Elección y adaptación al Sistema SAMCEF de un Método de Continuación para la resolución de sistemas estáticos no lineales

Ever Barbero
Alberto Cardona
Sergio Idelsohn
Grupo de Tecnología Mecánica
Instituto de Desarrollo Tecnológico para la Industria Química
C.C. 91, 3000 Santa Fé, Argentina

RESUMEN

En este trabajo se expone la formulación de los Métodos de Continuación para luego introducir una modificación al método de plano normal constante, la cual mejora la robustez y propiedades de convergencia del algoritmo. Se presenta el algoritmo final que fue instalado en el modulo no lineal estático del sistema SAMCEF. Finalmente se presentan algunos ejemplos para mostrar las capacidades del programa.

ABSTRACT

In this work we review the formulation of Continuation Methods. Next, we introduce a modification to the constant-normal-plane method and show how it improves the reliability and convergence characteristics of the algorithm. After that, we present the final algorithm, which was installed in the static, noninear module of the system SAMCEF. Finally, we include some selected examples in order to highlight the capabilities of the program.

1. CONTINUATION METHODS

The Structural Finite Element modelization taking into account geometrical and material non-linearities leads to non-linear systems as:

$$G(U) = S(U) \, . \, U - P = 0 \qquad (1)$$

with total displacement vector U, external loads vector P and stiffnesss matrix S.

For proportional loading, the loads may be expressed by one load factor λ and a vector of reference loads P; thus, in this case, eq (1) may be written as:

$$G(U) = S(U) \, . \, U - \lambda P = 0 \qquad (2)$$

In continuation methods the load parameter (λ) is treated as an unknown in addition to the structural displacements; then, a constraint equation is introduced to define the problem completely. In this way, a problem having N d.o.f. is replaced by one accounting N+1 unknowns. Eq (2) becomes:

$$G_i(U, \lambda) = 0 \qquad i = 1..N \qquad (3a)$$

$$G_{N+1}(U, \lambda) = 0 \qquad (3b)$$

Without loss of generality the constraint equation (3b) may be written [1] as:

$$G_{N+1}(U, \lambda) = g(U, \lambda) - \eta = 0 \qquad (4)$$

where η is the path parameter and the function g determines the method to be used, ie. load control, displacement control, etc.

Notation of reference [2] has also been used here (see fig 1):
- total values for any configuration are indicated by a left superscript.
- values referred to the beginning of current increment (last equilibrium point) are denoted by a right superscript indicating the iteration number.
- incremental values for one iteration are preceded by the symbol Δ.

With the continuation formulation, the non-linear problem is stepwise linearized and the linearization error is corrected by additional equilibrium iterations. On assuming that the equilibrium configuration m (corresponding to increment m) has already been determined, a predictor scheme is used to advance toward the m+1 increment, that is:

$$^{1}U = {}^{m}U + [{}^{m}J]^{-1} \lambda^{1} P \qquad (5)$$

and then, a corrector scheme is used to search the equilibrium configuration m+1, that is:

$$ {}^{j}U = {}^{i}U - [{}^{i}J]^{-1} G^{i} \qquad (6)$$

where j=i+1 is the current iteration, J is the Jacobian matrix and G is the residue taken from eq (2).

