# DEVELOPMENT OF SOLUTION TECHNIQUES FOR NONLINEAR STRUCTURAL ANALYSIS

FINAL REPORT

Contract NAS8-29625

September 30, 1974

Reproduced by NATIONAL TECHNICAL INFORMATION SERVICE US Department of Commerce Springfield, VA. 22151

Prepared by

BOEING AEROSPACE COMPANY RESEARCH AND ENGINEERING DIVISION SEATTLE, WASHINGTON 98124

R. G. Vos – Technical Leader

J. S. Andrews – Program Leader

Prepared For-

### PRICES SUBJECT TO CHANGE

### NATIONAL AERONAUTICS AND SPACE ADMINISTRATION GEORGE C. MARSHALL SPACE FLIGHT CENTER MARSHALL SPACE FLIGHT CENTER, ALABAMA 35812

(NASA-CR-120525) DEVELOPMENT OF SOLUTION TECHNIQUES FOR NONLINEAR STRUCTURAL ANALYSIS Final Report (Boeing Aerospace Co., Seattle, Wash.)

N75-11371

**Unclas** 

CSCL 20K G3/39 02589



, D180-18325-1

### DEVELOPMENT OF SOLUTION TECHNIQUES FOR NONLINEAR STRUCTURAL ANALYSIS

FINAL REPORT

Contract NAS8-29625

September 30, 1974

### Prepared by

BOEING AEROSPACE COMPANY RESEARCH AND ENGINEERING DIVISION SEATTLE, WASHINGTON 98124

R. G. Vos - Technical Leader J. S. Andrews - Program Leader

Prepared for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION GEORGE C. MARSHALL SPACE FLIGHT CENTER MARSHALL SPACE FLIGHT CENTER, ALABAMA 35812



### ABSTRACT

Nonlinear structural solution methods in the current research literature are classified according to order of the solution scheme, and it is shown that the analytical tools for these methods are uniformly derivable by perturbation techniques. A new perturbation formulation is developed for treating an arbitrary nonlinear material, in terms of a finite-difference generated stress-strain expansion. Nonlinear geometric effects are included in an explicit manner by appropriate definition of an applicable strain tensor. A new finite-element pilot computer program PANES (Program for Analysis of Nonlinear Equilibrium and Stability) is presented for treatment of problems involving material and geometric nonlinearities, as well as certain forms on nonconservative loading.



### TABLE OF 'CONTENTS

SECTION		
	TABLE OF CONTENTS	iii
1.0	INTRODUCTION	1-1
1.1	General Philosophy and Evaluation of Methods	1-1
1.2	Previous Developments and Present Work	1-3
2.0	DETERMINATION OF THE EQUILIBRIUM PATH: A GENERAL- IZATION OF STATIC PERTURBATION TECHNIQUES	2-1
2.1	Description of Nonlinearities	2-1
2.2	Formulation of Equilibrium Equations	2-3
2.3	Solution of Equilibrium Equations	2-7
3.0	DETERMINATION OF BIFURCATION AND THE POSTBUCKLING PATH	3-1
3.1	Description of the Postbuckling Path	3-1
3.2	Formulation of Postbuckling Equilibrium Equations	3-2
3.3	Solution of Postbuckling Equilibrium Equations	3-10
4.0	THE PANES FINITE ELEMENT PROGRAM AND NONLINEAR SOLUTION ROUTINES	4-1
4.1	General Program Characteristics	4-1
4.2	PANES Nonlinear Analysis Routines	4-2
4.3	Summary of PANES Input Data	4-8
4.4	Summary of PANES Output	4-15
-5.0	ILLUSTRATIVE PROBLEMS	5-1
5.1	Snap-Through Truss	5-1
5.2	Simple Pressure Membrane	5-4
5.3	Toroidal Membrane	5-9
5.4	Asymmetric Buckling Model	5-13

THE BOEING COMPANY

SECTION

SECTIO	N		PAGE
6.0	CONCLUSIONS 2	AND RECOMMENDATIONS	6-1
7.0	REFERENCES AND BIBLIOGRAPHY		
	APPENDIX A:	FINITE DIFFERENCE EXPANSIONS	A-1
	APPENDIX B:	NONCONSERVATIVE LOADING EFFECTS	B-1

THE BOEING COMPANY

#### 1.0 INTRODUCTION

Purpose and Scope of the Study - The present research was undertaken to develop improved techniques for solution of structures with material and geometric nonlinearities, including the limit point and bifurcation behavior which occurs in buckling and collapse problems. Because the effectiveness of such solution techniques has been found to depend strongly on the method used for generating the nonlinear equations, e.g., creation of the system Jacobian matrix, improved equation generation techniques were also emphasized. Available nonlinear analysis methods were evaluated for their current capabilities and their projected long term potentials, and the methods judged to be most promising formed a starting point for development of the techniques presented in this report. Corresponding FORTRAN subroutines were developed and incorporated into the pilot computer program PANES (acronym of the Program for Analysis of Nonlinear Equilibrium and Stability) for checkout and evaluation. The equation generation and solution techniques are within the framework of the finite element structural discretization method.

1.1 General Philosophy and Evaluation of Methods

<u>Criteria</u> - Structural solution methods available in the current literature were initially evaluated for this study based on four general criteria:

- A high degree of <u>automation</u> which minimizes the burden on the user.
- 2. <u>Cost effectiveness</u> for large size problems.
- 3. The use of an effective <u>incremental</u> technique which allows the user to follow and plot the structural response path.
- Achievement of <u>accuracy</u> by a self-correcting characteristic, which assures that the true solution is approached at each point where results are desired.

### THE BOEING COMPANY

During the study it was decided that recent advancements in structural theory made it timely to broaden the applicability of the developed equation generation and solution techniques by including a fifth requirement:

5. An efficient treatment of large-strain problems, and of arbitrary nonlinear elastic or inelastic materials.

<u>Classification of Methods</u> - The current literature contains a very broad variety of nonlinear solution methods, and even the specialized requirements for nonlinear structural solutions have not resulted in consensus on a best method or methods. On the other hand, certain types of highly nonlinear problems are presently receiving considerable attention (for example, stability analyses and large-strain effects with arbitrary nonlinear materials), and such problems tend to eliminate certain methods from consideration while giving some direction to future research and development.

Most of the nonlinear structural solution methods can be broadly grouped into three classes:

- 1. Methods which use only the initial (constant) stiffness of the structure, and rely on iteration with the calculation of residual (unbalanced) forces to achieve the correct solution. The loading may be applied incrementally or in a single step.
- 2. Methods which form the Jacobian (tangent stiffness) matrix at a series of load increments, without iteration; also, various combined incremental and iterative algorithms which update the Jacobian at each step or periodically.
- 3. Higher-order methods (perturbation approaches or various numerical integration schemes) which employ higherorder derivative relations in addition to the firstorder Jacobian coefficients.

THE BOEING COMPANY

In many respects, the above ordering of classes is according to increasing sophistication and greater capability. For example, the class 3 methods are especially suited for analysis of complicated limit point and postbuckling problems. As might be expected, the historical development of nonlinear solution methods has shown some tendency to progress from original class 1 techniques to those of classes 2 and 3.

1.2 Previous Developments and Present Work

Historical Development - Early finite-element work in nonlinear structural analysis began with a paper by Turner, Dill, Martin and Melosh (1960). This work incorporated nonlinear geometric effects within the so-called "geometric" stiffness matrix, and various incremental and iterative solution procedures were recognized by the authors. In the initial attempts at nonlinear solutions which followed, there was a natural tendency to generalize existing linear capabilities. This usually led to iterative approaches and use of the initial constant stiffness matrix, with calculation of the nonlinearities as additional load terms. The geometric stiffness matrix, however, formed the basis for eigenvalue buckling analyses. A more consistent and theoretical basis for the geometric stiffness matrix was investigated by Gallagher and Padlog (1963), who used a strain-energy derivation with the same displacement functions for both the linear and nonlinear stiffness terms. Several formulations for nonlinear beam and plate analysis soon followed. Mallett and Marcal (1968) presented a unifying basis for formulating large displacement problems, by deriving the total strain energy as a function of nodal displacements and including previously neglected nonlinear terms. Meanwhile, developments were proceeding in the area of material nonlinearities, for example the plasticity work of Argyris (1965) and Marcal (1968), and nonlinear elastic analysis by Oden and Kubitza (1967).

### THE BUEING COMPANY

A consensus on solution methods did not appear, however, and different approaches were emphasized by different research groups. Zienkiewicz and coworkers popularized some of the class 1 solution methods, and Zienkiewicz et al. (1969) presented a particular method called the "initial stress" residual-load method. This work was followed more recently by techniques for improving the interative convergence of such methods, e.g. Nayak and Zienkiewicz (1972). However, the paper by Zienkiewicz and Nayak (1971) presents a quite general formulation for various class 2 methods with application to combined geometric and material large-strain nonlinearities. Considerable work in geometric and material nonlinearity has been done at Brown University, for example Marcal (1969), and McNamara and Marcal (1971). Researchers there have tended to favor class 2 methods without iteration, although the use of one or two iterations in each load step has been suggested as a way of increasing accuracy. Combined incremental and interative class 2 techniques (Newton Raphson iteration, for example) have been employed for analysis of highly nonlinear material and geometric nonlinearities, including some stability problems, by Oden and Key (1970), Sandidge (1973), and Key (1974). It would certainly appear that for such problems the class 1 methods at least are highly unsuitable. A number of nonlinear survey and development papers have been written at Texas A & M University, including Haisler et al. (1971), Stricklin et al. (1972), Stricklin et al. (1973) and Tillerson et al. (1973). These papers provide a detailed investigation of various class 1 and 2 methods, as well as certain class 3 methods which rely on numerical integration schemes. Although perturbation procedures are not tested by these authors, it is suggested in one of the early papers that perturbation techniques would be very time consuming for cases with large degrees of freedom, while a more recent paper notes that these techniques require further evaluation before they will be accepted by structural analysts. Many other researchers working in the areas of limit point and bifurcation stability problems, however, have concentrated on perturbation methods, reviving the

THE BOEING COMPANY

original theoretical developments in that area by Koiter (1945). Haftka et al. (1970) use an extension of Koiter's perturbation theory in a solution approach called the "modified structure method." Morin (1970) uses perturbation techniques in developing higher-order predictor and corrector algorithms for analysis of geometrically nonlinear shells. Gallagher and Mau (1972) and Mau and Gallagher (1972) establish procedures for limit point and postbuckling analysis based on perturbation expansions and the evaluation of determinants, which employ a combination of class 1, 2 and 3 solution techniques. A number of other perturbation developments of a more theoretical nature are included in the references and bibliography section of this report.

Much of the present diversity in nonlinear solution methods can be attributed to a desire to further investigate the potentials of all methods and to compare the results obtained from them. However, the comparisons and evaluations which are presented often disagree in their conclusions as to the effectiveness of a particular method. It must be surmised that the evaluation of nonlinear solution methods is necessarily influenced by the previous experiences and preferences of the researcher, by the degree of sophistication in his various solution method tools, and by the type of problems toward which his interests are directed.

<u>Direction of Present Work</u> - Because the present work was directed toward obtaining techniques whose applicability included the more highly nonlinear structures, a decision was made to eliminate from consideration the constant-stiffness methods of class 1. Although schemes have been proposed for extending these methods to more severe nonlinearities, it must be said that the arguments given are not convincing. In fact, when the structural system has advanced into a highly nonlinear state, the initial constant portion of the stiffness does not really possess any more significance than that provided by an arbitrary positive definite matrix; it can not be expected that a technique based on this

1-5

### THE BOEING COMPANY

matrix will be of any significant value in advancing the solution beyond the current state. It was also decided in the present work to reject those methods of a non self-correcting nature, i.e., methods which do not involve an iterative calculation of the unbalanced or "residual" forces, which gives an indication of accuracy and allows the solution to be improved. Although such methods are sometimes effective, they can lead to serious errors in the computed results, especially for path dependent problems. A third group of methods eliminated from consideration were those which use the solution data generated at several previous solution points. Such methods essentially extrapolate the previous data, either by some numerical integration formula or by a curve fitting approach. These methods require storage of previous data and are usually not self-starting. However the main objection to their use would seem to be that the same type of capability is provided by perturbation methods, which more accurately evaluate the path direction and are more generally applicable to a wide range of highly nonlinear problem types (e.q., those involving path discontinuities such as bifurcation points).

With these considerations, the methods which remain for development include methods of "incremental loading", Newton Raphson iteration and its modifications involving only periodic updating of the Jacobian, and higher-order methods including various orders of predictor and corrector algorithms. In order to make the current methods applicable to cases of large strain and arbitrary nonlinear materials, the equation generation process is accomplished in the present work by a finite difference expansion procedure. It is found that generation of the nonlinear equations by this means within a perturbation context provides a unifying basis for definition of the nonlinear solution terms, including as special cases the first-order Newton Raphson and incremental loading methods, as well as almost an unlimited variety of higher-order solution techniques. The perturbation

THE BOEING COMPANY

procedures have the advantage of a sound theoretical basis in classical developments, and lend themselves readily to both limit point and postbuckling problems as well as to simple nonlinear behavior without critical points.

THE BOEING COMPANY

## 2.0 DETERMINATION OF THE EQUILIBRIUM PATH: A GENERALIZATION OF STATIC PERTURBATION TECHNIQUES

In this Section the theory and techniques are developed for following the nonlinear equilibrium path of a structure under prescribed loading. It is assumed that the equilibrium path is continuous and unique, although limit point behavior is allowed (the non-uniqueness due to bifurcation of the equilibrium path is considered in Section 3. The development follows the "static perturbation method" which was recognized and established in concrete form by Sewell (1965). The present work generalizes previous structural solution techniques based on the method to allow effective treatment of arbitrary nonlinear materials. The resulting formulation is shown to provide a quite general and unifying basis for solution of nonlinear structures, including geometric and material nonlinearities as well as certain forms of nonconservative loading. A summary of the formulation is contained in the paper by Vos (1974).

### 2.1 Description of Nonlinearities

An important characteristic of the present method is a preliminary separation of the nonlinear material and geometric effects, which minimizes the required number of perturbation expansion terms, and also increases numerical accuracy.

<u>Material Effects</u> - The nonlinear material effects are described by expanding the stress about a known equilibrium configuration:

$$\sigma_{i} = \sigma_{i}^{*} + D0_{ij}^{*}\Delta\varepsilon_{j} + D1_{ijk}^{*}\Delta\varepsilon_{j}\Delta\varepsilon_{k} + \cdots \qquad (2-1)$$

which provides the stress,  $\sigma$ , in terms of the incremental strain,  $\Delta \epsilon$ . Here and throughout this work, an asterisk (\*) denotes quantities evaluated at a known equilibrium state, and  $\Delta$  denotes an incremental quantity. In (2-1)  $\sigma$ \* is the initial stress, while

THE BOEING COMPANY

DO<sup>\*</sup> and D1\* are 2nd and 3rd order incremental stress-strain tensors, respectively. This type of expansion can be developed numerically for a general nonlinear elastic or inelastic material, by an efficient finite difference or Taylor series evaluation.<sup>†</sup> Complete symmetry of the D tensors can be used to advantage if they are derivable from a strain-enery function, or in certain other cases such as that of associative plasticity. These considerations are discussed by Zienkiewicz and Nayak (1971) in a development which employs only the 2nd order (DO) tensor. In any case, the tensors can be made symmetric in the j, k and any higher order indices.

<u>Geometric Effects</u> - The nonlinear geometric effects are included through a definition of finite element displacement functions and an appropriate strain tensor, giving

$$\theta_{i} = G_{ij}q_{j} \qquad (2-2a)$$

$$\varepsilon_{i} = A \theta_{ij} \theta_{j} + \frac{1}{2} A l_{ijk} \theta_{j} \theta_{k}$$
 (2-2b)

Here q are the element generalized (nodal) displacements of an element, G is obtained by differentiating the assumed displacement functions,  $\theta$  are the displacement derivatives at any point, while A0 and Al are constant coefficients which define the strain tensor with Al<sub>ijk</sub> = Al<sub>ikj</sub>. The explicit form (2-2b) of the strain tensor will be seen to simplify later manipulations.

+ The best approach developed thus far is a forward difference scheme, which requires a minimum number of function evaluations and allows an arbitrary difference size for each independent variable. Explicit coefficients have been derived for such expansions of either linear, quadratic or cubic form, and in terms of an arbitrary number of variables. Details are provided in Appendix A. It may be noted that for certain problems involving incompressible materials, the hydrostatic stress must be included as an indepdent parameter in addition to the strains.

THE BOEING COMPANY

Advantages of Present Approach - The present approach defines all nonlinearities through the form of (2-1) and (2-2), rather than through a direct expansion of the nodal displacements such as that used in the investigation of Oden and Key (1970). The present approach appears to offer substantial advantages, because it allows implementation of perturbation theories into limit point and bifurcation analysis, without involving a huge number of terms and formidable algebraic operations. As a practical matter, it should also be noted that a numerical expansion based on displacements often causes severe problems with accuracy of the expansion coefficients, due to large differences in magnitude between individual displacement limits (e.g., between the membrane and bending freedoms of a plate or shell), and the selection of accurate finite difference sizes then becomes difficult. Accuracy is more easily obtained in an expansion of the type (2-1), because the strain limits tend to be of the same order of magnitude.

2.2 Formulation of Equilibrium Equations

<u>Virtual Work Statement</u> - The principle of virtual work, which is valid for arbitrary nonlinear materials and nonconservative systems, is employed to obtain equilibrium equations for the system of finite elements. The formulation is developed here for a conservative system, and nonconservative effects are treated in Appendix B. The equivalence of external and internal virtual work, relates the generalized nodal forces p and displacements q of a particular finite element, in the element equilibrium equation

$$\delta q_{i} p_{i} = \int_{V} \delta \varepsilon_{a} \sigma_{a} \, dV \qquad (2-3)$$

which holds along any equilibrium path in the neighborhood of the reference equilibrium (\*) configuration. Here  $\delta \epsilon$  and  $\delta q$ are kinematically consistent variations, and from equation (2-2)

$$\delta \theta_{i} = G_{ij} \delta q_{j} \qquad (2-4a)$$

THE BOEING COMPANY

$$\delta \varepsilon_{i} = (AO_{ij} + Al_{ijk}\theta_{k}) \delta \theta_{j} \equiv A_{ij}\delta \theta_{j} \equiv B_{ij}\delta q_{j}$$
(2-4b)

The integral in equation (2-3) is taken over the volume of the element, and it is to be noted that a proper definition of stress and strain is required to give the correct evaluation of internal work. One approach for accomplishing this is a formulation of Lagrangian strain and second Piola-Kirchoff stress integrated over the undeformed volume, e.g. see Oden and Key (1970).

<u>Basic Equilibrium Equation</u> - Substituting for  $\delta \varepsilon$ , and noting that (2-3) must be satisfied for arbitrary variations  $\delta q$ , provides the basic equilibrium equation for the element, as

$$p_{i} = \int_{V} B_{ai} \sigma_{a} dV = \int_{V} (AO_{am} + Al_{amn} \theta_{n}) \sigma_{a} dV \qquad (2-5a)$$

In order to merge the element equations into the system equations, the usual type of finite element transformation is applied. The system forces and displacements will be denoted by the capitals P and Q, respectively, and the system basic equilibrium equation corresponding to (2-5a) is written as

$$P_{i} = \int_{V} B_{ai}\sigma_{a} dV = \int_{V}^{\sigma_{ai}} (A0_{am} + Al_{amn}\theta_{n})\sigma_{a} dV \qquad (2-5b)$$

where now it must be understood that the integral is summed over all elements while applying the proper element-system nodal transformations. With this understanding, the element and system quantities will here be used interchangeably.

Derivative Relations - Equations (2-5) may now be differentiated as many times as desired with respect to some suitable path parameter. Toward that end, it is useful to record here the following typical derivative relations, where an overdot (\*) denotes differentiation with respect to the path parameter.

### THE BUEING COMPANY

N 1 1 4

$$\dot{\sigma}_{a} = DO_{ab}^{*}\dot{\varepsilon}_{b} + 2DI_{ab}^{*}\dot{\varepsilon}_{b}\Delta\varepsilon_{c} + \dots$$

$$\ddot{\sigma}_{a} = DO_{ab}^{*}\ddot{\varepsilon}_{b} + 2DI_{abc}^{*}\ddot{\varepsilon}_{b}\Delta\varepsilon_{c} + 2DI_{abc}^{*}\dot{\varepsilon}_{b}\dot{\varepsilon}_{c} + \dots$$

$$\dot{\varepsilon}_{a} = B_{ai}\dot{q}_{i}$$

$$\ddot{\varepsilon}_{a} = B_{ai}\dot{q}_{i} + \dot{B}_{ai}\dot{q}_{i} = B_{ai}\ddot{q}_{i} + AI_{amn}\dot{\theta}_{m}\dot{\theta}_{n}$$

$$(2-6a)$$

and at the reference configuration ( $\Delta \epsilon = 0$ ), we have

$$\ddot{\sigma}_{a}^{*} = DO_{ab}^{*} \dot{\tilde{e}}_{b}^{*}$$

$$(2-6b)$$

$$\ddot{\sigma}_{a}^{*} = DO_{ab}^{*} \dot{\tilde{e}}_{b}^{*} + 2Dl_{abc}^{*} \dot{\tilde{e}}_{b}^{*} \dot{\tilde{e}}_{c}^{*}$$

First Order Equilibrium Equation - Differentiating equation (2-5b) once, and evaluating at the reference equilibrium (\*) configuration, gives

Substituting from relations (2-6b) gives

$$\dot{P}_{i}^{*} = \int (G_{mi}\sigma_{a}^{*}Al_{amn}G_{nj}\dot{q}_{j}^{*} + B_{ai}^{*}D0_{ab}^{*}B_{bj}^{*}\dot{q}_{j}^{*}) dV \qquad (2-7b)$$

This is the first order equilibrium equation, which may be written in the form

$$\dot{P}_{i}^{*} = K0_{ij}^{*}\dot{Q}_{j}^{*}$$
 (2-7c)

where

 $KO_{ij}^{*} = \int (G_{mi}\sigma_{a}^{*}Al_{amn}G_{nj} + B_{ai}^{*}DO_{ab}^{*}B_{bj}^{*}) dV$ and V

$$B_{ai}^{\star} = G_{mi}A_{am}^{\star} = G_{mi}(A0_{am} + A1_{amn}\theta_{n}^{\star})$$

### THE BOEING COMPANY

The "tangent stiffness" relation (2-7c) is equivalent to the incremental matrix formulation of Zienkiewicz and Nayak (1971), although the tensor form given here shows perhaps more clearly the symmetry and differentiability properties of the tangent (Jacobian) matrix KO\*. The first contribution to KO\* is due to the initial stresses during changing geometry, and is always symmetric in form. The second contribution is due to the incremental stress-strain relation, and its symmetry depends on symmetry of the matrix DO\*.

Second Order Equilibrium Equation - A second differentiation of (2-5b) and evaluation at the reference configuration, gives

$$\dot{P}_{i}^{*} = \int_{V} (\ddot{B}_{ai}^{*}\sigma_{a}^{*} + 2\dot{B}_{ai}^{*}\dot{\sigma}_{a}^{*} + B_{ai}^{*}\dot{\sigma}_{a}^{*}) dV$$
 (2-8a)

Substituting from relations (2-6b) gives

$$\dot{P}_{i}^{*} = \int \{G_{mi}\sigma_{a}^{*}Al_{amn}G_{nj}\dot{q}_{j}^{*} + 2G_{mi}Al_{amn}\dot{\theta}_{n}^{*} DO_{ab}^{*}\dot{\epsilon}_{b}^{*} \}$$
$$+ B_{ai}^{*}(DO_{ab}^{*}B_{bj}^{*}\dot{q}_{j}^{*} + DO_{ab}^{*}Al_{brs}\dot{\theta}_{r}^{*}\dot{\theta}_{s}^{*} + 2Dl_{abc}^{*}\dot{\epsilon}_{b}^{*}\dot{\epsilon}_{c}^{*})\} dV (2-8b)$$

This is the second order equilibrium equation, which may be written in the form

$$\ddot{P}_{i}^{*} = KO_{ij}^{*}\dot{Q}_{j}^{*} + Pl_{i}^{*}$$
 (2-8c)

where Pl is a psuedo force term given by

$$Pl_{i}^{*} = \int_{V} G_{mi} \{ DO_{ab}^{*} (A_{am}^{*}Al_{brs}^{\dagger} \dot{\theta}_{r}^{*} \dot{\theta}_{s}^{*} + 2Al_{amn} \dot{\epsilon}_{b}^{*} \dot{\theta}_{n}^{*} )$$
$$+ 2Dl_{abc}^{*} A_{am}^{*} \dot{\epsilon}_{b}^{*} \dot{\epsilon}_{c}^{*} \} dV$$

### 2.3 Solution of Equilibrium Equations

Incremental Load and Path Parameters - An increment of conservative loading is defined by

$$\Delta P_{i} = \Lambda P_{i}^{\circ}, P_{i} = \Lambda P_{i}^{\circ}, \text{ etc.}$$
(2-9)

using a variable load parameter  $\Lambda$  and constant nodal load distribution P°.

Taylor series expansions are then used to approximate both the incremental load parameter  $\Lambda$  and displacements  $\Delta Q$ :

$$\Lambda = \Lambda * \mathbf{S} + \frac{1}{2} \Lambda * \mathbf{S}^2 + \dots \qquad (2-10a)$$

$$\Delta Q_{i} = \dot{Q}_{i}^{*} S + \frac{1}{2} \ddot{Q}_{i}^{*} S^{2} + \dots \qquad (2-10b)$$

In order to handle limit point situations within the present formulation, the path parameter S is here taken as defined by

$$s^{2} = i KO_{ij}^{*} \Delta Q_{i} \Delta Q_{j} \ge 0, i = \pm 1$$
 (2-11)

with the requirement that KO\*, evaluated at the beginning of each load increment, be nonsingular (either positive or negative definite). Without any loss of generality, additional requirements imposed at every path point, S, are that

```
\dot{s} = \dot{s}^2 = 1
\ddot{s} = \ddot{s} = \dots = 0
```

Successive differentiation of (2-11) provides the relations

$$2\dot{s}\dot{s} = i KO_{ij}^{*}(\dot{Q}_{i}\Delta Q_{j} + \Delta Q_{i}\dot{Q}_{j})$$

$$2\dot{s}^{2} = i KO_{ij}^{*}(\ddot{Q}_{i}\Delta Q_{j} + 2\dot{Q}_{i}\dot{Q}_{j} + \Delta Q_{i}\ddot{Q}_{j})$$

$$2-7$$

$$(2-12a)$$

DO 6000 2140 ORIG 4/71

BOEING COMPANY

$$0 = \mathbf{i} \operatorname{KO}_{ij}^{*} ( \ddot{\mathcal{Q}}_{i} \Delta \mathcal{Q}_{i} + 3 \ddot{\mathcal{Q}}_{i} \dot{\mathcal{Q}}_{j} + 3 \dot{\mathcal{Q}}_{i} \ddot{\mathcal{Q}}_{j} + \Delta \mathcal{Q}_{i} \ddot{\mathcal{Q}}_{j} )$$

and evaluation at the reference state (S =  $\Delta Q_i = 0$ ), yields

$$\dot{s}^{2} = 1 = i \ KO_{ij}^{*}\dot{Q}_{i}^{*}\dot{Q}_{j}^{*}$$
  
 $0 = i \ KO_{ij}^{*}(\ddot{Q}_{i}^{*}\dot{Q}_{j}^{*} + \dot{Q}_{i}^{*}\ddot{Q}_{j}^{*})$  (2-12b)

(2 - 12a)

It may be noted that relations (2-12) hold for the general case of an unsymmetric KO\* matrix.

Determination of Rate Quantities - In order to implement various solution techniques, the equilibrium equations (2-7c) and (2-8c) must be used to determine the load and displacement rates. Multiplying (2-7c) by  $\dot{Q}^*$ , and making use of (2-9) and (2-12b), gives

$$i = \Lambda * \dot{Q} * P_{i}^{\circ}$$
 (2-13a)

Solving (2-7c) for  $\dot{Q}^*$  gives

$$KO_{ij}^{*-1}\Lambda *P_{j}^{\circ} = \dot{Q}_{i}^{*} \equiv \Lambda *Q_{i}^{\circ}$$
(2-13b)

and substituting  $\hbar^*Q^\circ$  for  $\dot{Q}^*$  in (2-13a) gives

$$\Lambda^{\star 2} = i/(Q_i^{\circ}P_i^{\circ}) \qquad (2-13c)$$

where now i is chosen to make  $\Lambda^{\star 2}$  positive. Multiplying (2-8c) by  $\dot{Q}^{\star}$ , and again making use of (2-9) and (2-12b), gives

$$\dot{Q}_{i}^{*}\Lambda^{*}P_{i}^{\circ} = KO_{ij}^{*}\dot{Q}_{i}^{*}\dot{Q}_{j}^{*} + \dot{Q}_{i}^{*}Pl_{i}^{*} = -KO_{ij}^{*}\ddot{Q}_{i}^{*}\dot{Q}_{j}^{*} + \dot{Q}_{i}^{*}Pl_{i}^{*}$$
 (2-14a)

THE BOEING COMPANY

Solving (2-8c) for  $\ddot{Q}^*$  gives

$$\tilde{Q}_{i}^{*} = KO_{ij}^{*-1} (\tilde{\Lambda}^{*}P_{j}^{\circ} - Pl_{j}^{*}) \equiv \tilde{\Lambda}^{*}Q_{i}^{\circ} - Ql_{i}^{*}$$
 (2-14b)

Substituting (2-14b) into (2-14a) with the use of (2-13c) then provides the result

$$\dot{\Lambda}^{*} = i\dot{\Lambda}^{*2} (Pl_{i}^{*}Q_{i}^{\circ} + P_{i}^{\circ}Ql_{i}^{*})/2$$
 (2-14c)

after which  $Q^*$  is obtained directly from (2-14b).

It is to be noted that a solution for the rates  $\Lambda^*$ ,  $\Lambda^*$ ,  $Q^*$  and  $Q^*$ (and higher order rates if desired by similar calculations) requires only a single formation and decomposition of the matrix KO\*.

Solution Procedures - Once the load and displacement rates have been determined to a desired order, many different solution procedures can be applied in tracing the nonlinear equilibrium path of the structure. The first order rates allow solution by methods of incremental loading (with or without evaluation of residual forces and corrective iterations), and Newton Raphson iteration where the Jacobian is re-evaluated at each iteration. Various combinations of incrementation and interation, with periodic updating of the Jacobian are of course possible. The second order rates allow the use of a second order predictor. The additional cost of the 2nd order predictor is associated with the Pl psuedo-force term, whose evaluation is performed at the elemental level with a cost roughly proportional to that of a single "residual force" evaluation. The cost of evaluating Pl by the form of (2-8c) is only linearly proportional to the number of integration points within an element, so that this technique is effective even for elements having complex geometry and large degrees of freedom.

THE BOEING COMPANY

Such a predictor has been found to be very useful during the present study, and although the PANES computer program allows use of various predictor-corrector options, the second order predictor almost always appears to be more efficient for cases of substantial nonlinearity. Since the second order rates are valid at any reference equilibrium configuration, they may be applied in a corrector technique, at a state where the system is in "equilibrium" under the applied loads plus a set of unbalanced residual loads. Thus convergence could be considerably accelerated if the second order relations were computed and used at each iteration, although the cost per iteration would also increase considerably. Higher order predictorcorrector relations are obviously possible as well, and the best type of solution capability would probably be a program in which more or less arbitrary options are allowed for the order of predictor and corrector, the frequency with which the Jacobian is updated, and the number of iterations to be performed per update. Although these considerations will not be discussed in any more detail here, the PANES program is at least a step in that direction, and makes available various options using the first and second order rate relations.

