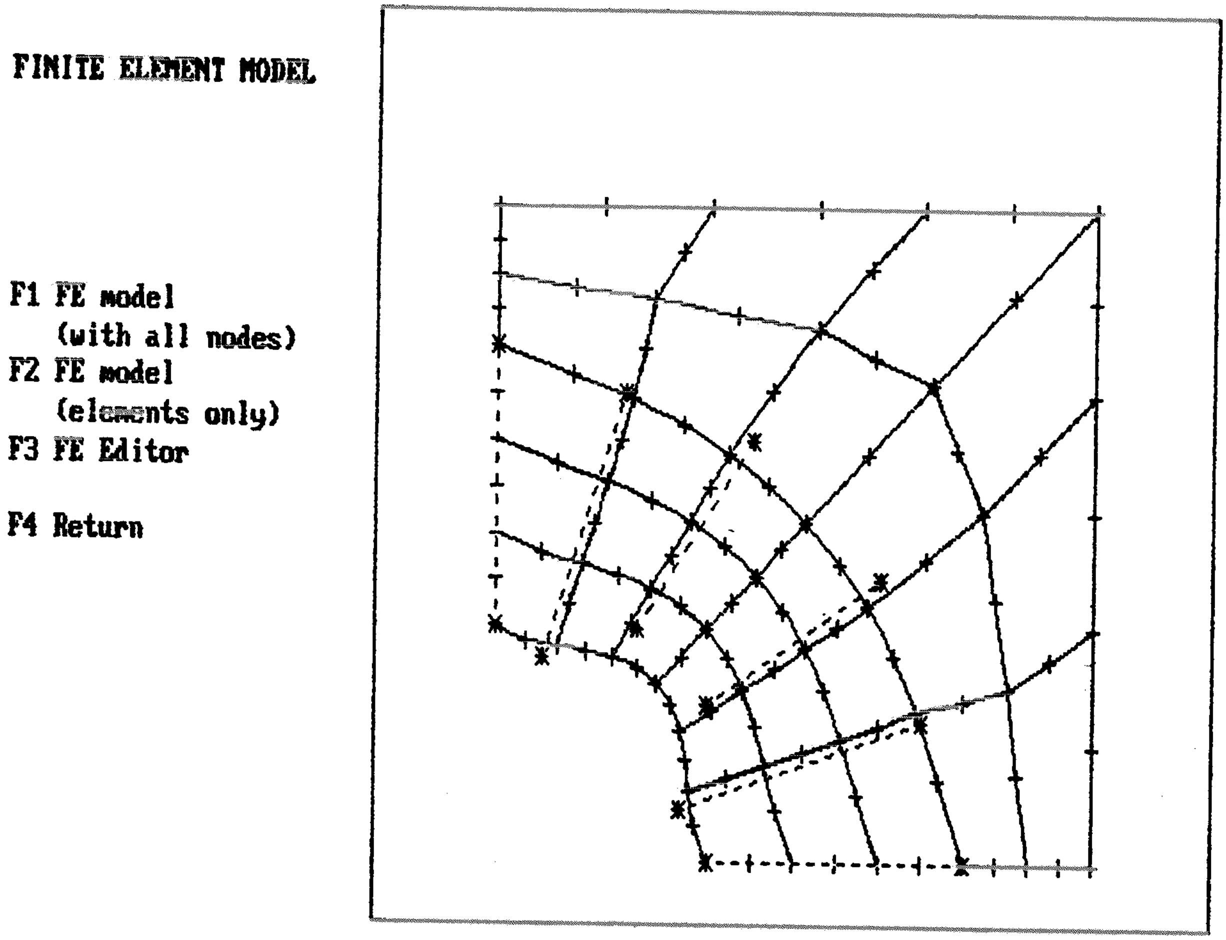

FINITE ELEMENT MODEL
F1 FE model
(with all nodes)
F2 FE model
(elements only)
F3 FE Editor
F4 Return

Fig.5b

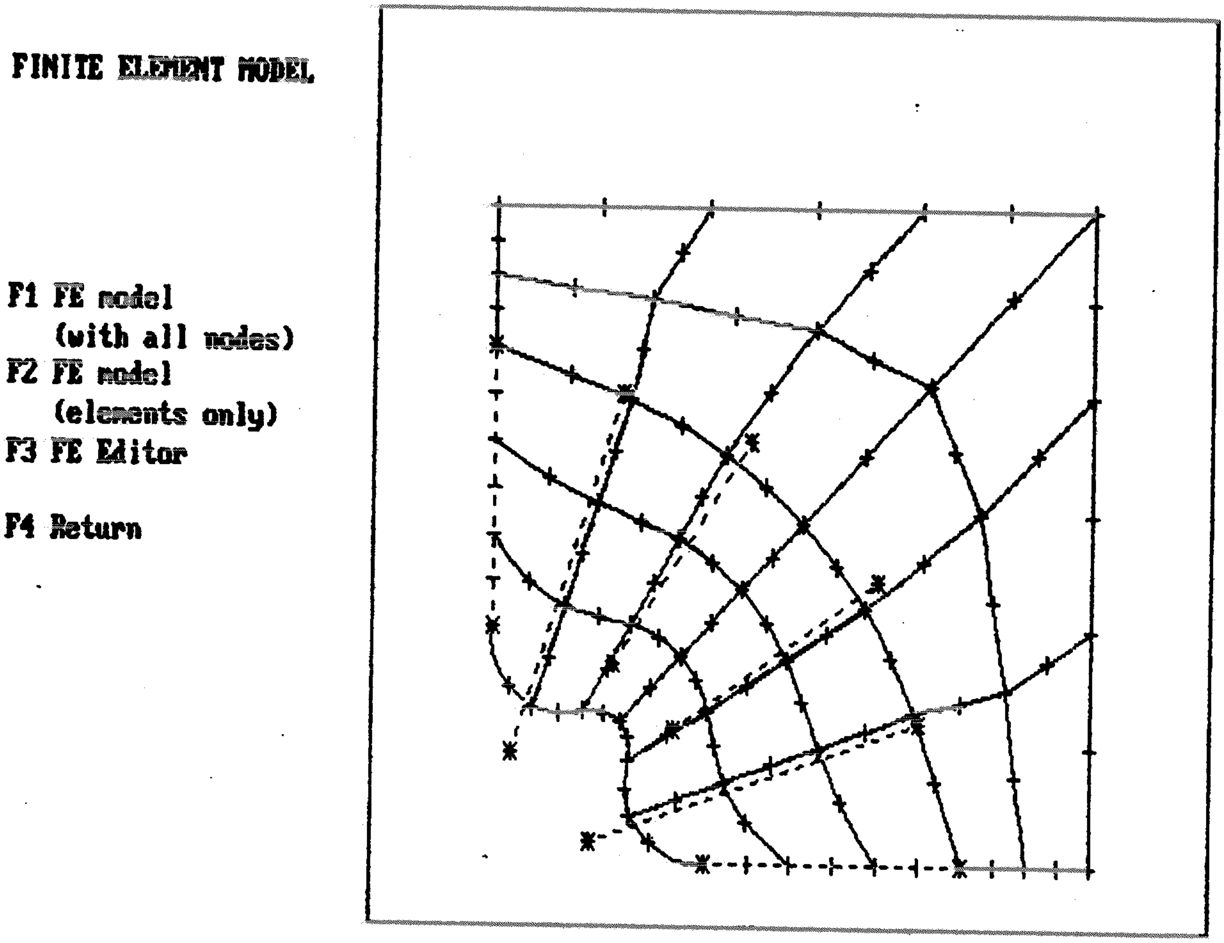

FINITE ELEMENT MODEL
F1 FE model
(with all nodes)
F2 FE model
(elements only)
F3 FE Editor
F4 Return