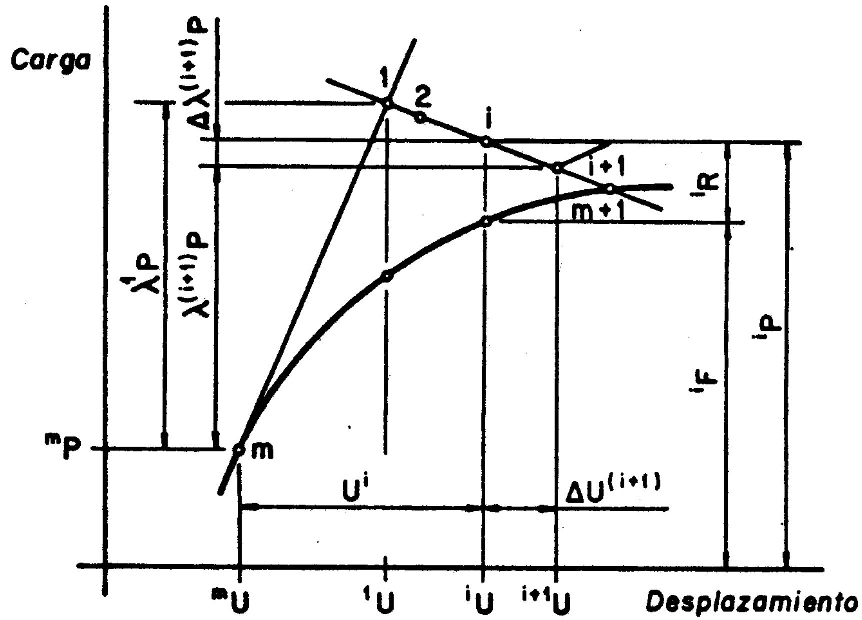

Fig 1. The Constant-Normal-Plane Method for one d.o.f. sistem

If one neglects variations of the loads due to geometrical changes, the Jacobian matrix is the same as the tangent stiffness matrix, usually available in general-purpose, non-linear finite element codes, that is:

$$ J = \frac{\partial G}{\partial U} = \frac{\partial}{\partial U} [\, S\,(U)\ .\ U - \lambda P \,] = K \qquad (7)$$

therefore eq (6) may be written as:

$$ {}^{i}K\ ({}^{j}U - {}^{i}U) = - G^{i} \qquad (8)$$

then using eq (2), introducing incremental displacements U^{j}, internal forces ${}^{i}F$ and external loads ${}^{i}P$; and taking into account that the load level may change during iterations, we have:

$$ K^{*}\ .\ \Delta U^{j} = \Delta P^{j} + {}^{i}P - {}^{i}F \qquad (9)$$

where the superscript * indicates that updating of the stiffness matrix may be made at some previous iteration as in the modified Newton-Raphson technique. Now, introducing out-of-balance forces $R^i = {}^iP - {}^iF$ and taking into account that for proportional loading $P^j = \Delta\lambda^j P$, we have:

$$K^* \ \Delta U^j = \Delta\lambda^j \ P + R^i \tag{10}$$

2. DISPLACEMENT CONTROL METHOD

In the displacement control method [3], a single displacement, eg. U_k, is chosen to control the stepwise advance on the solution. The path parameter η in eq (4) is equated to U_k and then, the constraint eq (3b) becomes $\eta = U_k$.

In the former versions of the method the system (10) was partitioned isolating the k_{th} equation, corresponding to the controlled displacement U_k, thus removing the singularity at critical point. Later it was noted [4] that it is not very likely to obtain exactly the singular point; therefore, it was proposed to use the whole tangent stiffness matrix in eq (10) without partitioning it.

Next, eq (10) is solved in two steps:

$$K^* \ . \ \Delta U^{jI} = P^i \tag{11a}$$

$$K^* \ . \ \Delta U^{jII} = R^i \tag{11b}$$

and then both solutions are added to form the displacement increment:

$$\Delta U^j = \Delta\lambda^j \ \Delta U^{jI} + \Delta U^{jII} \tag{12}$$

This vector includes also the prescribed component:

$$\Delta U^j{}_k = \Delta\lambda^j . \Delta U^{jI}{}_k + \Delta U^{jII}{}_k \tag{13a}$$

and taking into account that $\overline{U}_k$ does not change during iterations, we have:

$$U^j{}_k = \overline{U}_k \ \delta_{ij} \tag{13b}$$

The eq (13a,b) is used in the predictor scheme (j=1) to find the first incremental load parameter, and assuming perfect convergence at the last increment, we have:

$$\Delta\lambda^1 = \frac{\overline{U}_k}{U^{1I}{}_k} \qquad \text{for } j = 1 \tag{14a}$$

Finally eq (13a,b) is again used to find the change in load parameter during corrector iterations as follows:

$$\Delta\lambda^j = \frac{\Delta U^{jII}{}_k}{\Delta U^{1I}{}_k} \qquad \text{for } j > 1 \tag{14b}$$