Limit Point and Step Size Considerations - A major advantage of a 2nd order predictor is that, with little increase in computational effort, it provides greatly increased prediction accuracy and allows larger load steps to be taken. In addition, it enables the traversing of limit points and provides various techniques for automatic selection of the load step size.

In the vicinity of a limit point, the load rate relation

$$\dot{\Lambda} = \dot{\Lambda}^* + \ddot{\Lambda}^* S \qquad (2-15a)$$

2 - 10

THE BOEIND

is used. Also from (2-10a) the path value for given  $\Lambda$  is

$$S = \{ -\Lambda^{*} + (\Lambda^{*2} + 2\Lambda\Lambda^{*})^{1/2} \} / \Lambda^{*} \ge 0 \qquad (2-15b)$$

At a limit point  $\dot{\Lambda} = 0$ , so that from (2-15a) the critical path value is

$$S^{C} = -\Lambda * / \Lambda *$$
 (2-15c)

Using these relations the limit point can be traversed when  $\Lambda$  is within some specified fraction of its critical value.

With regard to automatic selection of a general load step size, the following predictor relationships are noted.

$$\Delta \Lambda = \dot{\Lambda} S + \frac{1}{2} \ddot{\Lambda} S^2 \qquad (2-16a)$$

$$\Delta Q_{i} = \dot{Q}_{i} S + \frac{1}{2} \ddot{Q}_{i} S^{2}$$
 (2-16b)

Here the quadratic terms give an indication of the accuracy of the linear predictor, but because of the truncation of higher order terms there is no indication of accuracy for the quadratic predictor. The rationale used in the PANES program implementation is therefore to select a load step size which limits the quadratic contributions to some specified factor times the linear contributions. Specifically, change in slope of the load parameter during a load step is approximated by

$$\Delta \dot{\Lambda} = \dot{\Lambda} S \qquad (2-17a)$$

and the ratio of slope change to average slope is

$$\Delta \dot{\Lambda} / \dot{\Lambda}_{average} = \ddot{\Lambda} S / (\dot{\Lambda} + \frac{1}{2} \ddot{\Lambda} S) \qquad (2-17b)$$

THE BOEING COMPANY

This slope ratio is specified as a given allowable magnitude, in order to prevent over-prediction in (2-16a) of the behavior beyond accurate values. A similar step size restriction is employed based on (2-16b) for displacement rates.

THE BOEING COMPANY

#### 3.0 DETERMINATION OF BIFURCATION AND THE POSTBUCKLING PATH

This section considers the identification of bifurcation points in the load-displacement path of a structure, and the prediction of the postbuckling path beyond these points. The formulation follows the appraoch of Section 2 for representing geometric nonlinearities and an arbitrary nonlinear material. The effects of nonconservative load on bifurcation and postbuckling are treated in Appendix B.

Description of the Postbuckling Path 3.1

We consider behavior of the type shown in Figure 3-1, which is a plot of the incremental load parameter  $\Lambda$  for a structure versus its incremental displacements AQ, shown here conceptually for a single degree of freedom system. The point O represents a reference equilibrium configuration (Q = Q\*,  $\Lambda = \Delta Q = 0$ ). Travel along the "fundamental" and "postbuckling" paths is measured by suitable path parameters S and R, respectively. Thus S has a value of zero at point O, while R takes on a zero value at the critical bifurcation point C.



Figure 3-1: Fundamental and Postbuckling Paths

We follow the terminology of Mau and Gallagher (1972) and use a "sliding coordinate" system to describe the various fundamental and postbuckling quantities. For a given value of  $\Lambda$ , a point on the fundamental path has associated quantities whose values are denoted by ()<sup>f</sup>, while <u>additional</u> values at the corresponding point on the postbuckling path are denoted by ()<sup>p</sup>. Thus <u>total</u> values on the postbuckling path are denoted by ()<sup>f</sup> + ()<sup>p</sup>, and we write for the postbuckling path

$$Q = Q^{f} + Q^{p}$$

$$\Delta Q = \Delta Q^{f} + Q^{p}$$

$$\Delta \varepsilon = \Delta \varepsilon^{f} + \varepsilon^{p}$$

$$\sigma = \sigma^{f} + \sigma^{p}$$
etc.
$$(3-1)$$

where the  $\Lambda$  quantities are increments from the fundamental reference configuration.

We will refer to the ()<sup>f</sup> and ()<sup>p</sup> values as the "fundamental" and "postbuckling" values, respectively, and to their sums as the "total" values.

### 3.2 Formulation of Postbuckling Equilibrium Equations

Basic Equilibrium Equation - Because the postbuckling path is an equilibrium path, and equation (2-5b) is valid for a point on any equilibrium path, we may write the postbuckling equilibrium equation as

$$P_{i} = P_{i}^{f} + P_{i}^{p} = \int_{V} B_{ai}\sigma_{a} dV = \int_{V} (B_{ai}^{f} + B_{ai}^{p}) (\sigma_{a}^{f} + \sigma_{a}^{p}) dV \quad (3-2)$$

### THE BOEING COMPANY

Recognizing that  $P_i = P_i^f$  for a given value of A with conservative loading, and subtracting out terms in equation (3-2) which are zero because they collectively satisfy the fundamental equilibrium equation, provides the desired form of the postbuckling equilibrium equation as

$$P_{i}^{p} = 0 = \int \left\{ B_{ai}^{f} \sigma_{a}^{p} + B_{ai}^{p} (\sigma_{a}^{f} + \sigma_{a}^{p}) \right\} dV$$
(3-3)

<u>Derivative Relations</u> - We now record the following typical derivative relations, where a prime ()' denotes differentiation with respect to the postbuckling path parameter R.

$$\begin{split} \sigma_{a}^{p} &= \sigma_{a} - \sigma_{a}^{f} = D0_{ab}^{*} \varepsilon_{b}^{p} + D1_{abc}^{*} (2\varepsilon_{b}^{p} \Delta \varepsilon_{c}^{f} + \varepsilon_{b}^{p} \varepsilon_{c}^{p}) \\ \sigma_{a}^{'p} &= D0_{ab}^{*} \varepsilon_{b}^{'p} + D1_{abc}^{*} (2\varepsilon_{b}^{'p} \Delta \varepsilon_{c}^{f} + 2\varepsilon_{b}^{p} \varepsilon_{c}^{'f} + 2\varepsilon_{b}^{p} \varepsilon_{c}^{'p}) \\ \sigma_{a}^{'p} &= D0_{ab}^{*} \varepsilon_{b}^{'p} + D1_{abc}^{*} (2\varepsilon_{b}^{'p} \Delta \varepsilon_{c}^{f} + 4\varepsilon_{b}^{'p} \varepsilon_{c}^{'f} + 2\varepsilon_{b}^{p} \varepsilon_{c}^{'f} \\ &+ 2\varepsilon_{b}^{'p} \varepsilon_{c}^{'p} + 2\varepsilon_{b}^{p} \varepsilon_{c}^{'p}) \\ \sigma_{a}^{'p} &= D0_{ab}^{*} \varepsilon_{b}^{'pp} + D1_{abc}^{*} (2\varepsilon_{b}^{'p} \Delta \varepsilon_{c}^{f} + 6\varepsilon_{b}^{'p} \varepsilon_{c}^{'f} + 6\varepsilon_{b}^{'p} \varepsilon_{c}^{'f} \\ &+ 2\varepsilon_{b}^{p} \varepsilon_{c}^{'p} + 2\varepsilon_{b}^{p} \varepsilon_{c}^{'p}) \\ \sigma_{a}^{'p} &= D0_{ab}^{*} \varepsilon_{b}^{'pp} + D1_{abc}^{*} (2\varepsilon_{b}^{'pp} \Delta \varepsilon_{c}^{f} + 6\varepsilon_{b}^{'p} \varepsilon_{c}^{'f} + 6\varepsilon_{b}^{'p} \varepsilon_{c}^{'f} \\ &+ 2\varepsilon_{b}^{p} \varepsilon_{c}^{'p} + 2\varepsilon_{b}^{p} \varepsilon_{c}^{'pp} + 2\varepsilon_{b}^{p} \varepsilon_{c}^{'pp}) \\ \varepsilon_{a}^{'p} &= \varepsilon_{a}^{'} - \varepsilon_{a}^{'f} = B_{ai} q_{i}^{'} - B_{ai}^{f} q_{i}^{'f} \\ \varepsilon_{a}^{'p} &= \delta_{ai}^{'q} q_{i}^{'} + B_{ai} q_{i}^{'p} - B_{ai}^{'q} q_{i}^{'f} \\ \varepsilon_{a}^{'pp} &= B_{ai}^{'q} q_{i}^{'} + 2B_{ai}^{'q} q_{i}^{'p} - B_{ai}^{'p} q_{i}^{'p} - 2B_{ai}^{'p} q_{i}^{'p} \\ \varepsilon_{a}^{'pp} &= B_{ai}^{'q} q_{i}^{'pp} + 2B_{ai}^{'pp} q_{i}^{'pp} - B_{ai}^{'p} q_{i}^{'pp} - 2B_{ai}^{'p} q_{i}^{'pp} \\ \varepsilon_{a}^{'pp} &= B_{ai}^{'p} q_{i}^{'pp} + 2B_{ai}^{'pp} q_{i}^{'pp} - B_{ai}^{'pp} q_{i}^{'pp} - 2B_{ai}^{'pp} q_{i}^{'pp} \\ \varepsilon_{a}^{'pp} &= B_{ai}^{'pp} q_{i}^{'pp} + 2B_{ai}^{'pp} q_{i}^{'pp} - B_{ai}^{'pp} q_{i}^{'pp} \\ \varepsilon_{a}^{'pp} &= B_{ai}^{'pp} q_{i}^{'pp} + 2B_{ai}^{'pp} q_{i}^{'pp} - B_{ai}^{'pp} q_{i}^{'pp} \\ \varepsilon_{a}^{'pp} &= B_{ai}^{'pp} q_{i}^{'pp} + 2B_{ai}^{'pp} q_{i}^{'pp} - B_{ai}^{'pp} q_{i}^{'pp} \\ \varepsilon_{a}^{'pp} &= B_{ai}^{'pp} q_{i}^{'pp} + 2B_{ai}^{'pp} q_{i}^{'pp} \\ \varepsilon_{a}^{'pp} &= B_{ai}^{'pp} q_{i}^{'pp} + 2B_{ai}^{'pp} q_{i}^{'pp} \\ \varepsilon_{a}^{'pp} &= B_{ai}^{'pp} q_{i}^{'pp} \\ \varepsilon_{a}^{'pp} &$$

and at the critical point ( $\sigma_a = \sigma_a^f$ ,  $B_{ai} = B_{ai}^f$ ,  $\sigma_a^p = B_{ai}^p = 0$ ),

THE BOEING COMPANY

we have

$$\begin{aligned} \sigma_{a}^{i p} &= D0_{ab}^{*} \varepsilon_{b}^{i p} + 2D1_{abc}^{*} \varepsilon_{b}^{i p} \Delta \varepsilon_{c}^{f} = D0_{ab}^{*} \varepsilon_{b}^{i p} \\ \sigma_{a}^{i p} &= D0_{ab}^{*} \varepsilon_{b}^{i p} + D_{abc}^{*} (2\varepsilon_{b}^{i p} \Delta \varepsilon_{c}^{f} + 4\varepsilon_{b}^{i p} \varepsilon_{c}^{i f} + 2\varepsilon_{b}^{i p} \varepsilon_{c}^{i p}) \\ &= D0_{ab}^{*} \varepsilon_{b}^{i p} + D1_{abc}^{*} (4\varepsilon_{b}^{i p} \varepsilon_{c}^{f} + 2\varepsilon_{b}^{i p} \varepsilon_{c}^{i p}) \\ \sigma_{a}^{i p} &= D0_{ab}^{*} \varepsilon_{b}^{i p} + D1_{abc}^{*} (2\varepsilon_{b}^{i p} \Delta \varepsilon_{c}^{f} + 6\varepsilon_{b}^{i p} \varepsilon_{c}^{i f}) \\ &+ 6\varepsilon_{b}^{i p} \varepsilon_{c}^{i f} + 6\varepsilon_{b}^{i p} \varepsilon_{c}^{i p}) \\ &= D0_{ab}^{*} \varepsilon_{b}^{i p} + D1_{abc}^{*} (6\varepsilon_{b}^{i p} \varepsilon_{c}^{f} + 6\varepsilon_{b}^{i p} \varepsilon_{c}^{i f}) \\ &+ 6\varepsilon_{b}^{i p} \varepsilon_{c}^{i p} + D1_{abc}^{*} (6\varepsilon_{b}^{i p} \varepsilon_{c}^{f} + 6\varepsilon_{b}^{i p} \varepsilon_{c}^{i f}) \\ &+ 6\varepsilon_{b}^{i p} \varepsilon_{c}^{i p}) \\ &\varepsilon_{a}^{i p} = B_{ai}^{f} q_{i}^{i p} = B_{ai}^{*} q_{i}^{i p} + A1_{amn}^{0 p} \theta_{m}^{p} \Delta \varepsilon_{n}^{f} \\ &\varepsilon_{a}^{i p} = B_{ai}^{f} q_{i}^{i p} + B_{ai}^{i} q_{i}^{i } - B_{ai}^{f} q_{i}^{i f} - 2B_{ai}^{f} q_{i}^{i f} - 2B_{ai}^{f} q_{i}^{i f} \\ &+ A1_{amn}^{*} (2\theta_{m}^{p} \theta_{n}^{i p} + A1_{amn}^{*} (3\theta_{m}^{*} \theta_{n}^{i f} + 3\theta_{m}^{*} \theta_{n}^{*})^{f} \end{aligned}$$

First Order Equilibrium Equation - Differentiating the basic postbuckling equilibrium equation (3-3), and evaluating at the critical bifurcation point, gives

$$P_{i}^{P} = 0 = \int_{V} (B_{ai}^{f} \sigma_{a}^{P} + B_{ai}^{P} \sigma_{a}) dV \qquad (3-5a)$$

Substituting for  $\sigma'^p$  and  $B'^p$  gives

$$0 = \int_{V} \{B_{ai}^{f}(D0_{ab}^{*}\varepsilon_{b}^{*P} + 2D1_{abc}^{*}\varepsilon_{b}^{*P}\Delta\varepsilon_{c}^{f}) + G_{mi}^{Al}amn_{n}^{\theta}e_{n}^{*P}a\} dV \quad (3-5b)$$

Substituting for  $B^f_{ai}$ , and using the relations which express  $\theta'^{\,p}$  and  $\epsilon'^{\,p}$  in terms of  $q'^{\,p}$ , gives

$$0 = \int_{V} [(B_{ai}^{*} + G_{mi}A_{amn}\Delta\theta_{n}^{f}) \{DO_{ab}^{*}(B_{bj}^{*}q_{j}^{*P} + A_{bns}\theta_{n}^{*P}\Delta\theta_{s}^{f}) + 2DI_{abc}^{*}\epsilon_{b}^{*P}\Delta\epsilon_{c}^{f}\} + G_{mi}A_{amn}^{*}\{G_{nj}q_{j}^{*P}\sigma_{a}^{*} + \theta_{n}^{*P}(DO_{ab}^{*}\Delta\epsilon_{b}^{f} + DI_{abc}^{*}\Delta\epsilon_{b}^{f}\Delta\epsilon_{c}^{f})\}] dV$$
(3-5c)

This is the first order postbuckling equilibrium equation, which may be written in the form

$$0 = K0_{ij}^{*} Q_{j}^{*} + Pl_{i}^{1}$$
(3-5d)

where again

$$K0_{ij}^{*} = \int_{V} (G_{mi}\sigma_{a}^{*}A1_{amn}G_{nj} + B_{ai}^{*}D0_{ab}^{*}B_{bj}^{*}) dV$$

and

$$Pl_{i}^{1} = \int_{V} G_{mi} \{ D0_{ab}^{*} (A_{am}^{*}Al_{bns}\theta_{n}^{*}P_{\Delta}\theta_{s}^{f} + Al_{amn}\varepsilon_{b}^{*}P_{\Delta}\theta_{n}^{f} + Al_{amn}\varepsilon_{b}^{*}P_{\Delta}\theta_{n}^{f} + Al_{amn}\Delta\varepsilon_{b}^{f}\theta_{n}^{*}P \} + Dl_{abc}^{*} (2A_{am}^{*}\varepsilon_{b}^{*}P_{\Delta}\varepsilon_{c}^{f} + 2Al_{amn}\varepsilon_{b}^{*}P_{\Delta}\varepsilon_{c}^{f}\Delta\theta_{n}^{f} + Al_{amn}\Delta\varepsilon_{b}^{f}\Delta\varepsilon_{c}^{f}\Theta_{n}^{*}P \} \} dV$$

Equation (3-5d) is an eigenequation form, in terms of the unknown critical values  $\Delta\theta$  and  $\Delta\epsilon$ , which is suitable for solution by power iteration.

THE BOEING COMPANY

Alternatively, the eigenequation may be written in the form

$$0 = K O_{ij} Q_{j}^{P} = (K O_{ij}^{*} + \Delta K O_{ij}) Q_{j}^{P}$$
(3-5e)

where KO is the Jacobian at the critical point, and is given by

$$KO_{ij} = \int_{V} (G_{mi}\sigma_{a}Al_{amn}G_{nj} + B_{ai}DO_{ab}B_{bj}) dV$$

with the  $\sigma$ , D0 and B quantities evaluated at the critical point. (3-5e) may be solved directly for the eigenvector Q'<sup>P</sup>, provided that the critical values of  $\sigma$ , D0 and B have been previously determined (as in the method proposed by Mau and Gallagher (1972). This equation may also be solved by expressing  $\Delta K0$  as a Taylor series expansion in the fundamental path parameter, giving a form suitable for solution by one of the many "direct" eigensolution methods.

<u>Second Order Equilibrium Equation</u> - A second differentiation of the postbuckling equilibrium equation (3-3) and evaluation at the critical point, gives

$$P_{i}^{\prime} P_{i}^{P} = 0 = \int_{V} \{ 2B_{ai}^{\prime} \sigma_{a}^{\prime} P_{ai}^{P} + B_{ai}^{\prime} \sigma_{a}^{\prime} P_{ai}^{P} + B_{ai}^{\prime} \sigma_{a}^{\prime} + 2B_{ai}^{\prime} (\sigma_{a}^{\prime} f_{a}^{+} \sigma_{a}^{\prime}) \} dV$$
(3-6a)

Substituting for  $\sigma''^p$  and  $B''^p$  gives

$$0 = \int_{V} \{2G_{mi}Al_{amn}\theta_{n}^{\dagger}D0_{ab}\varepsilon_{b}^{\dagger}P + B_{ai}D0_{ab}\varepsilon_{b}^{\dagger}P + Dl_{abc}(4\varepsilon_{b}^{\dagger}P\varepsilon_{c}^{\dagger}f + 2\varepsilon_{b}^{\dagger}P\varepsilon_{c}^{\dagger}P)\} + G_{mi}Al_{amn}\theta_{n}^{\dagger}P_{a} + 2G_{mi}Al_{amn}\theta_{n}^{\dagger}P$$
$$D0_{ab}(\varepsilon_{b}^{\dagger}f + \varepsilon_{b}^{\dagger}P)\} dV \qquad (3-6b)$$

### THE BOEING COMPANY

Using the relations which express  $\theta''^p$  and  $\epsilon''^p$  in terms of  $q''^p$  gives

$$0 = \int_{V} \left[ 2G_{mi}Al_{amr}\theta'_{r}^{f}D0_{ab}\varepsilon'_{b}^{p} + B_{ai}D0_{ab}B_{bj}q'_{j}^{p} + G_{mi}A_{am} \right]$$

$$\left\{ D0_{ab} \left( 2Al_{bns}\theta'_{n}^{p}\theta'_{s}f + Al_{bns}\theta'_{n}^{p}\theta'_{s}^{p} \right) + Dl_{abc} \left( 4\varepsilon_{b}^{p}\varepsilon_{c}^{f} + 2\varepsilon_{b}^{p}\varepsilon_{c}^{p} \right) \right\}$$

$$+ G_{mi}\sigma_{a}Al_{amn}G_{nj}q'_{j}^{p} + 2G_{mi}Al_{amn}\theta'_{n}^{p} \left\{ D0_{ab} \left( \varepsilon_{b}^{f} + \varepsilon_{b}^{p} \right) \right\} \right] dV \qquad (3-6c)$$

This is the second order postbuckling equilibrium equation, which may be written in the form

$$0 = K0_{ij}Q_{j}''^{p} + 2S'P2_{i}^{l} + P2_{i}^{2}$$
(3-6d)

where K0 is again the Jacobian evaluated at the critical point, and

$$P2_{i}^{1} = f_{V}G_{mi} \{D0_{ab}(A_{am}Al_{bns}\theta_{n}^{*P}\theta_{s}^{*f} + Al_{amr}\varepsilon_{b}^{*P}\theta_{r}^{*f} + Al_{amr}\varepsilon_{b}^{f}\theta_{r}^{*P}) + 2Dl_{abc}A_{am}\varepsilon_{b}^{*P}\varepsilon_{c}^{f}\} dV$$

$$P2_{i}^{2} = f_{V}G_{mi} \{D0_{ab}(A_{am}Al_{bns}\theta_{n}^{*P}\theta_{s}^{*P} + 2Al_{amr}\varepsilon_{b}^{*P}\theta_{r}^{*P}) + 2Dl_{abc}A_{am}\varepsilon_{b}^{*P}\varepsilon_{c}^{*P}\} dV$$

The term S' in equation (3-6d) is the derivative of the fundamental path parameter with respect to the postbuckling path parameter, and occurs because of the substitutions

$$\theta_n^{f} = \dot{\theta}_n^{f} s'$$
  
 $\varepsilon_a^{f} = \dot{\varepsilon}_a^{f} s'$ 

(3-6d) may be solved for the postbuckling displacement second derivatives Q''<sup>p</sup>, and for the path derivative S'.

.

.

•

-

.

\_\_\_\_

### THE BOEING COMPANY

+ 
$$3G_{mi}Al_{amn}\theta_{n}^{p} \{DO_{ab}(\varepsilon_{b}^{\prime}, f + \varepsilon_{b}^{\prime}, p) + Dl_{abc}(2\varepsilon_{b}^{f}\varepsilon_{c}, f + 4\varepsilon_{b}^{p}\varepsilon_{c}, f + 4\varepsilon_{b}^{p}\varepsilon_{c}, f + 2\varepsilon_{b}^{p}\varepsilon_{c}, p)\}]dV$$
  
(3-7c)

This is the third order postbuckling equilibrium equation, which may be written in the form

.

$$0 = KO_{ij}Q_{j}^{\prime}P_{i} + 3S^{\prime}P2_{i}^{1} + 3P3_{i}$$
(3-7d)

where

$$P3_{i} = \int_{V} G_{mi} [DO_{ab} \{S^{i}^{2} (A_{am}Al_{bns}\theta_{n}^{i} P_{\theta}^{i} f + Al_{amr} \varepsilon_{b}^{i} P_{\theta}^{i} f + Al_{amr} \varepsilon_{b}^{$$

The term S'' occurs in (3-7d) after making the substitutions

$$\theta_{n}^{\dagger} = \theta_{n}^{f} S^{\dagger}$$

$$\theta_{n}^{\dagger} = \theta_{n}^{f} S^{\dagger} + \theta_{n}^{f} S^{\dagger}^{2}$$

$$\varepsilon_{a}^{\dagger} = \varepsilon_{a}^{f} S^{\dagger}$$

$$\varepsilon_{a}^{\dagger} = \varepsilon_{a}^{f} S^{\dagger} + \varepsilon_{a}^{f} S^{\dagger}^{2}$$

3-9

.

THE BOEING COMPANY

Equation (3-7d) may be solved for the postbuckling displacement second derivatives  $Q'''^p$ , and for the path second derivative S''.

3.3 Solution of Postbuckling Equilibrium Equations

The postbuckling equilibrium equations (3-5d, 6d and 7d) may be solved sequentially to yield the displacement and load derivatives necessary for construction of the postbuckling path. These equations have been formulated here for the general case of an unsymmetric KO Jacobian matrix, and the effects of nonconservative loading are discussed in Appendix B. The solution of the second and higher order equations for the unsymmetric case present some practical difficulties, however. Therefore, in contrast to the general solution outlined in Section 2 for the fundamental equations, the solution given here for the postbuckling equations will be presented for the case of a symmetric KO matrix.

First Order (Bifurcation) Solution - The first order equation (3-5d) may be solved for the eigenvector  $Q'^p$  of postbuckling displacements, and for the critical value of the fundamental path parameter S. The initial step is to relate the unknown critical displacement increments AQ to the eigenvalue S, using the previously computed fundamental displacement derivatives:

$$Q_{i}^{f} = \dot{Q}_{i}^{\star}S + \frac{1}{2}\ddot{Q}_{i}^{\star}S^{2}$$
(3-8)

In addition to the nonlinearity inherent in this relation, the eigenequation is nonlinear for other reasons:

1. Although increments in the displacement derivatives  $\Delta \theta^{f}$ and displacements  $\Delta Q^{f}$  are linearly proportional, the strain increments vary nonlinearly, i.e.  $\Delta \varepsilon_{a}^{f} = A_{am}^{*} \Delta \theta_{m}^{f} + \frac{1}{2} \Delta \theta_{m}^{f} \Delta \theta_{n}^{f}$ .

2. There are  $\Delta^2$  terms ( $\Delta \epsilon^{f} \Delta \theta^{f}$ ,  $\Delta \epsilon^{f} \Delta \epsilon^{f}$ ) in the eigenequation

THE BOEING COMPANY

due to consideration of the nonlinear material effects (effects of the D1\* matrix).

Because of these nonlinearities an eigensolution by direct iteration may not converge. Particular difficulty may be expected during the first few iterations, when the estimated eigenvector contains significant proportions of higher modes for which the  $\Delta \theta^{f}$  at some locations in the structure could be much larger than the corresponding  $\theta^*$ . Also for such higher modes, the contribution to  $\Delta Q$  by S<sup>2</sup> may be large and the  $\Delta^{2}$  terms may be large relative to  $\Delta$  terms. It is therefore necessary to solve first the linear eigen problem, obtained by dropping all nonlinear terms. When convergence has been achieved to within a specified accuracy, iteration is continued with inclusion of all terms until convergence to the desired nonlinear eigensolution.

<u>Higher Order Solutions</u> - With the critical point now defined by the critical value of S, the higher order postbuckling equations may be solved by formation and decomposition of the critical point Jacobian K0. To accomplish the solution, a definition of the postbuckling path parameter R is required. We here follow the general approach of Mau and Gallagher (1972) and take R to be one of the postbuckling displacements, say  $Q_m^P$ . In the PANES program, m is taken as the index of the largest component of the eigenvector  $Q'^P$ . We then impose the requirements at every path point, R, that

$$R' = 1$$
  

$$R'' = R''' = \dots = 0$$
 (3-9)

Although the matrix K0 is singular, this constraint of the mth degree of freedom allows the matrix to be decomposed. A somewhat different approach than this is suggested by Haftka et al. (1970), involving the introduction of an additional constraint equation to make the K0 matrix effectively nonsingular. That approach

THE BOEING COMPANY

however increases the size of the matrix, and no real advantage is seen. The present approach retains the sparsity of K0.

At this point the eigenvector  $Q'^{p}$  is again determined, using the constrained K0 matrix. This is done to achieve consistency in the calculation of  $Q'^{p}$  and the higher order derivatives determined later, as well as for greater accuracy. In terms of the symbolic inverse K0<sup>-1</sup>:

$$0 = K0_{ij}Q_{j}^{P}$$
, with  $Q_{m}^{P} = 1$  (3-10a)

$$Q'_{i}^{p} = K0_{ij}^{-1}(0)$$
 (3-10b)

The second order equation is

$$0 = KO_{ij}Q_{j}'P + 2S'P2_{j}^{1} + P2_{i}^{2}, \text{ with } Q_{m}'P = 0 \quad (3-1)$$

Premultiplying by Q'<sup>p</sup>, and using the symmetry of K0 with  $K0_{ij}Q_{j}^{p} = 0$ , results in

$$0 = Q_{i}^{P} K Q_{ij} Q_{j}^{P} + 2S' Q_{i}^{P} P 2_{i}^{1} + Q_{i}^{P} P 2_{i}^{2} = 2S' Q_{i}^{P} P 2_{i}^{1} + Q_{i}^{P} P 2_{i}^{2}$$
(3-11b)

which then gives

$$S'^{C} = -Q_{i}'^{P}P2_{i}^{2}/2Q_{i}'^{P}P2_{i}^{1}$$
(3-11c)

and 
$$Q'_{i}'^{p} = K0^{-1}_{ij}(-25'P2^{1}_{j} - P2^{2}_{j}) \equiv -25'Q2^{1}_{i} - Q2^{2}_{i}$$
 (3-11d)

The third order equation is

$$0 = KO_{ij}Q_{j}^{\prime}P_{j}^{P} + 3(S^{\prime}P_{i}^{1} + P_{i}^{3}), \text{ with } Q_{m}^{\prime}P_{m}^{P} = 0 \qquad (3-12a)$$

Premultiplying by Q'<sup>p</sup> as before, results in

$$0 = S'' Q_{i}'^{p} P 2_{i}^{l} + Q_{i}'^{p} P 3_{i}$$
(3-12b)
which then gives

$$S'' = -Q_{i}'^{p} P 3_{i} / Q_{i}'^{p} P 2_{i}^{l}$$
(3-12c)

and 
$$Q'_{i}''^{p} = KO_{ij}^{-1}(-3S''P2_{i}^{1} - 3P3_{i}) \equiv -3(S''Q2_{i}^{1} + 3Q3_{i})$$
 (3-12d)

With the critical point derivatives of S and  $Q^p$  known, the postbuckling path can be constructed. The variation of load parameter A with postbuckling path R, is defined by

$$\Lambda' = \dot{\Lambda} S'$$

$$\Lambda'' = \dot{\Lambda} S'' + \ddot{\Lambda} S'^{2} \qquad (3-13)$$

THE BOEING COMPANY

4.0 FINITE ELEMENT PROGRAM AND NONLINEAR SOLUTION ROUTINES

4.1 General Program Characteristics

The major goal of this research effort was the development of improved nonlinear solution techniques and subroutines. It was decided that the most effective way of accomplishing this goal was to develop a practical nonlinear finite element program, into which the various subroutines could be incorporated for checkout and verification. This has been accomplished, and the resulting finite element program has been given the acronym PANES (Program for Analysis of Nonlinear Equilibrium and Stability). Although PANES is a pilot program and is by no means a general structural analyzer (it utilizes only the constant strain triangle element, for 2-D in-plane or 3-D membrane analysis) it demonstrates all of the basic techniques and operations necessary for nonlinear analysis by more general types of finite elements. The program handles geometric nonlinearities and arbitrary nonlinear elastic materials (including very large strain cases), as well as certain forms of nonconservative loadings, i.e. those due to follower-force pressure loadings where the surfaces change in size and orienta-Extension of the program to cases of inelastic materials tion. is considered to be relatively straightforward, with the introduction of appropriate stress-strain constitutive relations.

The present pilot version of PANES has three basic capabilities:

- Analysis of nonlinear structures without critical points, i.e. tracing of simple nonlinear equilibrium paths under a specified general (non-proportional) loading. Various solution techniques are available, with automatic calculation of load step sizes.
- Traversing of limit (maximum and minimum) type critical points, with automatic continuation of the load-path history.

THE BOEING COMPANY

3. Determination of bifurcation type critical points, and prediction of the postbuckling behavior and direction of travel, by means of path derivatives computed at the bifurcation point. Automatic switching from the fundamental path to a postbuckling path, and continuation along the postbuckling path, have not yet been included. Thus the postbuckling capability should be regarded as still in a developmental stage.

#### 4.2 PANES Nonlinear Analysis Routines

This section describes briefly the purpose and capabilities of the program subroutines, in the order in which they appear in the PANES program. Some of these are basic finite element routines, while others are specialized routines needed for generating and solving nonlinear structural equations.