Fig.5d

F1 OBJECTIVE FUNCTION

F2 DESIGNVARIABLE

F3 CONSTRAINTS

F4 RETURN

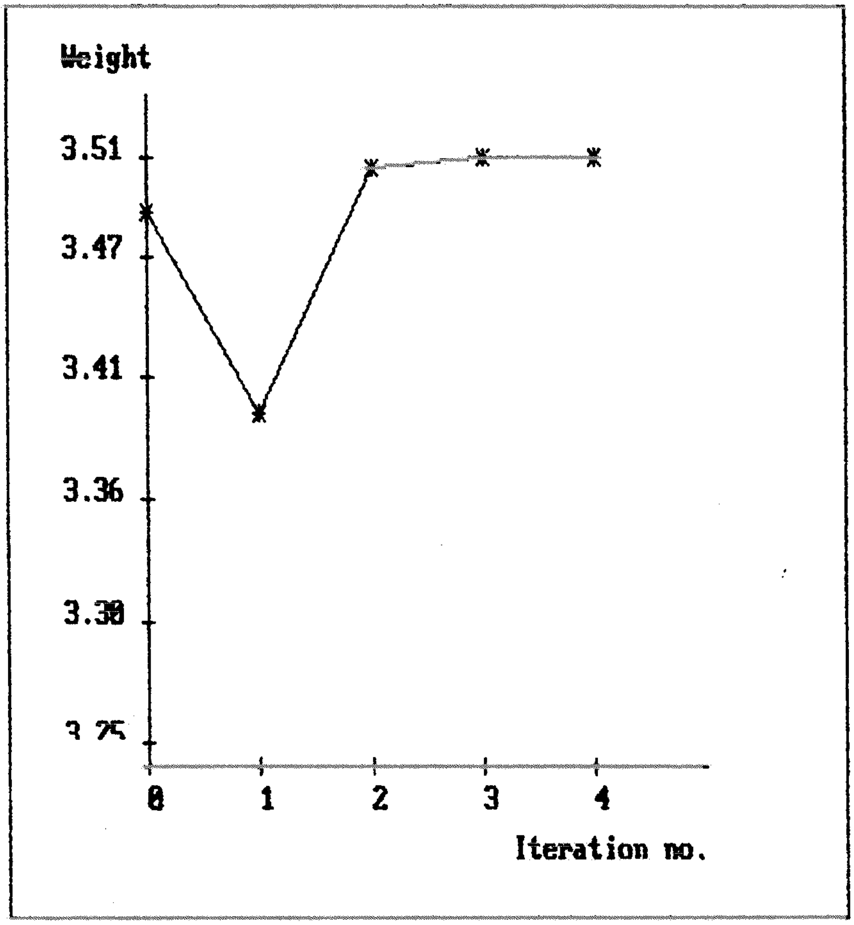


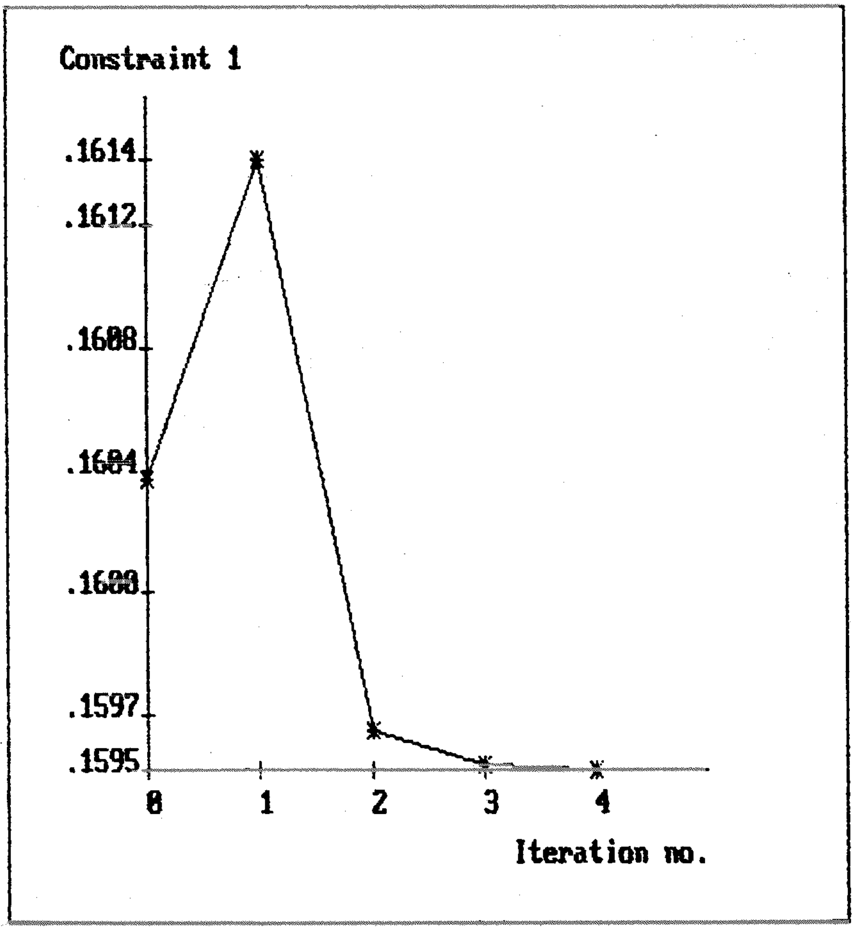
Figure 5c

F1 OBJECTIVE FUNCTION

F2 DESIGNVARIABLE

F3 CONSTRAINTS

F4 RETURN



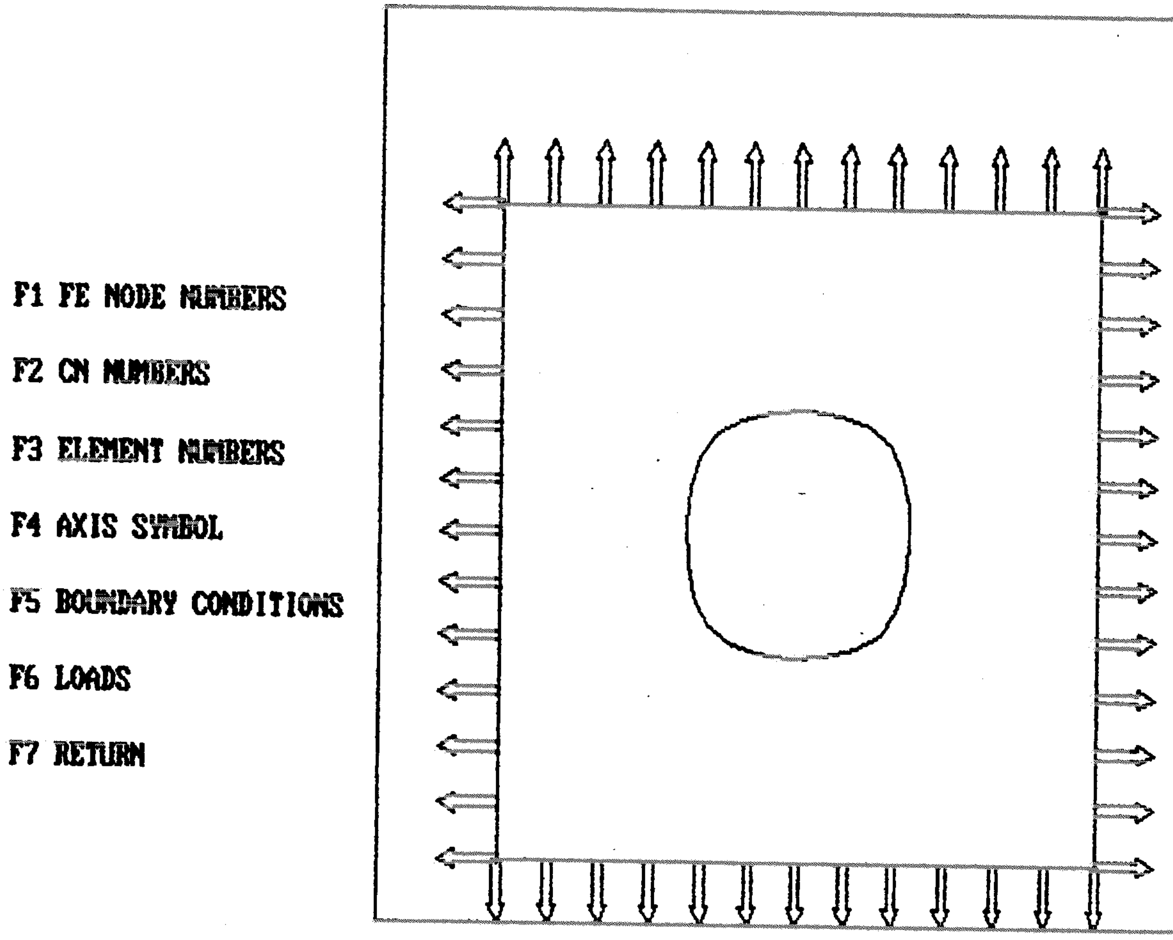

F1 FE NODE NUMBERS
F2 CN NUMBERS
F3 ELEMENT NUMBERS
F4 AXIS SYMBOL
F5 BOUNDARY CONDITIONS
F6 LOADS
F7 RETURN

Fig.6a

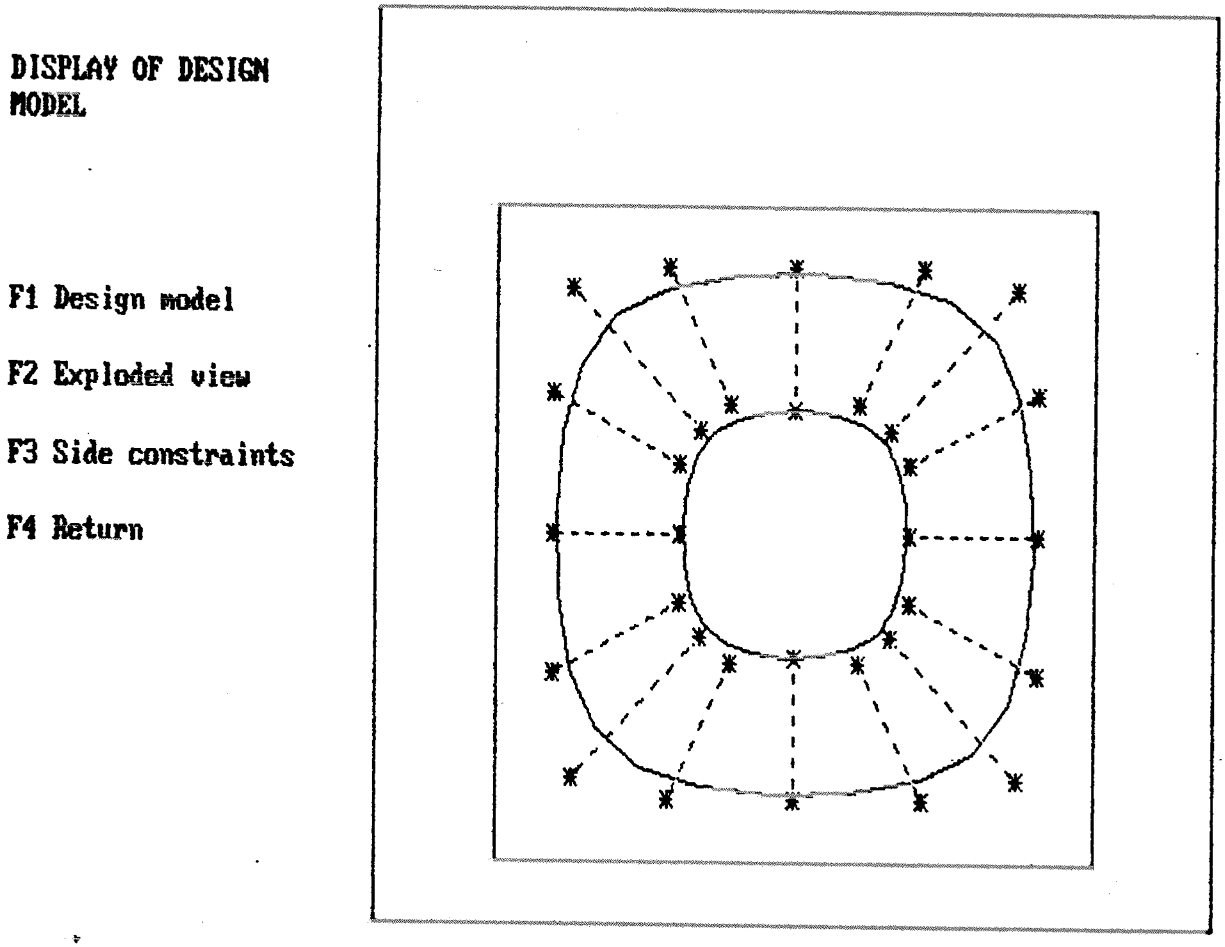

DISPLAY OF DESIGN
MODEL
F1 Design model
F2 Exploded view
F3 Side constraints
F4 Return