3. CONSTANT-NORMAL-PLANE METHOD

For the constant-normal-plane method [5,6] the constraint eq (3b) is:

$$[\Delta U^{1}]^{T}\,[\Delta U^{1}] + (\Delta\lambda^{1})^{2} = \eta^{2} \quad \text{for } j = 1 \tag{15a}$$

$$[\Delta U^{1}]^{T}\,[\Delta U^{j}] + \Delta\lambda^{1}\,.\,\Delta\lambda^{j} = 0 \quad \text{for } j > 1 \tag{15b}$$

where η is, as in eq (4), the path parameter. In this way, the new equilibrium configuration (m+1) will be at the intersection of the solution path with the plane defined by eq (15a), which is normal to the tangent at point m (see fig 1) and it is located at a distance η from this point.

As in the displacement control method, eqs (11a,b) are used to find ΔU^{jI} and ΔU^{jII} and eq (12) is used to form ΔU^{j}. Next, eq (12) is inserted in eq (15a); then, assuming again perfect convergence at the last increment, we have:

$$\Delta\lambda^{1} = \frac{\eta}{(1 + [\Delta U^{1I}]^{T}.[\Delta U^{1I}])^{1/2}} \quad \text{for } j=1 \tag{16a}$$

and from eq (15b), we have:

$$\Delta\lambda^{j} = -\,\frac{[\Delta U^{1I}]^{T}.[\Delta U^{jII}]}{(\Delta\lambda^{1} + [\Delta U^{1I}]^{T}.[\Delta U^{jI}]} \quad \text{for } j>1 \tag{16b}$$

4. SCALING FACTOR

Eqs (14a,b) and therefore eq (15a,b) are formed with displacements and the load parameter, which have different dimensions. This fact introduces a dependence of the algorithm on the magnitudes used in the formulation of the problem.

Let us examine eq (16b) for a system with one d.o.f., hence all variables involved are scalars,

$$\text{If} \quad \Delta\lambda^{1} \ll [\Delta U^{1}].[\Delta U^{jI}] \tag{17}$$

$$\text{then} \quad \Delta\lambda^{j} = -\,\frac{\Delta U^{jI}}{\Delta U^{jII}} \tag{18}$$

which is the same as eq (14b); thus, the algorithm becomes the displacement control one. This is a severe drawback because the displacement is not really controlled since eq (14a) is not used, and the algorithm tries to maintain the displacement calculated at the first iteration (predictor), which may not correspond to any neighboring equilibrium point.

Besides, if $\Delta\lambda^1 >> [\Delta U^1].[\Delta U^{jI}]$ (19a)

also $\Delta\lambda^1 >> [\Delta U^1].[\Delta U^{jII}]$ (19b)

and then $\Delta\lambda^j == 0$ (20)

In this way, the algorithm becomes the load control method and it loses all utility near limit points.

It was proposed in [7,8] to neglect the increment of load parameter $\Delta\lambda^1$ in eq (15). This reduces by one the number of variables used to define the surface on which problem solution is searched. For one d.o.f. systems eq (16b) is reduced to eq (18) which is the same as eq (14b) leading again to displacement control method.

In this work, a scaling factor alpha is introduced in such a way that the load parameter has the same weight as a typical displacement $\|\Delta U\|/N$; that is, for increment m+1, alpha is defined as:

$$\alpha = \frac{\| {}^{m}U - {}^{m-1}U \|}{N} \; \frac{{}^{m+1}\eta}{{}^{m}\eta} \qquad (21)$$

where $\| \;\|$ indicates usual Euclidean norm over the N d.o.f. Now, constraint eqs (15a,b) are rewritten as:

$$[\Delta U^1]^T [\Delta U^1] + \alpha^2.(\Delta\lambda^1)^2 = \eta^2 \quad \text{for } j=1 \qquad (22a)$$

$$[\Delta U^1]^T [\Delta U^j] + \alpha^2.\Delta\lambda^1.\Delta\lambda^j = 0 \quad \text{for } j>1 \qquad (22a)$$