<u>BIGS</u> - Initializes program variables (serves as the calling subroutine for most of the input data reading routines). Also provides problem restart capability by reading or writing the restart tape.

READRS - Reads data file numbers and start or restart codes.

<u>READO</u> - Reads problem identification title. Also reads incremental and iterative constants, such as those relating to the predictor and corrector types, the finite difference expansions for nonlinear materials, and the techniques for continuation of the equilibrium path through limit points.

<u>READ1</u> - Reads basic structural codes and values, and material constants for each material.

READC - Reads user-defined special nodal coordinate systems.

THE BOEING COMPANY

<u>READM</u> - Reads mesh data, including nodal locations and coordinate system codes, and element data.

<u>READK</u> - Reads codes to determine degrees of freedom with specified forces, displacements or constraints.

<u>READP</u> - Reads two load reference curves which define distribution of the applied generalized nodal loads.

<u>READPR</u> - Reads a pressure load reference curve which defines the distribution of the applied pressure loads (one intensity for each element).

<u>READI</u> - Reads incremental load data, including the nodal load and pressure load curve factors for the total load at the end of each increment.

<u>HEAD</u> - Writes a heading output for each load increment step, including load parameter value, number of iterations required and accuracy achieved.

<u>OUTLIM</u> - Predicts and outputs limit point values for the load parameter, and nodal forces and displacements.

OUTPQ - Outputs nodal forces and displacements.

OUTE - Outputs element strains.

<u>QFILL</u> - Uses vector of system-level nodal displacements Q to form vector of element-level nodal displacements q for an element.

<u>PFILL</u> - Takes vector of element-level nodal forces p and adds them to system-level force vector P.

<u>DFILL</u> - Uses element nodal displacement vector q to compute vector of displacement derivatives  $\theta$  within the element.

<u>EFILL</u> - Uses element displacement derivatives vector  $\theta$  to compute vector of strains  $\varepsilon$  within the element.

<u>AFILL</u> - Uses element displacement derivatives  $\theta$  to form Lagrangian AO or Al matrix within the element.

<u>GFILL</u> - Uses element displacement functions to form the  $\theta$ -q transformation matrix G.

<u>MTRAN</u> - Matrix transformation routine, which performs operations of the type  $K_{ij} = D_{ab}B_{ai}B_{bj}$  for given D and B matrices.

<u>ROTO</u> - Transforms element displacements or forces, from either nodal to element or element to nodal coordinate system.

<u>ROTK</u> - Transforms element stiffness matrix from element to nodal coordinate system.

<u>FORCE</u> - Computes internal nodal generalized forces corresponding to given nodal displacements.

<u>PFORCE</u> - Computes applied nodal force loadings, using nodal load reference curves and corresponding load factors.

<u>EFORCE</u> - Computes nodal forces due to applied pressure loadings, using pressure reference curve and factor, and the current area and orientation of each element (determined from geometry and current displacements).

<u>ERCOMP</u> - Computes and outputs error norm for each residual force iteration, using applied (external) forces and computed (internal) forces.

THE BOEING COMPANY

STRAIN - Computes strains for each element using geometry and nodal displacements.

ENERGY - Evaluates the strain energy density for an element at given strain components. This routine will in general be a user supplied routine based on the types of materials being used in the structure.

<u>EVAL</u> - Performs the function (strain energy) evaluations at the current strain state, and at the required adjacent "perturbed" states necessary to establish a strain energy expansion in terms of incremental strains. EVAL calls the STRAIN routine for evaluation, and defines the evaluation points by using the userspecified finite difference sizes. A first, second or third order expansion may be specified, and the corresponding function evaluations are returned in the form of a vector.

<u>U2FORM</u> - Forms coefficients for a general second order Taylor series expansion, using function values provided by EVAL. Used to develop the strain-energy related tensors  $\sigma_i$  and DO<sub>ij</sub> for a material at current deformation state.

<u>U3FORM</u> - Similar to U2FORM, but forms coefficients for a general third order expansion. Develops the tensors  $\sigma_i$ , DO<sub>ij</sub> and Dl<sub>ijk</sub>.

<u>UFILL</u> - Calling routine which calls either U2FORM or U3FORM, depending on desired expansion order.

<u>CFORM</u> - Forms the contribution to the Jacobian stiffness matrix due to the nonconservative pressure loadings.

<u>GENER8</u> - Generates the elemental Jacobian matrix using the current geometry and the material tensors  $\sigma_i$  and  $DO_{ij}$ . Also adds contributions from CFORM if loading is nonconservative.

<u>USUM1</u> - Performs a summing operation between a second or third order tensor function and its vector argument, to give a vector.

<u>USUM21</u> - Performs a summing operation between a third order tensor function and its two (different) vector arguments, to give a vector.

<u>PlCOMP</u> - Computes the nonlinear load term Pl\*, required in generating the second order fundamental equilibrium equations.

<u>RATES</u> - Computes the first and second order fundamental load parameter and displacement rates.

<u>STEP</u> - Provides automatic calculation of a fundamental path load step size, and techniques for traversing limit points.

<u>EIGEN1</u> - Computes the psuedo force term  $Pl^1$ , for use in the inverse power iteration eigensolution process.

EIGEN - Eigensolution routine for inverse power iteration. Calls EIGEN1 routine.

<u>POST2</u> - Computes the second order postbuckling psuedo force terms  $P2^{1}$  or  $P2^{2}$ .

POST3 - Computes the third order postbuckling psuedo force term P3.

<u>PRATES</u> - Computes the first and second order postbuckling load and displacement rates, and third order displacement rate, at the bifurcation point.

<u>VDOT, VCROSS, VLENTH, VNORM</u> - Vector subroutines for computing dot product, cross product, length, and normalizing a vector, respectively.

THE BOEING COMPANY

<u>MERGE</u> - Merges elemental Jacobians into system Jacobian, with provision for constrained degrees of freedom. Forms general unsymmetric Jacobian matrix.

DECOMP - Decomposes unsymmetric Jacobian using Gauss wavefront type procedure. Takes advantage of sparsity but uses total square matrix for storage without packing or external storage devises. (This is a small pilot version decomposition routine.)

SOLVE - Performs forward and backward substitution for unsymmetric Jacobian matrix to provide solution vectors.

-----

4.3 Summary of PANES Input Data

A pictorial of the PANES input deck is shown in Figure 4-1. The input data consists of the following three general types:

Type C: Data on the usual card file. These are data which are needed for each start or restart.

- Type I: Data on File I. These are basic structural data for a given problem, such as material properties and mesh data. They are the same for all load increments and are needed only when starting.
- Type II: Data on File II. File II is not used in the current PANES version. It is provided for possible future use as a file of incremental data (e.g. additional nodal and thermal load data).

The data included on each file are described below. Formats are consistent with FORTRAN IV conventions.

C-1. Start-restart code and data file numbers:

- a. "START" if new problem, or "RESTART" if restarting.
- b. If starting give unit number for file I.
- c. Unit number for file II (need not be given).
- d. Unit number for output file (e.g. printer).

THE BOEING COMPANY





- e. If restarting give load increment number from the end of which a restart is to be made.
- f. If restarting give input restart-tape unit number.
- g. If data is to be saved for future restart give output restart-tape unit number.Format (A4, 6X, 6I5)
- C-2. Problem I.D. title.

Format (20A4).

- C-3. Program control constants (any constant left blank is assigned a default value):
  - a. Specified order of material incremental stress-strain expansion to be used (2 is exact for linear material, maximum order is 3). Default order is 3.
  - Solution predictor type. Type 1 = 1st order,
     Type 2 = 2nd order.

Default type = 2.

c. Maximum number of Jacobian updates per load increment step.

Default = 0.

d. Maximum number of residual force iterations per Jacobian update.

Default = 5.

THE BOEING COMPANY

e. Maximum allowable residual force error norm.

Default =  $1 \times 10^{-8}$ .

Format (4110, F10.0)

C-4. Perturbation difference magnitudes for evaluating strain energy.

a. Difference for computing stiffnesses.

Default =  $1 \times 10^{-3}$ .

b. Difference for computing forces.

Default =  $1 \times 10^{-8}$ .

Format (2F10.0)

- C-5. Program control constants
  - Number of increment subdivisions to be performed as load nears a limit value. Default = 3.
  - B. Ratio of limit load to load increment values at which
     limit point is to be traversed. Default = 0.1.
  - c. Increment step size limitation, computed from slope of load parameter versus path parameter curve, and equal to change in slope divided by average slope.

Default = 0.5.

d. Maximum load increment step size (used especially in unloading), and defined as a factor times the specified load increment.

Default = 1.0.

e. Maximum fraction of current load increment by which load is allowed to reduce after passing a maximum limit point.

Format (I10, 4F10.0)

- I-1. Basic structure definition
  - a. Code for element pressure loads. Code 0 = no pressures, Code 1 = pressures. Default code is 0.
  - b. Degree of freedom per node (2 or 3). No default value.

c. Default thickness for all elements.

Format (2110, F10.0)

I-2. Material property definitions. For each material give material I.D. number, and 2 material constants for use by the strain energy evaluation routine.

Format (I10, 2F10.0)

------

Blank card after data for last material.

THE BOEING COMPANY

I-3. For each special Cartesian coordinate system: the identification number (integer > 2) and counter-clockwise angle (degrees) from basic system X-axis to special system x-axis.

'Format (I10, F10.0)

Blank card after last coordinate system.

I-4. For each node: Node number; identification number of coordinate system to define location; X, Y and Z (or R, θ and Z); identification number of coordinate system to define displacements. (Coordinate I.D. number 0 implies the basic Cartesian system, 1 implies the basic cylindrical system).

Format (215, 3F10.0, 15)

Blank card after last node.

I-5. For each element: element number, material number, thickness, three node numbers (counter-clockwise order).

If thickness is left blank, default value from I-lc is used.

Format (215, F10.0, 315)

. . . . .

Blank card after last element.

I-6. For each DOF with specified displacement or constraint:

If specified displacement, give node number and component (1, 2 or 3) number;

THE BOEING COMPANY

If specified constraint, give node and component number, and independent node and component to which DOF is constrained (independent component number is + for specified force, - for specified displacement). User has option of from 1 to 4 values per card.

Format (4(415))

I-7. Nodal load reference vectors.

Number of vectors (for current program version must be 2)

Format (I10)

For each nonzero component of load vector: node number, component number (1 = X or R, 2 = Y or  $\theta$ , 3 = Z), force or displacement value. User has option of from 1 to 4 values per card.

Format (4(215, F10.0))

Blank card after last value of each vector.

I-8. Pressure Load Reference Vector. (Input only if pressure code in data item I-1 is nonzero.

Number of vectors (for current program version must be 1)

Format (I10)

For each nonzero component of pressure load vector: element number, pressure intensity. User has option of from 1 to 4 values per card.

Format (4(I10, F10.0))

Blank card after last value of vector.

THE BOEING COMPANY

C-6. Incremental load data

Number of load increments

Format (I10)

For each load increment: solution predictor type (if left blank, value from C-3b is used), the cumulative factors to be applied to the nodal load reference vectors, pressure value for all elements. Pressure is applied in element positive z-coordinate direction.

Format (I10, 3F10.0)

C-7. Final blank card.

Problems may be run consecutively (first data item for each problem follows immediately after last item of preceding problem). Final blank card indicates that all problems have been read.

4.4 Summary of PANES Output

The description of PANES output is conveniently divided into two parts. The first is primarily an echo check of the input data, and the second part consists of output results for each load increment.

4.4.1 Echo Check of Input Data

Initial Output - The first page of PANES output for a problem is essentially an echo check of input items C-1 to C-5, I-1 and I-2. An indication is given as to whether the problem is being started or restarted. If it is restarted then the previous increment number is given, from the end of which the restart is

progressing. Next the problem I.D. title is printed, followed by the various control constants and finite difference magnitudes (DFE and DFF). The limit point related control constants (MJUMP, JUMPR, SLOPED, FLAMAX and LAMIN) are then printed. Finally the basic structural quantities from I-1, and the material property constants from I-2 are printed.

<u>Special Coordinate Systems</u> - These are the user-defined direction (special Cartesian) systems of input data item I-3. Quantities printed are the system number, and counter-clockwise angle (in degrees) from the basic X axis to the special-system x axis.

<u>Node Definitions</u> - The information given in input item I-4 is printed. Values are the node number, location coordinate system number (0 = basic Cartesian, 1 = basic cylindrical), X or R coordinate, Y or  $\theta$  (degrees) coordinate, Z coordinate, direction coordinate system number (0 = basic Cartesian, 1 = basic cylindrical, >1 = number of special user defined system).

<u>Element Definitions</u> - The information given in input item I-5 is printed. Values are the element number, material I.D. number, element thickness, the three element node numbers in counterclockwise order, and the computed element area.

Force-Displacement-Constraint Prescriptions - These are the codes given in input data item I-6. Quantities printed are the dependent node and component number, and independent node and component number. (If specified displacement, no independent numbers are given).

Nodal Load Reference Vectors - For each input component of the two load vectors from input item I-7, the node number, component number, and load value are printed.

THE BOEING COMPANY

<u>Pressure Load Reference Vector</u> - For each input component of the pressure load vector from input item I-8, the element number and pressure intensity are printed.

<u>Incremental Load Data</u> - Quantities related to input data item C-6 are printed. First is printed the number of load increments to be run. Then for each increment is given the increment number, input or default value for the predictor type, and factors to be applied to the two nodal load reference vectors and the pressure load reference vector.

4.4.2 Results for Each Load Increment

Interative Error Values - An error norm computed at the end of each iteration is printed. The error norm is obtained by a ratio of unbalanced (residual) forces to total forces.

Increment Heading - The load increment and load step numbers are printed, along with the load increment and load step values at the end of the step. Following this are the nodal load reference vector factors, the element pressure vector factor, the predictor type for the increment, the maximum allowable number of Jacobian updates and the number performed during this load step, the maximum allowable number of iterations per update and the number performed since the last update, and the maximum allowable error norm and the error norm actually achieved.

Forces and Displacements - The cumulative nodal displacements and corresponding internal forces are output. The node number is printed, followed by the U, V and W (or R,  $\theta$ , Z) components of force and displacement.

<u>Strains</u> - The cumulative element strains are output. The element number is printed, followed by the XX, YY, and XY strains in the element coordinate system.

Limit Point Output - When a limit point is traversed, the predicted value of the incremental limit load parameter is output, followed by the predicted limit forces and displacements, and strains.

<u>Bifurcation Point Output</u> - When an eigenvalue solution is performed to determine a critical point, the eigenvalue computed for each inverse power iteration is printed, along with the location of the maximum value in the eigenvector.

<u>Decomposition Output</u> - Whenever the Gauss decomposition routine is called, it prints the value (sign and base 10 logarithm) of the Jacobian stiffness determinant.

HE BOEING COMPANY

# 5.0 ILLUSTRATIVE PROBLEMS

Four example problems problems are presented here in order to illustrate various aspects of nonlinear equilibrium and stability theory, and to demonstrate use of the developed nonlinear subroutines and the PANES finite element program. Section 5.1 describes a snap-through truss problem with geometric nonlinearity and a maximum and a minimum limit point. Section 5.2 describes a simple pressure membrane with nonconservative type loading (changing load area), resulting in a maximum limit point. The toroidal membrane in Section 5.3 is a fairly difficult problem, involving nonlinear (Mooney) material with followerforce loads and changing load areas. It results in very large displacements and strains, and a maximum and a minimum limit This problem demonstrates some unique capabilities of point. the PANES program. Finally, Section 5.4 describes a simple bifurcation/postbuckling model, with asymmetric behavior.

#### 5.1 Snap-Through Truss

This is a problem similar to that used as a test case by several researchers in nonlinear structural analysis, e.g., Haisler et al. (1971). The system consists of a single inclined bar (or one half of a symmetric two-member truss) as shown in Figure 5-1. The bar has length 1.0 with axial stiffness  $AE = 2 \times 10^7$  (Hookean material), is inclined initially at a slope of 1:100, and is subjected to a vertical end load P. The PANES program idealization of this system used two constant strain triangle elements, with modulas  $E = 2 \times 10^8$  and thickness = 0.1. Node 4 was constrained to have vertical displacement equal to that at node 2, so that a system with essentially one degree of freedom (vertical displacement Q) is obtained.

The expression for the axial strain,  $\varepsilon$ , is given by

$$\varepsilon = -.01Q + 0.5Q^2$$

(5.1-la)

00 0000 0140 0010 - +1--





and stress,  $\sigma$ , is given by

$$\sigma = -2,000,000Q + 100,000,000Q^2$$
 (5.1-1b)

Thus the axial force is  $-200,000Q + 10,000,000Q^2$ , from which it may by shown that the vertical applied force, P, is given by the following basic equilibrium equation.

$$P = 2000Q - 300,000Q^{2} + 10,000,000Q^{3}$$
 (5.1-2)

Differentiating with respect to Q, and evaluating at a reference equilibrium configuration (Q\*, P\*), gives the first order equilibrium equation

$$\dot{\mathbf{P}}^* = 2000\dot{\mathbf{Q}}^* - 600,000\mathbf{Q}^*\dot{\mathbf{Q}}^* + 30,000,000\mathbf{Q}^*\dot{\mathbf{Q}}^*$$
 (5.1-3a)

or

$$\dot{P}^* = K0^*\dot{Q}^*$$
 (5.1-3b)

where the Jacobian stiffness is

$$K0^* = 2000 - 600,0000^* + 30.000.0000^*^2$$

A second differentiation and evaluation of equation (5.1-2) results in the second order equilibrium equation

$$P^* = K0^*Q^* + P1^*$$
 (5.1-4)

where the psuedo force Pl is defined by

$$P1* = 600,000\dot{q}*^2 + 60,000,000Q*\dot{q}*^2$$

Using the equilibrium equations (5.1-3) and (5.1-4), the P-Q path history can be computed by various incremental and iterative approaches, including identification of limit points.

(Of course for this simple one-degree-of-freedom system, the path can be obtained immediately from the basic equilibrium equation (5.1-2)). The P-Q path history is shown in Figure 5-1. The PANES program solution was accomplished using six user-specified load increments (P = 1.0, 2.0, 3.0, 4.0, 5.0, 10.0), although about 30 additional load steps were selected automatically (mostly to achieve the desired accuracy in locating and traversing the limit point regions). Most load steps required only one or two residual force iterations, with use of a second order predictor.

#### 5.2 Simple Pressure Membrane

One of the simplest problems which can be used to illustrate some of the effects of nonconservative loading in stability analysis is the simple pressure membrane shown in Figure 5-2. The system consists of a flat membrane 2.0 wide by 1.0 high with unit thickness, and subjected to a uniform pressure intensity A on one side. The ends of the membrane slide along the 45° supports, and are constrained to move together equally in the X direction. This gives a single degree-of-freedom system, with X-direction force P and displacement Q, and the membrane undergoes a uniform stretching in the Y direction. The solution was verified by a finite element analysis using the PANES The finite element mesh consisted of the two constant program. strain triangle elements shown in Figure 5-2, with X-direction displacements at nodes 2-4 constrained to equal the displacement at node 1. Zero displacements were enforced in the Z direction.

Considering large displacement and large strain effects, the stretch in the Y direction is denoted by  $\lambda$ , and is equal to the change in length divided by original length. The strain-energy density U, measured per unit undeformed volume, is taken to be defined by the function

$$U = C_{1}(\lambda - 1)^{2} + C_{2}(\lambda - 1)^{4} = C_{1}Q^{2} + C_{2}Q^{4}$$
 (5.2-1)



Note that we could define a Lagrangian strain  $\epsilon$ , and stress-like quantity  $\sigma,$  by

$$\varepsilon = \frac{1}{2}(\lambda^2 - 1)$$
  
$$\sigma = \frac{\partial U}{\partial \lambda} = \frac{\partial U}{\partial Q} = 2C_1Q + 4C_2Q^3$$

However, this is not necessary in the present problem, for which the force can be derived directly from the strain-energy function.

The basic equilibrium equation for the system is written using the equivalence of external force (defined in terms of the pressure loading  $\Lambda$ ) and internal force (defined as the derivative of strain energy with respect to displacement Q):

$$P = 2 \Lambda (1+Q) = 2(2C_1Q + 4C_2Q^3)$$
 (5.2-2)

Choosing specific values for the material constants  $C_1 = 10$ and  $C_2 = -1$ , the basic equilibrium equation may be written as

$$\Lambda (1+Q) = 20Q - 4Q^3$$
 (5.2-3)

Differentiating equation (5.2-3) provides the first order equilibrium equation

$$\dot{\Lambda}$$
 (1+Q) +  $\Lambda \dot{Q}$  = 20 $\dot{Q}$  - 12 $Q^2 \dot{Q}$  (5.2-4a)

or at an equilibrium configuration ( $\mathbb{N}^*$ ,  $\mathbb{Q}^*$ ) we can write

$$\dot{\Lambda} * = ((20 - 12Q*^2 - \Lambda *)/(1+Q*)) \dot{Q}^*$$
 (5.2-4b)

THE BOEING COMPANY

From this equation it can be seen that the value of a Jacobian stiffness K0\*, relating  $\Lambda$  \* with  $\dot{Q}$ \*, is

$$K0^* = (20 - 12Q^{*2} - \Lambda^*)/(1+Q^*) \qquad (5.2-4c)$$

(For simplicity we have here defined the Jacobian relative to  $\ddot{\Lambda}$  rather than  $\dot{P})$  .

Differentiating equation (5.2-3) a second time gives the second order equilibrium equation, as

 $\ddot{\Lambda}$  (1+Q) +  $2\dot{\Lambda}\dot{Q}$  +  $\Lambda\ddot{Q}$  = 20 $\ddot{Q}$  - 24Q $\dot{Q}^2$  - 12Q $^2\ddot{Q}$  (5.2-5a)

or at an equilibrium configuration we can write

$$\tilde{\Lambda}^* = K0^*\tilde{Q}^* + P1^*$$
 (5.2-5b)

where the psuedo force Pl is given by

$$P1^{*} = (-24Q^{*}\dot{Q}^{*}^{2} - 2\dot{\Lambda}^{*}\dot{Q}^{*})/(1+Q^{*})$$
 (5.2-5c)

To illustrate the actual behavior of the system, we now choose Q as the path parameter, and without loss of generality specify at every point the conditions  $\dot{Q} = 1$  and  $\ddot{Q} = 0$ . To aid in determination of a limit type critical point, we have the condition  $\dot{\Lambda} * = 0$  at the limit point. Using this condition, and solving equation (5.2-4b) with the use of (5.2-3), we find a limit point at ( $\Lambda * = 8.0$ , Q\* = 1.0). The entire  $\Lambda$  -Q path history may be determined by various incremental and iterative predictor-corrector schemes, and is shown in Figure 5-2. The PANES program solution to this problem used a second-order predictor with residual-force corrective iterations.



#### 5.3 Toroidal Membrane

The problem illustrated here is a toroidal membrane under internal pressure, shown in Figure 5-3. This structure exhibits a highly nonlinear type of behavior with very large displacements and strains, and both a maximum and a minimum limit point. The second-order predictor was employed in a PANES program solution, along with residual-force corrective iterations to achieve equilibrium at each load step. Both the predictor and corrector incorporated all changing load area and follower-force effects. (This gave an unsymmetric K0\* matrix, whose decomposition was obtained by the Gauss wavefront procedure. It was apparent that the unsymmetric effects became large enough that their inclusion was necessary for convergence).

The torus was assumed to be of Mooney material with constants  $C_1 = 80$  and  $C_2 = 20$ , and was analyzed using plane-stress membrane elements. Geometry and displacement components are defined in Figure 5-3. The torus has major radius 10, minor radius 2, and thickness .05. Cylindrical coordinates were employed to model a wedge-shaped segment of the major circumference, of from 2 to 10 degrees arc. Constraints were employed in the radial and vertical directions in order to equalize corresponding displacements along the two sides of the wedge.

Table 5-1 summarizes computed values of key displacements for the user-defined (input) pressure increments and the computed limit pressures, obtained with three different meshes. (N denotes the number of subdivisions over one half of the minor circumference). The indicated convergence with mesh refinement is of the kind to be expected, with finer meshes giving a more flexible structure, and resulting generally in somewhat larger displacements and lower limit point values. Computer run times ranged from 1 minute (IBM central processor time) for mesh N = 4, to 7 minutes for mesh N = 12. These times should be regarded in a

qualitative fashion only, since for example much of the time was spent in solving the linear unsymmetric stiffness equations and this time could be reduced by use of a production type equation solver.

More detailed results for the fine mesh (N = 12) are shown in Table 5-2. There the data columns represent respectively the pressure, load increment number (user-specified increment), step number (where the PANES program automatically divided the specified increment into a number of smaller steps), number of residual-force type iterations performed in order to achieve the required accuracy, and values of the radial and vertical displacements at key points. Figure 5-3 gives very interesting plots of two key load-displacement paths, and indicates that no difficulties were caused by a displacement which followed an extremely irregular "doubling back" type of path, including sharp curvature sections. The basic results for this problem are corroborated by another solution to the same problem by Key (1974), who developed a finite element program with a Newton Raphson solution technique, and obtained results for pressure levels up to near the first limit point.

It may be of use for comparison/test purposes to mention corresponding results obtained by increasing the major radius from 10 to 12. Maximum and minimum limit points occurred at pressures, h, of 4.355 and 4.125, respectively, while maximum displacements (at h = 5.0) increased roughly 20 percent.

5-9

----

THE BUEING COMPANY



Figure 5-3: TORUS SOLUTION RESULTS

BOEING COMPANY

TABLE 5-1: TORUS RESULTS (MESH/CONVERGENCE CHARACTERISTICS)

h	U <sub>A</sub>	U <sub>B</sub>	₩ <sub>B</sub>	UC	
$\begin{array}{r} .10\\ .50\\ 1.0\\ 2.0\\ 3.0\\ 4.0\\ 4.5\\ \dagger\dagger 4.688\\ \dagger\dagger 4.458\\ 5.0\end{array}$	0065 0399 0827 162 216 168 .057 .653 2.032 -3.142	.0015 .0114 .0315 .109 .281 .732 1.438 2.891 10.056 22.203	.0120 .0529 .1080 .243 .436 .784 1.203 1.959 6.787 20.007	.0076 .0484 .1130 .301 .631 1.371 2.436 4.543 15.551 35.578	N = 4
$\begin{array}{r} .10\\ .50\\ 1.0\\ 2.0\\ 3.0\\ 4.0\\ \dagger + 4.473\\ \dagger + 4.244\\ 4.5\\ 5.0\end{array}$	0062 0414 0870 171 223 129 .641 1.946 -1.665 -4.746	.0013 .0112 .0323 .117 .316 .908 2.892 10.374 19.076 27.973	.0135 .0571 .1157 .261 .476 .904 1.967 7.039 15.972 24.446	.0067 .0489 .1169 .320 .693 1.640 4.531 15.951 29.895 42.678	N = 6
.10 .50 1.0 2.0 3.0 4.0 ++ 4.357 ++ 4.127 4.5 5.0	0065 0431 0902 176 226 088 .636 1.903 -2.895 -5.510	.0013 .0115 .0333 .123 .340 1.050 2.889 10.607 22.706 32.327	.0136 .0584 .1189 .270 .499 .991 1.964 7.199 18.904 27.005	.0070 .0505 .1210 .334 .736 1.853 4.518 16.239 34.581 47.404	N = 12

++ Limit points

....

TABLE 5-2: TORUS INCREMENTAL RESULTS (N = 12)

	h	INCR.	STEP	ITER.	υ <sub>A</sub>	U <sub>B</sub>	<sup>w</sup> в	U <sub>C</sub>
	.00625	1	1	7	00026	.00004	.00121	.00028
†	.10		2	6	0065	.0013	.0136	.0070
t	.50	2	1	3	0431	.0115	.0584	.0505
†	1.0	3	1	2	0902	.0333	.1189	.1210
	1.5	4	1	2	135	.069	,188	.213
†	2.0		2	2	176	.123	.270	.334
	2.5	5	1	2	209	.207	.370	,499
	2.75	5	2	1	220	.266	.430	.606
t	3.0		3	1	226	.340	.499	.736
	3.360	6	1	2	219	.490	.621	.986
	3.632		2	2	193	.658	.742	1.254
	3.816		3	1	155	.819	.849	1.503
	3.908		4	1	127	.922	.914	1.661
+	4.0		5	1	088	1.050	.991	1.853
	4.108	7	1	1	023	1.248	1.106	2.147
	4.187		2	1	.049	1.446	1.216	2.439
	4.243		3	'1	.123	1.641	1.322	2.723
	4.284		4	1	.197	1.829	1.421	2,995
	4.309		5	1	.268	2.005	1.513	3.250
	4.327		6	1	.335	2.167	1.596	3.483
	4.339		7	1	.395	2.313	1,671	3.692
	4.346		8	1	.448	2.441	1.736	3.875
	4.351		9	1	.494	2.549	1.791	4.031
	4.354		10	0	.532	2.640	1.837	4.161
++	4.357		-	-	.636	2.889	1.964	4.518
	4.354		11	6	.795	3.273	2,160	5.067
	4.329		12	2	1.094	4.018	2,546	6.137
	4.279		13	2	1.472	5.063	3.117	7.654
	4.229		14	1	1.784	6,117	3.740	9.210
	4.204		15	1	1.921	6.715	4.121	10.108
	4.179		16	1	2.040	7.413	4,595	11.172
	4.158		17	1	2.110	8.143	5.125	12.302
	4.145		18	1	2,124	8.725	5.577	13.218
	4.137		19	1	2.107	9.194	5,958	13.962
	4.133		20	1	2.076	9.568	6.272	14.559
	4.130		21	0	2.040	9.860	6.526	15.029
++	4.127		-	-	1.903	10.607	7.199	16.239
	4.130		22	5	1.659	11.544	8.083	17.760
	4.153		23	´ 2	1.036	13.257	9.785	20.534
	4.327		24	3	-1.387	18.766	15.270	28.987
†	4.5		25	2	-2.895	22.706	18.904	34,581
†	5.0	8	1	5	-5.510	32.327	27.005	47.404

+ Input pressure loads. Intermediate pressures were selected automatically by the program.

++ Limit points.

# 5.4 Asymmetric Buckling Model

The problem illustrated here is that used by several investigators of bifurcation and postbuckling behavior, e.g., Thompson (1970). The model consists of a spring and rigid bar as shown in Figure 5-4. The asymmetric postbuckling behavior is due to the decreasing resisting moment arm of the spring force about point O, as the top of the bar deflects to the right.

The conservative load A is applied vertically to the top of the bar. The spring is initially inclined at 45 degrees, and has constant stiffness K. This is a single degree of freedom system, defined by the horizontal displacement Q. The vertical component of distance from O to B at any time is equal to  $\sqrt{1-Q^2}$ , and the horizontal distance from A to B is 1+Q, from which the length of the spring is found to be  $\sqrt{2(1+Q)}$ . The moment arm of the spring force about point O is then determined as  $\sqrt{1-(1+Q)/2}$ . Equating the external applied and internal resisting moments gives the basic postbuckling equilibrium equation for the system as

$$Q = K(\sqrt{2(1+Q)} - \sqrt{2}) \sqrt{1 - (1+Q)/2} = K(\sqrt{1+Q} - 1) \sqrt{1-Q}$$
(5.4-1a)

or

$$\Lambda = K(\sqrt{1-Q^2} - \sqrt{1-Q})/Q$$
 (5.4-1b)

Evaluating  $\Lambda$  from this expression using a small finite difference in Q, gives the critical load value as

$$\Lambda = \frac{1}{2} K \qquad (5.4-2)$$

THE BOEING COMPANY

Similarly a second-order finite difference evaluation gives the critical asymmetric load rate as

$$\frac{\partial \Lambda}{\partial Q} = -\frac{3}{8} K \tag{5.4-3}$$

The location of the critical (bifurcation) point was verified by a PANES program solution.