Fig.6b

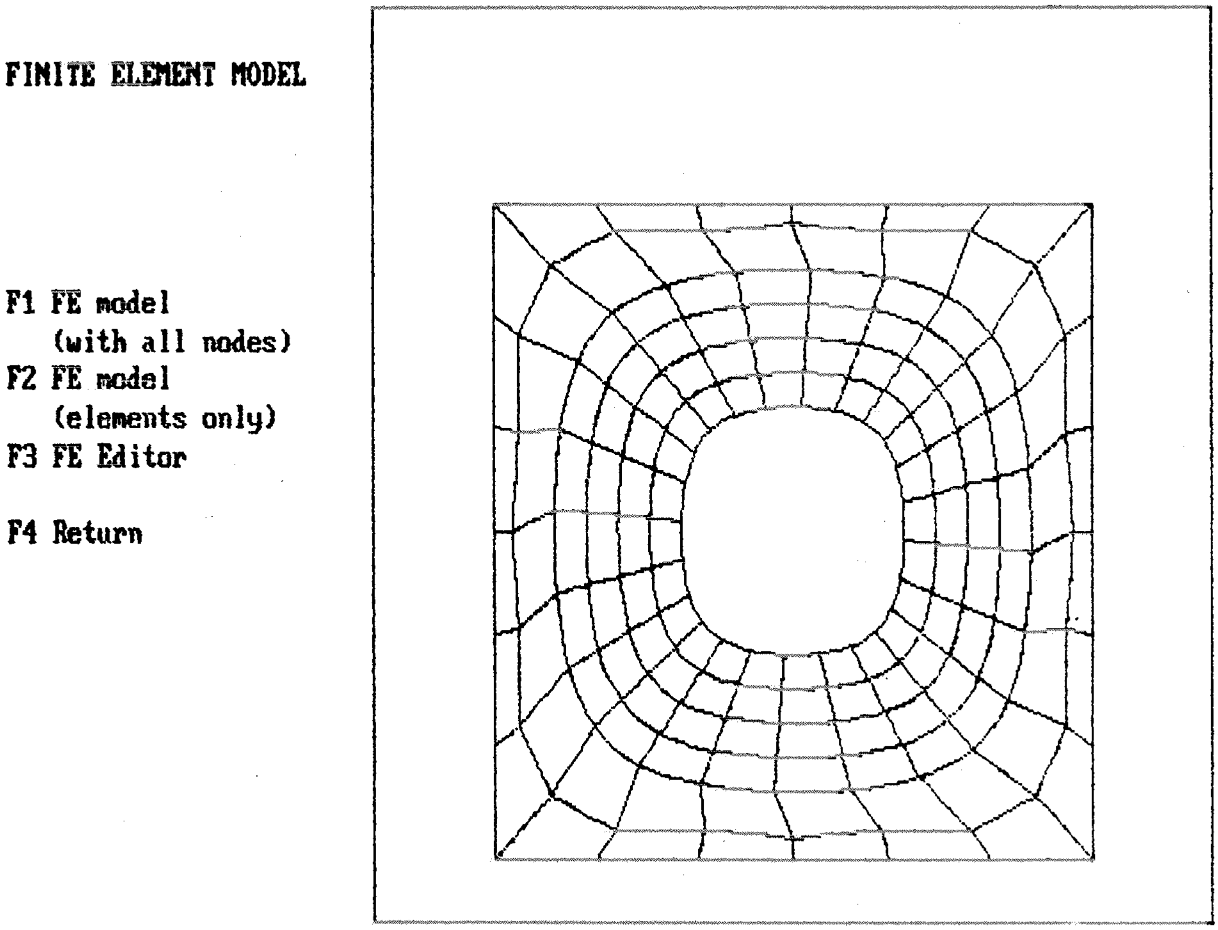

FINITE ELEMENT MODEL
F1 FE model
(with all nodes)
F2 FE model
(elements only)
F3 FE Editor
F4 Return

Fig.6c

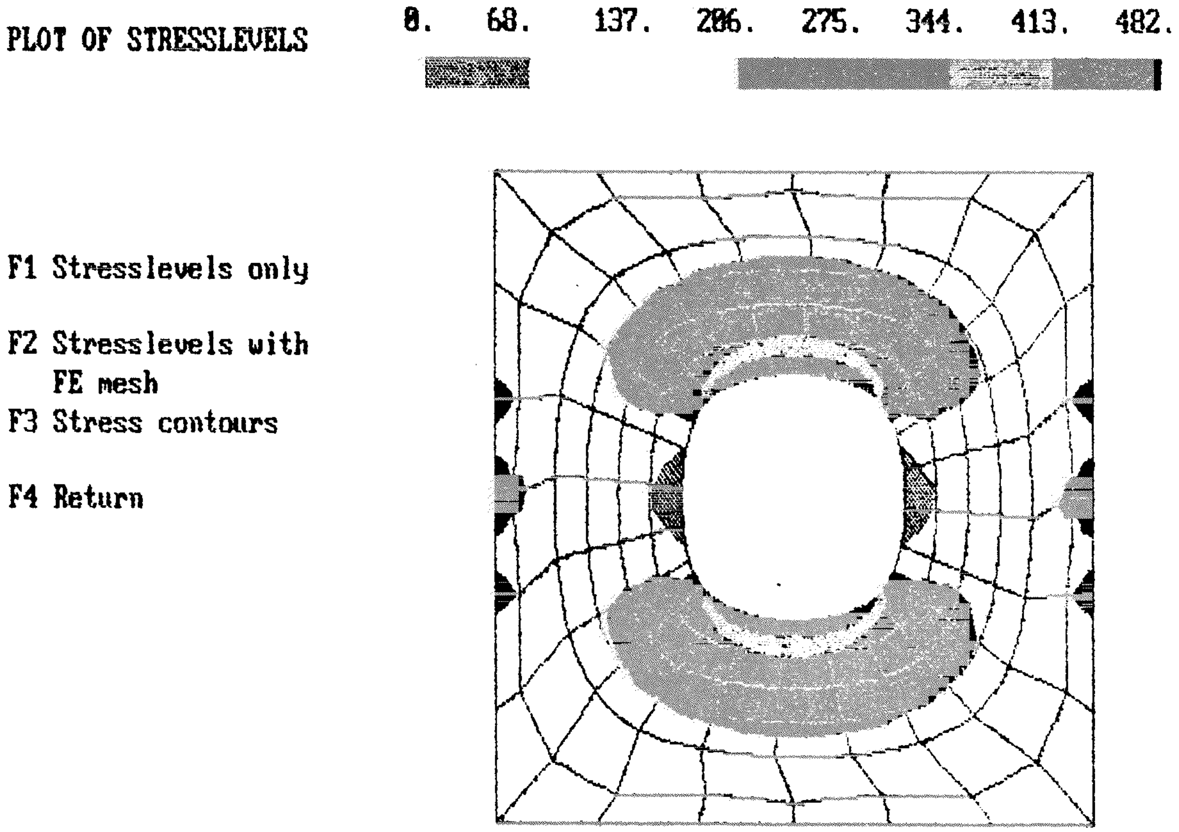

PLOT OF STRESSLEVELS
0.
68.
137.
206.
275.
344.
413.
482.
F1 Stresslevels only
F2 Stresslevels with
FE mesh
F3 Stress contours
F4 Return

Fig.6d

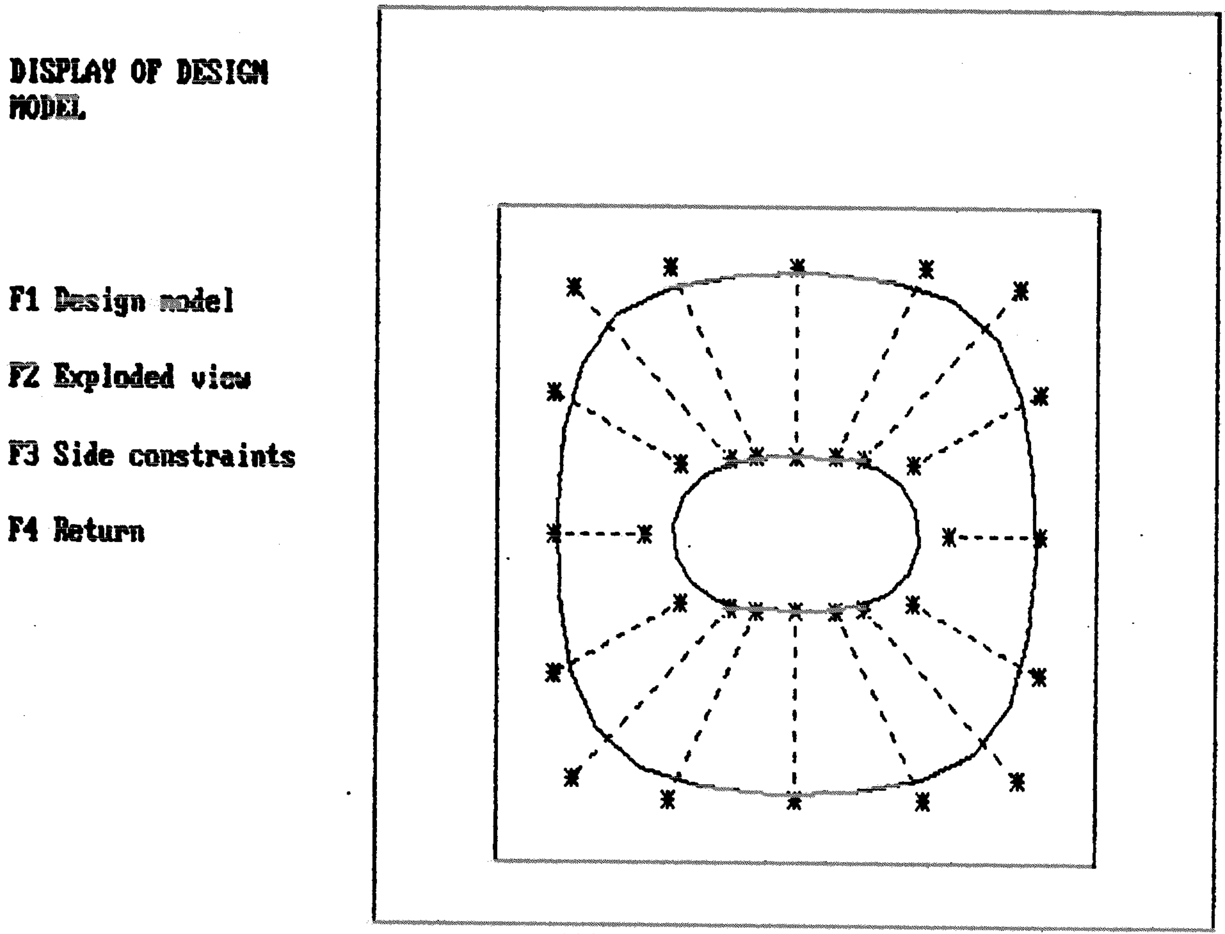

DISPLAY OF DESIGN MODEL
F1 Design model
F2 Exploded view
F3 Side constraints
F4 Return