which lead to:

$$\Delta\lambda^1 = \frac{\eta}{(\alpha^2 + [\Delta U^{1I}]^T.[\Delta U^{1I}])^{1/2}} \quad \text{for } j=1 \quad (23a)$$

$$\Delta\lambda^j = \frac{[\Delta U^{1I}]^T.[\Delta U^{jII}]}{(\alpha^2 \Delta\lambda^1 + [\Delta U^{1I}]^T.[\Delta U^{jI}]} \quad \text{for } j>1 \quad (23b)$$

One has to note that, for systems with many d.o.f. for which $[\Delta U^1]^T.[\Delta U^{jI}] >> \Delta\lambda^1$, the modified algorithm tends to that proposed in [7,8]. But, on any other cases, scaling factor helps to solve scaling problems. In this way, users need not be aware of considerations about dimensions to formulate problems and magnitudes for reference loads.

5. AUTOMATIC INCREMENTAL PROCEDURE

The algorithm may trace solution paths automatically when the step (path parameter) is specified for each increment in an automatic way. The step size for any increment is scaled to the step size of the previous increment on relating the number of iterations in the last increment (i) to a user defined number (#i). The most usual scheme is:

$$ {}^{m}\eta = {}^{m-1}\eta \, . \, SQRT(\#i/i) \qquad (24) $$

η is also limited by the user input δ and γ so that:

$$ \delta < \frac{{}^{m}\eta}{{}^{1}\eta} < \gamma \qquad (25) $$

Tangent stiffness matrix reevaluation is controlled by three parameters i^1, i^2, i^3, so that reevaluation occurs for iterations i^1, i^2, and $i^2 + n\, i^3$, for all n until a maximum iteration number i^{max} is reached. Introduction of the last parameter is to prevent failed convergence difficult points. The user may overcome such difficult points on restarting with a prescribed path parameter or using a displacement control step.

Free combinations of load, displacement, and global (constant-normal-plane) control steps are allowed. The reference load P is defined at the first increment (load control) and may be redefined at any increment on specifying another load control step; this allows flexibility for loading.

Convergence is accepted after a norm for residue [9] is bellow a user defined parameter e_1; tangent stiffness matrix is discontinued when the norm is bellow another parameter e_2.

Unloading is initiated whenever a negative pivot is found during frontal solution for the first iteration of each increment. When stiffening occurs, reloading is automatically initiated since tangent stiffnes matrix becomes again definite positive.

6. ALGORITHM

```
var reeval,load-control,displacement-control,
    global-control : logical;
begin
m:=0; η:=1; P:=^oP;                                  {set reference load}
while m<=m^max do begin
  i:=0; m:=m+1;
  K^i:=^(m-1)K; R^i:=^(m-1)R;
  set method for current increment;
  if load-control then P:=^mP;                               {see text}
  while i<=imax and e>e_1 do begin                 {start iterations}
    j:=i+1;          {i:last iteration, j:current iteration}
    reeval:=((j=i_1) or (j=i_2) or (j=i_2+n i_3)) and (e>e_2);
    if reeval then update K^i;
    if (reeval and not load-control)
           or (load-control and j=1)
      then solve K^i. ΔU^jI = P
      else copy ΔU^jI = ΔU^iI;
    solve K^i. ΔU^jII = - R^i;
    if load-control then if j=1 then Δλ^j:=1 else  Δλ^j:=0;
    if displacement-control then
                               Ū_k                       ΔU^jII_k
      if j=1 then  Δλ^j:= -------- else  Δλ^j:= - --------;
                             U^jI_k                      ΔU^jI_k
    if global-control then
      if  j=1  then begin
        α :=α_1.η; {set scaling factor for current incr.}
                              η
        Δλ^j  = --------------------------------------; end
                  (α^2 + [ΔU^1I]^T.[ΔU^1I] )^(1/2)

                               [ΔU^iI]^T.[ΔU^jII]
        else  Δλ^j  = - ---------------------------------;
                         (α^2.Δλ^i + [ΔU^iI]^T.[ΔU^jI]

    ΔU^j  =   Δλ^j.ΔU^jI + ΔU^jII;          {form disp. vector}
    U^j:=U^i + ΔU^j;             {update disp. for current incr.}
    ^jU:=^iU+ ΔU^j;                  {update total displacements}
    α_1:= U^j/N/η;                        {update scaling factor}
    ^jP:=^iP+ Δλ^j P;                              {update loads}
    find out-of-balance forces R^j;
    evaluate norm of residue;         {convergence parameter}
    i:=j;
    end while;                                   {next iteration}
  η:=η.sqrt(#i/i);
  check for condition eq (25);
  end while;                                     {next increment}
end.
```