THE BOEING COMPANY

## 6.0 CONCLUSIONS AND RECOMMENDATIONS

<u>Conclusions</u> - The present work provides improved techniques for solution of structures with material and geometric nonlinearities. FORTRAN subroutines have been developed, and incorporated into a new nonlinear finite-element program called PANES (<u>Program for Analysis of Nonlinear Equilibrium and Stability</u>). A new approach is developed for representing an arbitrary nonlinear material in terms of a finite-difference generated stress-strain expansion, and is considered to be of major significance. This approach leads to formulation of perturbation-type equilibrium equations of any desired order, and is effective even for numerically integrated finite elements with large degrees of freedom. The formulation should provide a unifying basis for design of many nonlinear structural analysis programs.

The present PANES program is a pilot version, capable of analyzing problems with large strain and arbitrary nonlinear elastic materials, and provides membrane finite elements (two or three degrees of freedom per node) with nonconservative pressure loading. It includes automated techniques which have been developed for selection of load step sizes, and for locating and traversing maximum and minimum limit-type critical points. Subroutines are also included for location of bifurcation-type critical points on a general nonlinear prebuckling path, and for determining the symmetric or asymmetric postbuckling behavior. The postbuckling capabilities have not yet been completely automated and tested, however, and should be regarded as being in a developmental stage.

<u>Recommendations</u> - The PANES program solution routines provide a significant pilot capability for analysis of structures with highly nonlinear material and geometric effects, and should now

6-1

-----

THE BOEING COMPANY

be extended and evaluated for a wider class of practical structures. Listed below are recommendations for future work.

- 1. The first priority should be given to improving and verifying the PANES postbuckling subroutines. This effort should include the generalization of program logic to accept additional types of finite elements, and in particular the addition of simple bar elements to the program. These elements will simplify the study of postbuckling results, and are desireable for initial verification because of the somewhat complex nature of the new nonlinear material postbuckling theory. Automated techniques should be incorporated for branching to the postbuckling path, similar to the existing PANES techniques for traversing limit points.
- 2. The program should be extended to incorporate a number of higher-order finite elements. Important candidate elements are plates, shells, and isoparametric-solid elements. Such elements will greatly extend the analysis capabilities of the program, and also importantly demonstrate the effectiveness of the new nonlinear solution techniques for elements which are numerically integrated and have large degrees of freedom.
- 3. PANES now handles an arbitrary nonlinear elastic material, by use of the proper material strain-energy definition. Formation of the stress-strain expansion relation should be generalized to cases of inelastic material, i.e., those materials for which a strain-energy function does not exist. The concept of this generalization is not difficult, but some study is required to develop an effective algorithm for forming the higher-order stress-strain expansion terms.
- 4. A number of largely theoretical improvements should be studied. These include the treatment of multiple and closely-spaced critical points (as often occur in an optimally designed light-weight structure), and the method

THE BOEING COMPANY

of postbuckling behavior solution for cases of an unsymmetric Jacobian stiffness matrix. Incorporation of a third-order fundamental path predictor also appears desireable, especially for use in prediction of bifurcationtype critical points. The nonlinear eigenvalue solution for these points is somewhat costly, and would have to be performed less often with the higher-order predictor. The program size capability should be increased to handle the expected range of practical nonlinear structural problems. This involves some reorganization of the main program logic, and the addition of a production-type linear equation solver such as the Gauss-wavefront routines used in the BOPACE elastic-plastic-creep program (Vos and

5.

----

Armstrong (1973)).
- 7.0 REFERENCES AND BIBLIOGRAPHY
- 1. Argyris, J. H., Fried, I., and Scharpf, D. W., (1968), "The TET 20 and TEA 8 Elements for the Matrix Displacement Method," Journal of the Royal Aeronautical Society, Vol. 72, pp. 618-623.
- Connor, J., and Morin, N., (1970), "Perturbation Techniques in the Analysis of Geometrically Nonlinear Shells," High Speed Computing of Elastic Structures, Proceedings of Symposium, Liege, Belgium, Vol. 2, pp. 683-705.
- 3. Britvec, S. J., and Chilver, A. H., (1963), "Elastic Buckling of Rigidly-Jointed Braced Frames," Journal of the Engineering Mechanics Division, ASCE, Vol. 89, pp. 217-255.
- Ecer, A., (1973), "Finite Element Analysis of the Postbuckling Behavior of Structures," AIAA Journal, Vol. 11, pp. 1532-1538.
- 5. Gallagher, R. H., and Mau, S. T., (1972), "A Method of Limit Point Calculation in Finite Element Structural Analysis," NASA CR-2115, National Aeronautics and Space Administration, Washington, D.C.
- Gallagher, R. H., and Padlog, J., (1963), "Discrete Element Approach to Structural Instability Analysis," AIAA Journal, Vol. 1, pp. 1437-1439.
- 7. Haftka, R. T., Mallett, R. H., and Nachbar, W., (1971), "Adaption of Koiter's Method to Finite Element Analysis of Snap-Through Buckling Behavior," International Journal of Solids and Structures, Vol. 7, pp. 1427-1445.
- Haftka, R. T., Mallett, R. H., and Nachbar, W., (1970), "A Koiter-Type Method for Finite Element Analysis of Nonlinear Structural Behavior, Volume I: The Modified Structure Method," AFFDL-TR-70-130, Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio.
- 9. Haisler, W. E., Stricklin, J. A., and Stebbins, F. J., (1971), "Development and Evaluation of Solution Procedures for Geometrically Nonlinear Structural Analysis by the Direct Stiffness Method," AIAA Paper No. 71-356, AIAA/ASME 12th Structures, Structural Dynamics and Materials Conference, Anaheim, California.
- Huseyin, K., (1973), "The Multiple-Parameter Perturbation Technique for the Analysis of Non-Linear Systems," International Journal of Non-Linear Mechanics, Vol. 8, pp. 431-443.

7-1

THE BOEING COMPANY

- 11. Hutchinson, J. W., (1973), "Post-Bifurcation Behavior in the Plastic Range," Journal of the Mechanics and Physics of Solids, Vol. 21, pp. 163-190.
- 12. Hutchinson, J. W., (1973), "Imperfection Sensitivity in the Plastic Range," Journal of the Mechanics and Physics of Solids, Vol. 21, pp. 191-204.
- Johns, K. C. and Chilver, A. H., (1971), "Multiple Path Generation at Coincident Branching Points, "International Journal of Mechanical Sciences, Vol. 13, pp. 899-910.
- 14. Key, J. E., (1974), "A Note on the Numerical Solution of Certain Problems in Finite Elasticity by the Finite Element Method," paper to be published in International Journal for Numerical Methods in Engineering.
- 15. Koiter, W. T., (1945), "Over de Stabiliteit van het Elastisch Evenwicht," Ph.D. Thesis, Technische Hogeschool, Delf, Holland. English translation (1970), "The Stability of Elastic Equilibrium," AFFDL-TR-70-25, Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio.
- 16. Levinson, M., and Burgess, I. W., (1971), "A Comparison of Some Simple Constitutive Relations for Slightly Compressible Rubber-Like Materials," International Journal of Mechanical Sciences, Vol. 13, pp. 563-572.
- 17. Mallett, R. H., and Marcal, P. V., (1968), "Finite Element Analysis of Nonlinear Structures," Journal of the Structural Division, ASCE, Vol. 94, pp. 2081-2105.
- 18. Marcal, P.V., (1969), "Large Deflection Analysis of Elastic-Plastic Plates and Shells," Brown University Technical Report No. N00014-67-A-0191-0006/1.
- 19. Marchenko, G. A., (1966), "Solution of Nonconservative Problems in the Theory of Elastic Stability by Ritz's Method," Isvestiya VUZ. Aviatsionnaya Tekhnika, No. 3, pp. 62-68.
- 20. Mau, S. T., and Gallagher, R. H., (1972), "A Finite Element Procedure for Nonlinear Prebuckling and Initial Postbuckling Analysis," NASA CR-1936, National Aeronautics and Space Administration, Washington, D.C.
- 21. McNamara, J. F., and Marcal, P. V., (1971), "Incremental Stiffness Method for Finite Element Analysis of the Nonlinear Dynamics Problem," Proceedings, O.N.R. International Symposium on Numerical and Computer Methods in Structural Mechanics, University of Illinois.

- 22. Morin, N., (1970), "Nonlinear Analysis of Thin Shells," Massachusetts Institute of Technology, Department of Civil Engineering, Report R70-43.
- 23. Murray, D. W., and Wilson, E. L., (1969), "Finite Element Large Deflection Analysis of Plates," Journal of the Engineering Mechanics Division, ASCE, Vol. 95, pp. 143-165.
- 24. Nayak, G. C., and Zienkiewicz, O. C., (1972), "Note on the 'Alpha' - Constant Stiffness Method for the Analysis of Non-Linear Problems," International Journal for Numerical Methods in Engineering, Vol. 4, pp. 579-582.
- 25. Nemat-Nasser, S., and Shatoff, H. D., (1973), "Numerical Analysis of Pre- and Post-Critical Response of Elastic Continua at Finite Strains," Computers and Structures, Vol. 3, pp. 983-999.
- 26. Oden, J. T., and Key, J. E. (1970), "Numerical Analysis of Finite Axisymmetric Deformations of Incompressible Elastic Solids of Revolution," International Journal of Solids and Structures, Vol. 6, pp. 497-518.
- 27. Osias, J. R., and Swedlow, J. L., (1974), "Finite Elasto-Plastic Deformation-I: Theory and Numerical Examples," International Journal of Solids and Structures, Vol. 10, pp. 321-339.
- 28. Prasad, S. N., and Herrmann, G., (1972), "Adjoint Variational Methods in Nonconservative Stability Problems," International Journal of Solids and Structures," Vol. 8, pp. 29-40.
- 29. Remmler, K. L., Cawood, D. W., Stanton, J. A., and Hill, R., (1966), "Solutions of Systems of Nonlinear Equations," Final Report to NASA Marshall Space Flight Center by Lockheed Missiles and Space Company, HREC 0178-1, A-78333.
- 30. Sandidge, D. W., (1973), "Stability and Postbuckling Behavior of Hyperelastic Bodies at Finite Strain by the Finite Element Method," M. S. Thesis, University of Alabama in Huntsville, Huntsville, Alabama.
- 31. Schmit, L. A., Bogner, F. K., and Fox, R. L., (1968), "Finite Deflection Structural Analysis Using Plate and Shell Discrete Elements," AIAA Journal, Vol. 6, pp. 781-791.
- 32. Sewell, M. J., (1965), "The Static Perturbation Technique in Buckling Problems," Journal of the Mechanics and Physics of Solids, Vol. 13, pp. 247-263.

7-3

THE BOEING COMPANY

- 33. Straight, J. W., (1968), "Solution to Beam Vibrations Problems with Mixed Response-Excitation Input Information," AIAA/ ASME 9th Structures, Structural Dynamics and Materials Conference, Palm Springs, California.
- 34. Stricklin, J. A., Haisler, W. E., and Von Riesemann, W. A., (1972), "Computation and Solution Procedures for Nonlinear Analysis by Combined Finite Element-Finite Difference Methods," Computers and Structures, Vol. 2, pp. 955-974.
- 35. Stricklin, J. A., Haisler, W. E., and Von Riesemann, W. A., (1973), "Evaluation of Solution Procedures for Material and/or Geometrically Nonlinear Structural Analysis," AIAA Journal, Vol. 11, pp. 292-299.
- 36. Thompson, J. M. T., (1963), "Basic Principles in the General Theory of Elastic Stability," Journal of the Mechanics and Physics of Solids, Vol. 11, pp. 13-20.
- 37. Thompson, J. M. T., (1969), "A General Theory for the Equilibrium and Stability of Discrete Conservative Systems," Zeitschrift für angewandte Mathematik und Physik," Vol. 20, pp. 797-846.
- 38. Thompson, J. M. T., (1970), "A New Approach to Elastic Branching Analysis," Journal of the Mechanics and Physics of Solids, Vol. 18, pp. 29-42.
- 39. Thompson, J. M. T., and Hunt, G. W., (1971), "A Theory for the Numerical Analysis of Compound Branching," Zeitschrift für angewandte Mathematik und Physik," Vol. 22, pp. 1001-1015.
- 40. Tillerson, J. R., Stricklin, J. A., and Haisler, W. E., (1973), "Numerical Methods for the Solution of Nonlinear Problems in Structural Analysis," ASME Symposium on Numerical Solution of Nonlinear Structural Problems, Detroit, Michigan.
- 41. Turner, M. J., Dill, E. H., Martin, H. C., and Melosh, R. J., (1960), "Large Deflections of Structures Subjected to Heating and External Loads," Journal of the Aero/Space Sciences, Vol. 27, pp. 97-106.
- 42. Vos, R. G., (1970), "Finite element analysis of plate buckling and postbuckling," Ph.D. Thesis, Rice University, Houston, Texas.
- 43. Vos, R. G., (1974), "Finite Element Solution of Nonlinear Structures by Perturbation Technique," First International Conference on Numerical Methods in Nonlinear Mechanics, University of Texas at Austin, Austin, Texas.

- 44. Vos, R. G., and Armstrong, W. H., (1973), "BOPACE Theoretical Manual," Report on Space Shuttle Main Engine Computer Program Development, to NASA Marshall Space Flight Center by Boeing Aerospace Company, D5-17266-1.
- 45. Vos, R. G., and Vann, W. P., (1973), "A Finite Element Tensor Approach to Plate Buckling and Postbuckling," International Journal for Numerical Methods in Engineering, Vol. 5, pp. 351-365.
- 46. Walker, A. C., (1969), "A Method of Solution for Nonlinear Simultaneous Algebraic Equations," International Journal for Numerical Methods in Engineering, Vol. 1, pp. 177-180.
- 47. Zienkiewicz, O. C., and Nayak, G. C., (1971), "A General Approach to Problems of Large Deformation, and Plasticity Using Iso-Parametric Elements," 3rd Conference on Matrix Methods in Structural Mechanics, Wright-Patterson Air Force Base, Ohio.
- 48. Zienkiewicz, O. C., Valliappan, S., and King, I. P., (1969), "Elasto-Plastic Solutions of Engineering Problems 'Initial Stress,' Finite Element Approach," International Journal for Numerical Methods in Engineering, Vol. 1, pp. 75-100.

THE BOEING COMPANY

#### APPENDIX A: FINITE DIFFERENCE EXPANSIONS

For the type of nonlinear solution techniques utilized in this work, it is necessary to generate the Jacobian stiffness matrix as well as various force-type vectors associated with residuals and nonlinear predictor quantities. In the general case involving nonlinear materials, these quantities cannot be effectively determined by an explicit process, and must be generated numerically. The numerical representation may be based on a direct expansion of the generalized forces in terms of displacements (e.g. the method for Jacobian generation used by Oden and Key (1970)), or, as in the approach used here, it may be based on an expansion of stresses in terms of strains. In any case the procedure requires the expansion of a dependent function of several independent variables, about a known reference point.

An effective expansion procedure has been developed in the present work by use of a Taylor series, in which the expansion coefficients (partial derivatives) are evaluated using finite difference expressions. After a study of various alternatives, it was concluded that the most efficient scheme involves forward differences rather than central differences, because the forward differences result in simple formulas and require a minimum number of function evaluations. In addition, if an approximate solution path increment is known, i.e. if the approximate increments which will occur in the independent variables are known, then a more accurate function representation can be obtained with the forward difference scheme by selecting the appropriate difference values. Difference coefficients are derived here for expansions of linear, quadratic and cubic form.

Linear Form - Coefficients for a linear expansion correspond to those in a two-point forward difference formula. The derivation is rather trivial, but it serves to illustrate the basic procedure.

A-1

The Taylor series expansion of an arbitrary function f, in terms of independent variables  $x_i$ , is

$$f = f^{l} + f_{i} \Delta x_{i}$$
 (A-1)

where  $f_i = \hat{f}/\partial x_i$  denotes the partial derivative of f with respect to the ith independent variable, and  $\Delta$  denotes an incremental quantity. The unique types of terms may be derived by considering only one of the independent variables, which we denote simply by x. Referring to Figure A-1, we describe the values of f at points 1 and 2 by the linear expansion

$$\begin{cases} f^{1} \\ f^{2} \\ f^{2} \end{cases} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \qquad \begin{cases} f^{1} \\ f_{x} \Delta x \end{cases}$$
 (A-2)

Inversion of this relation gives explicit definition to the difference coefficients, in the form of the matrix in the inverse relation:

$$\begin{cases} f^{1} \\ f_{x} \Delta x \end{cases} = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \begin{cases} f^{1} \\ f^{2} \end{cases}$$
 (A-3)

or

$$f_{x} = (-f^{1} + f^{2}) / \Delta x \qquad (A-4)$$





Quadratic Form - Coefficients for the quadratic expansion correspond to those in a three-point forward difference formula. Because the expansion involves terms no higher than second order, the unique types of difference coefficients can be derived by considering only two of the independent variables, say x and y. The corresponding Taylor series expansion for the function f(x,y)is given by the expression

$$f = f^{1} + f_{x}\Delta x + f_{y}\Delta y + \frac{1}{2} f_{xx}(\Delta x)^{2} + f_{yx}\Delta y\Delta x + \frac{1}{2} f_{yy}(\Delta y)^{2} \quad (A-5)$$



Referring to Figure A-2, we write

$$\begin{bmatrix} f^{1} \\ f^{2} \\ f^{3} \\ f^{4} \\ f^{5} \\ f^{6} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1/2 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1/2 \\ 1 & 2 & 0 & 2 & 0 & 0 \\ 1 & 1 & 1/2 & 1 & 1/2 \\ 1 & 0 & 2 & 0 & 0 & 2 \end{bmatrix} \begin{cases} f^{1} \\ f_{x} \Delta x \\ f_{y} \Delta y \\ f_{xx} (\Delta x)^{2} \\ f_{yx} \Delta y \Delta x \\ f_{yy} (\Delta y)^{2} \end{bmatrix}$$
 (A-6)

and inverting the above relation gives

THE BOEING COMPANY

$$\begin{cases} f^{1} \\ f_{x} \Delta x \\ f_{y} \Delta y \\ f_{y} \Delta y \\ f_{xx} (\Delta x)^{2} \\ f_{yx} \Delta y \Delta x \\ f_{yy} (\Delta y)^{2} \end{cases} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -3/2 & 2 & 0 & -1/2 & 0 & 0 \\ -3/2 & 0 & 2 & 0 & 0 & -1/2 \\ 1 & -2 & 0 & 1 & 0 & 0 \\ 1 & -1 & -1 & 0 & 1 & 0 \\ 1 & 0 & -2 & 0 & 0 & 1 \end{bmatrix} \begin{cases} f^{1} \\ f^{2} \\ f^{3} \\ f^{4} \\ f^{5} \\ f^{6} \\ f^{6} \end{cases}$$
 (A-7)

<u>Cubic Form</u> - Coefficients in the cubic expansion correspond to those in a four-point forward difference formula. Because the expansion terms are no higher than third order, the unique coefficient types can be derived by writing the function in terms of only three independent variables, say x, y and z:

$$\mathbf{f} = \mathbf{f}^{1} + \mathbf{f}_{\mathbf{x}} \Delta \mathbf{x} + \mathbf{f}_{\mathbf{y}} \Delta \mathbf{y} + \mathbf{f}_{\mathbf{z}} \Delta \mathbf{z} + \frac{1}{2} \mathbf{f}_{\mathbf{x}\mathbf{x}} (\Delta \mathbf{x})^{2} + \dots + \frac{1}{6} \mathbf{f}_{\mathbf{x}\mathbf{x}\mathbf{x}} (\Delta \mathbf{x})^{3}$$
$$+ \dots + \frac{1}{6} \mathbf{f}_{\mathbf{z}\mathbf{z}\mathbf{z}} (\Delta \mathbf{z})^{3} \qquad (A-8)$$



Figure A-3: Cubic Difference Expansion

A-4

	Re	efe	rr	:i)	ng	t	0	F	ig	uı	ce.	A	- 1	3,	W	e	W	ri	te	ž			• •
$\left[ f^{1} \right]$		<b>[</b> 1	. (	) (	0.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Ţ	$\int f^1$
$f^2$		1	1	0	0	$\frac{1}{2}$	0	0	0	0	0	$\frac{1}{6}$	• 0	0	0	0	0	0	0	0	0		f <sub>v</sub> ∆x
$f^3$		1	0	1	0	0	0	$\frac{1}{2}$	0	0	0	0	0	0	$\frac{1}{6}$	0	0	0	0	0	0		f <sub>ν</sub> Δy
$f^4$		1	0	0	1	0	0	0	0	0	$\frac{1}{2}$	0	0	0	0	0	0	0	0	0	$\frac{1}{6}$		f <sub>z</sub> Δz
$f^5$		1	2	0	0	2	0	0	0	0	0	$\frac{4}{3}$	0	0	0	0	0	0	0	0	0		$f_{xx}(\Delta x)^2$
$f^6$		1	1	1	0	$\frac{1}{2}$	1	$\frac{1}{2}$	0	0	0	$\frac{1}{6}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{6}$	0	0	0	0	0	0		f <sub>yx</sub> ΔyΔx
$f^7$		1	0	2	0	0	0	2	0	0	0	0	0	0	$\frac{4}{3}$	0	0	0	0	0	0		$f_{yy}(\Delta y)^2$
$f^8$		1	1	0	1	$\frac{1}{2}$	0	0	1	0	$\frac{1}{2}$	$\frac{1}{6}$	0	0	0	$\frac{1}{2}$	0	0	$\frac{1}{2}$	0	$\frac{1}{6}$		$f_{zx} \Delta z \Delta x$
f <sup>9</sup>		1	0	1	1	0	0	$\frac{1}{2}$	0	1	$\frac{1}{2}$	0	0	0	$\frac{1}{6}$	0	0	$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{6}$		f <sub>zv</sub> ∆z∆y
f <sup>10</sup>		1	0	0	2	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	$\frac{4}{3}$		$f_{zz}(\Delta z)^2$
$f^{11}$	1	1	3	0	0	<u>9</u> 2	0	0	0	0	0	$\frac{9}{2}$	0	0	0	0	0	0	0	0	0		$f_{xxx}(\Delta x)^3$
$f^{12}$		1	2	1	0	2	2	$\frac{1}{2}$	0	0	0	$\frac{4}{3}$	2	1	$\frac{1}{6}$	0	0	0	0	0	0		$f_{yxx} \Delta y (\Delta x)^2$
$f^{13}$		1	1	2	0	$\frac{1}{2}$	2	2	0	0	0	$\frac{1}{6}$	1	2	$\frac{4}{3}$	0	0	0	0	0	0		$f_{yyx}(\Delta y)^2 \Delta x$
$f^{14}$		1	0	3	0	0	0	$\frac{9}{2}$	0	0	0	0	0	0	$\frac{9}{2}$	0	0	0	0	0	0		$f_{yyy}(\Delta y)^3$
f <sup>15</sup>		1	2	0	1	2	0	0	2	0	$\frac{1}{2}$	$\frac{4}{3}$	0	0	0	2	0	0	1	0	$\frac{1}{6}$		$f_{zxx}^{\Delta z (\Delta x)^2}$
$f^{16}$		1	1	1	1	$\frac{1}{2}$	1	$\frac{1}{2}$	1	1	$\frac{1}{2}$	$\frac{1}{6}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{6}$	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{6}$		f <sub>zyx</sub> ∆z∆y∆x
$f^{17}$		1	0	2	1	0	0	2	0	2	$\frac{1}{2}$	0	0	0	$\frac{4}{3}$	0	Ò	2	0	1	$\frac{1}{6}$		f <sub>zyy</sub> Δz(Δy) <sup>2</sup>
f <sup>18</sup>		1	1	0	2	$\frac{1}{2}$	0	0	2	0	2	$\frac{1}{6}$	0	0	0	1	0	0	2	0	$\frac{4}{3}$		$f_{zzx}(\Delta z)^2 \Delta x$
f <sup>19</sup>		1	0	1	2	0	0	$\frac{1}{2}$	0	2	2	0	0	0	$\frac{1}{6}$	0	0	1	0	2	$\frac{4}{3}$		$f_{zzy}(\Delta z)^2 \Delta y$
$f^{20}$		1	0	0	3	0	0	0	0	0	9 2	0	0	0	0	0	0	0	0	0	$\frac{9}{2}$		$f_{zzz}(\Delta z)^3$

(A-9)

THE BOEING COMPANY

A-5

THE BOEING COMPANY

The inverse of this matrix is the matrix of difference coefficients, and is given below.

	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$-\frac{11}{6}$	- 3	0	0		$\frac{3}{5}$ 0	0	0	0	0	$\frac{1}{3}$	0	0	0	0	0	0	0	0	0
$-\frac{11}{6}$	- 0	3	0	0	0	$-\frac{3}{2}$	0	0	0	0	0	0	$\frac{1}{3}$	0	0	0	0	0	0
$-\frac{11}{6}$	- 0	0	3	0	0	0	0	0	$-\frac{3}{2}$	0	0	0	0	0	0	0	0	0	$\frac{1}{3}$
2	-5	0	0	4	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0
2	$-\frac{5}{2}$	$-\frac{5}{2}$	0	$\frac{1}{2}$	3	$\frac{1}{2}$	0	0	0	0	$-\frac{1}{2}$	$-\frac{1}{2}$	0	0	0	0	0	0	0
2	0	-5	0	0	0	4	0	0	0	0	0	0	-1	0	0	0	0	0	0
2	$-\frac{5}{2}$	0	$-\frac{5}{2}$	$\frac{1}{2}$	0	0	3	0	$\frac{1}{2}$	0	0	0	0	$-\frac{1}{2}$	0	0	$-\frac{1}{2}$	0	0
2	0	$-\frac{5}{2}$	$-\frac{5}{2}$	0	0	$\frac{1}{2}$	0	3	$\frac{1}{2}$	0	0	0	0	0	0	$-\frac{1}{2}$	0	$-\frac{1}{2}$	0
2	0	0	-5	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	-1
-1	3	0	0	- 3	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
-1	2	1	0	-1	- 2	0	0	0	0	0	1	0	0	0	0	0	0	0	0
-1	1	2	0	0	- 2	-1	0	0	0	0	0	1	0	0	0	0	0	0	0
-1	0	3	0	0	0	- 3	0	0	0	0	0	0	1	0	0	0	0	0	0
-1	2	0	1	-1	0	0	- 2	0	0	0	0	0	0	1	0	0	0	0	0
-1	1	1	1	0	-1	0	-1	-1	0	0	0	0	0	0	1	0	0	0	0
-1	0	2	1	0	0	- 1	0	- 2	0	0	0	0	0	0	0	1	0	0	0
-1	1	0	2	0	0	0	-2	0	-1	0	0	0	0	0	ò	0	1	0	0
-1	0	1	2	0	0	0	0	<del>-</del> 2	-1	0	0	0	0	0	0	0	0	1	0
-1	0	0	3	0	0	0	0	0	<b>-</b> 3	0	0	0	0	0	0	0	0	0	1

A-6

\_...

THE BOEING COMPANY

Organization of Terms - The number of unique terms in a symmetric tensor of order M and dimension N, is summarized as follows.

Order	Number of Terms
0	1
1	N
2	N(N+1)/2
3	N(N+1)(N+2)/6
, M	N(N+1)(N+2) (N+M-1)/M!

These relations also give the number of unique Mth order partial derivatives involved in an Mth order Taylor series expansion of N variables. For example, the coefficients of a second-order expansion in three variables (x,y,z) are the 3(3+1)/2 = 6 second partial derivatives (xx,yx,yy,zx,zy,zz). The total number of terms required in the expansion is the sum of the numbers of partial derivatives of each order, for example a second-order expansion in three variables requires 1 + 3 + 3(3+1)/2 = 10 total terms, and therefore a minimum of 10 function evaluations to determine the coefficient (partial derivative) values.

The required coefficients for an expansion are conveniently organized into a one dimensional array, for example for a function of three variables the array is

 $(f^{1} f_{1} f_{2} f_{3} f_{11} f_{21} f_{22} f_{31} \cdots f_{111} \cdots)$ 

The terms are then easily retrieved from this array using the tensor relations given above. Thus using  $L_i$ ,  $L_{ij}$ ,  $L_{ijk}$ ,  $\cdots$  to denote the location of the respective i, ij, ijk,  $\cdots$  derivative terms within the array, we have for a function of N variables

A-7

<u>General Discussion</u> - It may be observed that the forward difference scheme outlined here requires a minimum number of function evaluations, i.e. one plus the number of partial derivatives involved in the expansion. Along with its other advantages of possible improved accuracy in certain types of situations, this would seem to indicate that the forward difference formulas presented here constitute the best approach for the required function expansions.

An interesting alternative derivation of the difference coefficient matrices is possible by a procedure used in the finite element method. The function to be expanded may be thought of as the quantity (say displacement) being interpolated within the finite element. The function is then defined in terms of its values at the nodes (i.e. the independent variable values) times the corresponding shape functions. The required partial derivatives can then be evaluated explicitly at the "origin", i.e. a corner node at which the sides of the element form an orthogonal coordinate system, simply by differentiating the element shape functions. Of course the appropriate finite element shape functions must be available, but if they are then this process allows derivation of the difference formulas without inversion of a matrix. These considerations were initially responsible for the selection of the forward difference approach, and the second and third order coefficients were evaluated in this manner, for example the third order coefficients using the TET 20 element of Argyris et al. (1968).

A-8

THE BOEING COMPANY

#### APPENDIX B: NONCONSERVATIVE LOADING EFFECTS

<u>General Considerations</u> - Several types of nonconservative loading can occur in finite element analysis, for example where the applied generalized nodal forces depend on the system displacements, velocities, or other displacement or deformation parameters. The nonconservative effects considered in the present work are those due to nodal forces which are a function of the nodal displacements, and in particular the effects of pressure loadings where the pressurized surface undergoes significant changes in area and orientation. General formulations may be developed for these cases, in terms of area integrals and pressure intensities. However, in order to illustrate completely the basic effects, the nonconservative load terms are derived here for the special case of a constant-strain-triangle (CST) finite element.



Figure B-1: CST Element for Nonconservative Loading

-----

<u>CST Element Definitions</u> - Referring to Figure B-1, consider the CST element with nodes 1, 2 and 3, and subjected to a normal pressure loading of intensity h. The vectors  $V_{21}$  and  $V_{31}$  connect the nodes 1-2 and 1-3, respectively. The cross product  $V_{21} \times V_{31}$ is then a vector with (positive) direction normal to the element surface and with magnitude equal to twice the element area.

Coordinates in the initial configuration are defined by the components x,y,z, and corresponding displacements by u,v,w. Then by defining for node k, the quantities

$$cx_{k} = (x_{k} + u_{k}) - (x_{1} + u_{1}) + cy_{k} = (y_{k} + v_{k}) - (y_{1} + v_{1}) + cz_{k} = (z_{k} + w_{k}) - (z_{1} + w_{1})$$
(B-1a)

we may write the vectors V as

$$V_{21} = (cx_2, cy_2, cz_2) V_{31} = (cx_3, cy_3, cz_3)$$
(B-1b)

J

while their rates are given by

$$\dot{\mathbf{v}}_{21} = (\dot{\mathbf{u}}_2 - \dot{\mathbf{u}}_1, \dot{\mathbf{v}}_2 - \dot{\mathbf{v}}_1, \dot{\mathbf{w}}_2 - \dot{\mathbf{w}}_1) \\ \dot{\mathbf{v}}_{31} = (\dot{\mathbf{u}}_3 - \dot{\mathbf{u}}_1, \dot{\mathbf{v}}_3 - \dot{\mathbf{v}}_1, \dot{\mathbf{w}}_3 - \dot{\mathbf{w}}_1)$$
(B-1c)

B.1 Fundamental Equilibrium Formulation

In this section the nonconservative loading effects are formulated for the CST element, and used to generalize the fundamental equilibrium equations in Section 2 to analysis of nonconservative systems.