Fig.6e

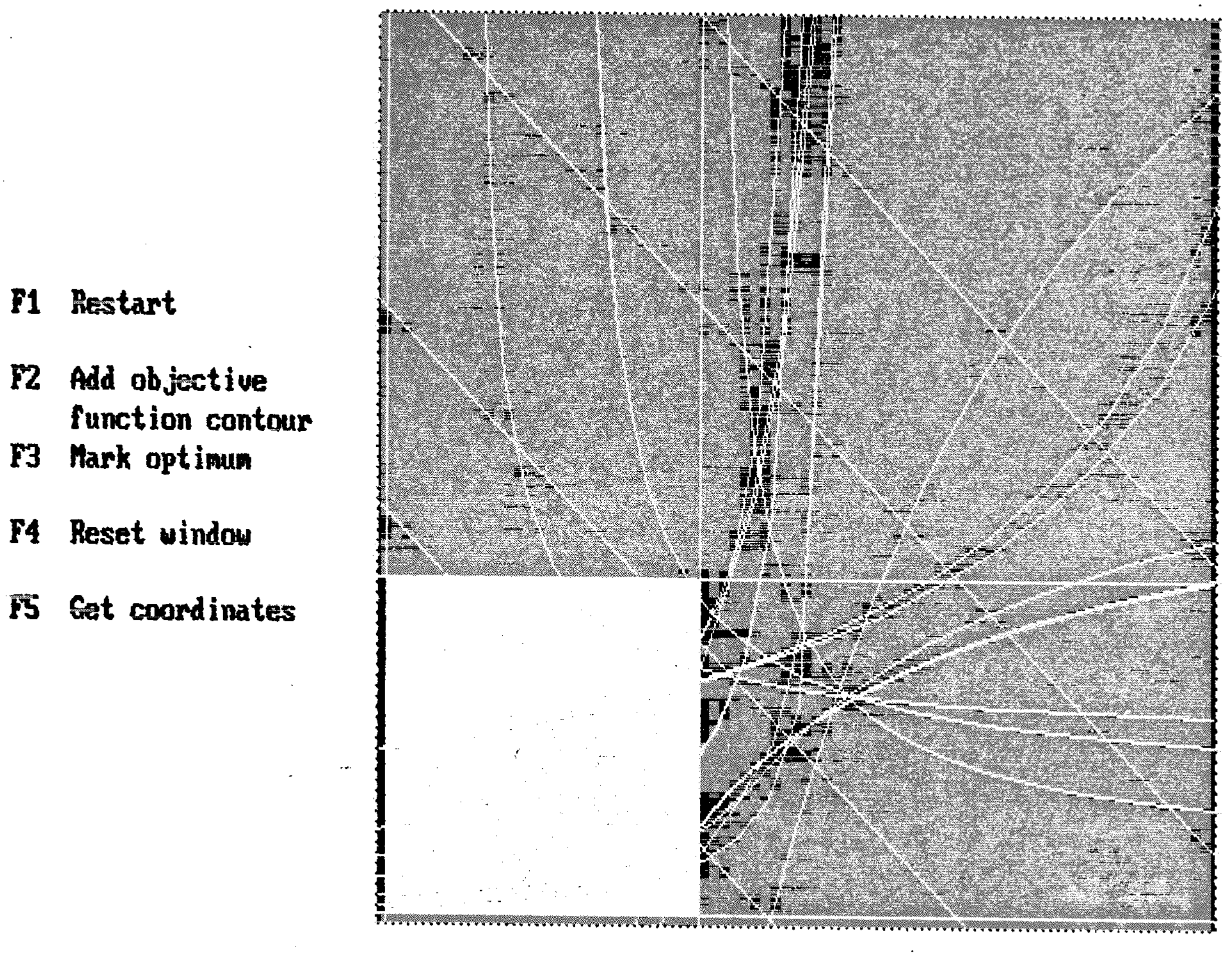

F1 Restart
F2 Add objective function contour
F3 Mark optimum
F4 Reset window
F5 Get coordinates

Fig.6f

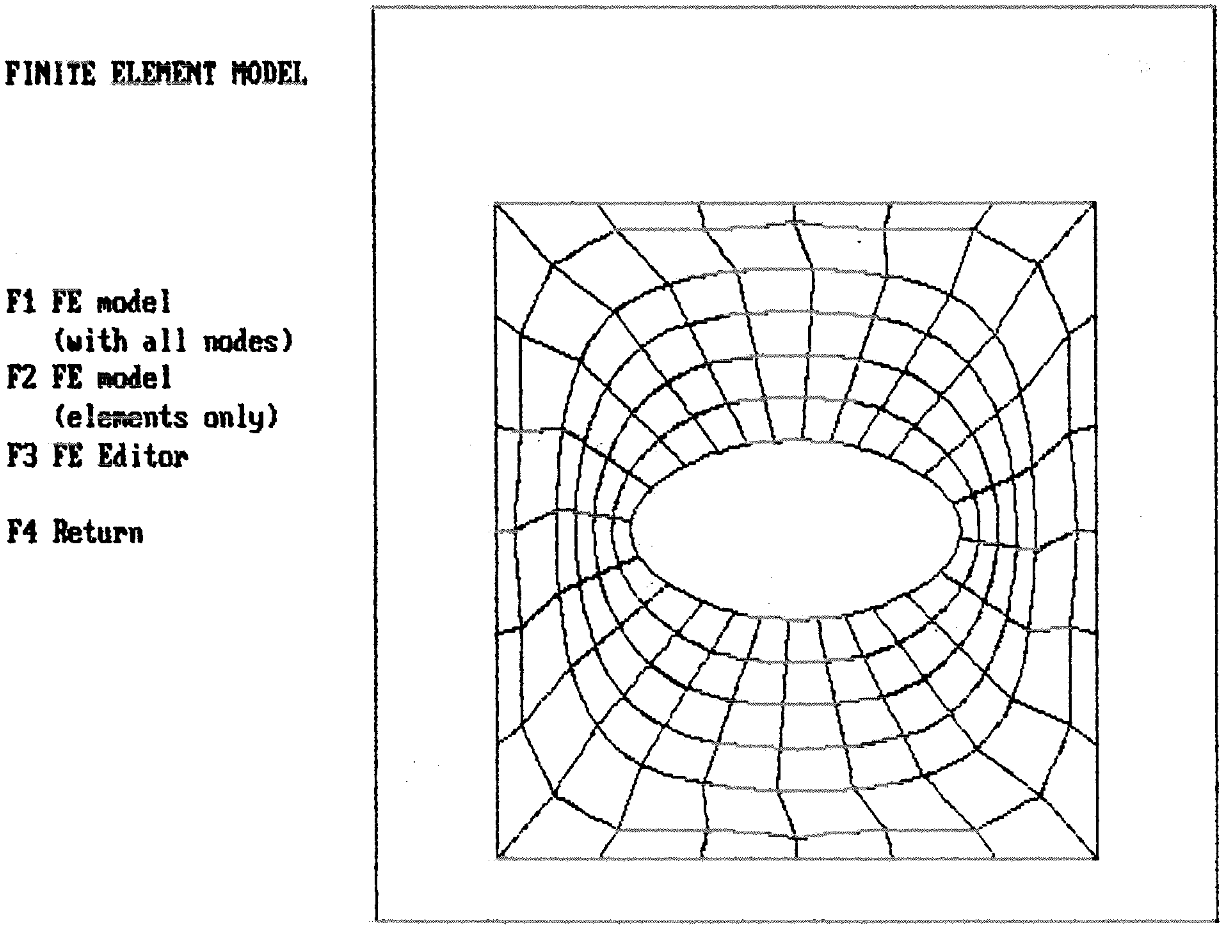

FINITE ELEMENT MODEL
F1 FE model
(with all nodes)
F2 FE model
(elements only)
F3 FE Editor
F4 Return