NOTE: this algorithm is written in PASCAL like notation but readers only have to be awarded of use of semicolon to end each statement and use of words begin end to refer a group of statements as a one.

7. EXAMPLES

These examples have been analyzed on a VAX 11/780 computer using the Finite Element Code SAMCEF [10]. The geometrical non-linearity is based on the Total Lagrangian Formulation. Shell structures are idealized by special three-dimensional elements developed in [11]. Full Newton iterations were used throughout the examples, although use of the modified Newton Raphson technique is possible.

Figure 2 shows a typical snap through characteristic. The mechanical model was analyzed in reference [12]. Table I shows the performance obtained with both the original scheme ($\alpha=1$) and the scheme proposed here. A first load control step of 0.3 KN is is imposed and further increments are performed with the global control' method. The step size is chosen automatically with eq (24).

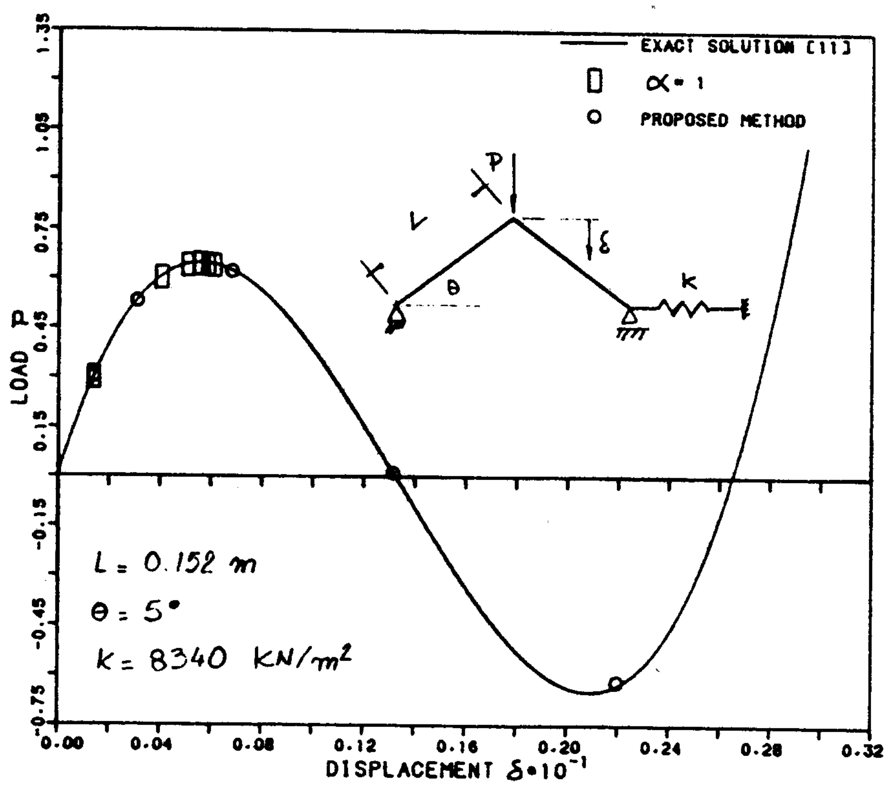

Fig 2. Load vs Displacement diagram showing snap through characteristic

It can be seen from table I that without scaling, convergence failed for the third increment and the automatic solution procedure is no longer possible. The step size (η) have to be fixed at a very low value by user input. Several trials and previous knowledge of the solution are necessary to select this input. Otherwise, a very little step have to be employed with the consequent high solution cost.