ļ

Basic Load Equation - For a uniform pressure loading, each node of the CST element has an equivalent concentrated load vector p, given by

$$p_{i} = \frac{1}{6} h (V_{21} \times V_{31})_{i}$$
 (B-2a)

The nodal force during an increment of loading is defined by the basic nonconservative load equation

$$p_{i} = \frac{1}{6} (h^{*} + \Lambda h^{\circ}) (V_{21} \times V_{31})_{i}$$
 (B-2b)

where  $h^*$  is the pressure intensity at the reference equilibrium configuration,  $h^\circ$  is a constant pressure distribution and  $\Lambda$  is the incremental load parameter.

First-Order Load Equation - Differentiating equation (B-2b) with respect to the fundamental path parameter, gives

$$\dot{p}_{1} = \frac{1}{6}\dot{\Lambda} h^{\circ}(V_{21} \times V_{31})_{1} + \frac{1}{6} (h^{*} + \Lambda h^{\circ})(\dot{V}_{21} \times V_{31} + V_{21} \times \dot{V}_{31})_{1}$$
Evaluating at the reference equilibrium configuration
$$(B-3a)$$

 $(\Lambda = 0, V = V^*, \text{ etc.})$  gives

$$\dot{p}_{i}^{*} = \frac{1}{6}\dot{\Lambda}^{*} h^{\circ} (V_{21}^{*} \times V_{31}^{*})_{i} + \frac{1}{6}h^{*} (V_{21}^{*} \times V_{31} + V_{21}^{*} \times V_{31}^{*}) (B-3b)$$

This is the first order nonconservative load equation. To put it into the desired form, we write the vector q of element displace-ments as

$$q = (u_1 v_1 w_1 u_2 v_2 w_2 u_3 v_3 w_3)$$

from which it follows using equations (B-1), that

$$\frac{1}{6} h (V_{21} \times \dot{V}_{31} + \dot{V}_{21} \times V_{31})_{i} = C_{ij} \dot{q}_{j}$$

------

THE BOEING COMPANY

where

$$C_{ij} = \frac{1}{6} h \begin{bmatrix} \frac{cz_2}{2} & -cy_2 & 0 & cz_3 & -cy_3 & 0 \\ -cz_3 & +cy_3 & cx_2 & -cy_3 & -cy_3 & -cy_3 & -cy_2 & cy_2 \\ -cz_2 & 0 & -cx_3 & -cz_3 & 0 & cx_3 & -cz_2 & 0 \\ -cy_3 & +cx_3 & -cx_3 & -cx_3 & 0 & -cy_2 & cx_2 & 0 \end{bmatrix}$$

The desired form of the first order nonconservative load equation is then

$$\dot{p}_{i}^{*} = \Lambda \dot{p}_{i}^{\circ} + C_{ij}^{*} \dot{q}_{j}^{*}$$

$$p_{i}^{\circ} = \frac{1}{6} h^{\circ} (V_{21}^{*} \times V_{31}^{*})_{i}$$
(B-3c)

where

The first part of  $p^*$  in (B-3c) is a usual nodal load rate, and occurs in the nonconservative form of equilibrium equation (2-7c) as a contribution to the load term P° (see(2-9)). The second part of  $p^*$  occurs in the nonconservative form of (2-7c) as an unsymmetric contribution to the Jacobian stiffness K0\* (note that since  $C_{ij}^*$  occurs on the left-hand-side of (2-7c), it must be subtracted from K0\*).

<u>Second-Order Load Equation</u> - A second differentiation of (B-2b) gives

$$\ddot{\mathbf{p}}_{i} = \frac{1}{6}\ddot{\Lambda}\mathbf{h}^{\circ} (\mathbf{V}_{21} \times \mathbf{V}_{31})_{i} + \frac{1}{3}\dot{\Lambda}\mathbf{h}^{\circ} (\dot{\mathbf{V}}_{21} \times \mathbf{V}_{31} + \mathbf{V}_{21} \times \dot{\mathbf{V}}_{31})_{i} + \frac{1}{6} (\mathbf{h}^{*} + \Lambda \mathbf{h}^{\circ}) (\ddot{\mathbf{V}}_{21} \times \mathbf{V}_{31} + \mathbf{V}_{21} \times \ddot{\mathbf{V}}_{31} + 2\dot{\mathbf{V}}_{21} \times \dot{\mathbf{V}}_{31}) \qquad (B-4a)$$

THE BOEING COMPANY

Evaluating at the reference equilibrium configuration provides

$$\ddot{p}_{1}^{*} = \frac{1}{6} \ddot{\Lambda}^{*} h^{\circ} (v_{21}^{*} \times v_{31}^{*})_{1} + \frac{1}{3} \dot{\Lambda}^{*} h^{\circ} (\dot{v}_{21}^{*} \times v_{31}^{*} + v_{21}^{*} \times \dot{v}_{31}^{*})_{1}$$
$$+ \frac{1}{6} h^{*} (\ddot{v}_{21}^{*} \times v_{31}^{*} + v_{21}^{*} \times \ddot{v}_{31}^{*} + 2\dot{v}_{21}^{*} \times \dot{v}_{31}^{*})_{1} \qquad (B-4b)$$

This is the second order nonconservative load equation, which may be written in the form

$$\dot{p}_{i}^{*} = C_{ij}^{*} \dot{q}_{j}^{*} + pl_{i}^{*}$$
(B-4c)  
where  $pl_{i}^{*} = \frac{1}{6} \ddot{\Lambda}^{*} h^{\circ} (V_{21}^{*} \times V_{31}^{*})_{i} + \frac{1}{3} \dot{\Lambda}^{*} h^{\circ} (\dot{V}_{21}^{*} \times V_{31}^{*})_{i}$   
 $+ V_{21}^{*} \times \dot{V}_{31}^{*})_{i} + \frac{1}{3} h^{*} (\dot{V}_{21}^{*} \times \dot{V}_{31}^{*})_{i}$ 

### B.2 Bifurcation and Postbuckling Formulation

<u>Basic Load Equation</u> - Development of the nonconservative load effects for bifurcation and postbuckling follows the fundamental relations of section B.l. Using equation (B-2b), we may write the nodal force during an increment of loading on the postbuckling path, as

$$p_{i} = p_{i}^{f} + p_{i}^{p} = \frac{1}{6} (h^{*} + \Lambda h^{\circ}) ((V_{21}^{f} + V_{21}^{p}) \times (V_{31}^{f} + V_{31}^{p}))_{i} (B-5a)$$

Since this relation will be used to establish a nonconservative form of the postbuckling equilibrium equation (3-3), the fundamental load contribution must be subtracted from (B-5a). We then obtain

$$p_{i}^{p} = \frac{1}{6} (h^{*} + \Lambda h^{\circ}) (V_{21}^{f} \times V_{31}^{p} + V_{21}^{p} \times V_{31}^{f} + V_{21}^{p} \times V_{31}^{p})_{i}$$
(B-5b)

First-Order Load Equation - Differentiating equation (B-5b) with respect to the postbuckling path parameter, gives

$$p_{i}'^{p} = \frac{1}{6} \wedge h^{\circ} (v_{21}^{f} \times v_{31}^{p} + v_{21}^{p} \times v_{31}^{f} + v_{21}^{p} \times v_{31}^{p})_{i}$$

$$+ \frac{1}{6} (h^{*} + \wedge h^{\circ}) (v_{21}'^{f} \times v_{31}^{p} + v_{21}^{f} \times v_{31}'^{p} + v_{21}'^{p} \times v_{31}^{f} + v_{21}^{p} \times v_{31}'^{f}$$

$$+ v_{21}'^{p} \times v_{31}^{p} + v_{21}^{p} \times v_{31}'^{p})_{i} \qquad (B-6a)$$

Evaluating at the critical bifurcation point ( $V^{p} = 0$ ), gives  $p_{1}'^{p} = \frac{1}{6} (h^{*} + \Lambda h^{\circ}) (V_{21}^{f} \times V_{31}'^{p} + V_{21}'^{p} \times V_{31}^{f})$  (B-6b)

Substituting for V<sup>f</sup> gives

$$p_{1}'^{p} = \frac{1}{6} (h^{*} + \Lambda h^{\circ}) (V_{21}^{*} \times V_{31}'^{p} + \Delta V_{21}^{f} \times V_{31}'^{p} + V_{21}'^{p} \times V_{31}^{*} + V_{21}'^{p} \times V_{31}'^{p} + V_{31}'^{p} + V_{$$

and using the relations which express  $V'^{p}$  in terms of  $q'^{p}$ , gives  $p'_{i}^{p} = C'_{ij}q'_{j}^{p} + \frac{1}{6}h^{*}(\Delta V^{f}_{21} \times V'^{p}_{31} + V'^{p}_{21} \times \Delta V^{f}_{31})_{i}$  $+ \frac{1}{6} \wedge h^{\circ}(V'^{*}_{21} \times V'^{p}_{31} + \Delta V^{f}_{21} \times V'^{p}_{31} + V'^{p}_{21} \times V'^{*}_{31} + V'^{p}_{21} \times \Delta V^{f}_{31})_{i}$ 

This is the first order postbuckling nonconservative (B-6d) load expression, which may be written in the form

$$p_{i}'^{p} = C_{ij}^{*}q_{j}'^{p} + pl_{i}^{1}$$
 (B-6e)

where

$$pl_{i}^{1} = \frac{1}{6} \wedge h^{\circ} (V_{21}^{*} \times V_{31}^{*p} + V_{21}^{*p} \times V_{31}^{*})_{i}$$
$$+ \frac{1}{6} (h^{*} + \wedge h^{\circ}) (\Delta V_{21}^{f} \times V_{31}^{*p} + V_{21}^{*p} \times \Delta V_{31}^{f})_{i}$$
$$= \frac{1}{B-6}$$

, **j** 

Expression (B-6e) provides the necessary nonconservative addition to the load term, in the first order postbuckling equation (3-5d). The Jacobian KO\* again becomes nonsymmetric due to the subtraction of the C\* terms. Thus a nonlinear, nonconservative eigen equation is produced, which may be solved by the same approaches discussed in Section 3 for the symmetric problem.

Second-Order Load Equation - A second differentiation of (B-5b) with respect to the postbuckling path parameter, gives

$$p_{1}^{\prime} {}^{p} = \frac{1}{6} \Lambda^{\prime} {}^{h} {}^{o} (v_{21}^{f} x v_{31}^{p} + v_{21}^{p} x v_{31}^{f} + v_{21}^{p} x v_{31}^{p})_{i}$$

$$+ \frac{1}{3} \Lambda^{\prime} {}^{h} {}^{o} (v_{21}^{f} x v_{31}^{p} + v_{21}^{f} x v_{31}^{p} + v_{21}^{p} x v_{31}^{f} + v_{21}^{p} x v_{31}^{p})_{i}$$

$$+ \frac{1}{6} (h^{*} + \Lambda h^{o}) (v_{21}^{\prime} x v_{31}^{p} + 2v_{21}^{\prime} x v_{31}^{p} + v_{21}^{f} x v_{31}^{p} + v_{21}^{\prime} x v_{31}^{f} + v_{21}^{\prime} x v_{31}^{\prime} + v$$

Evaluating at the critical point (V  $^{\rm p}$  = 0), with the critical value of h = h\* +  $\Lambda\,h^\circ,$  gives

$$p_{i}^{\prime} p_{i}^{p} = \frac{1}{3} \wedge h^{\circ} (v_{21}^{f} x V_{31}^{p} + V_{21}^{\prime} x V_{31}^{f})_{i} + \frac{1}{6} h (2 V_{21}^{f} x V_{31}^{p} + V_{21}^{f} x V_{31}^{\prime} + V_{21}^{\prime} x V_{31}^{f} + 2 V_{21}^{\prime} x V_{31}^{\prime} + 2 V_{21}^$$

Using the relations which express V''<sup>p</sup> in terms of q''<sup>p</sup>, gives  $p'_{i}'^{p} = \frac{1}{3} \wedge 'h^{\circ} (v_{21}^{f} \times v_{31}'^{p} + v_{21}'^{p} \times v_{21}^{f})_{i} + c_{ij}q'_{j}'^{p} + \frac{1}{6}h(2v'_{21}^{f} \times v'_{31}^{p} + 2v'_{21}^{p} \times v'_{31}^{f} + 2v'_{21}^{p} \times v'_{31}^{p})_{i} \qquad (B-7c)$ 

This is the second order postbuckling nonconservative load expression, which may be written in the form

$$p_{i}'^{P} = C_{ij}q_{j}'^{P} + 2S'^{P}2_{i}^{1} + p2_{i}^{2}$$
(B-7d)

where

$$p_{1}^{1} = \frac{1}{6} \dot{h} h^{\circ} (v_{21}^{f} x V_{31}^{p} + V_{21}^{p} x V_{31}^{f}) i$$
$$+ \frac{1}{6} h (\dot{v}_{21}^{f} x V_{31}^{p} + V_{21}^{p} x \dot{V}_{31}^{f}) i$$

and

$$p2_{i}^{2} = \frac{1}{3}h(V_{21}^{P} \times V_{31}^{P})$$
 i

Third-Order Load Equation - A third differentiation of (B-5b) and evaluation at the critical point, provides

$$p_{1}^{\prime} p_{1}^{\prime} = \frac{1}{2} \wedge p_{1}^{\prime} (v_{21}^{f} x v_{31}^{\prime p} + v_{21}^{\prime p} x v_{31}^{f}) i$$

$$+ \frac{1}{2} \wedge p_{1}^{\prime} (v_{21}^{f} x v_{31}^{\prime p} + 2v_{21}^{\prime f} x v_{31}^{\prime p} + v_{21}^{\prime p} x v_{31}^{f} + 2v_{21}^{\prime p} x v_{31}^{\prime f}$$

$$+ 2v_{21}^{\prime p} x v_{31}^{\prime p}) i$$

$$+ \frac{1}{6} (v_{21}^{f} x v_{31}^{\prime \prime p} + 3v_{21}^{\prime f} x v_{31}^{\prime p} + 3v_{21}^{\prime f} x v_{31}^{\prime p} + v_{21}^{\prime p} x v_{31}^{f} + 3v_{21}^{\prime p} x v_{31}^{\prime f}$$

$$+ 3v_{21}^{\prime p} x v_{31}^{\prime f} + 3v_{21}^{\prime p} x v_{31}^{\prime p} + 3v_{21}^{\prime p} x v_{31}^{\prime p} + 3v_{21}^{\prime p} x v_{31}^{\prime p}$$

$$(B-8a)$$

Using the relations which express V'' in terms of q'', gives

$$p_{i}^{\prime \prime \prime ,p} = \frac{1}{2} \Lambda^{\prime \prime \prime h} (v_{21}^{f} x v_{31}^{\prime p} + v_{21}^{\prime p} x v_{31}^{f})_{i} + \frac{1}{2} \Lambda^{\prime \prime h} (v_{21}^{f} x v_{31}^{\prime p})_{i} + \frac{1}{2} \Lambda^{\prime \prime h} (v_{21}^{f} x v_{31}^{\prime p})_{i} + \frac{1}{2} \lambda^{\prime \prime h} (v_{21}^{f} x v_{31}^{\prime p})_{i} + \frac{1}{2} \lambda^{\prime \prime h} (v_{21}^{f} x v_{31}^{\prime p})_{i} + v_{21}^{\prime p} x v_{31}^{f})_{i} + 2 v_{21}^{\prime p} x v_{31}^{\prime p})_{i} + c_{ij} q_{j}^{\prime \prime \prime p}$$

$$+ \frac{1}{2} h (v_{21}^{\prime \prime f} x v_{31}^{\prime p} + v_{21}^{\prime f} x v_{31}^{\prime p} + v_{21}^{\prime p} x v_{31}^{\prime f} + v_{21}^{\prime p} x v_{31}^{\prime f} + v_{21}^{\prime p} x v_{31}^{\prime p})_{i} + v_{21}^{\prime p} x v_{31}^{\prime p} + v_{21}^{\prime p} x v_{31}^{\prime p})_{i} \qquad (B-8b)$$

This is the third order postbuckling nonconservative load expression, which may be written in the form

$$p_{i}^{\prime} p_{j}^{\prime} = C_{ij}q_{j}^{\prime} p_{j}^{\prime} + 3(S^{\prime}p_{i}^{2} + p_{i}^{3})$$
 (B-8c)

where

$$p_{3_{1}} = s^{2} \left\{ \frac{1}{6} \ddot{\lambda} h^{\circ} (v_{21}^{f} x v_{31}^{*p} + v_{21}^{*p} x v_{31}^{f})_{i} + \frac{1}{3} \dot{\lambda} h^{\circ} (\dot{v}_{21}^{f} x v_{31}^{*p} + v_{21}^{*p} \dot{v}_{31}^{f})_{i} \right\} \\ + \frac{1}{6} h (\ddot{v}_{21}^{f} x v_{31}^{*p} + v_{21}^{*p} \ddot{v}_{31}^{f})_{i} \right\} \\ + s^{*} \left\{ \frac{1}{6} \dot{\lambda} h^{\circ} (v_{21}^{f} x v_{31}^{*p} + v_{21}^{*p} x v_{31}^{f} + 2 v_{21}^{*p} x v_{31}^{*p})_{i} \right\} \\ + \frac{1}{6} h (\dot{v}_{21}^{f} x v_{31}^{*p} + v_{21}^{*p} x \dot{v}_{31}^{f})_{i} \right\} \\ + \frac{1}{6} h (\dot{v}_{21}^{f} x v_{31}^{*p} + v_{21}^{*p} x \dot{v}_{31}^{f})_{i} \right\}$$

APPENDIX C: PANES PROGRAM LISTING

This appendix contains a FORTRAN IV listing of the PANES (Program for Analysis of Nonlinear Equilibrium and Stability) program. Following the program listing is a listing of input data for the torus problem described in section 5.3.

C	**********	*00000040
C	P A N E SIPROGRAM FOR ANALYSIS OF NONLINEAR FOUTLIBRIUM/STABILITY	00000040
С	R.G. VOS, THE BOEING COMPANY, PHONE 773-2638, KENI, WASHINGTON	00000060
C	PANES IBM 360 VERSION (78 DOF) DATED 09/30/74	000000000
С	*****	00000070
C	APPLICABLE TO NONLINEAR NONCONSERVATIVE HYPERFLASTIC SYSTEMS	00000000
C	CST ELEMENT, 26 NODES. 78 DOF. 24 FLEMENTS, 20 LOAD INCREMENTS	00000090
C	5 MATERIALS, 2 OR 3 DOF PER NODE.	
С	NODES ARE LOCATED IN BASIC-CARTESIAN OR CYLINDRICAL COORDINATER	00000110
С	AND DISPLACEMENTS ARE IN BASIC CYLINDRICAL, OR SPECIAL CARTESTAN	00000120
	INTEGER UIN1, UIN2, UOUT, UINRS, UOUTRS	00000130
	INTEGER NOD, NEL .NN .NE .NS . (IRD . I . M. HIMP . N. HIMP	00000150
· .	INTEGER IPRESS, PRED. IPRED(20) . MAXUP, MAXIT, NMAT, IMAT(26)	00000150
	INTEGER NINCR.INCR.ISTEP.ITER.FIND13.241.KED(79) TDET	00000150
	DOUBLE PRECISION PEACT(2.20) PREF(2.78) PO(2) PI(2) PA(2)	00000170
	DOUBLE PRECISION PREACT(20) PRREF(24) PRO DP1 PPA	0810000
	DOUBLE PRECISION PP(78), CO(78), P(78), O(78), PC(78), PRO1781, UNING 701	00000190
	COUBLE PRECISION LSIGN.RI. RRI. PATH. FRR. EPPMAY. DET	00000200
	DOUBLE PRECISION LAM. LANS. LANR. FLANAY, LANIN, HINDE D. HIND SLODED	00000210
	DOUBLE PRECISION COORDA(5) F(5) NULS) CORES DEF	00000220
	DOUBLE PRECISION GCOS(9,26), COORD(3,26), KMAT(70, 70)	00000230
	DOUBLE PRECISION T(24) FGEOM(3, 24) FET(3, 24)	00000240
	COMMON/COMNEL/NEL/COMNS/NS/COMORD/ORD/CONDEE/DEE/COMPET/CDT	00000250
	COMMON/COMME/NE/COMMAT/IMAT/COMENU/E.NU/COMCC/CO	00000260
		00000270
		00000280
С	START PROBLEM	00000290
-	1 CALL READESIUTNI-HIN2-HOUT THERE HELENES	00000300
	CALL READOLS-HOHT. ORD. PRED. NAVID NAVIT ERRNAV DEE DEE	00000310
	18.11MPA.41MPR.SLOPED.ELAMAY.LAMIN	00000320
	FELUINES_GT_01G0_TO_11	0000330
C	COLD START	00000340
•	CALL REGELATION TODESS NE NHAT F NU	00000350
	ICCORDA.NOD.NEL.COOPD.CCOS.IMAT T FIND FCCOM MED DOGE DOGE	00000360
	21 STGN, PP, OG, P1, OPIN	00000370
	N.311MO = N.11MD+3	0000380
		00000390
c	READ RESTART TABE	00000400
U	11 CALL RIGS/2. HINRS, INCO. IDDECC NO AMAT D NO.	00000410
	1000RDA.NOD.NEL.COOPD COCK IMAT I SING FOROM WED DOWN	00000420
	ICCONTRACTION AND A CONTRACT I A CONTRACT I A CONTRACT I A CONTRACT	00000430

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR.

C-2

4

С

С

		,	I I	
ċ				
			2LSIGN.PP.CQ.P1.PR1)	00000440
			NJUMP = MJUMP+1	00000450
			16 IF(UOUTRS.EQ.0)GO TO 17	00000460
		C	WRITE RESTART TAPE	00000470
			CALL BIGS[3, UOUTRS, 0, IPRESS, NF, NMAT, E, NU,	0000480
			1COORDA,NOD,NEL,COORD,GCOS,IMAT,T,ELNO,EGEOM,KFD,PREF,PRREF,	00000490
			2LSIGN, PP, QQ, P1, PR1)	00000500
		C '	GENERAL PROGRAM FLOW	00000510
			17  NS = 3	00000520
			NN = NOD*NF	00000530
		C	READ INCREMENTAL LOAD FACTOR DATA	00000540
			CALL READI(5,UOUT,NINCR,PRED,IPRED,PFACT,PRFACT)	00000550
		C	BEGIN LOAD INCREMENT LOOP	00000560
			DU 1000 INCR=1,NINCR	00000570
		С	LAM = INCREMENTAL LOAD PARAMETER (MAXIMUM VALUE 1.0).	00000580
		C	LAMR = LOAD YET TO BE APPLIED = 1.0 - LAM.	00000590
		C	LAMS = LOAD STEP PARAMETER = FRACTION OF LAMR TO BE APPLIED.	00000600
		_	LAM = 0.00	00000510
e -		С	LSIGN = +, - FOR LOADING, UNLOADING SITUATION.	00000620
	ဂု	-	IFILSIGN.LI.O.DOISIUP 101	00000000
	ώ	L	PO(1),PI(1),PA(1) = LUAD FALTURS FUR LUAD REFERENCE VECTOR 1.	00000450
		۲. C	PRU, PRI, PRA = PRESSURE LUAU FACIURS FUR PRESSURE REFERENCE VECTOR	00000830
		L c	0,1 DENULE VALUES AT START,END OF INCREMENT. A DENOTED ACTUAL ADDITED VALUE WHICH AT THIS DOINT - O VALUE	00000670
		L	A DENULES ACTUAL APPLIED VALUE WATCH AT THIS POINT - O VALUE.	00000680
			$\frac{1}{2} \frac{1}{2} \frac{1}$	00000690
			PU(1) = PI(1) $PU(1) = PI(1)$	00000700
			$P_{1}(z) = PFAUT(z)z(UUU)$	00000710
			$\frac{200 \text{ PA(1)} \sim \text{PU(1)}}{000 \sim 001}$	00000720
			DD1 - DDEACTIINCD1	00000730
			$\rho p \Lambda = \rho p \Lambda$	00000740
			ISTEP = 0	00000750
		c	BEGIN LOAD STEP	00000760
		0	201  ISTEP = ISTEP+1	00000770
			NUP = <b>0</b>	0000780
			$LAMR \neq 1.00-LAM$	00000790
		C	SET UPPER BOUND FOR ABSOLUTE VALUE OF LOAD STEP SIZE LAMS.	008000
		••	LAMS = 1.00	00000810
			IF(LSIGN.LT.O)LAMS = FLAMAX/LAMR	0000082Q
		С	CALL PFORCE TO GIVE APPLIED CONSERVATIVE NODAL LOADS P.	00000830

	C	CALL EFORCE TO GIVE APPLIED NONCONSERVATIVE NORMAL LOADS T	
		CALL PEORCE(P), PREF, NN, P)	00000840
		CALL EFORCE(IPRESS, PRI, OPREE, OC) NEL NO. NEL SING ON	00000850
	C	COMPUTE LOAD STEP NODAL LOADS THESE ADDITED LOAD	00000860
	C	LOAD FOR SPECIFIED FORCE DOF ADDITED DISCHARTER AND	00000870
	C	DISPLACEMENT FOR SPECIFIED DISPLACEMENT - CURRENT	00000880
		DO 210 [=1.NN	00000890
		[F(KFD(I),GT,O)C] = P(I) + O(I) = PO(I)	00000900
		IF(KFD(1), T, O)C = P(T) = OO(T)	00000910
		210 P(I) = C	00000920
	C	FORM JACOBIAN AT BEGINNING OF LOAD STED	00000930
		CALL MERGE (KMAT. FIND. KED. NEL NN. NEL	00000940
		CALL DECOMPIEMAT.NN.KED.IDET DETN	00000950
	C	FUNDAMENTAL PATH PREDICTOR CODE	00000960
		IF(IPRED(INCR)_GF_2)GO_TO_241	00000970
	С	APPLY LINEAR PREDICTOR FOR LOAD STEP	00000980
		CALL SOLVE (KMAT_NN_KED.P.O)	00000990
		LAMS = LSTGN * LAMS	00001000
		DO 230 I=1.NN	00001010
Ģ.		$230 \ \Theta(1) = 1 \Delta MS \neq O(1)$	00001020
4		GC TO 301	00001030
	С	APPLY QUADRATIC PREDICTOR FOR LOAD STED	00001040
		241 CALL RATES (KMAT. PDUM. NN. KED. P. BPA. DP1   STCH DL DD1 DO DD1	00001050
		RJUMP = JUMPR/LAMR	00001060
		CALL STEPILSTON RI RRI NN. RO. DDO R HIND M HIND A HIND CLODER	00001070
		1PATH+LAMS)	0001080
	C	IF LIMIT POINT WAS TRAVERSED IT E LANS - ON OUTBUT LITTE - THE	00001090
		IF(LAMS.EQ.Q.DO)	00001100
		ICALL OUTLIMIG.NOD.NEL.NN.NE.ELNO ECERM FET CO. D. O. DET	00001110
		2RL, RRL+RQ, RRQ, LAM, LAMRI	00001120
	С	****	00001130
	С	****	¢COOO1140
	С	THIS SECTION OF CODE IS TEMPORY DOSTRUCKLING CUECKOUT OPEN	*00001150
		INTEGER BCODE.PCODE. IPOST. IIPOST	00001160
		DCUBLE PRECISION SCRITHICRITHINGTHINGTE CODOTLATON CONSTRAINTS	00001170
		1LPOST1, $LPOST2$ , $OOPOS11781$ , $OOPOS21781$ , $OOPOS21781$ , $OOOS21781$ , $OOOOS21781$ , $OOOOOS21781$ , $OOOOOS21781$ , $OOOOOS21781$ , $OOOOOOOS21781$ , $OOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO$	00001180
		DOUBLE PRECISION PDUM2(78), PDUM2(79), PDUM2(79)	00001190
	C	SET BCODE=1 TO GET NON INFAR FIGEN SOLUTION FOR DIFUSCATION	00001200
	C	ALSO SET PCODE=1 TO GET ADDITIONAL DOCTDUCK THE DATE OF THE	00001210
		BCODE = 0	00001220
			00001230

p

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

•

.

	PCODE = 1	00001240
	IF(BCODE.EQ.0)GO TO 279	00001250
	CALL EIGENIUOUT, KMAT, PDUM, Q, NN, KFD, PRA, PRI, RL, RRL, RQ, RRQ,	00001260
	25,15,1.D-2,1.D-5,SCRIT,QQPOS1,IPOST)	00001270
	CALL OUTPQ(UOUT,NOD,NF,QQPOS1,QQPOS1)	00001280
	LCRIT = RL*SCRIT + .5DO*RRL*SCRIT**2	00001290
	IF(PCODE.EQ.0)GD TO 279	00001300
	PERFORM POSTBUCKLING SOLUTION	00001310
	LCRIT = RL*SCRIT	00001320
	LDOT1 = RL	00001330
	LD0T2 = 0.D0	00001340
	DC 265 I=1,NN	00001350
	QCRIT(I) = QQ(I) + RQ(I) * SCRIT	00001360
	QQDOT1(1) = RQ(1)	00001370
265	QQD0T2(1) = 0.00	00001380
	IF(IPRED(INCR).LT.2)GO TO 271	00001390
	LCRIT = LCRIT + .5DO*RRL*SCRIT**2	00001400
	LDOT1 = LDOT1 + RRL*SCRIT	00001410
	LDOT2 = RRL	00001420
	DO 270 I=1,NN	00001430
	CCCRIT(I) = CQCRIT(I) + .5DO*RRQ(I)*SCRIT**2	00001440
	QQDOTI(I) = QQDOTI(I) + RRQ(I) * SCRIT	00001450
270	GQDOT2(I) = RRQ(I)	00001460
	CALL OUTPQ(UOUT,NOD,NF,QQDOT1,QQDOT2)	00001470
	DO 268 I=1,NN	00001480
	C = QQ(1)	00001490
	QQ(I) = QQCRIT(I)	00001500
268	(CCRIT(I) = C	00001510
271	CALL MERGE(KMAT,ELNO,KFD,NEL,NN,NF)	00001520
	DO 272 I=1,NN	00001530
	C = QQ(I)	00001540
	QQ(1) = QQCRIT(1)	00001550
272	QQCRIT(I) = C	00001560
	IIPOST = KFD(IPOST)	00001570
	KFD(IPOST) = -IPOST	00001580
	CALL DECOMP(KMAT,NN,KFD,IDET,DET)	00001590
	CALL PRATES(KMAT, PDUM, PDUM2, PDUM3, PDUM4, NN, IPUST, KFD, SCRIT,	00001600
	LCCCRIT,QQDOT1,QQDOT2,LCRIT,LDOT1,LDOT2,PR0,PRA,	00001610
ć	2QQPOS1,LPOST1,QQPOS2,LPOST2,QQPOS3)	00001620
	CALL OUTPQ(UOUT,NOD,NF,QQCRIT,QQPOS1)	00001630

**C-**5

C

}

.