Fig.7a

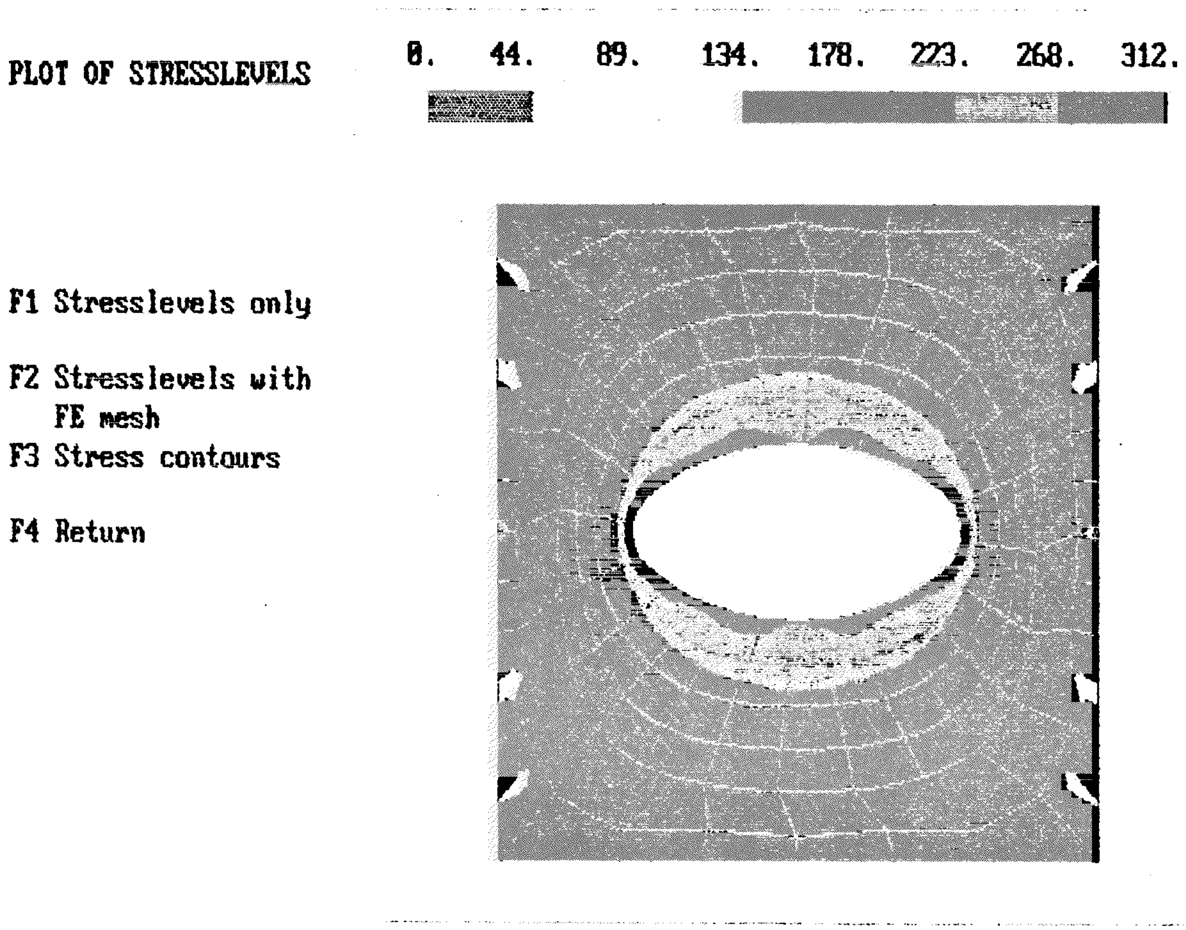

PLOT OF STRESSLEVELS
0.
44.
89.
134.
178.
223.
268.
312.
F1 Stresslevels only
F2 Stresslevels with
FE mesh
F3 Stress contours
F4 Return

Fig.7c

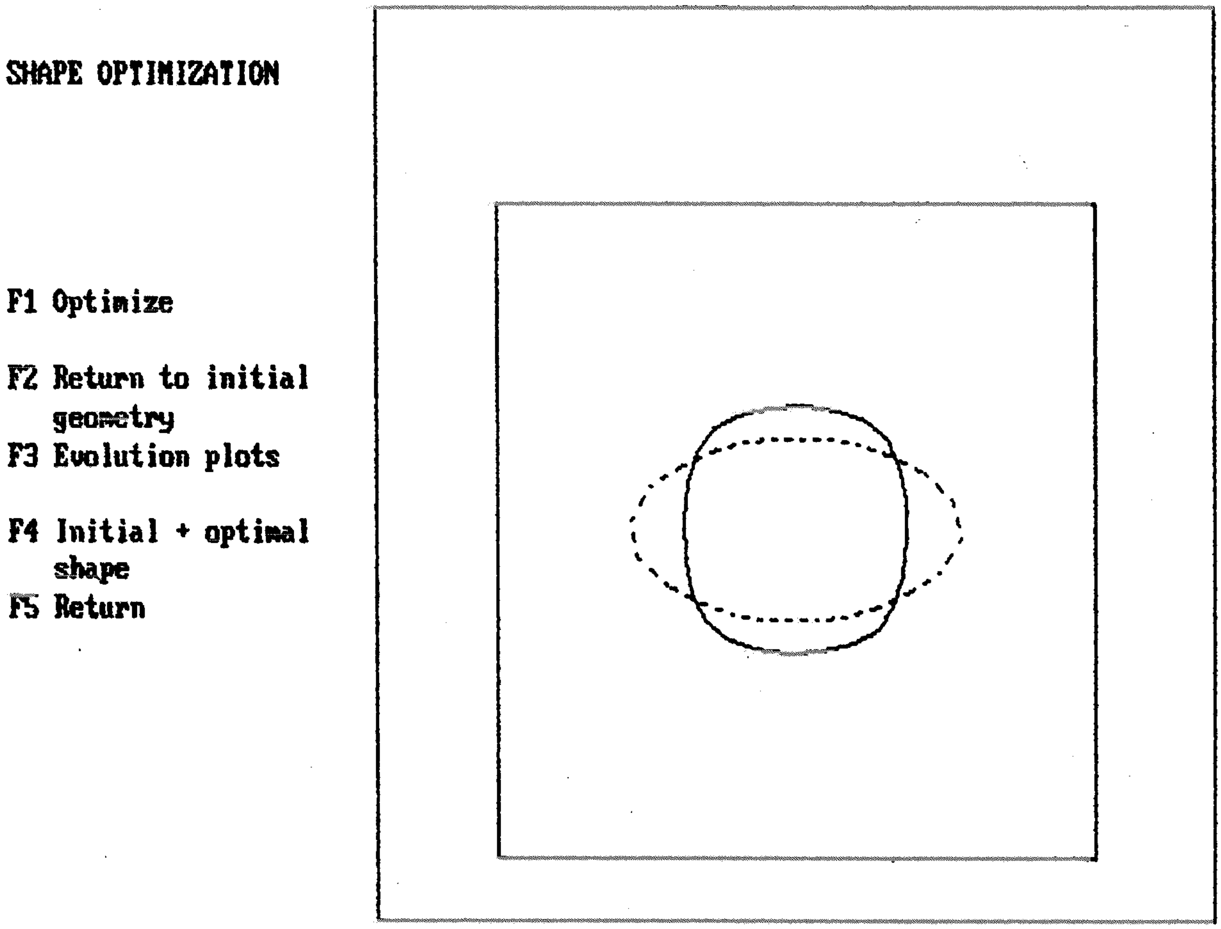

SHAPE OPTIMIZATION
F1 Optimize
F2 Return to initial geometry
F3 Evolution plots
F4 Initial + optimal shape
F5 Return

Fig.7b

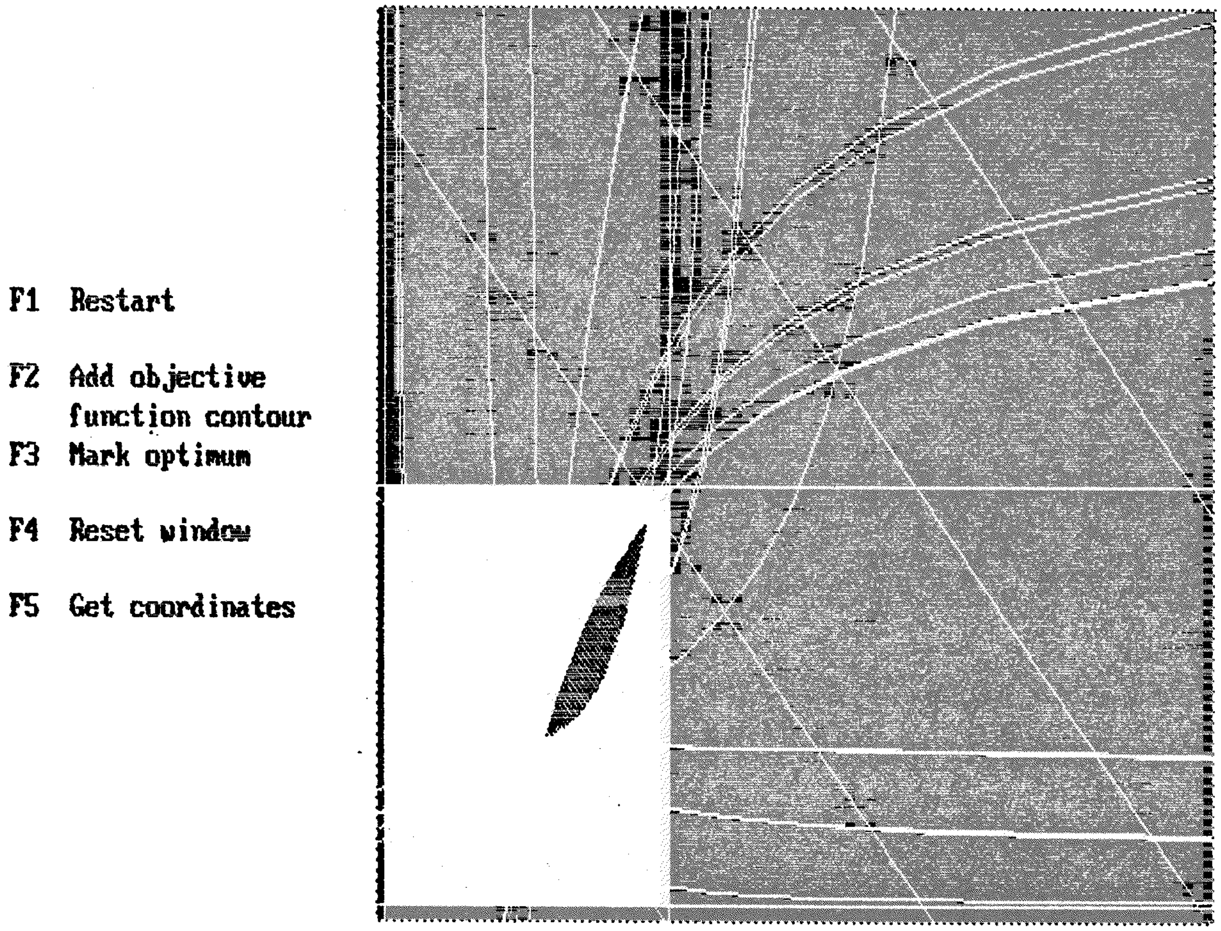

F1 Restart
F2 Add objective function contour
F3 Mark optimum
F4 Reset window
F5 Get coordinates