TABLE I. Iterative performance for example 1 (figure 2)

	$\alpha = 1$		proposed $\alpha \neq 1$	
step	η	iter	η	iter
1	1.41	8	1.41	8
2	1.58	10	1.58	3
3	0.5	10	2.88	4
4	0.2	10	4.56	4
5	0.2	7	7.21	4
6	0.2	6	11.40	4

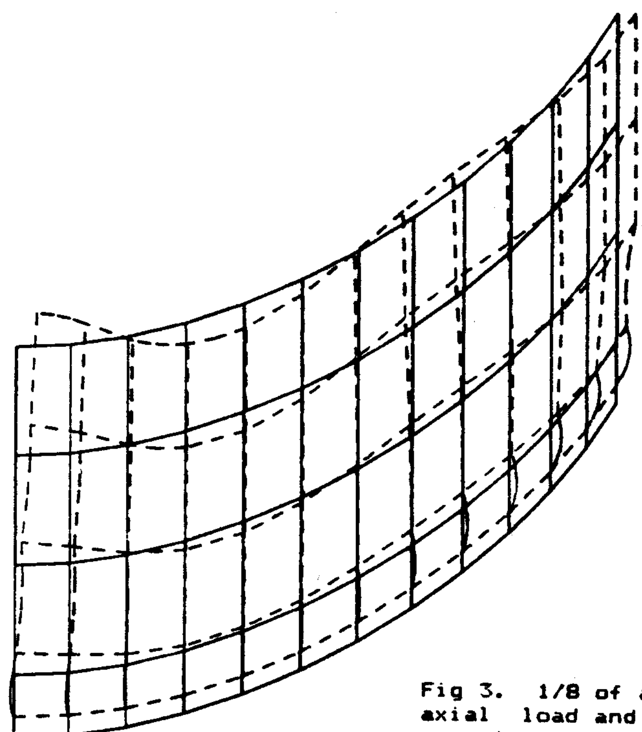

Fig 3. 1/8 of a cylinder with axial load and lateral sea-wave impact. Deformed shape at critical load (axial)

Sometimes, it is possible to avoid this type of difficulties making carefully studied choices of both the reference load and the units system used to describe the model. In this particular case if one choose millimeters and a first load increment of 0.21 KN, the scaling factor is approximately one, giving a well conditioned constraint equation for the first increment. However, as the deformation of the structure proceeds, variables involved in the constraint equation change, and its initial well conditioning may vanish.

When the scaling factor is used, the solution is obtained in an automatic way with a minimum of increments and few iterations per increment.

One has to note the difference between the number of iterations employed for the first increment and the corresponding number employed for the second one. This suggest us to replace the first load control step by a global control one, with excellent results.

A cylindrical shell under a sea wave impact was analyzed in order to assess the algorithm with a large system.