			CALL OUTPQ(UOUT,NOD,NF,QQPOS2,QQPOS3)	00001660
	_		KFD(1POST) = IIPOST	00001450
	С		************	*00001650
	C		******************	*00001600
	C		PATH CONTINUATION CODE	00001690
		279	CONTINUE	00001680
			DO 280 I=1,NN	00001090
	_	280	Q(I) = PATH + RQ(I) + .5D0 + PATH + + 2 + RRQ(I)	00001700
	C		ACD STEP LOAD LAMS TO INCREMENT LOAD SUM LAM	00001720
		-	LAMS = LAMS*LAMR	00001720
		301	LAM = LAM + LAMS	00001750
		_	DO 305 I=1+2	00001750
		305	PA(I) = PO(I) + LAM + (P1(I) - PO(I))	00001750
			CALL PFORCE(PA, PREF, NN, P)	00001700
			DC 310 I=1+NN	00001790
			IF(KFD(1),GT,O)QQ(1) = QQ(1) + Q(1)	00001780
			$IF(KFD(1) \cdot LT \cdot O)QQ(1) = P(1)$	00001190
		310	CONTINUE	00001800
			$PRA = PRO + LAM \neq (PR1 - PRO)$	00001810
Ģ			ITER = $0$	00001820
φ.			GC TC 451	00001850
	C		BEGIN ITERATION LCOP	00001850
		311	ITER = ITER+1	00001850
			IF(ITER.GT.1)GO TO 401	00001880
~ H	C		FORM JACOBIAN AT APPROXIMATE END OF STEP	00001840
E			CALL MERGE(KMAT,ELNO,KFD,NEL,NN,NF)	00001890
GP			CALL DECOMP(KMAT, NN, KFD, IDET, DET)	00001890
ΖÖ		401	CALL SOLVE(KMAT,NN,KFD,P,Q)	00001900
AC	С		UPDATE INTERNAL FORCES PP AND DISPLACEMENTS CO.	00001910
20	С		COMPUTE APPLIED EXTERNAL LOADS (CONSERVATIVE P.NONCONSERVATIVE O)	00001920
PA BI			DU 410 I=1,NN	00001930
<u> 9</u> E		410	QQ(1) = QQ(1) + Q(1)	00001940
29		451	CALL FORCE(NEL, NN, NF, ELNO, DFF, QQ, PP)	00001950
ୂଇ			CALL PFORCE(PA, PREF, NN, P)	00001900
2 Y			CALL EFORCE(IPRESS, PRA, PRREF, QQ, NEL, NN, NF, FINO, O)	00001970
. Зн			DO 460 I=1,NN	00001980
変田			C = 0.00	00001330
			IF(KFD(1),GT,O)C = P(1) + Q(1) - PP(1)	00002000
•		460	P(1) = C	00002010
			CALL ERCOMP(UOUT,NN,KED, PP,P,ERR)	00002020
				VUUU2030

÷

C	END ITERATION LOOP	00002040
	IF(ERR.LE.ERRMAX)GO TO 901	00002050
	IF(ITER.LT.MAXITIGO TO 311	00002060
C	IF THIS POINT IS REACHED, MAX. NC. ITERATIONS OCCUR	00002070
	IF(NUP.GE.MAXUP)GO TO 901	00002080
	NUP = NUP+1	00002090
	ITER = $0$	00002100
	GO TO 311	00002110
C	OUTPUT INCREMENTAL STEP RESULTS	00002120
	901 CALL HEAD(UOUT, INCR, ISTEP, LAMS, LAM, PFACT(1, INCR), PFACT(2, INCR),	00002130
	1PRFACT(INCR), IPRED(INCR), MAXUP, NUP, MAXIT, ITER, ERRMAX, ERR)	00002140
	CALL OUTPQ(UOUT,NOD,NF,PP,GQ)	00002150
	CALL STRAIN(NEL,ELND,EGEOM,NF,QQ,EET)	00002160
	CALL OUTE(UOUT,NEL,EET)	00002170
C	END LOAD STEP	00002180
	IF(LAM.LT.LAMIN)GO TO 1	00002190
	IF(LAM.LT999D0)G0 TO 201	00002200
	IF(UOUTRS.EQ.0)GO TO 1000	00002210
С	WRITE INCREMENTAL RESTART TAPE	00002220
	CALL BIGS(3,UOUTRS,INCR, IPRESS,NF,NMAT,E,NU,	00002230
	1CCORDA,NOD,NEL,COORD,GCOS,IMAT,T,ELNO,EGEOM,KFD,PREF,PRREF,	00002240
	2LSIGN, PP, QQ, P1, PR1)	00002250
	LOOO CONTINUE	00002260
C	END LOAD INCREMENT LOOP	00002270
	GO TO 1	00002280
	END	00002 <b>290</b>

		SUBROUTINE BIGS(KODE, I1, I2, IPRESS, NF, NMAT, E, NU,	00002300
		1CCORDA,NOD+NEL,COORD.GCOS,IMAT,T,ELNO,EGEOM,KFD,PREF,PRREF,	00002310
		2LSIGN, PP, QQ, P1, PR1)	00002320
	C	KCDE = 1,2,3 = INITIALIZE, READ RESTART, WRITE RESTART.	00002330
	C	I1 = INPUT OR RESTART INPUT-OUTPUT FILE UNIT NUMBER.	00002340
	C	<pre>12 = OUTPUT FILE UNIT NUMBER, OR INCREMENT NUMBER FOR RESTART.</pre>	00002350
	C	IPRESS = NONCONSERVATIVE CODE = 0,1 FOR NO PRESSURE, PRESSURE.	00002360
	C	NF = NUMBER OF FREEDOMS PER NODE.	00002370
	C	NMAT = NUMBER OF NATERIALS.	00002380
-	C	E,NU = MATERIAL CONSTANTS (E.G. ELASTIC MODULUS, POISSON'S RATIO).	00002390
	C	CCORDA(I) = ANGLE FOR SPECIAL CCORDINATE SYSTEM I.	00002400
	С	NOD,NEL = NUMBER OF NODES,ELEMENTS.	00002410

C-7.

C       GCOS(J,I) = DIRECTION COSINES FOR NODE I.       GCO2430         C       IMAT(I) = MATERIAL NUMBER FOR ELEMENT I.       GCO2440         C       I(I) = THICKNESS OF ELEMENT I.       GCO2440         C       ELMC(J,I) = NODE NUMBERS FOR ELEMENT I.       GCO2440         C       EGEOM(J,I) = GEOMETRY FOR ELEMENT I (BASE,HEIGHT,PART BASE).       GCO2440         C       KFO(I) = FORCE-DISPLACEMENT-CONSTRAINT SPECIFICATION FUR DOF I.       GCO2440         C       PREF(J,I) = NODAL LOAD AT CUF I FOR LOAD REFERECNC VECTOR J.       GCO2490         C       PREF(I,I) = INTENSITY ON ELEMENT I FOR PRESSURE REFERENCE VECTOR J.       GCO2400         C       PREF(I) = INTENSITY ON ELEMENT I FOR PRESSURE REFERENCE VECTOR.       00002510         C       PREF(I) = LOAD PARAMETER IS INCREASING, DECREASING.       00002510         C       PREFIJ = CURRENT NODAL DISPLACEMENT AT DOF I.       00002530         C       PAI = LOAD FACTOR TO BE APPLIED TO PREF(I).       00002550         C       PAI = LOAD FACTOR TO BE APPLIED TO PREF(I).       00002550         C       PAI = LOAD FACTOR TO BE APPLIED TO PREF(I).       00002550         DUBLE PRECISION THO BE APPLIED TO PREF(I).       00002560         CUUBLE PRECISION THORS,UCUTRS).(J2.UOUT,INCR)       00002550         ITTIGER INS.S.VCUTRS).(J2.UOUT,INCR)       000022560	Ç	COORD(J+I) = COORDINATES OF NODE I.	00002420
C       IMAT(I) = MATERIAL NUMBER FOR ELEMENT I.       00002440         C       T(I) = THICKNESS OF ELEMENT I.       00002440         C       ELNO[J,I) # NODE NUMBERS FOR ELEMENT I.       00002440         C       EGEOM(J,I) = GEOMETRY FOR ELEMENT I.       00002460         C       EGEOM(J,I) = GEOMETRY FOR ELEMENT I.       (BASE,MEIGHT,PART BASE).       00002460         C       EGEOM(J,I) = GEOMETRY FOR ELEMENT I.       (BASE,MEIGHT,PART BASE).       00002460         C       EGEOM(J,I) = GEOMETRY FOR ELEMENT I.       (BASE,MEIGHT,PART BASE).       00002460         C       PREF(I) = NODAL LOAD AT OUF I.       OACC RESULT       00002510         C       PREF(I) = INTENSITY ON ELEMENT I TOR PRESSURE REFERENCE VECTOR.       00002510         C       QUIRENT INTERNAL NCDAL FORCE AT COF I.       00002530         C       QUIL) = CURRENT NODAL DISPLACEMENT AT DOF I.       00002540         C       PRI = LOAD FACTOR TO BE APPLIED TO PREF(1).       00002550         INTEGER KODE, II, T2, IPRESS,NF, NMAT,NCD,NEL, IMAT(1), ELCC(3,1), KFD(1)00002560       00002560         CUUVALENCE (J),UN,UINRS,UOUTRS).(J2,UOUT,INCR)       00002560         CUUBLE PRECISION THICK       00002560         CUUBLE PRECISION THICK       00002660         INTEGER INC, I, J,NN       00002660 <td< td=""><td>С</td><td>GCOS(J,I) = DIRECTION COSINES FOR NODE I.</td><td>00002420</td></td<>	С	GCOS(J,I) = DIRECTION COSINES FOR NODE I.	00002420
C       T(1) = THICKNESS OF ELEMENT I.       00002450         C       ELNO(J,I) = NODE NUMBERS FOR ELEMENT I.       00002450         C       EGEDM(J,I) = GORMETRY FOR ELEMENT I. (BASE, MEIGHT, PART BASE).       00002470         C       KFO(I) = FORCE-DISVLACEMENT-CONSTRAINT SPECIFICATION FOR DOF I.       00002480         C       PREF(J,I) = NODAL LOAD AT COF I FOR LOAC REFERECNE VECTOR J.       00002480         C       PREF(I) = INTENSITY ON ELEMENT I FOR PAESSURE REFERENCE VECTOR J.       00002510         C       LSIGN = +,- IF LOAD PARAMETER IS INCREASING, DECREASING.       00002510         C       Q(I) = CURRENT INTERNAL NODAL FORCE AT COF I.       00002530         C       Q(I) = CURRENT NODAL DISPLACEMENT AT DOF I.       00002550         C       PRI = LOAD FACTOR TO BE APPLIED TO PREF(I).I.       00002550         C DUBLE PRECISION CIO, BE APPLIED TO PREF(I).       00002550         C DUBLE PRECISION CIO, NURS, UDUTRS).4(J2, UDUT, INCR)       00002560         CUBLE PRECISION THICK       00002580         INTEGER IN., I, NN       00002610         COUBLE PRECISION THICK       00002610         CUBLE PRECISION THICK       00002650         INTEGER IN., ULIN, ULINRS, UCUTRS).4(J2, UCUT, INCR)       00002610         CUBLE PRECISION THICK       00002630         INTEGER ILI, UN, ULI	С.	IMAT(1) = MATERIAL NUMBER FOR ELEMENT 1.	00002430
C       ELN0(J,1) = NODE NUMBERS FOR ELEMENT 1.       00002460         C       EGEOM(J,1) = GEOMETRY FOR ELEMENT 1 (BASE, HEIGHT, PART BASE).       00002470         C       KPD(I) = FORCE-DISPLACEMENT-CONSTRAINT SPECIFICATION FOR DOF 1.       00002480         C       PREF(I) = NODAL LOAD AT OUF I FOR LOAC REFERECNE VECTOR J.       00002500         C       LSIGN = +,- IF LOAD PARAMETER IS INCREASING, DECREASING.       00002500         C       CQ(I) = CURRENT INTERNAL NCDAL FORCE AT DOF I.       00002500         C       QQ(I) = CURRENT INTERNAL NCDAL FORCE AT DOF I.       00002530         C       PQ(I) = CURRENT NODAL DISPLACEMENT AT DOF FI.       00002550         C       PQ(I) = CURRENT NODAL DISPLACEMENT AT DOF FI.       00002550         INTEGER KODE, IL, 12, IPRESS, NF, NMAT, NCD, NEL, IMAT(I), ELNC(3, I), KFD(I)0002560       00002550         COUBLE PRECISION E(1), NU(I), COORDA(1), CCORD 13, 1), GCCS (9, 1),       00002560         CUIVALENCE (J1, UIN, UINKS, UCUTRS, J2, UCUT, INCR       00002660         INTEGER MC, I, J, NN       00002660         CUUVALENCE (J1, UIN, UINKS, UCUTRS, J2, UCUT, INCR       00002660         CALL READC(UIN, UOUT, IPRESS, NF, THICK, NMAT, E, NU)       00002660         CALL READC(UIN, UOUT, NOD, NF, KFD)       00002660         CALL READCHUIN, UOUT, NOD, NF, KFD)       00002710         CALL	С	T(I) = THICKNESS OF ELEMENT 1.	00002440
C       EGEOM(J,1) = GEOMETRY FOR ELEMENT I (BASE, HEIGHT, PART BASE).       00002470         C       KFD(1) = FORCE-DISPLACEMENT-CONSTRAINT SPECIFICATION FOR DOF I.       00002480         C       PREF(J,1) = NODAL LOAD AT DOF I FOR LOAC REFERECNE VECTOR.       00002490         C       PREF(I) = INTENSITY ON ELEMENT I FOR PARESSURE REFERENCE VECTOR.       00002400         C       PREF(I) = INTENSITY ON ELEMENT I FOR PARESSURE REFERENCE VECTOR.       00002510         C       PP(1) = CURRENT INTERNAL NCDAL FORCE AT DOF I.       00002530         C       Q(1) = CURRENT INTERNAL NCDAL FORCE AT DOF I.       00002540         C       PRI = LOAD FACTOR TO BE APPLIED TO PREF(J).       00002540         DRI = LOAD FACTOR TO BE APPLIED TO PREF(I).       00002560         DCUBLE PRECISION E(1),NUNI).COORDA(1).CCORD(1).H.PLNC(3,1).KFD(1)COC02560         DCUBLE PRECISION E(1).NUNI).CONTACI).SIGN.PP(1).Q0(1).PI(1).PRI       00002570         LT(1).EGEOM(3,1).PREF(2,1).PRREF(1).SIGN.PP(1).Q0(1).PI(1).PRI       00002580         CUIVALENCE [J].UIN.JUNRS.UUTRS).JCJUUT.INCR       00002600         INTEGER INC.I.J.NN       00002600         INTEGER INC.I.J.NN       00002600         CUELE PRECISION THICK       00002600         J2 = 12       00002600         INTEGER INC.I.J.NN       00002600         CALL READCUUN.JUOT.FUCORDA <td>C</td> <td>ELNO(J+I) = NODE NUMBERS FOR ELEMENT T.</td> <td>00002450</td>	C	ELNO(J+I) = NODE NUMBERS FOR ELEMENT T.	00002450
C       KFD(1) = FORCE-DISPLACEMENT-CONSTRAINT SPECIFICATION FOR DOF 1.       00002480         C       PREF(J,I) = NODAL LOAD AT OUF I FOR LOAD REFERENCE VECTOR J.       CC002490         C       PREF(I) = INTENSITY ON LEMENT I FOR PLASSURE REFERENCE VECTOR J.       CC002490         C       LSIGN = +,- IF LOAD PARAMETER IS INCREASING, DECREASING.       00002510         C       LSIGN = +,- IF LOAD PARAMETER IS INCREASING, DECREASING.       00002500         C       VII = CURRENT INTERNAL NCDAL FORCE AT COF I.       00002530         C       P(1) = CURRENT NODAL DISPLACEMENT AT DOF I.       00002550         C       PI(J) = LOAD FACTOR TO BE APPLIED TO PREF(J,I).       00002550         INTEGER KODE, II, 12, IPRESS, NF, NMAT, NDD, NEL, IMAT(1), ELNC(3,1), KFD(1) D(002560       DOUBLE PRECISION E(1), NU(1), COORDA(1), CCORD(3,1), GCCS(9,1),       00002570         IT(I), EGEOM(3,1), PREF(2,1), PREF(1), LSIGN, PP(1), QQ(1), P1(1), PR1       C0002580       CO002600         INTEGER INC, I, J, NN       CC002600       INTEGER INC, I, J, NN       C0002660         CUUVALENCE (J1, UIN, UINRS, UCUTRS, J2, UCUT, INCR       C0002660       CO002660         IF(KODE, NE, I)GO TO 101       C0002660       CALL READI(UIN, UOUT, NDO, NE, KFD)       C0002660         C       INTEGER INALIZE VARIABLES       C0002660       CALL READI (UIN, UOUT, NOD, NE, KFD)       C00002710 <tr< td=""><td>C</td><td>EGEOM(J.I) = GEOMETRY FOR FLEMENT I (BASE HEIGHT BART BASE)</td><td>00002460</td></tr<>	C	EGEOM(J.I) = GEOMETRY FOR FLEMENT I (BASE HEIGHT BART BASE)	00002460
C       PREF(J,I) = NODAL LOAD AT OUF I FOR LOAD REFERENCE VECTOR.       00002490         C       PRREF(I) = INTENSITY ON ELEMENT I FOR PRESSURE REFERENCE VECTOR.       00002500         C       LSIGN = +, IF LOAD PARAMETER IS INCREASING, DECREASING.       00002510         C       PP(I) = CURRENT INTERNAL NEDAL FORCE AT COF I.       00002530         C       CQII) = CURRENT INTERNAL NEDAL FORCE AT COF I.       00002540         C       PQ(I) = CURRENT NDDAL DISPLACEMENT AT DOF I.       00002540         C       PILJ = LOAD FACTOR TO BE APPLIED TO PREF(J,I).       00002540         C       DAD FACTOR TO BE APPLIED TO PREF(I).       00002550         INTEGER KDDE, II, 12, IPRESS, NF, NMAT, NDD, NEL, IMAT(I), ELNC(3,1), KFD(I)00002560       00002560         C       CUBLE PRECISION F(1), PREF(1), LSIGN, PP(1), QG(1), PIL), PRI C0002580       00002560         FQUIVALENCE (J1, UIN, UINRS, UCUTRS, J2, UCUT, INCR)       00002660         INTEGER INC, I, J, NN       00002660         CUBLE PRECISION THICK       00002660         CALL READIUIN, UOUT, IPRESS, NF, THICK, NMAT, E, NU1       00002660         CALL READI (UIN, UOUT, IPRESS, NF, THICK, NMAT, E, NU1       00002660         CALL READT (UIN, UOUT, IPRESS, NF, THICK, NMAT, E, NU1       00002660         CALL READT (UIN, UOUT, NOD, NE, KFD)       00002710         CALL READT (UIN, UOU	) C	KFD(I) = FORCE-DISPLACEMENT-CONSTRAINT SPECIFICATION CON DOC 1	00002470
C       PRREF(I) = INTENSITY ON ELEMENT I FOR PRESSURE REFERENCE VECTOR. 00002500         C       LSIGN = +,- IF LOAD PARAMETER IS INCREASING. ECTOR. 00002510         C       P(I) = CURRENT INTERNAL NCDAL FORCE AT COF I. CC002520         C       CQ(I) = CURRENT NODAL DISPLACEMENT AT DOF I. C0002530         C       P(I) = CURRENT NODAL DISPLACEMENT AT DOF I. 00002530         C       P(I) = LOAD FACTOR TO BE APPLIED TO PREF(J). 00002540         DOUBLE PRECISION FIL: PRESS,NF,NMAT,NOD,NEL,IMAT(I),ELNC(3,1),KFD(I)C0002560         DOUBLE PRECISION E(1),NU(1),COORDA(1),CCORD(3,1),GCCS(9,1), 00002570         INTEGER KODE,II,I2,IPRESS,NF,NMAT,NOD,NEL,IMAT(I),ELNC(3,1),KFD(I)C0002560         DOUBLE PRECISION FE(1),SIGN,PP(1),Q(1),PI(1),PRI         ONO02560         CUIVALENCE (J),UIN,UINRS,UOUTRS),(J2,UOUT,INCR)         ONO02610         INTEGER INC,I,J,NN         OC002640         INTITALIZE VARIABLES         C         CALL READI(UIN,UOUT,IDRAL)         CALL READI(UIN,UOUT,NOD,NEL,COORDA,COORD,GCOS,         CM002640         IMAT,THECK,JL,RAMETER IS, ING,PREF)         CALL READP(UIN,UOUT,NOD,NF,KFD)         CALL READP(UIN,UOUT,NOD,NF,KFD)         CALL READP(UIN,UOUT,NOD,NF,KFD)         CALL READP(UIN,UOUT,NOD,NF,KFD)         CALL READP(UIN,UOUT,NOD,NF,KFD)         CO002710 <td>C</td> <td>PREF(J,I) = NODAL LOAD AT ONE I FOR LOAD DECERTORIE VECTOR</td> <td>00002480</td>	C	PREF(J,I) = NODAL LOAD AT ONE I FOR LOAD DECERTORIE VECTOR	00002480
C       LSIGN = +,- IF LOAD PARAMETER IS INCREASING, DECREASING, 00002510         C       PP(1) = CURRENT INTERNAL NEDAL FORCE AT DOF I.       CC002520         C       GQ(1) = CURRENT NODAL DISPLACEMENT AT DOF I.       00002530         C       P1(J) = LOAD FACTOR TO BE APPLIED TO PREF(J,I).       00002540         C       P1(J) = LOAD FACTOR TO BE APPLIED TO PREF(I).       00002550         INTEGER KODE, II, 12, IPRESS,NF,NMAT,NOD,NEL, IMAT(1),ELNC(3,1),KFD(1)00002570       00002570         DUBLE PRECISION E(1),NU(1),COORDA(1),CCCRD(3,1),GCCS(9,1),       00002570         ITTI),GECDM(3,1),PREF(2,1),PREF(1),LSIGN,PP(1),QQ(1),P1(1),PR1       C0002580         FCUIVALENCE (J,I,UIN,UINRS,UCUTRS,J2,UCUT,INCR)       00002610         CUBLE PRECISION THICK       00002610         DCUBLE PRECISION THICK       00002630         J1 = I1       00002640         J2 = 12       00002640         CALL READI(UIN,UOUT,IPRESS,NF,THICK,NMAT,E,NU)       00002640         CALL READG(UIN,UOUT,ORDA)       00002670         CALL READG(UIN,UOUT,NOD,NF,KFD)       00002710         CALL READG(UIN,UOUT,NOD,NF,KFD)       00002710         CALL READF(UIN,UOUT,NOD,NF,KFD)       00002710         CALL READF(UIN,UOUT,NOD,NF,KFD),PREF)       00002710         CALL READF(UIN,UOUT,NOD,NF,KFD),PREF)       000002710	C	PRREF(I) = INTENSITY ON FLEMENT I FOR DOESSURE DEFENSION WESTER	00002490
C       PP(1) = CURRENT INTERNAL NCDAL FORCE AT COF 1.       C0002510         C       QQ(1) = CURRENT NODAL DISPLACEMENT AT DOF 1.       00002530         C       P1(J) = LOAD FACTOR TO BE APPLIED TO PREF(J).       00002540         C       PR1 = LOAD FACTOR TO BE APPLIED TO PREF(J).       00002560         COUBLE PRECISION TO BE APPLIED TO PREF(I).       00002560         COUBLE PRECISION E(1).NU(1).COORDA(1).COORD(3)1).GCCS(9,1).       00002570         ITT1).EGEOM(3,1).PREF(2,1).PREF(2,1).FCCG(3,1).FCCG(3,1).FCD(1).       00002570         INTEGER J1.UIN.UINS.UCUTRS.J2.UUT.INCR       00002590         INTEGER INC.I.J.NN       00002600         INTEGER INC.I.J.NN       00002600         INTEGER INC.I.J.NN       00002650         CCUBLE PRECISION THICK       00002650         J2 = 12       00002650         CALL READ(UIN.UUT.FRESS.NF.THICK.NMAT.E.NU)       00002660         CALL READ(UIN.UOUT.FRESS.NF.THICK.NMAT.E.NU)       00002660         CALL READ(UIN.UOUT.NDD.NEL.COORDA.COORD.GCOS.       00002660         CALL READMUIN.UOUT.NDD.NEL.COORDA.COORD.GCOS.       00002600         IMTTALIZE VARIABLES       00002710         CALL READMUIN.UOUT.NDD.NE.COORDA.COORD.GCOS.       00002710         CALL READMUIN.UOUT.NDD.NE.COORDA.COORD.GCOS.       00002710         CALL READMUIN.UOUT.	C	LSIGN = + - IF LOAD PARAMETER IS INCREASING OFFICIALS	00002500
C       GQ(I) = CURRENT NODAL DISPLACEMENT AT DOF I.       CCC02520         C       PI(J) = LOAD FACTOR TO BE APPLIED TO PREF(J,I).       00002540         C       PRI = LOAD FACTOR TO BE APPLIED TO PREF(I).       00002550         INTEGER KODE,II,I2,IPRESS,NF,NMAT,NCO,NEL,IMAT(I),ELNC(3,I),KFD(I)00002560       00002570         DCUBLE PRECISION E(I),NU(I),COORDA(I),COCRD(3,I),GCCS(9,I),       00002570         IT(I),EGEOM(3,I),PREF(2,I),PRREF(I),LSIGN,PP(I),QQ(I),PI(I),PRI       CC002580         EQUIVALENCE (J),UIN,UINRS,UCUTRS,J2,UGUT,INCR)       00002610         CUBLE PRECISION THICK       00002610         CUBLE PRECISION THICK       00002640         INTEGER INC,I,J,NN       00002640         INTEGER INC,I,J,NN       00002640         CUBLE PRECISION THICK       00002640         J2 = I1       00002640         J2 = I2       00002640         IF(KODE.NE.1)GO TO 101       00002670         CALL READI(UIN,UOUT,IPRESS,NF,THICK,NMAT,E,NU)       00002670         CALL READI(UIN,UOUT,NDD,NEL,COORDA,COORD,GCOS,       00002680         CALL READK(UIN,UOUT,NOD,NF,KFD)       00002710         CALL READF(UIN,UOUT,NOD,NF,KFD,PREF)       00002710         CALL READP(UIN,UOUT,NOD,NF,KFD,PREF)       00002710         CALL READP(UIN,UOUT,NOD,NF,KFD,PREF)       00002770	C	PP(1) = CURRENT INTERNAL NODAL EDOCE AT DOC T	00002510
C       Pilj = LOAD FACTOR TO BE APPLIED TO PREF(J,I).       00002530         C       PRI = LOAD FACTOR TO BE APPLIED TO PREF(I).       00002540         INTEGER KODE, II, I2, IPRESS, NF, NMAT, NOD, NEL, IMAT(I), ELNC(3,1), KFD(I) 0C002560       0002540         DOUBLE PRECISION E(1), NU(1), COORDA(1), COORD(3,1), GCCS(9,1),       00002570         IT(I), EGEOM(3,1), PREF(2,1), PRREF(1), LSIGN, PP(1), QQ(1), PI(1), PRI       00002590         INTEGER JI, UIN, UINRS, UOUTRS, J2, UGUT, INCR)       00002610         INTEGER INC, I, J, NN       00002610         CCUBLE PRECISION THICK       00002610         DCUBLE PRECISION THICK       00002610         J2 = I2       00002660         IF(KODE.NE.1)GO TO 101       00002660         C       INITALIZE VARIABLES       00002660         CALL READC(UIN, UOUT, COORDA)       00002670         CALL READC(UIN, UOUT, COORDA)       00002670         CALL READK(UIN, UOUT, NDO, NE, KED)       00002710         CALL READK(UIN, UOUT, NOD, NE, KED)       00002710         CALL READK(UIN, UOUT, NOD, NE, KED, PREF)       000002710         DLSIGN = 1.DO	С	QQ(1) = CURRENT NODAL DISPLACEMENT AT DOG T	00002520
C       PR1 = LOAD FACTOR TO BE APPLIED TO PREF(1).       00002540         INTEGER KODE, II, I2, IPRESS, NF, NMAT, NCD, NEL, IMAT(1), ELNC(3,1), KFD(1)0002560       00002570         DUBLE PRECISION E(1), NU(1), COORDA(1), COORD(3,1), GCCS(9,1),       00002570         IT(1), EGEM(3,1), PREF(2,1), PREF(1), LSIGN, PP(1), QQ(1), P1(1), PR1       00002580         INTEGER JI, UIN, UINRS, UOUTRS, J2, UOUT, INCR)       00002610         INTEGER INC, I, J, NN       00002610         COUBLE PRECISION THICK       00002610         J1 = 11       00002630         J2 = 12       00002640         IF(KODE.NE.1)GO TO 101       00002640         CALL READI(UIN, UOUT, IPRESS, NF, THICK, NMAT, E, NU)       00002660         CALL READI(UIN, UOUT, NDD, NEL, COORDA, COORD, GCOS,       00002670         CALL READK (UIN, UOUT, NDD, NF, KFD)       00002710         CALL READK (UIN, UOUT, NDD, NF, KFD, PREF)       00002710         CALL READK (UIN, UOUT, NDD, NF, KFD, PREF)       00002710         CALL READP (UIN, UOUT, NDD, NF, KFD, PREF)       00002710         CALL READP (UIN, UOUT, NDD, NF, KFD, PREF)       00002710         CALL READP (UIN, UOUT, NDD, NF, KFD, PREF)       00002730         LSIGN = 1.00       00002740         NN = NOD*NF       00002750         CC 50 I = 1, NN       00002770 <t< td=""><td>Ċ</td><td>P1(1) = 10AD FACTOR TO BE ADDITED TO DESCLUTE</td><td>00002530</td></t<>	Ċ	P1(1) = 10AD FACTOR TO BE ADDITED TO DESCLUTE	00002530
INTEGER KODE, II, IZ, IPRESS, NF, NMAT, NOO, NEL, IMAT(I), ELNC(3,1), KFD(I)00002560         DOUBLE PRECISION E(1), NU(1), COORDA(1), CCORD(3,1), GCCS(9,1),       00002570         IT(1), EGEOM(3,1), PREF(2,1), PREF(1), LSIGN, PP(1), QQ(1), P1(1), PR1       00002580         EQUIVALENCE (J, U, UI, NUTRS, UOUTRS), (J2, UOUT, INCR)       00002610         DCUBLE PRECISION THICK       00002610         DCUBLE PRECISION THICK       00002610         J1 = I1       00002640         J2 = J2       00002640         IF(KODE-NE-1)GO TO 101       00002670         CALL READI(UIN, UOUT, IPRESS, NF, THICK, NMAT, E, NU)       00002670         CALL READI(UIN, UOUT, IPRESS, NF, THICK, NMAT, E, NU)       00002670         CALL READM(UIN, UOUT, NDD, NEL, COORDA, COORD, GCDS,       00002710         CALL READK (UIN, UOUT, NDD, NF, KFD)       00002710         CALL READP (UIN, UOUT, NDD, NF, KFD)       00002710         CALL READP (UIN, UOUT, ND, NF, KFD, PREF)       00002710         CALL READP (UIN, UOUT, ND, NF, KFD, PREF)       00002710         LSIGN = 1.DO       00002740         NN = NOD*NF       00002760         C50 50 1=1,NN       00002760         C60 01=1,2       00002760         C60 01=1,2       00002760         C60 01=1,2       00002760         C70 00002	С	PR1 = 10AD FACTOR TO BE ADDITED TO DEPERTUAL.	00002540
DUBLE PRECISION E(1), NU(1), COURDA(1), COURD(3,1), GECS(9,1),       00002570         LT(1), GEGM(3,1), PREF(2,1), PREF(1), LSIGN, PP(1), QU(1), P1(1), PR1       00002580         EQUIVALENCE (J1, UIN, UINRS, UCUTRS), (J2, UOUT, INCR)       00002590         INTEGER JI, UIN, UINRS, UCUTRS, J2, UOUT, INCR)       00002610         DCUBLE PRECISION THICK       00002620         J1 = 11       00002630         J2 = 12       00002640         IF(KODE.NE.1)GO TO 101       00002650         CALL READI (UIN, UOUT, S, NF, THICK, NMAT, E, NU)       00002660         CALL READI (UIN, UOUT, IPRES, NF, THICK, NMAT, E, NU)       00002670         CALL READC (UIN, UOUT, NOD, NEL, COORDA, COURD, GCOS,       00002670         CALL READK (UIN, UOUT, NOD, NF, KFD)       00002710         CALL READK (UIN, UOUT, NOD, NF, KFD, PREF)       00002710         CALL READF (UIN, UOUT, NOD, NF, KFD, PREF)       00002710         CALL READF (UIN, UOUT, NOD, NF, KFD, PREF)       00002710         LSIGN = 1, DO       00002740         NN = NOD*NF       00002740         DC 50 I=1, NN       00002770         CD GOL       00002770         CD GOL       00002770         CO GOL       00002770         DC GALL READP (UIN, UOUT, NOD, NF, KFD, PREF)       00002730         DC SO I=1,		INTEGER KODE, TI, TO, TODESS, NE NHAT NOD NEL THATTAN SUNDLE	00002550
IT(1), EGEON(3,1), PREF(2,1), PREF(1), ISIGN, PP(1), QQ(1), P1(1), PRI       00002570         EQUIVALENCE (J1, UIN, UINRS, UOUTRS), (J2, UOUT, INCR)       00002590         INTEGER J1, UIN, UINRS, UOUTRS, J2, UGUT, INCR       00002610         INTEGER INC, I, J, NN       00002610         CUUVALENCE (J1, UIN, UINRS, UOUTRS, J2, UGUT, INCR)       00002610         J1 = I1       00002630         J2 = I2       00002640         IF(KODE, NE, I)GO TO 101       00002650         CALL READI (UIN, UOUT, IPRESS, NF, THICK, NMAT, E, NU)       00002660         CALL READC (UIN, UOUT, COORDA)       00002670         CALL READC (UIN, UOUT, NOD, NEL, COORDA, COORD, GCOS,       00002690         I IMAT, THICK, T, ELNO, EGEUM)       00002710         CALL READK (UIN, UOUT, NOD, NF, KFD)       00002710         CALL READK (UIN, UOUT, NOD, NF, KFD, PREF)       00002710         CALL READK (UIN, UOUT, NOD, NF, KFD, PREF)       00002710         CALL READF (UIN, UOUT, NOD, NF, KFD, PREF)       00002730         LSIGN = 1.00       00002740         NN = NOD*NF       00002760         DC 50 I=1, NN       00002760         DC 50 I=1, NN       00002760         DC 60 I=1, 2       00002760         DC 60 I=1, 2       00002760         DC 60 I=1, 2       0000276		DEUBLE PRECISION E(1) NU(1) COORDA(1) COORDA(1) PRESIDENT	100002560
ECUIVALENCE (J1,UIN,UINRS,UOUTRS),(J2,UOUT,INCR)       00002590         INTEGER J1,UIN,UINRS,UOUTRS,J2,UOUT,INCR)       00002610         INTEGER INC,I,J,NN       00002610         CCUBLE PRECISION THICK       00002610         J2 = 11       00002630         J2 = 12       00002640         INITIALIZE VARIABLES       00002670         CALL READI(UIN,UOUT,IPRESS,NF,THICK,NMAT,E,NU)       00002670         CALL READI(UIN,UOUT,IOD,NEL,COORDA,COORD,GCOS,       00002690         IPAT,THICK,T,ELNO,EGEUM)       00002710         CALL READF(UIN,UOUT,NOD,NF,KFD)       00002710         CALL READF(UIN,UOUT,NOD,NF,KFD,PREF)       00002710         CALL READF(UIN,UOUT,NOD,NF,KFD,PREF)       00002740         NN = NOD*NF       00002760         DC 50 I=1,NN       00002760         PQ(I) = 0.00       00002770         CO 60 I=1,2       00002780         0C 60 I=1,2       00002780         0C 60 I=1,2       00002780         0C 60 I=1,2       00002780         0C 60 PR1 = 0.00       00002800		$[1(1)_{*} EGEOM(3, 3)_{*} PPEE/2 = 1)  \text{obscr}(3) = (510) \text{for } (5$	00002570
INTEGER JI, UIN, UINRS, UCUTRS, J2, UCUT, INCR)       00002590         INTEGER JI, UIN, UINRS, UCUTRS, J2, UCUT, INCR)       00002610         INTEGER INC, I, J, NN       00002610         CUBLE PRECISION THICK       00002620         J1 = 11       00002630         J2 = I2       00002640         INTIALIZE VARIABLES       00002670         CALL READI (UIN, UOUT, FRESS, NF, THICK, NMAT, E, NU)       00002670         CALL READK (UIN, UOUT, COORDA)       00002670         CALL READK (UIN, UOUT, NOD, NE, COORDA, COORD, GCOS,       00002710         CALL READK (UIN, UOUT, NOD, NF, KFD)       00002710         CALL READK (UIN, UOUT, NOD, NF, KFD)       00002710         CALL READK (UIN, UOUT, NOD, NF, KFD, PREF)       00002710         CALL READK (UIN, UOUT, NOD, NF, KFD, PREF)       00002730         LSIGN = 1.00       00002740         NN = NOD*NF       00002760         CC 50 I=1, NN       00002760         PP(I) = 0.00       00002760         OC 60 I=1, 2       00002760         OC 60 I=1, 2       00002770         OO PR1 = 0.00       00002800		ECUIVALENCE (1), HIN HINDS HOUTDON (1), HOUT THEN	00002580
INTEGER INC,I.J,NN       G0002600         INTEGER INC,I.J,NN       00002610         CUBLE PRECISION THICK       G0002620         J1 = I1       00002640         J2 = I2       00002640         INTEGER INC,I.J,OOTO 101       00002640         C       INTIALIZE VARIABLES       00002640         CALL READI(UIN,UOUT,IPRESS,NF,THICK,NMAT,E,NU)       00002670         CALL READIUIN,UOUT,OORDA)       00002670         CALL READM(UIN,UOUT,NOD,NEL,COORDA,COORD,GCOS,       00002700         CALL READM(UIN,UOUT,NOD,NF,KFD)       00002710         CALL READP(UIN,UOUT,NOD,NF,KFD,PREF)       00002710         CALL READP(UIN,UOUT,NOD,NF,KFD,PREF)       00002720         IF(IPRESS.GT.O)CALL READPR(UIN,UOUT,NEL,PRREF)       00002730         LSIGN = 1.DO       00002750         NN = NOD*NF       00002750         CC 50 I=1,NN       00002750         PP(1) = 0.DO       00002760         PP(1) = 0.DO       00002760         PP(1) = 0.DO       00002760         PR1 = 0.DO       00002800		INTEGER 11 UIN UINES UCUIRS 10 UCUI INCR)	00002590
CLUBLE PRECISION THICK       00002610         CLUBLE PRECISION THICK       00002640         J1 = I1       00002640         J2 = I2       00002640         INITIALIZE VARIABLES       00002650         CALL READI(UIN,UOUT,IPRESS,NF,THICK,NMAT,E,NU)       00002670         CALL READ(UIN,UOUT,COORDA)       00002680         CALL READM(UIN,UOUT,NOD,NEL,COORDA,COORD,GCDS,       00002680         CALL READM(UIN,UOUT,NOD,NEL,COORDA,COORD,GCDS,       00002700         CALL READK(UIN,UOUT,NOD,NF,KFD)       00002710         CALL READK(UIN,UOUT,NOD,NF,KFD)       00002710         CALL READP(UIN,UOUT,NOD,NF,KFD,PREF)       00002710         LSIGN = 1.D0       00002750         NN = NDD*NF       00002750         CC 50 I=1,NN       00002760         PP411 = 0.D0       00002760         OU 60 I =1,2       00002780         OU 60 I =1,2       00002780         OU 60 I =1,2       00002780         OU 60 P1(1) = 0.D0       00002780         OU 6002780       00002780         OU 60028800       000028800		INTEGER INC. T I AM	00002600
J1 = I1       00002630         J2 = I2       00002640         IF(KODE.NE.1)GO TO 101       00002640         C       INITIALIZE VARIABLES       00002660         CALL READI (UIN, UOUT, IPRESS, NF, THICK, NMAT, E, NU)       00002670         CALL READC (UIN, UOUT, IORDA)       00002680         CALL READK (UIN, UOUT, NOD, NEL, COORDA, COORD, GCOS,       00002690         IMAT, THICK, T, ELNO, EGEUM)       00002700         CALL READK (UIN, UOUT, NOD, NF, KFD)       00002710         CALL READP (UIN, UOUT, NOD, NF, KFD, PREF)       00002710         CALL READP (UIN, UOUT, NOD, NF, KFD, PREF)       00002710         LSIGN = 1.00       00002740         NN = NOD*NF       00002750         DC 50 I=1,NN       00002760         PP(I) = 0.00       00002770         DC 60 I=1,2       00002770         00 0002780       00002770         DC 60 I=1,2       00002770         00 PR1 = 0.00       00002810		PEURIE DECISION TOTAV	00002610
J2 = 12       00002630         J2 = 12       00002640         IF(KODE.NE.1)GO TO 101       00002650         INITIALIZE VARIABLES       00002660         CALL READI(UIN,UOUT,IPRESS,NF,THICK,NMAT,E,NU)       00002670         CALL READC(UIN,UOUT,COORDA)       00002680         CALL READM(UIN,UOUT,NDD,NEL,COORDA,COORD,GCOS,       00002690         IMAT,THICK,T,ELNO,EGEUM)       00002710         CALL READM(UIN,UOUT,NOD,NF,KFD)       00002710         CALL READP(UIN,UOUT,NOD,NF,KFD,PREF)       00002710         IF(IPRESS.GT.0)CALL READPR(UIN,UOUT,NEL,PRREF)       00002730         LSIGN = 1.D0       00002740         NN = NDD*NF       00002760         DC 50 I=1,NN       00002760         PP(I) = 0.D0       00002770         DC 60 I=1,2       00002770         C0 Q(I) = 0.D0       00002780         DC 60 PI(I) = 0.D0       00002780         DC 60 PI(I) = 0.D0       00002780         D0 PI(I) = 0.D0       00002800         PR1 = 0.D0       00002810		1.11 = 11	00002620
IF(KODE.NE.1)GO TO 101       00002640         IF(KODE.NE.1)GO TO 101       00002650         C       INITIALIZE VARIABLES       00002660         CALL READ1(UIN, UDUT, IPRESS,NF,THICK,NMAT,E,NU)       00002670         CALL READC(UIN,UDUT,COORDA)       00002680         CALL READM(UIN,UDUT,NOD,NEL,COORDA,COORD,GCOS,       00002690         CALL READK(UIN,UOUT,NOD,NEL,COORDA,COORD,GCOS,       00002700         CALL READK(UIN,UOUT,NOD,NF,KFD)       00002710         CALL READP(UIN,UOUT,NOD,NF,KFD,PREF)       00002710         CALL READP(UIN,UOUT,NOD,NF,KFD,PREF)       00002730         LSIGN = 1.00       00002740         NN = NDD*NF       00002750         DC SO I=1,NN       00002760         PP(I) = 0.00       00002760         DC 60 I=1,2       00002750         OC 0002760       00002760         PO 00       00002760         PO 00       00002760         PO 00       00002750         DC 50 I=1,NN       00002760         OC 60 I=1,2       00002770         50 QQ(I) = 0.00       00002780         OC 0002800       00002810		12 - 12	00002630
C       INITIALIZE VARIABLES       00002650         CALL READI(UIN, UOUT, IPRESS,NF,THICK,NMAT,E,NU)       00002670         CALL READI(UIN,UOUT,COORDA)       00002680         CALL READM(UIN,UOUT,COORDA)       00002690         CALL READM(UIN,UOUT,NOD,NEL,COORDA,COORD,GCOS,       00002700         CALL READK(UIN,UOUT,NOD,NF,KFD)       00002710         CALL READK(UIN,UOUT,NOD,NF,KFD,PREF)       00002720         IF(IPRESS.GT.0)CALL READPR(UIN,UOUT,NEL,PRREF)       00002730         LSIGN = 1.DO       00002760         NN = NDD*NF       00002760         DC 50 I=1,NN       00002760         PP(I) = 0.DO       00002780         DC 60 I=1,2       00002790         60 P1(I) = 0.DO       0002800         PR1 = 0.DO       00002810		92 - 12 IE/KODE NE 1100 TO 201	00002640
C       INTITACIZE VARIABLES       C0002660         CALL READI(UIN,UOUT, IPRESS,NF,THICK,NMAT,E,NU)       00002670         CALL READC(UIN,UOUT,COORDA)       00002680         CALL READM(UIN,UOUT,NOD,NEL,COORDA,COORD,GCOS,       00002690         IPAT,THICK,T,ELNO,EGEUM)       00002710         CALL READK(UIN,UOUT,NOD,NF,KFD)       00002710         CALL READP(UIN,UOUT,NOD,NF,KFD,PREF)       00002720         IF(IPRESS.GT.0)CALL READPR(UIN,UOUT,NEL,PRREF)       00002730         LSIGN = 1.D0       00002750         NN = NDD*NF       00002760         PP(I) = 0.D0       00002760         D0       00002780         00       00002780         00       00002780         00       00002780         00       00002790         00       00002780         00       00002790         00       00002780         00       00002790         00       00002800         00002810       00002810	ſ	IT INDEGNEGIJOU TU IUI	00002650
CALL READ(UIN, UOUT, IPRESS, NF, THICK, NMAT, E, NU)       00002670         CALL READC(UIN, UOUT, COORDA)       00002680         CALL READM(UIN, UOUT, NOD, NEL, COORDA, COORD, GCOS,       00002690         IIMAT, THICK, T, ELNO, EGEUM)       00002700         CALL READK(UIN, UOUT, NOD, NF, KFD)       00002710         CALL READF(UIN, UOUT, NOD, NF, KFD)       00002720         IF(IPRESS, GT.O)CALL READPR(UIN, UOUT, NEL, PRREF)       00002730         LSIGN = 1.DO       00002740         NN = NDD*NF       00002750         DC 50 I=1, NN       00002760         PP(I) = 0.DO       00002770         D0 0002780       00002770         D0 0002780       00002770         D0 PR1 = 0.DO       00002800	C	INTITALIZE VAKIABLES	00002660
CALL READC(01N,0001,C00RDA)       00002680         CALL READM(U1N,U0UT,N00,NEL,C00RDA,C00RD,GC0S,       00002690         11MAT,THICK,T,ELN0,EGE0M)       00002700         CALL READK(U1N,U0UT,N00,NF,KFD)       00002710         CALL READP(U1N,U0UT,N00,NF,KFD,PREF)       00002720         IF(IPRESS.GT.0)CALL READPR(U1N,U0UT,NEL,PRREF)       00002730         LSIGN = 1.00       00002760         NN = N0D*NF       00002760         DC 50 I=1,NN       00002760         PP(I) = 0.00       00002780         DU 60 I=1,2       00002790         60 P1(I) = 0.00       00002800         PR1 = 0.00       00002810		CALL READITUTIN, UUUT, IPRESS, NF, THICK, NMAT, E, NU)	00002670
CALL READMIDIN, UDUT, NUD, NEL, COORDA, COORD, GCOS,       00002690         1 IMAT, THICK, T, ELNO, EGEOM)       00002700         CALL READK (UIN, UOUT, NOD, NF, KFD)       00002710         CALL READP (UIN, UOUT, NOD, NF, KFD)       00002720         If(IPRESS.GT.O)CALL READPR(UIN, UOUT, NEL, PRREF)       00002730         LSIGN = 1.D0       00002740         NN = NDD*NF       00002750         DC 50 I=1,NN       00002760         PP(I) = 0.D0       00002770         50 QQ(I) = 0.D0       00002780         00002790       00002790         60 P1(I) = 0.D0       00002800         PR1 = 0.D0       00002810		CALL READCIDIN, UUUT, CUURDA)	00002680
CALL READK (UIN, UOUT, NOD, NF, KFD)       00002710         CALL READP (UIN, UOUT, NOD, NF, KFD, PREF)       00002720         IF(IPRESS.GT.0)CALL READPR(UIN, UOUT, NEL, PRREF)       00002730         LSIGN = 1.D0       00002740         NN = NOD*NF       00002750         DC 50 I=1, NN       00002760         PP(I) = 0.D0       00002780         DC 60 I=1, 2       00002790         60 P1(I) = 0.D0       00002800         PR1 = 0.D0       00002810		LALL READMIDIN+UUUI+NUU+NEL+COORDA+COORD+GCOS+	00002690
CALL READR(UIN, UUU1, NUD, NF, KFD)       00002710         CALL READP(UIN, UUUT, NDD, NF, KFD, PREF)       00002720         IF(IPRESS.GT.0)CALL READPR(UIN, UUUT, NEL, PRREF)       00002730         LSIGN = 1.D0       00002740         NN = NDD*NF       00002750         DC 50 I=1,NN       00002760         PP(I) = 0.D0       00002770         D0 60 I=1,2       00002780         60 P1(I) = 0.D0       00002800         PR1 = 0.D0       00002810		ALL SCARKING HOUT NOD AT WEEK	00002700
CALL READPIOIN, OUUT, NUU, NF, KFD, PREF)       00002720         IF(IPRESS.GT.0)CALL READPR(UIN, UOUT, NEL, PRREF)       00002730         LSIGN = 1.D0       00002740         NN = NOD*NF       00002750         DC 50 I=1, NN       00002760         PP(I) = 0.D0       00002770         D0 60 I=1, 2       00002780         60 P1(I) = 0.D0       00002790         PR1 = 0.D0       00002800		CALL READAIUTA, UUUT, NUD, NF, KFD)	00002710
IF(IPRESS.GI.0)CALL READPR(UIN,UOUT,NEL,PRREF)       00002730         LSIGN = 1.D0       00002740         NN = NDD*NF       00002750         DC 50 I=1,NN       00002760         PP(I) = 0.D0       00002770         50 QQ(I) = 0.D0       00002780         D0 60 I=1,2       00002790         60 P1(I) = 0.D0       00002800         PR1 = 0.D0       00002810		CALL READPIDIN, UUUT, NUU, NF, KFD, PREF)	00002720
LSIGN = 1.00       00002740         NN = NDD*NF       00002750         DC 50 I=1,NN       00002760         PP(I) = 0.00       00002760         50 QQ(I) = 0.00       00002780         00 60 I=1,2       00002780         60 P1(I) = 0.00       00002800         PR1 = 0.00       00002810		IFTIPRESS+GI+OJCALL READPR(UIN+UOUT+NEL+PRREF)	00002730
NN = NDU*NF       00002750         GC 50 I=1,NN       00002760         PP(I) = 0.00       00002770         50 QQ(I) = 0.00       00002780         DC 60 I=1,2       00002790         60 P1(I) = 0.00       00002800         PR1 = 0.00       00002810		LSIGN = 1.00	00002740
DC 50 I=L,NN       00002760         PP(I) = 0.00       00002770         50 QQ(I) = 0.00       00002780         00 60 I=1,2       00002790         60 P1(I) = 0.00       00002800         PR1 = 0.00       00002810		$NN = NUU \neq NF$	00002750
$\begin{array}{llllllllllllllllllllllllllllllllllll$		UU 50 I=L;NN	00002760
$\begin{array}{rcl} 50 & QQ(1) &= 0.00 \\ 00 & 60 & I = 1,2 \\ 60 & P1(1) &= 0.00 \\ PR1 &= 0.00 \\ PR1 &= 0.00 \end{array} \qquad \begin{array}{r} 00002780 \\ 00002800 \\ 00002810 \end{array}$		PP(1) = 0.00	00002770
00 60 1=1,2         00002790           60 P1(1) = 0.00         00002800           PR1 = 0.00         00002810		20 We(F) = 0.00	00002780
80 P1(1) = 0.00 PR1 = 0.00 00002800 00002810		$00.60 \ 1=1+2$	00002790
PRI = 0.00 00002810		00  PI(1) = 0.00	00002800
		HKI = 0.00	00002810