Fig.7e

F1 OBJECTIVE FUNCTION

F2 DESIGNVARIABLE

F3 CONSTRAINTS

F4 RETURN

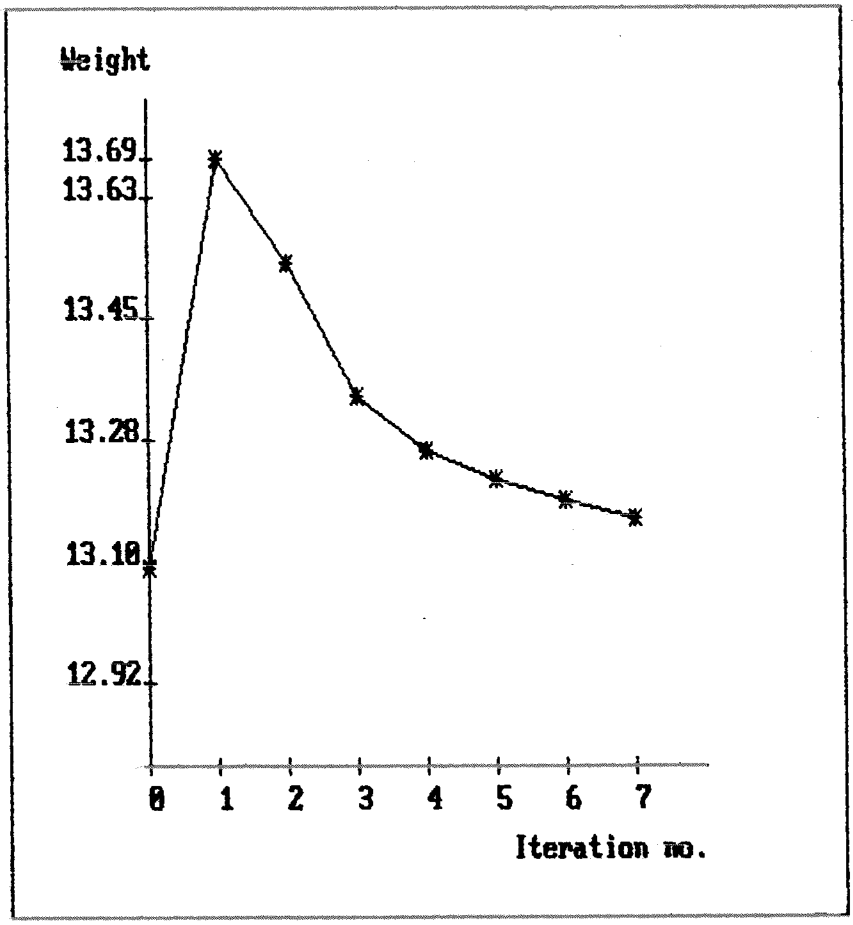


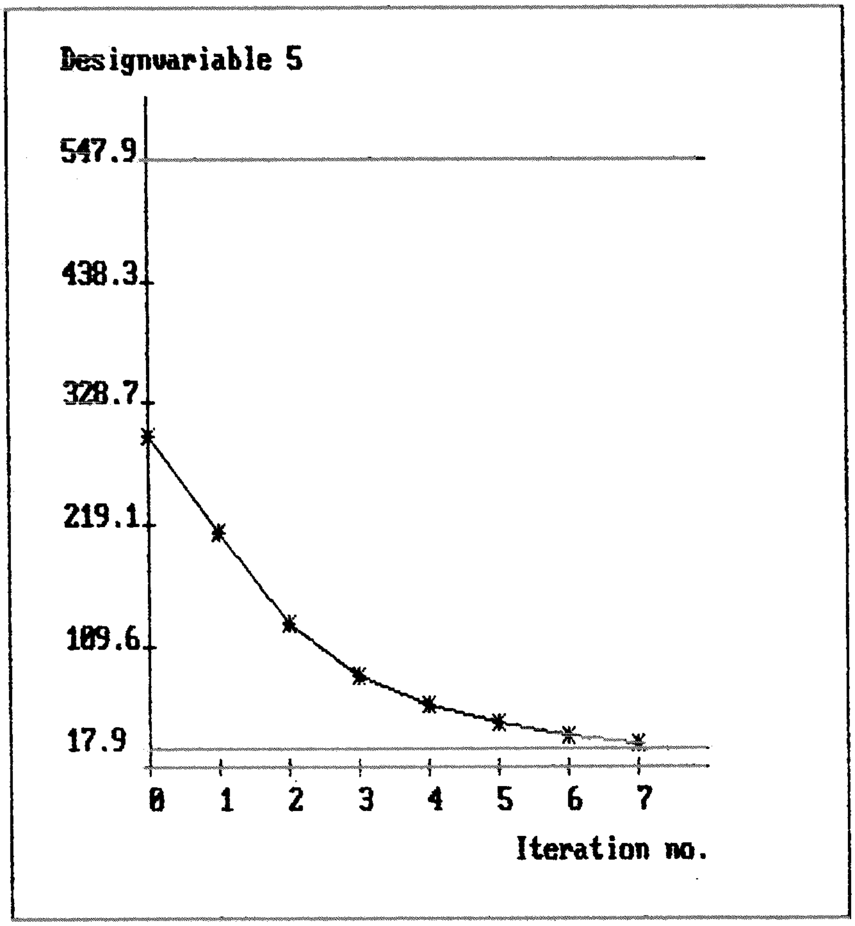
Figure 7d

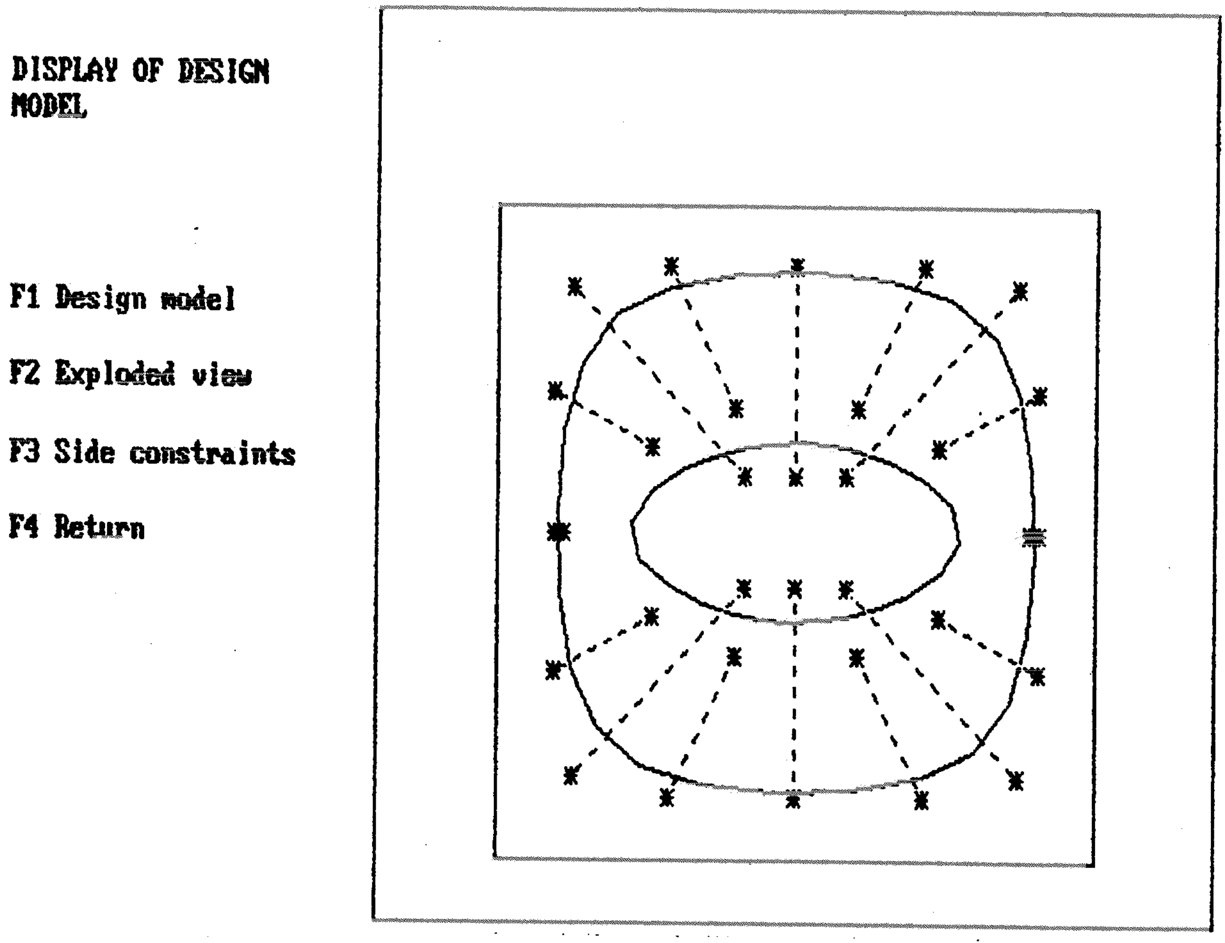
F1 OBJECTIVE FUNCTION

F2 DESIGNVARIABLE

F3 CONSTRAINTS

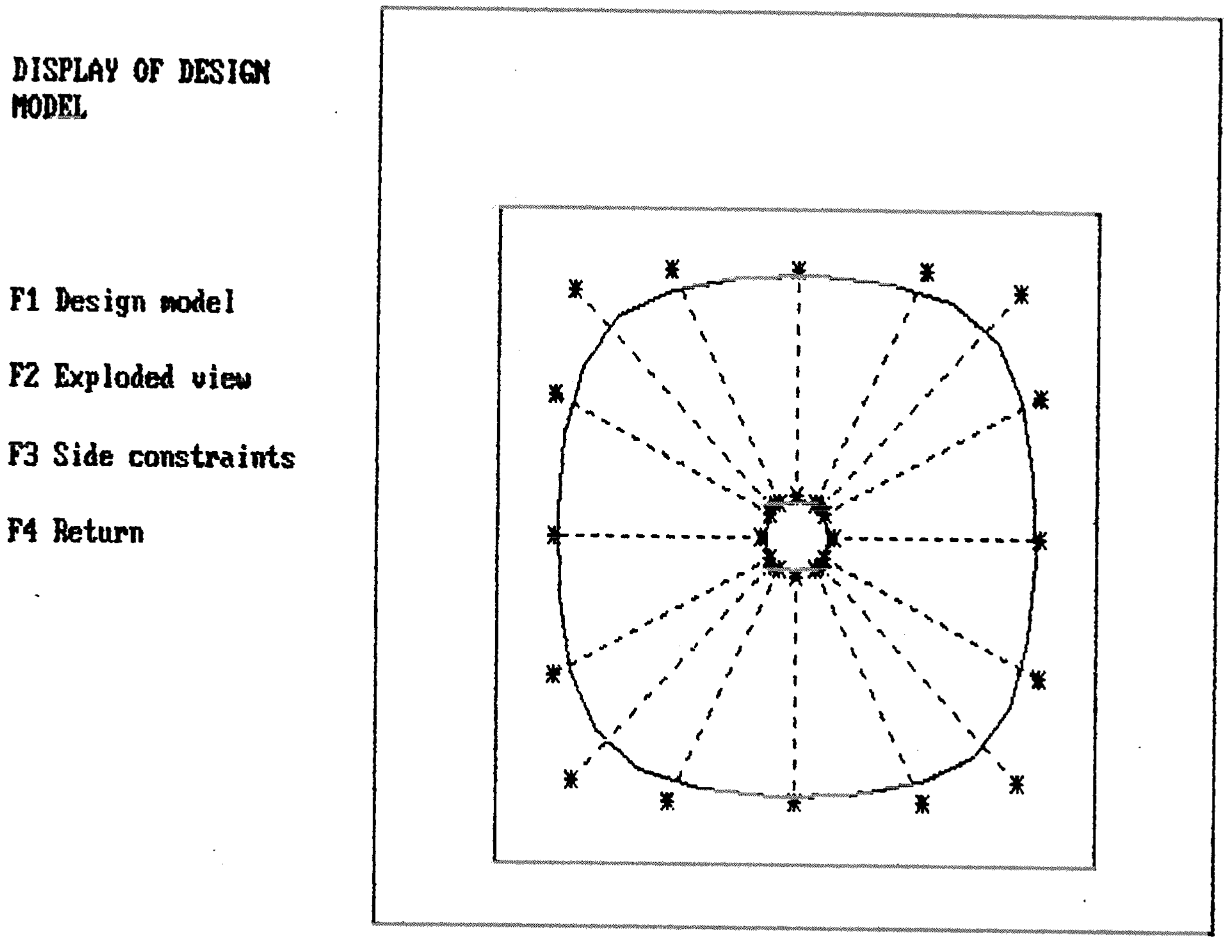