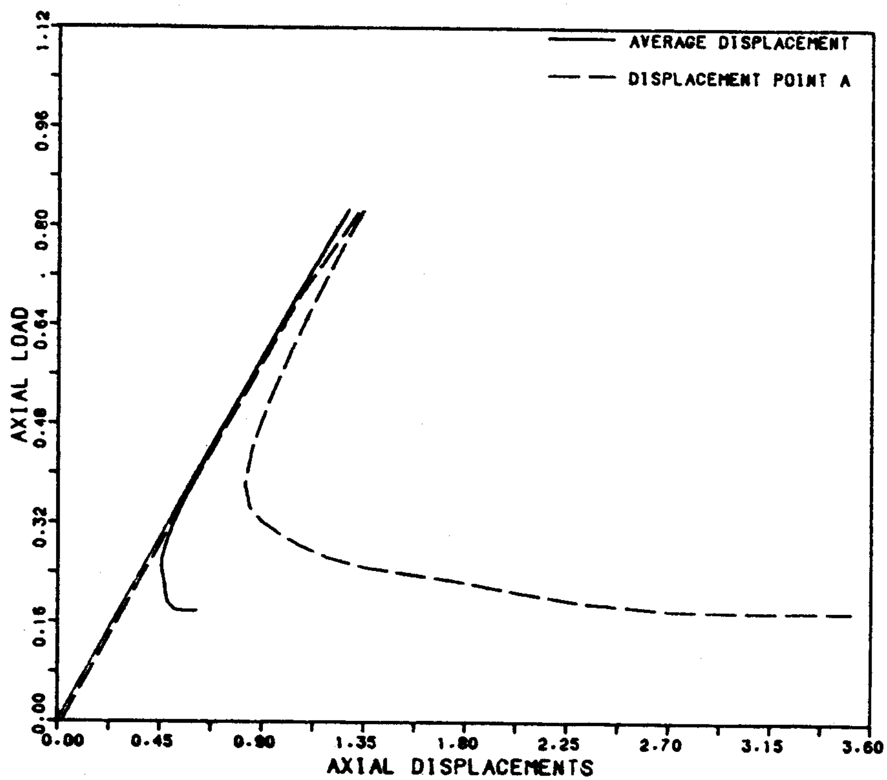

Fig 4. Load vs Deflection diagram for the end shortening of a cylinder with axial load and lateral perturbation