•

C-8

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

.

	RETURN	00002820
	101 IF(KODE.NE.2)GO TO 201	00002830
C	READ VARIABLES FROM RESTART TAPE	00002840
	READ(UINRS)IPRESS,NF,NMAT,(E(I),NU(I),I=1,NMAT),	00002850
	1NOD,NEL,((COORD(J,I),J=1,3),(GCOS(J,I),J=1,9),I=1,NOD),	00002860
	2(IMAT(1),T(1),(ELNO(J,1),EGEOM(J,1),J=1,3),PRREF(1),I=1,NEL)	00002870
	NN = NOD*NF	00002880
	READ(UINRS)(KFD(I),(PREF(J,I),J=1,2),I=1,NN)	00002890
	151 READ(UINRS)INC	00002900
	IF(INC.EQ.INCR)GD TO 161	00002910
	READ(UINRS)	00002920
	GC TO 151	00002930
	161 READ(UINRS)LSIGN, (PP(I), CQ(1), I=1, NN), (P1(I), I=1, 2), PR1	00002940
	REWIND UINRS	00002950
	RETURN	00002960
	201 IF(KODE.NE.3)RETURN	00002970
С	WRITE VARIABLES ONTO RESTART TAPE	00002980
	IF(INCR.GT.O)GO TO 251	00002990
	WRITE(UOUTRS)IPRESS,NF,NMAT,(E(I),NU(I),I=1,NMAT),	00003000
	<pre>INCD,NEL,((COORD(J,I),J=1,3),(GCOS(J,I),J=1,9),I=1,NOD),</pre>	00003010
	2(IMAT(I),T(I),(ELNO(J,I),EGEOM(J,I),J=1,3),PRREF(I),I=1,NEL)	00003020
	NN = NOD+NF	00003030
	wRITE(UOUTRS)(KFD(I),(PREF(J,I),J=1,2),I=1,NN)	00003040
	251 WRITE(UUUTRS)INCR	00003050
	WRITE(UOUTRS)LSIGN,(PP(I),CQ(I),I=1,NN),(P1(I),I=1,2),PR1	00003060
	RETURN	00003070
	END	00003080

.

		SUBROUTINE READRS(UIN1,UIN2,UOUT,INCR,UINRS,UOUTRS)	00003090
С		READ DATA FILE NUMBERS AND START-RESTART CODES.	00003100
С		UIN1,UIN2 = FILE UNIT NUMBER FOR INPUT DATA TYPE I,II.	00003110
С		UOUT = FILE UNIT NUMBER FOR OUTPUT DATA.	00003120
С		INCR = LOAD INCREMENT NUMBER FROM END OF WHICH RESTART IS MADE.	00003130
С		UINRS = INPUT RESTART TAPE UNIT NUMBER.	00003140
С		UCUTRS = OUTPUT RESTART TAPE UNIT NUMBER.	00003150
		INTEGER UIN1, UIN2, UOUT, INCR, UINRS, UOUTRS	00003160
		INTEGER START, STAR, REST	00003170
	101	FORMAT(A4,6X,6I5)	00003180
	201	FORMAT(1H1, STARTING PROBLEM)	00003190

C-9

,

.

202 FORMAT(1H1, RESTARTING PROBLEM FROM END OF LOAD INCREM	ENT . 15) 00003200
UATA STAR/*STAR*/+REST/*REST*/	00003210
REAULS, IOI)START, UIN1, UIN2, UOUT, INCR, UINRS, UOUTRS	00003220
IF(START.NE.STAR.AND.START.NE.REST)STAP 9999	00003220
IF(UIN2.LE.O.OR.UOUT.LE.O)STOP 101	00003230
IF(START-NE-STAR)GO TO 21	00003240
C COLD START REDUESTED	00003250
	00003260
	00003270
	00003280
	00003290
KETURN	00003200
C RESTART REQUESTED	00003300
21 IF(UINRS.LE.O)STOP 101	00003310
WRITE(6.202)INCR	00003320
RETURN	00003330
END	00003340
	00003350

• .

.

	SUBROUTINE READOLUL.UD.ORD. PRED. MAXUD. NAVIT FORMAN DEF. DEF.	
	INJUMP.JUMPR.SI OPED.ELAWAY, LAWING AND THANTI FERRMARY UPE, UPE, UPE,	00003360
C	READ PROBLEM IDENTIFICATION AND INCOMMENTAL AND	00003370
Ċ	ULID = INPUT OUTOUT THE WAY AND INCREMENTAL-ITERATIVE CONSTANTS.	00003380
č	$\Omega RD = MAYENUM TENSOD CODED TO DE VOTE$	00003390
ē	PRED = DEFAULT SOLUTION OPERIODE STRESS-STRAIN EXPANSION	1.00003400
ř	MAYUR - MANINUM ANNO FR. OF AND ANNO FR.	00003410
r	HAAUF = MAXIMUM NUMBER OF JACOBIAN UPDATES PER LOAD STEP.	00003420
Ċ	FRANCE MAXIMUM NUMBER OF RESIDUAL-LOAD ITERATIONS PER UPDATE.	00003430
с с	ERRMAX = MAXIMUM ALLOWABLE ERROR NORM.	00003440
C C	CFEIDFF = FINITE DIFFERENCE STEP SIZES TO BE USED IN COMPUTING	00003450
r L	STRESS-STRAIN TENSORS, FORCES.	00003430
C	MJUMP = NUMBER OF INCREMENT DIVISIONS TO PERFORM WHEN NEARING A	00003460
C	LIMIT POINT.	00003470
C	JUMPR = FRACTION OF LOAD INCREMENT PRECEDING LINIT POINT	00003480
C	AT WHICH LIMIT IS TO BE TRAVERSED	00003490
C	SLOPED = MAXIMUM SLOPE RATIO (CHANCE AVERACE) DURING LODG AND	00003500
С	FLAMAX = MAXIMUM FRACTION OF LOAD INCOUNTER TO DE TUDAN STEP.	00003510
C	NEGATIVE LOADING LAFTER MAYTHIN LINIT OCTUTE	00003520
č	LAMIN # MINIMUM INFGATIVES EDACTION OF STATES	00003530
č	ANALYSIS IS TERMINATED LAFTED WALLAUN OF LOAD INCREMENT AT WHICH	00003540
C	INTECED UT NO ODD NOTO MANY PRAVILIMIT POINT).	00003550
	DOUBLE DECLASSON FRANCE MAXUP, MAXIT, MJUMP	00003560
•	DUDDLE PRECISION ERRMAX, DEE, DEE, JUMPR, SLOPED, FLAMAX, LAMIN	00003570

.

C-10

INTEGER I, BLANK, IDENT(20)	00003580
101 FORMAT(20A4)	00003590
102 FORMAT(4110,F10.0)	00003600
103 FORMAT(2F10.0)	00003610
104 FORMAT(110,4F10.0)	00003620
201 FORMAT(1H0,20A4)	00003630
202 FORMAT(/IH , TENSOR ORDER = , 15/1H , PREDICTOR TYPE = , 15/	00003640
11H , MAXIMUM JACOBIAN UPDATES PER STEP = 1,15/	00003650
21H , MAXIMUM RESIDUAL LOAD CORRECTIVE ITERATIONS = 1,15/	00003660
31H , MAXIMUM ERROR NORM = ', E12, 5)	00003670
203 FORMAT(/1H , DFE = + E12.5/1H , DFF = + E12.5)	00003680
204 FORMAT(/1H , MJUMP = , 15/1H , JUMPR = , E12.5/	00003690
11H ,*SLOPED =*,E12.5/1H ,*FLAMAX =*,E12.5/1H ,*LAMIN =*,E12.5)	00003700
DATA BLANK/ 1/	00003710
READ(UI,101)(IDENT(I),I=1,20)	00003720
WRITE(U0,201)(IDENT(I),I=1,20)	00003730
READ(UI,102)ORD, PRED, MAXUP, MAXIT, ERRMAX	00003740
$IF(ORD \cdot EQ \cdot O)ORD = 3$	00003750
$IF(PRED_EQ_0)PRED = 2$	00003760
$IF(MAXUP_EQ_0)MAXUP = 0$	00003770
IF(MAXIT-EQ-0)MAXIT = 5	00003780
$IF(ERRMAX \cdot EQ \cdot O \cdot DO) ERRMAX = I \cdot D - 8$	00003790
WRITE(UU, 202)ORD, PRED, MAXUP, MAXIT, ERRMAX	00003800
REAU(U1,103)DFE,DFF	00003810
$IF(DFE \cdot EQ \cdot O \cdot DO)DFE = 1 \cdot D + 3$	00003820
IF(UFF+EQ+0+D0)DFF = 1+D-8	00003830
WRITE(U0,203)DFE,DFF	00003840
READ(UI, 104) MJUMP, JUMPR, SLOPED, FLAMAX, LAMIN	00003850
$IF(MJUMP \cdot EQ \cdot 0) MJUMP = 3$	00003860
$IFIJUMPR \cdot EQ \cdot 0 \cdot 00) JUMPR = 0 \cdot 100$	00003870
IF(SLUPED = 0.500	000038 <b>80</b>
IF(FLAMAX + EQ + 0 + 00) FLAMAX = 1 + D0	00003890
EFILAMIN-EU-UU)LAMIN = 0.00	00003900
WRITE(UU+2U4)MJUMP+JUMPR+SLUPED+FLAMAX+LAMIN	00003910
KETUKN TAD	00003920
END	00003930
SUBROUTINE READI(UI, UO, IPRESS, NF. THICK. NMAT. F. NU)	00003940
READ BASIC CODES AND CONSTANTS.	00003950

c-11

С

00003950

ı.