F4 RETURN





DISPLAY OF DESIGN MODEL

Fig.8a



DISPLAY OF DESIGN MODEL

Fig.8b

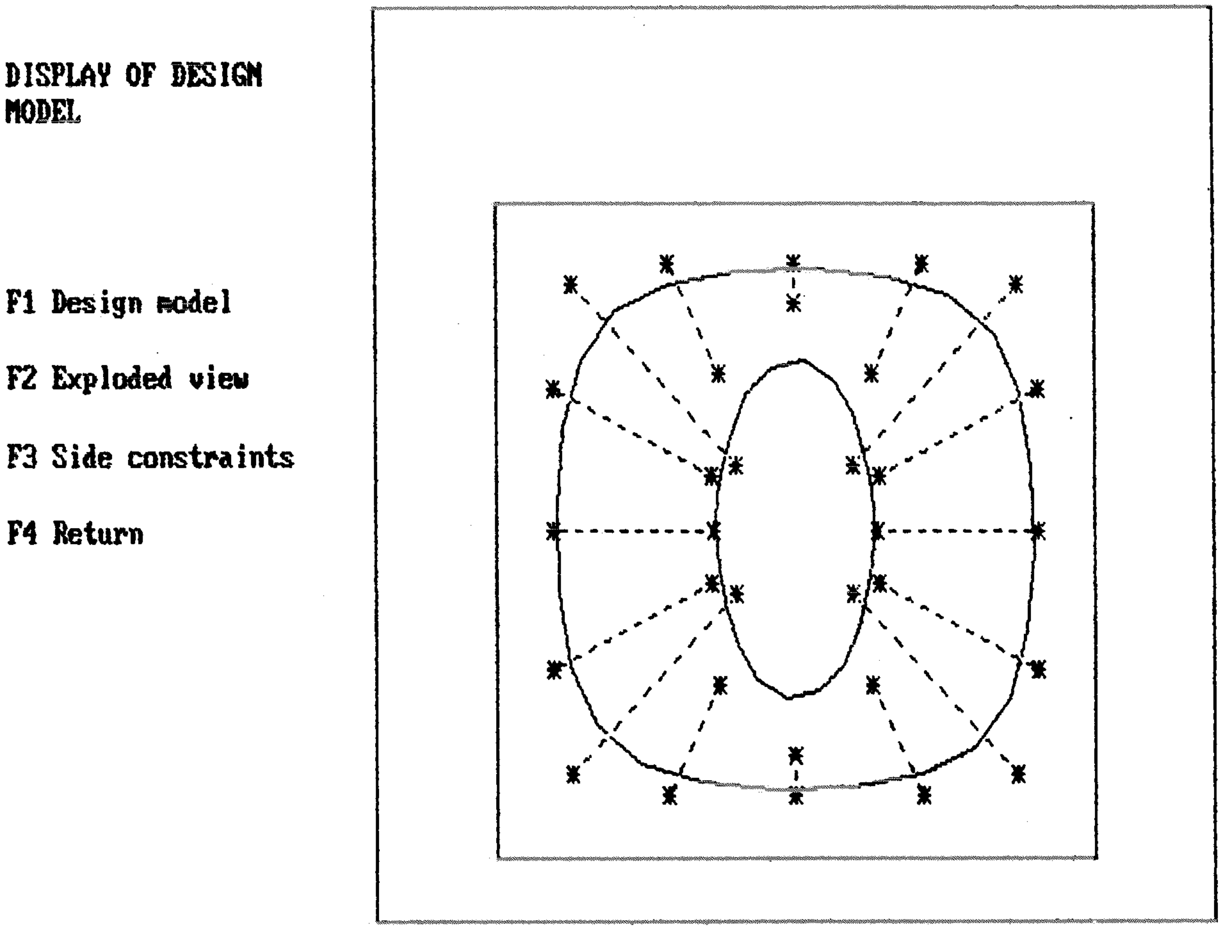

DISPLAY OF DESIGN MODEL
F1 Design model
F2 Exploded view
F3 Side constraints
F4 Return

Figure 8c

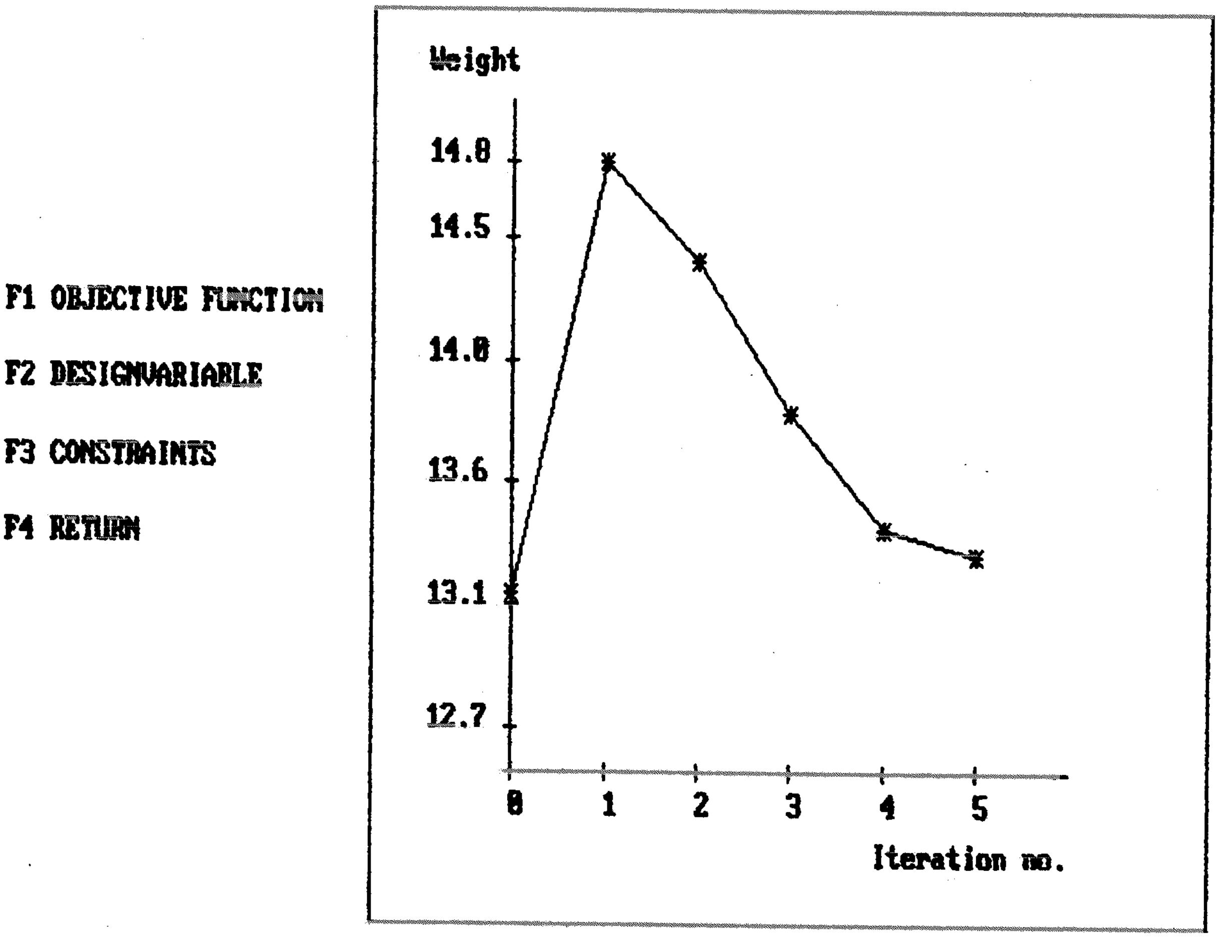
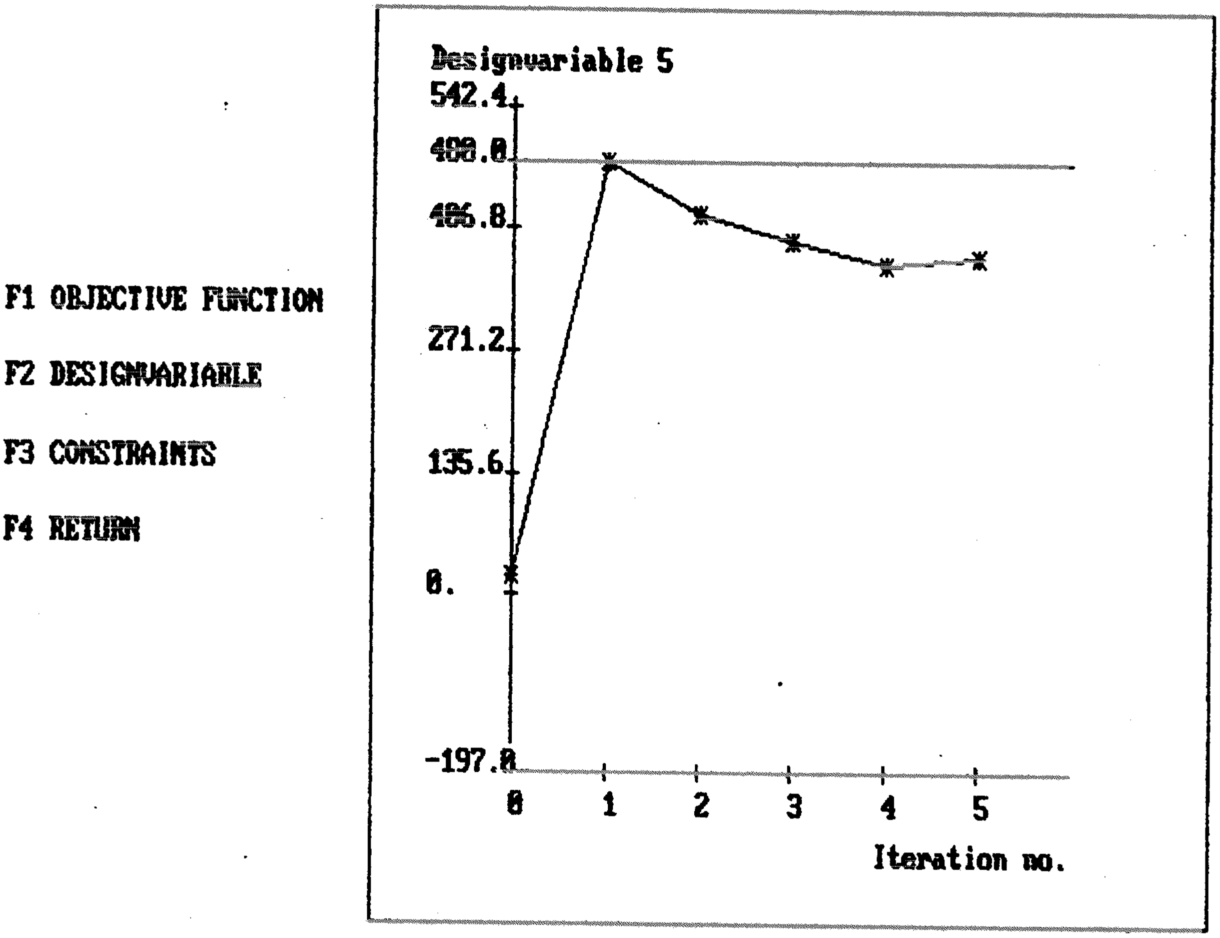