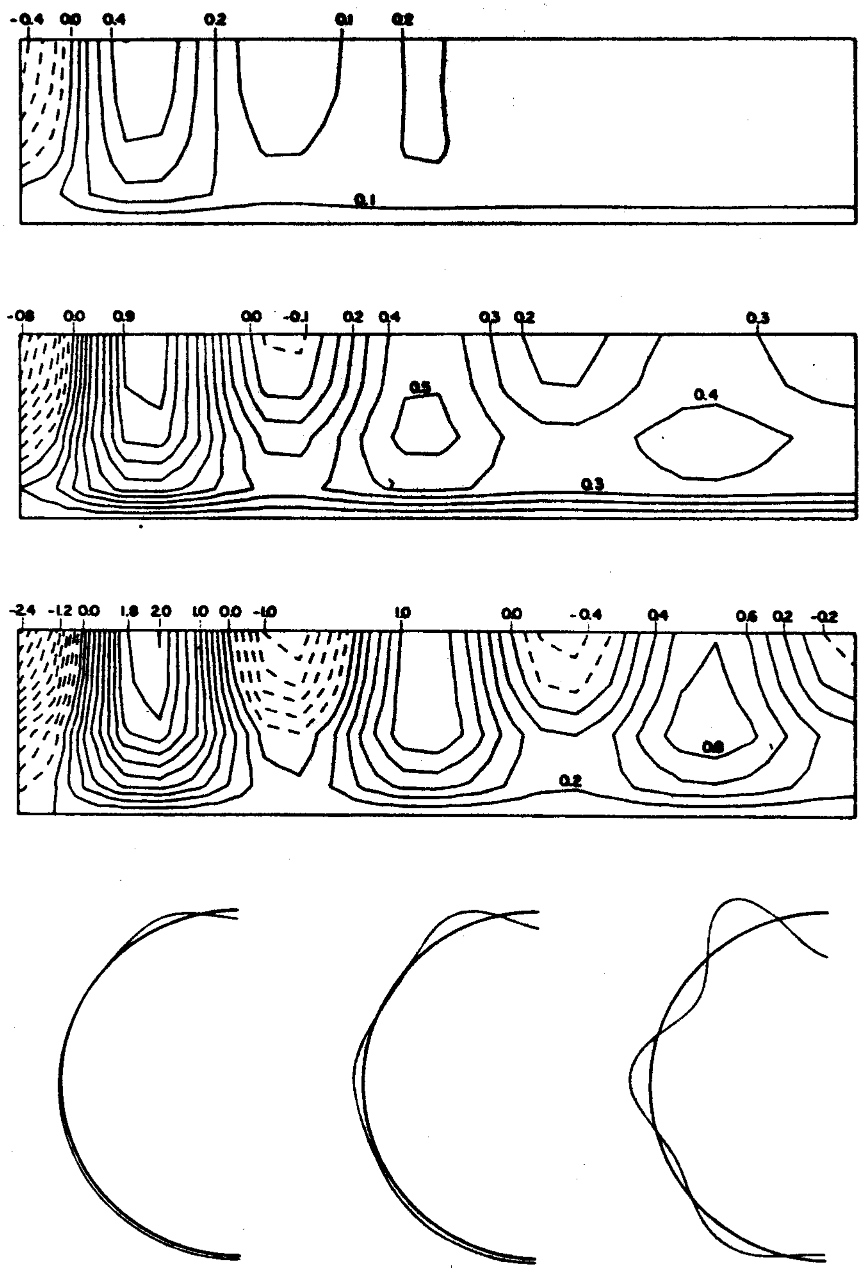

Fig 5. Contour plots for the radial displacements up to the critical load

The structure is simply supported at both ends. The nonlinear behavior under axial load is investigated. The axial load is normalized to the linear buckling value. A perturbation, representing the initial pressure distribution of a sea wave impact [13], is applied as energetically equivalent loads. They are applied laterally to the shell, before the axial loading be initiated.

One quarter of the shell is idealized by 96 quadratic shell elements. The first load-control increment introduces the lateral perturbation, and the second one defines the reference load p=0.35, but in this case variations of the load during iterations is allowed in order to improve the convergence.

Figure 3 shows the deformation of one quarter of the shell, near the limit point. The buckling mode has one half wave in the axial direction and six circumferencial waves. Contour plots in figure 5 show the development of the buckling modes up to the critical load (p=0.868) and figure 4 shows the load deflection diagram for the average end shortening of the shell. The entire solution was obtained with 30 steps.

8. REFERENCES

1 Risk E., Some Computational Aspects of the Stability Analysis of Nonlinear Structures, Computer Methods in Applied Mechanics and Engineering 47, 1984, 219/59

2 Ramm E., The Risk/Wempner Approach - An extension of the Displacement Control Method in Nonlinear Analises, Recent Advances in Non-Linear Computational Mechanics, Ed. by Hinton E., Owen D.R.J. and Taylor C., Pineridge Press 1982

3 Argyris J.H., Continua and Discontinua, Proceedings of the First Conference on Matrix Methods in Structural Mechanics., Wright-Patterson A.F.B., 1965, 11/189

4 Batoz J.L., Dath G., Incremental Displacement Algorithms for Nonlinear Problems, Int. Jou. Num. Meth. Engng. 14 , 1979, 1262/67

5 Risk E., The Application of the Newtons Method to the Problem of Elastic Stability, Jou. Appl. Mech., 39, 1972, 1060/66

6 Wempner G.A., Discrete Approximations Related to Nonlinear Theories of Solids, Int. Jour. Solid Struct., 7, 1971, 1581/99

7 Ramm E., Strategies for Tracing Nonlinear Responses Near Limit Points, Non-Linear Finite Element Analysis in Structural Mechanics, Ed. by Wunderlich W., Stein E. and Bathe K.J., Springer Verlag, 1981

8 Crisfield M.A., A Fast Incremental/Iterative Solution Procedure that handles "Snap-though", Comp. Struct., 13, 1981, 55/62

9 Nyssen C., Modélisation par Eléments Finis du Comportement Non Linéaire des Structures Aérospatiales, Université de Liège, 1978

10 SAMCEF. Système d'Analise des Milieux Continus par Elèments Finis, LTAS, Université de Liège.

11 Godinas A., Jetteur P., Laschet G., Developpement d'un Element Fini Tridimensionnel, Specialise pour l'etude des Coques dans le Domain Non Lineaire, Rapport SF-101, LTAS, 1981, Université de Liège

12 Croll J.G.A., Walker A.C., Elements of Structural Stability, Ed Macmillan.

13 Godoy L.A., Displacements Induced by Breaking Waves on Axially Loaded Cylindrical Shells. Applied Ocean Research, 1985, vol 7, n 3.