F1 OBJECTIVE FUNCTION
F2 DESIGNVARIABLE
F3 CONSTRAINTS
F4 RETURN
Weight
14.8
14.5
14.0
13.6
13.1
12.7
0
1
2
3
4
5
Iteration no.

Figure 8c


F1 OBJECTIVE FUNCTION
F2 DESIGNVARIABLE
F3 CONSTRAINTS
F4 RETURN
Designvariable 5
542.4
480.0
406.8
271.2
135.6
0.
-197.8
0
1
2
3
4
5
Iteration no.

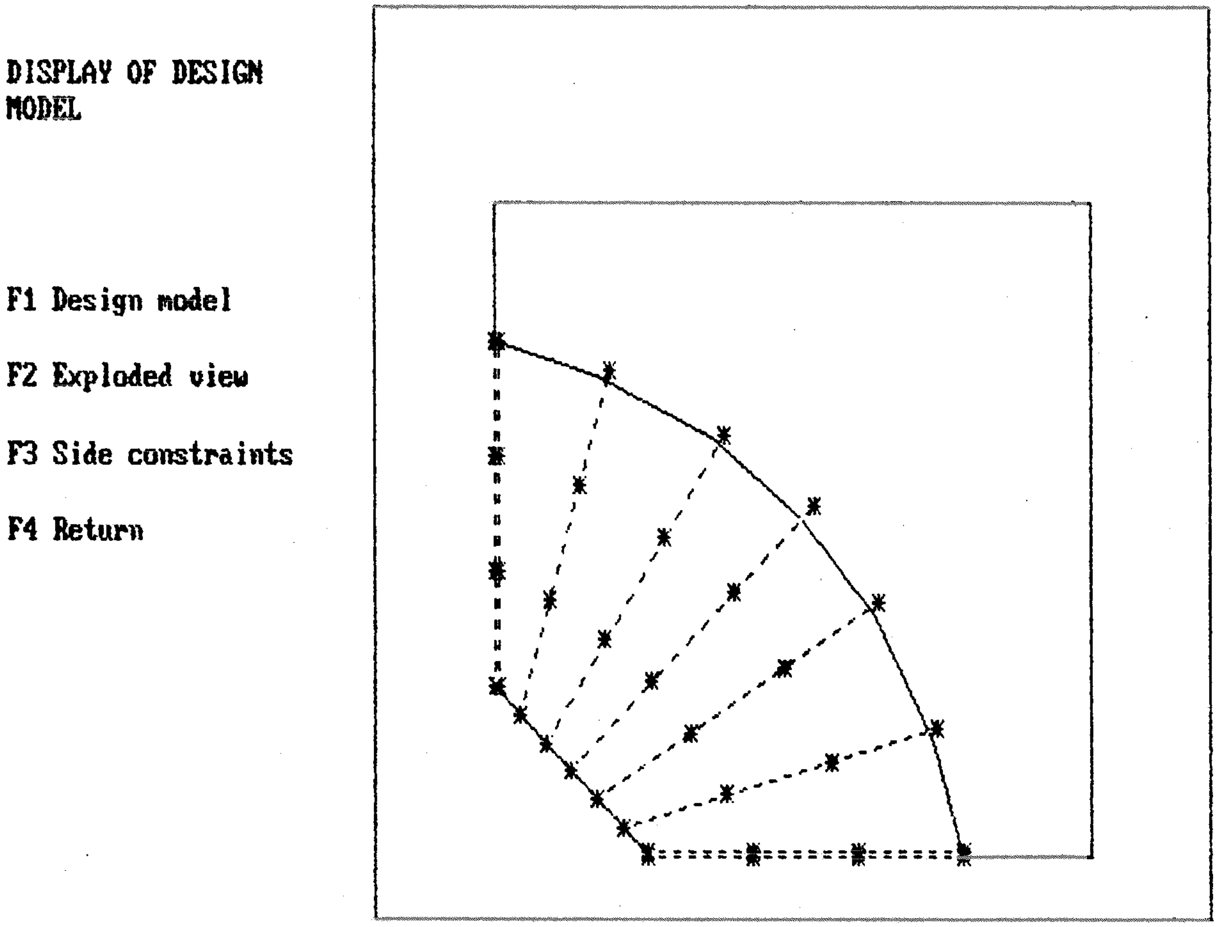

DISPLAY OF DESIGN MODEL
F1 Design model
F2 Exploded view
F3 Side constraints
F4 Return

Fig.9a

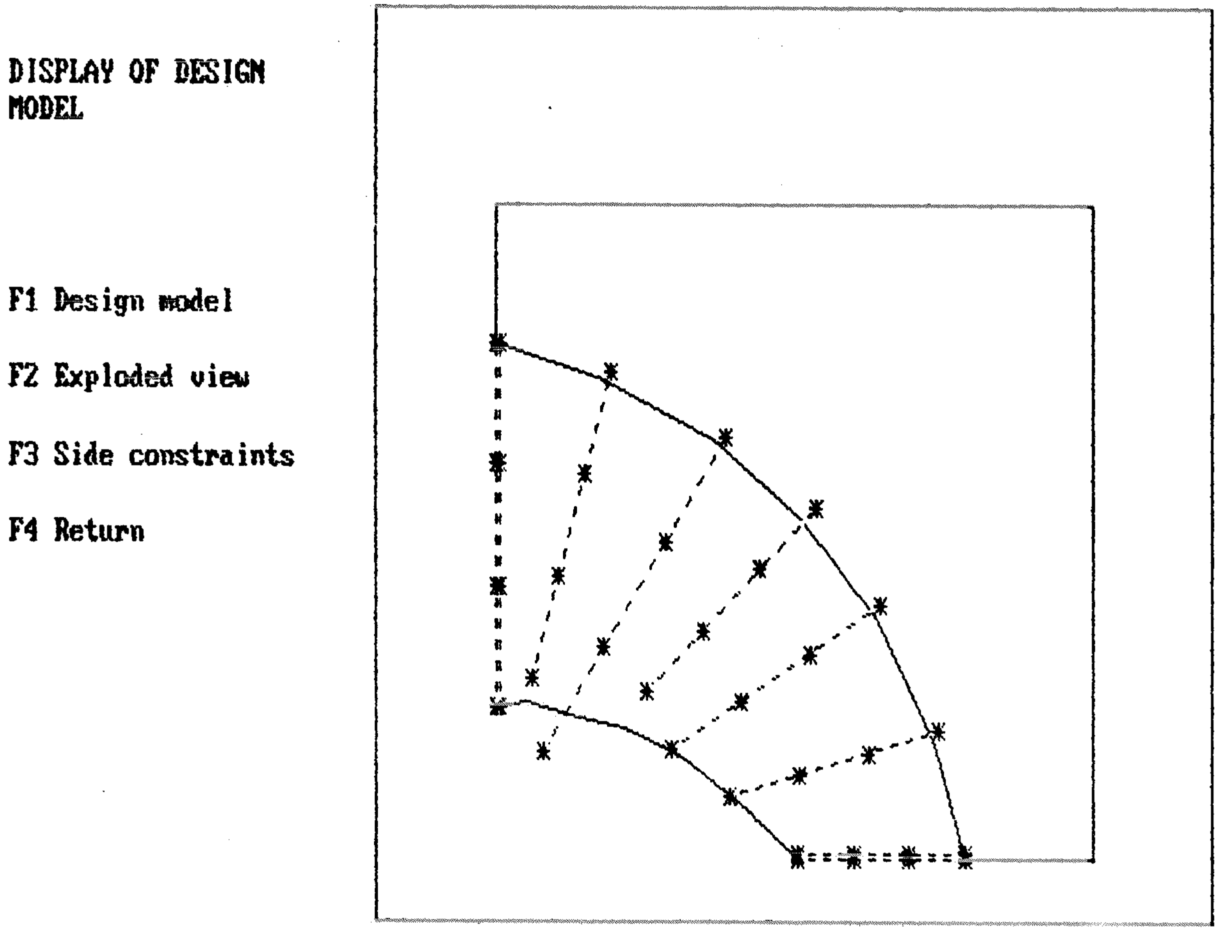

DISPLAY OF DESIGN MODEL
F1 Design model
F2 Exploded view
F3 Side constraints
F4 Return

Fig.9b