1

C		UI.UD = INPUT.DUTPUT FILE UNIT NUMBERS.	00000000
C		IPRESS = NONCONSERVATIVE CODE = $0.1$ EOP NO DRESSURE DRESSURE	00003980
Ċ		NE = NUMBER OF ERFEDOMS DER NODE	00003970 -
C		THICK = DEFAULT ELEMENT THICKNESS	00003980
Ē		NMAT = NHMAER OF MATCHIALC	00003990
ř		$F_{\rm A}NH = MATERIAL CONSTANTS (C. C. ELACTIC MODULUS DOLCOONAS DETECT$	00004000
Ŭ		INTEGER HI HO IDDESS NE AMAT	00004010
		THILDER UIDUDIPRESSANDANDAN DCHRIE DDECTETON THTER CALL MAAN	00004020
		DOUBLE PRECISION THICK, E(1), NU(1)	00004030
		DUUBLE PRECISIUN CI,C2	00004040
	101	INIEGEK MINI	00004050
	101	FURMA1(2110,F10.0)	00004060
	102	FURMAILI10,2F10,0}	00004070
	201	FURMATIIHO, PRESSURE CODE =', 15/1H , DOF PER NODE =', 15	00004080
		1/1H , DEFAULT THICKNESS = +, E12.5)	00004090
	202	FORMAT(/1H , 'MATERIAL CONSTANT CONSTANT')	00004100
	203	FORMAT(1H +15+5X+E12.5,1X,E12.5)	00004110
	•	READ(UI, 101) IPRESS, NF, THICK	00004120
		$IF(NF \cdot EQ \cdot O)NF = 3$	00004130
		IF(THICK.EQ.0.DO)THICK = 1.DO	00004140
		WRITE(U0,201)IPRESS,NF,THICK	00004150
		WRITE(U0,202)	00004160
		NMAT = 0	00004170
	40	READ(UI, 102) I, C1, C2	00004190
		IF(I.LE.O)RETURN	00004100
		WRITE(U0,203)1,C1,C2	00004190
		IF(I.LE.NMAT)GO TO 51	00004200
		N1 = NMAT+1	00004210
		D0 45 M=N1.I	00004220
		E(M) = 0.00	00004230
	45	NU(M) = 0.00	00004240
	-	NMAT = I	00004250
	51	F(1) = C1	00004260
		NH( $1 = C^2$	00004270
		60 TO 40	00004280
í			00004290
3			00004300
1	·		
2		SUBROUTINE DEADCIUT, UD. CCODOAN	
r		READ SPECIAL CARTESIAN COORDINATE SVETSHE	00004310
ř		NEAR SECTAL CARTESTAN CLUKUINATE STSTEMS.	00004320
		OTTOO - THEOTTOURUL FILE UNIT NUMBERS.	00004330
100			

C-12

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

C	CCORDA(I) = ANGLE FOR SPECIAL COORDINATE SYSTEM I.	00004340
	INTEGER UI, UO	00004350
	DOUBLE PRECISION COORDA(1)	00004360
	INTEGER I	00004370
	DCUBLE PRECISION ANGLE,F	00004380
	101 FORMAT(110,F10.0)	00004390
	201 FORMAT(//IH1, CARTESIAN COORDINATE SYSTEMS DEFINED /	00004400
	11H .*NUMBER X-AXIS ANGLE*)	00004410
	202 FORMAT(1H ,15,5X,F10.4)	00004420
	F = 3.141592653589793D0/180.D0	00004430
	WRITE(U0,201)	00004440
	00 5 I=1,5	00004450
	5  COORDA(I) = 0.00	00004460
	10 READ(UI, 101) I, ANGLE	00004470
	IF(I.LE.O)RETURN	00004480
	WRITE(U0,202)I,ANGLE	00004490
	COORDA(I) = ANGLE + F	00004500
	60 TO 10	00004510
	END	00004520
	SUBROUTINE READM (UI.UG.NOD.NEL.COORDA.COORD.GCOS.	00004530
	LIMAT, THICK, T, ELNO, EGEON)	00004540
C	READ MESH DATA.	00004550
С	UI,UC = INPUT,OUTPUT FILE UNIT NUMBERS.	00004560
C	NCD,NEL = NUMBER OF NODES,ELEMENTS.	00004570
C	COORDA(1) = ANGLE FOR SPECIAL COORDINATE SYSTEM I.	00004580
С	CCORD(J,I) = COORDINATES FOR NODE I.	00004590
С	GCOS(J,I) = DIRECTION COSINES FOR NODE I.	00004600
С	IMAT(I) = MATERIAL NUMBER FOR ELEMENT I.	00004610
C	THICK = DEFAULT THICKNESS.	00004620
С	T(I) = THICKNESS OF ELEMENT I.	00004630
C	ELNO(J,I) = NODE NUMBERS FOR ELEMENT I.	00004640
C	EGEOM(J,I) = GEOMETRY FOR ELEMENT I (BASE,HEIGHT,PART BASE).	00004650
	INTEGER UI;UO;NOD;NEL;IMAT(1);ELNO(3,1)	00004660
	COUBLE PRECISION COORDA(1),COORD(3,1),THICK,VLENTH,VDOT	00004670
	DOUBLE PRECISION GCOS(3,3,1), EGEOM(3,1), T(1)	00004680
	INTEGER I, J, M, LCOORD, DCOORD, N1, N2, N3	00004690
	DOUBLE PRECISION F,ANGLE,X,Y,Z,R,TT,B,B1,H,L31,A,V21(3),V31(3)	00004700
	101 FORMAT(215,3F10.0,15)	00004710

•

C-13

Ŧ

•

201 FORMAT(1H1, *** NODE ***/1H .* NO I D LCOCPD V 10.1	00004720
$17X_{\bullet}Y$ (THETA) 7 DCDORD X (R)*	00004730
202 FORMAT(1H +215+1X+15+2X+F12-5-1X-F12-5-1X F12-5-1X F12-5-2X IE)	00004740
203 FORMAT(1H1.* FEENENT */	00004750
11H + NO. I.D. MATERIAL THICKNESS AN	00004760
2 NODE 1 NODE 2 NODE 3 ADEALS	00004770
204 FORMAT(1H +215-5X-15-5X-F12-4-3X-15-5Y 15 5Y	00004780
F = 3.141592653589793D0/180, D0	00004790
WRITE(U0+201)	00004800
NCD = 0	00004810
6 READ(UI, 101) I.LCOORD.X.Y.Z.DCOORD	00004820
IF(I.LE.0)G0 T0 150	00004830
NOD = NOD + 1	00004840
WRITE(U0+202)NOD+I+ICOCRD-X-Y-Z-DCCOPO	00004850
$COURD(3 \cdot NOD) = Z$	0004860
IF(LCOORD.EQ.0)GO TO 7	00004870
ANGLE = Y*F	00004880
$Y = X \neq DS IN (ANGLE)$	00004890
$X = X \neq DCOS(ANGLF)$	00004900
7  CCORD(1, NOD) = x	00004910
COORD(2, NDD) = Y	00004920
IF(DCOORD-1)13,12,11	00004930
11 ANGLE = $COORDA(DCOURD)$	00004940
GCOS(1,1,NOD) = DCOS(ANGLE)	00004950
GCOS(1,2,NOD) = DSIN(ANGLE)	00004960
GO TO 100	00004970
12 R = DSQRT(X + 2 + Y + 2)	00004980
IF(R.EQ.O.DO)GO TO 13	00004990
GCOS(1,1,NOD) = X/R	00005000
GCOS(1+2+NOD) = Y/R	00005010
GO TO 100	00005020
13  GCOS(1+1+NOD) = 1+DO	00005050
GCOS(1,2,NOD) = 0.00	00005050
100  GCOS(1,3,NOD) = 0.00	00005040
GCOS(2,1,NOD) = -GCOS(1,2,NOD)	00005080
GCOS(2,2,NOD) = GCOS(1,1,NOD)	00005070
GCOS(2,3,NOD) = 0.DO	00005080
GCDS(3,1,NOD) = 0,DO	00005090
GCOS(3,2,NOD) = 0.00	00005100
	00005110
•	
--	-----------
GCOS(3+3+NDD) = 1+D0	20
GO TO 6	
150 WRITE(U0.203)	50 40
NEL = <b>0</b>	+U S 0
151 READ(UI+102)I+M+TT+N1+N2+N3	
IF([.LE.0]G0_T0_250	
$IF(IT \cdot EQ \cdot 0 \cdot DQ)IT = THICK 000051$	
NEL = NEL+1	
IMAT(NEL) = M	
T(NEL) = TT 000052	
000052 000052	
$V21(J) = CEORD(J \cdot N2) - CCORD(J \cdot N1)$	20
$170 \text{ V31(J)} = COORD(J \cdot N3) - CCORD(J \cdot N1)$	
B = VLENTH(V21)	+U 5 0
L31 = VLENTH(V31) (000052)	
$B1 = VD0T(V21 \cdot V31)/B = 000052$	
H = DSQRT(131**2 - B1**2) = 000052	
A = .500 + B + H	50 50
$EGEOM(1_{\bullet}NEL) = B$	70 20
$\Theta = E GEOM(2 \cdot NEL) = H = 000053$	
H = EGEOM(3, NEL) = B1 = 000053	
UT 200 WRITE(U0,204)NEL, I.M. TT.N1, N2, N3, A 000053	
ELNO(1, NEL) = N1	
ELNO(2,NEL) = N2	50
ELNO(3,NEL) = N3	
GO TO 151 000053	
250 RETURN 0000530	
0000336	

	SUBROUTINE READK(UI+UO+NOD+NF+KFD)	00005400
C	READ SPECIFIED FORCE-DISPLACEMENT-CONSTRAINT DEGREES OF FREEDOM.	00005410
С	UI,UO = INPUT,OUTPUT FILE UNIT NUMBERS.	00005420
C	NCD = NUMBER OF NODES.	00005430
C	NF = NUMBER OF FREEDOMS PER NODE.	00005440
С	<pre>KFD(I) = FORDE-DISPLACEMENT-CONSTRAINT SPECIFICATION FOR DOF I.</pre>	00005450
	INTEGER UI, UO, NOD, NF, KFD(1)	00005460
	INTEGER ISTOR(4), JSTOR(4), KSTOR(4), LSTOR(4)	00005470
	INTEGER I, J, K, L, M, NN, JLOC, LLOC	00005480
101	FORMAT(4(4[5))	00005490

· ·

1

	201	FORMAT(//1H1, SPECIFIED FORCE-DISPLACEMENT-CONSTRAINT DOF	00005500
		1/1H .*NODE I.D. COMPONENT NODE I.D. COMPONENT*)	00005510
	202	FCRMAT(1H +15+6X+15+6X+15)	00005520
		NN = NOD*NF	00005530
		WRITE(U0,201)	00005540
С		SET DEFAULT CODES TO SPECIFIED FORCE.	00005550
		DO 5 I=1,NN	00005560
	5	KFD(I) = I	00005570
	10	READ(UI,101)(ISTOR(K),JSTOR(K),KSTOR(K),LSTOR(K),K=1,4)	00005580
		DC 12 H=1+4	00005590
		IF(ISTOR(M).NE.O)GO TO 13	00005600
	12	CONTINUE	00005610
		GO TO 51	00005620
	13	DO 20 M=1+4	00005630
		I = ISTOR(M)	00005640
		IF(I.LE.0)GO TO 20	00005650
		J = JSTOR(M)	00005660
		K = KSTOR(M)	00005670
		L = LSTOR(M)	00005680
		WRITE(U0,202)I,J,K,L	00005690
		IF(K = Q = 0)K = I	00005700
		$IF(L \cdot EQ \cdot O)L = -J$	00005710
		JLOC = NF*(I-1) + J	00005720
		LLOC = L	00005730
		$IF(L_{\bullet}LT_{\bullet}O)LLOC = -L$	00005740
		LLOC = NF*(K-1) + LLOC	00005750
		$IF(L_LT_0)LLOC = -LLOC$	00005760
		KFD(JLOC) = LLOC	00005770
	20	CONTINUE	00005780
		GO TO 10	00005790
	51	RETURN	00005800
		END	00005810
		SUBROUTINE READP(UI,UO,NOD,NF,KFD,PREF)	00005820
С		READ LOAD REFERENCE CURVES.	00005830
C		UI,UC = INPUT,OUTPUT FILE UNIT NUMBERS.	00005840
C		NCD = NUMBER OF NODES.	00005850
¢		NF = NUMBER OF FREEDOMS PER NODE.	00005860
С		<pre>KFD(I) = FURCE+DISPLACEMENT-CONSTRAINT SPECIFICATION FOR DOF I</pre>	00005870

C-16

....

,

C	PREF(J,I) = NODAL LOAD AT DOF I FOR LOAD REFERENCE VECTOR J.	00005880
	INTEGER UI, UO, NOD, NF, KFD(1)	00005890
	DCUBLE PRECISION PREF(2,1)	00005900
	INTEGER ISTOR(4), JSTOR(4), ILOAD, NN, I, J, K, L	00005910
	DOUBLE PRECISION STOR(4)	60005920
1	O1 FORMAT(IIO)	00005930
1	02 FORMAT(4(215,F10.0))	00005940
2	OI FORMAT(IH1, NO. OF LOAD REFERENCE CURVES = 1,15)	00005950
2	02 FORMAT(1H0, LOAD REFERENCE CURVE NO. 1,15/	00005960
	11H / NODE COMPONENT LOAD*)	00005970
2	03 FORMAT(1H ,15,3X,15,E12.5)	00005980
	NN = NF*NOD	00005990
	READ(UI, 101)NLOAD	00006000
	WRITE(U0,201)NLOAD	00006010
	DG 1GU ILDAD=1,NLOAD	00006020
	WRITE(U0,202)ILOAD	00006030
	DO 5 I=1+NN	00006040
	$5 \text{ PREF(ILOAD_I)} = 0.00$	00006050
~	11 READ(UI,102)(ISTOR(K),JSTOR(K),STOR(K),K=1,4)	00006060
	DO 12 $K=1,4$	00006070
71	IF(ISTOR(K).NE.0)G0 TO 13	00006080
-	12 CONTINUE	00006090
	DQ 15 I=1,NN	00006100
	J = -KFD(I)	00006110
	IF(J.GT.O.AND.J.NE.I)PREF(ILOAD,I) = PREF(ILOAD,J)	00006120
	15 CONTINUE	00006130
	GO TO 100	00006140
	13 DC 20 K=1,4	00006150
	I = ISTOR(K)	00006160
	IF(I.LE.O)GD TO 20	00006170
	J = JSTOR(K)	00006180
	WRITE(UO+203)I,J,STOR(K)	00006190
	L = NF + (I - 1) + J	00006200
	PREF(ILDAD,L) = STUR(K)	00006210
	20 CONTINUE	00006220
• .	GE TO 11	00006230
. 10	DU CUNTINUE	00006240
	KETUKN STATES	00006250
	ENU	00006260

.

;

· j

		SUBROUTINE READPR(UI, UO, NEL, PRREF)		00006270
C		READ PRESSURE LOAD REFERENCE CURVE.		00006280
C		UI,UO = INPUT,OUTPUT FILE UNIT NUMBERS.		00006290
Ċ		NEL = NUMBER OF ELEMENTS.		00006300
С		PRREF(I) = INTENSITY ON ELEMENT I FOR PRESSURE REFERENCE VE	ECTOR.	00006310
		INTEGER UI, UD, NEL		00006320
		DOUBLE PRECISION PRREF(1)		00006330
		INTEGER ISTOR(4)+ILOAD+I,K		00006340
		DOUBLE PRECISION STOR(4)		00006350
	101	FORMAT(I10)		00006360
	102	FORMAT(4([10+F10+0])		00006370
	201	FORMAT(1H1, NO. OF PRESSURE LOAD REFERENCE CURVES = 1,15)		00006380
	202	FORMAT(1H0, PRESSURE LOAD REFERENCE CURVE NO. 1, 15/		00006390
		11H ,*ELEMENT PRESSURE!)		00006400
	203	FORMAT(1H ,15,3X,E12.5)		00006410
		READ(UI,101)NLOAD		00006420
		WRITE(U0,201)NLOAD		00006430
		DC 100 ILOAD=1,NLOAD		00006440
		WRITE(UD,202)ILOAD		00006450
		DO 5 I=1,NEL		00006460
	5	PRREF(I) = 0.00		00006470
	11	READ(UI, 102) (ISTOR(K), STOR(K), $K=1,4$ )		00006480
		00 12 K=1,4		00006490
		IFIISTOR(K).NE.0)GO TO 13		00006500
	12	CONTINUE		00006510
		GO TO 100		00006520
	13	DO 20 K=1,4		00006530
		I = ISTOR(K)		00006540
		IF(I.LE.0)G0 TO 20		00006550
		WRITE(U0,203)I,STOR(K)		00006560
		PRREF(I) = STOR(K)		00006570
	20			00006580
				00006590
	100			00006600
		KETUKN		00006610
		END		00006620

С	READ INCREMENTAL LOAD CATA.	00006640
С	UI,UO = INPUT,OUTPUT FILE UNIT NUMBERS.	00006650
С	NINCR = NUMBER OF LOAD INCREMENTS.	00006660
С	PRED = DEFAULT SOLUTION PREDICTOR ORDER.	00006670
С	IPRED(1) = SOLUTION PREDICTOR ORDER FOR LOAD INCREMENT I.	00006680
С	PFACT(J,I) = NODAL LOAD FACTOR FOR INCR. I AND REFERENCE VECTOR	J.00006690
С	PRFACT(I) = ELEMENT PRESSURE INTENSITY FACTOR FOR LOAD INCR. I.	00006700
	INTEGER UI, UO, NINCR, PRED, IPRED(1)	00006710
	DOUBLE PRECISION PFACT(2,1), PRFACT(1)	00006720
	101 FORMAT(IIO)	00006730
	102 FORMAT(110,3F10.0)	00006740
	201 FORMAT(///1H1, NO. OF LOAD INCREMENTS = ',15/	00006750
	11H , "INCREMENT", 2X, "PREDICTOR", 2X, "MECHANICAL CURVE FACTORS",	00006760
	16X, 'PRESSURE')	00006770
	202 FORMAT(1H ,15,6X,15,6X,E12.5,2X,E12.5,2X,E12.5)	C0006780
	READ(UI, 101)NINCR	00006790
	WRITE(U0,201)NINCR	00006800
	DE 100 INCR=1,NINCR	00006810
	READ(UI, 102) IPRED(INCR), PFACT(1, INCR), PFACT(2, INCR), PRFACT(INCR)	00006820
	IF(IPRED(INCR).EQ.0) IPRED(INCR) = PRED	00006830
	WRITE(UO,202)INCR, IPRED(INCR), PFACT(1, INCR), PFACT(2, INCR),	00006840
	1PRFACT(INCR)	00006850
	100 CONTINUE	00006860
	RETURN	00006870
	END	00006880
		~~~~~~~
	SUBROUTINE HEAD(UO, INCR, ISTEP, LAMS, LAM, FL, F2, FP,	00006890
	1PRED,MAXUP,NUP,MAXIT,ITER,ERRMAX,ERR)	00006900
<u> </u>	UDITE VERDINE FOR LOAD INCOCHENT ETER	

C-19

C C C

С

C

С

С

С

C

C

C

1

WRITE HEADING FOR LOAD INCREMENT STEP. 00006910 **UO = OUTPUT FILE UNIT NUMBER.** 00006920 INCR = LOAD INCREMENT NUMBER. 00006930 ISTEP = LOAD STEP NUMBER. 00006940 LAMS = LOAD STEP PARAMETER = FRACTION OF REMAINING LOAD INCREMENT.00006950 LAM = INCREMENTAL LOAD PARAMETER (MAXIMUM VALUE 1.0). 00006960 F1.F2 = NODAL LOAD FACTORS APPLIED TO REFERENCE VECTORS. 00006970 FP = ELEMENT PRESSURE LOAD FACTOR APPLIED TO REFERENCE VECTOR. 00006980 PRED = SOLUTION PREDICTOR ORDER. 00006990 MAXUP = SPECIFIED MAXIMUM NUMBER OF JACOBIAN STIFFNESS UPDATES. 00007000 NUP = NUMBER OF JACOBIAN UPDATES PERFORMED DURING THIS LOAD STEP. 00007010

C	MAXIT = MAXIMUM NUMBER OF RESIDUAL-FORCE ITERATIONS PER UPDATE.	00007020				
С	ITER = NUMBER OF ITERATIONS PERFORMED SINCE LAST UPDATE.	00007030				
С	ERRMAX = SPECIFIED MAXIMUM RESIDUAL-FORCE ERROR NORM.	00007040				
C	ERR = ACTUAL ERROR NORM OBTAINED.	00007050				
	INTEGER UD, INCR, ISTEP, PRED, MAXUP, NUP, MAXIT, ITER	00007060				
	DOUBLE PRECISION LAMS, LAM, F1, F2, FP, ERRMAX, ERR	00007070				
	201 FORMAT(1H1////1H ,*L 0 A D I N C R E M E N T*,15,	00007080				
	1 <sup>4</sup> , LOAD STEP <sup>1</sup> ,15,	00007090				
	2//1H ; INCREMENT LOAD PARAMETER =*;E12.5; 3*; STEP LOAD PARAMETER =*;E12.5;					
	202 FORMAT(//1H , MECHANICAL LOAD FACTORS = , 2E14.5)	00007120				
	207 FORMAT(1H , PRESSURE LOAD FACTOR = +, E14.5)	00007130				
	203 FORMAT(1H , PREDICTOR TYPE = 1,15)	00007140				
	204 FORMATCHH , SPECIFIED MAX. NO. JACOBIAN UPDATES = 1,15,	00007150				
	1 NO. UPDATES PERFORMED = 1,15)	00007160				
	205 FORMAT(1H , SPECIFIED MAX. NO. ITERATIONS PER UPDATE = 1,15,	00007170				
	1', NO. ITERATIONS PERFORMED SINCE LAST UPDATE =',15)	00007180				
	206 FORMAT(IH , SPECIFIED MAX. RESIDUAL FORCE ERROR = + E12.4,	00007190				
	1", ACTUAL ERROR =",E12.4)	00007200				
	WRITE(U0,201)INCR,ISTEP,LAM,LAMS	00007210				
	WRITE(U0+202)F1+F2					
	WRITE(U0,207)FP	00007230				
	WRITE(U0,203)PRED	00007240				
	WRITE(U0,204)MAXUP,NUP	00007250				
	WRITE(UO,205)MAXIT,ITER					
	WRITE(UO,206)ERRMAX,ERR	00007270				
	RETURN	00007280				
	END	00007290				
	SUBROUTINE OUTLIM(UO,NOD,NEL,NN,NF,ELNO,EGEOM,EET,QQ,P,Q,DFF,	00007300				
	LRL,RRL,RQ,RRQ,LAM,LAMR)	00007310				
С	OUTPUT LIMIT POINT DATA.	00007320				
С	UO = OUTPUT FILE UNIT NUMBER.	00007330				
C	NOD,NEL = NUMBER OF NODES,ELEMENTS.	00007340				
С	NN,NF = SYSTEM DOF,DOF PER NODE.	00007350				
С	ELNO(J,I) = NODE NUMBERS FOR ELEMENT I.	00007360				
C	EGEOM(J,I) = GEOMETRIC PROPERTIES FOR ELEMENT I.	00007370				
С	EET(J,I) = PREDICTED LIMIT STRAINS FOR ELEMENT I.	00007380				
С	<pre>CQ(I) = CUMULATIVE NODAL DISPLACEMENT FOR DOF I.</pre>	00007390				

. .

.

•

•

, .

•

С	D P,Q = TEMPORARY STORAGE VECTORS.	00007400
С	DFF = FINITE+DIFFERENCE SPACING USED IN COMPUTING FORCES.	00007410
C	RL,RRL = IST,2ND ORDER LOAD PARAMETER RATES.	00007420
С	RQ(I), RRQ(I) = 1ST, 2ND ORDER DISPLACEMENT RATES FOR DOF I.	00007430
C	LAM = INCREMENTAL LOAD PARAMETER (MAXIMUM VALUE 1.0).	C0007440
C	; LAMR = LOAD YET TO BE APPLIED = 1.0 - LAM.	00007450
	INTEGER UO,NOD,NEL,NN,NF,ELNO(3,1)	00007460
	DCUBLE PRECISION EGEOM(3,1),EET(3,1),QQ(1),P(1),Q(1),DFF	00007470
	DOUBLE PRECISION RL,RRL,RQ(1),RRQ(1),LAM,LAMR	00007480
	DCUBLE PRECISION SLIM, LAMLIM	00007490
	201 FORMAT(//1H , PREDICTED LIMIT POINT OCCURS AT LOAD INCREMENT	PARAM00007500
	1ETER = + E12.5/1H +	00007510
	2 THE FOLLOWING ARE PREDICTED LIMIT FORCES-DISPLACEMENTS-STRAI	NS*) 00007520
	SLIM = -RL/RRL	00007530
	LAMLIM = LAM + (SLIM*RL + .5DO*SLIM**2*RRL)*LAMR	00007540
	WRITE(U0,201)LAMLIM	00007550
	DC 10 I=1,NN	00007560
	10  Q(I) = QQ(I) + SLIM*RQ(I) + .5D0*SLIM**2*RRQ(I)	00007570
	CALL FORCE(NEL,NN,NF,ELNO,DFF,Q,P)	00007580
	CALL OUTPQ(UO,NOD,NF,P,Q)	00007590
	CALL STRAIN(NEL,ELNO,EGEOM,NF,Q,EET)	00007600
	CALL OUTE(UO,NEL,EET)	00007610
	RETURN	00007620
	END	000076 <b>30</b>
~	SUBROUTINE OUTPQ(U0,NOD,NF,P,Q)	00007640
C	, WRITE TUTAL FORCES AND DISPLACEMENTS.	00007650
Ĉ	UC = OUTPUT FILE UNIT NUMBER.	00007660
C.	NUD, NF = NUMBER OF NODES, DOF PER NODE.	00007670
U	, P(I),Q(I) = FURCE,DISPLACEMENT TO BE OUTPUT AT DOF I.	00007680
	INTEGER UD,NOD,NF	00007690
	DUUBLE PRECISION P(1)+C(1)	00007700
	INTEGER 1, J,K	00007710
	CUUBLE PRECISION STURIES	C0007720
	201 FURMAI(1H1,14X,22(1H*),* CUMULATIVE INTERNAL FORCES AND DISP	'LACEM00007730

J

3

1ENTS\*,2X,23(1H\*)/1H ,\*\*\* NODE \*\*\*,4X,17(1H\*),\* FURCES \*,16(1H\*),00007740 27X,13(1H\*),\* DISPLACEMENTS \*,13(1H\*), c0007750 3/1H ,10H ND. I.D.,5X,1HU,14X,1HV,14X,1HW,19X,1HU,14X,1HV,14X,1HW/00007760 1) 00007770

C-21

÷

J

	202	FURMAT(1H +215+2X+3E15+7+5X+3E15+7)	00007780
		WRITE(U0,201)	00007790
		DU 10 I=1+6	00007800
	10	SIOR(1) = 0.00	00007810
		EC 100 I=1,NOD	00007820
		DC 50 J=1,NF	00007830
		K = NF + (I - 1) + J	00007840
		STOR(J) = P(K)	00007850
	50	STOR(3+J) = Q(K)	00007860
	100	WRITE(U0,202)I,I,(STOR(J),J=1,6)	00007870
		RETURN	00007880
		"END	00007890
-		SUBROUTINE OUTE(UO,NEL,ET)	00007900
C		WRITE CUMULATIVE STRAINS ET.	00007910
C		UO = OUTPUT FILE UNIT NUMBER.	00007920
C		NEL = NUMBER OF ELEMENTS.	00007930
C		ET(J,I) = STRAINS TO BE OUTPUT FOR ELEMENT I.	00007940
		INTEGER UD, NEL	00007950
		COUBLE PRECISION ET(3,1)	00007960
		INTEGER II,I	00007970
	201	FORMATCIHI, "ELEMENT", 3X, 12(1H*), CUMULATIVE STRAINS *, 12(1H*),	00007980
	<b>.</b>	L/1H , NO. , 10X, 2HXX, 10X, 2HYY, 10X, 2HZZ, 10X, 2HXY)	00007990
	202	FURMAT(1H , 15, 3X, 2E12.4, 12X, E12.4)	0008000
		WRITE(U0,201)	00008010
		DO 100 II=1,NEL	00008020
	100	WRITE(U0,202)11,(ET(1,11),1=1,3)	00008030
		RETURN	00008040
		END	00008050
c		SUBRUUTINE GFILLUNF,ELNU,GG,G)	00008060
С С		FURM VECTUR OF ELEMENT DISPLACEMENTS Q FROM NODAL DISPLACEMENTS Q	Q00008070
C C		NF = DUF PER NUUE.	00008080
с С		ELNUIJ,I) = NODAL DIERA AGENENT AT DOG	00008090
с С		WAVER - NUMAL DISPLACEMENT AT DUF 1.	00008100
L		WILL - ELEMENT NUDAL DISPLACEMENT AT ELEMENT DOF I.	00008110
		INIEVER NETEENULIJ	00008120
		nnnore akerijina Artijatij	00008130

		CALL ROTG(NF,ELNO,G,O)	00008210
		RETURN	00008220
		END	00008230
		SUBROUTINE PFILL(NF,ELNC,P,PP)	00008240
С		FORM VECTOR OF NODAL FORCES PP FROM ELEMENT FORCES P.	00008250
С		NF = DOF PER NOCE.	00008260
С		ELNO(J,I) = NODE NUMBERS FOR ELEMENT I.	00008270
С		P(I) = ELEMENT NODAL FORCE AT ELEMENT DOF I.	00008280
C		PP(I) = NODAL FORCE AT COF I.	0.0008290
		INTEGER NF,ELNO(1)	00008300
		DCUBLE PRECISION P(1), PP(1)	00008310
		INTEGER I,NI,IO,JO,K	00008320
		CALL ROTQ(NF,ELNG,P,1)	00008330
		DC 100 I=1,3	00008340
		NI = ELNO(I)	00008350
		IO = NF * (NI - 1)	00008360
		JO = NF*(I-1)	00008370
		DO 100 K=1,NF	00008380
	100	PP(IO+K) = PP(IO+K) + P(JO+K)	00008390
		RETURN	00008400
		END	00008410
		SUBROUTINE DFILL(NF.EGEOM.Q.D)	00008420
C		COMPUTE ELEMENT DISPLACEMENT DERIVATIVES D FROM DISPLACEMENTS Q.	00008430
С		NF = DOF PER NODE.	00008440
С		EGEUM(J,I) = GEOMETRIC PROPERTIES FOR ELEMENT I.	00008450
С		Q(I) = ELEMENT NODAL DISPLACEMENT AT ELEMENT DOF I.	00008460
С		D(I) = ITH DISPLACEMENT DERIVATIVE (UX, VX, WX, UY, VY, WY).	00008470
		INTEGER NF	00008480
		DCUBLE PRECISION EGEOM(1),Q(1),D(1)	00008490

INTEGER I,NI,IO,JO,K	00008140
00 100 1=1,3	00008150
NI = ELNO(I)	00008160
$IO = NF \neq (NI - 1)$	00008170
$JO = NF \neq (I-1)$	00008180
DO 100 K=1.NF	00008190
100 Q(J0+K) = QQ(I0+K)	00008200
CALL ROTG(NF,ELNO,G,O)	00008210
RETURN	00008220
END	00008230

j.

1

·	10	INTEGER NF2,I DCUBLE PRECISION B.H.B1.B2.A2 NF2 = NF*2 B = EGEOM(1) H = EGEOM(2) B1 = EGEOM(3) B2 = B-B1 A2 = 1.DO/(B*H) D0 10 I=1.NF D(I) = A2*H*(-Q(I)+Q(NF+I)) D(NF+I) = A2*(-B2*C(I)-B1*Q(NF+I)+B*Q(NF2+I)) RETURN END	00008500 00008520 00008530 00008540 00008550 00008560 00008570 00008580 00008590 00008590 00008610 00008620
С С С С	10	SUBROUTINE EFILL(NF,D,ET) COMPUTE ELEMENT STRAINS ET FROM DISPLACEMENT DERIVATIVES D. NF = DOF PER NCDE. D(I) = ITH DISPLACEMENT DERIVATIVE (UX,VX,WX,UY,VY,WY). ET(I) = ITH LAGRANGIAN STRAIN COMPONENT (XX,YY,XY). INTEGER NF DCUBLE PRECISION D(1),ET(1) ET(1) = D(1) ET(2) = D(NF+2) ET(3) = D(2) + D(NF+1) DO 10 I=1,NF ET(1) = ET(1) + .5D0*D(I)**2 ET(2) = ET(2) + .5D0*D(NF+I)**2 ET(3) = ET(3) + D(I)*D(NF+I) RETURN END	00008630 00008640 00008660 00008660 00008670 00008690 00008700 00008710 00008710 00008720 00008750 00008750 00008760 00008770
С С С С С		SUBROUTINE AFILL(NF,D,A,KODE) KCDE=0 FORM LAGRANGIAN A1(I,J) = A1(I,J,K)*D(K). KCCE=1 FORM LAGRANGIAN A(I,J) = A0(I,J) + A1(I,J,K)*D(K). NF = DOF PER NODE. D(I) = ITH DISPLACEMENT DERIVATIVE (UX,VX,WX,UY,VY,WY). A GIVES LAGRANGIAN STRAINS (XX,YY,XY) FROM (UX,VX,WX,UY,VY,WY). INTEGER NF,KODE	00008790 00008800 00008810 00008820 00008830 00008840 00008850

-

C-24

,

DCUBLE PRECISION D(1),A(3,1)	00008860
INTEGER J,J1	00008870
DC 10 J=1,NF	00008880
JI = NF+J	00008890
A(1,J) = D(J)	0008900
A(1, J1) = 0.00	00008910
A(2,J) = 0.D0	00008920
A(2,J1) = D(J1)	00008930
A(3,J) = D(J1)	00008940
10 A(3, J1) = D(J)	00008950
IF(KODE.EQ.O)RETURN	00008960
A(1,1) = 1.00 + A(1,1)	00008970
A(2,NF+2) = 1.00 + A(2,NF+2)	00008980
A(3,2) = 1.00 + A(3,2)	0008990
A(3,NF+1) = 1.D0 + A(3,NF+1)	0009000
RETURN	00009010
END	00009020

.

. í

4

	SUBROUTINE GFILL(NF, EGEOM, G)	00009030
С	FORM DISPLACEMENT-DERIVATIVES FROM NODAL-DISPLACEMENTS MATRIX G	. 00009040
C	NF = DOF PER NODE.	00009050
C	EGEOM(J,I) = GEOMETRIC PROPERTIES FOR ELEMENT I.	00009060
C	G GIVES (UX,VX,WX,UY,VY,WY) FROM ELEMENT NODAL DISPLACEMENTS.	00009070
	INTEGER NF	00009080
	DOUBLE PRECISION EGEOM(1),G(6,1)	00009090
	INTEGER NF2,NF3,I,I1,I2	00009100
	DOUBLE PRECISION 8+H+B1+B2+A2	00009110
	NF2 = NF*2	00009120
	NF3 = NF*3+	00009130
	B = EGEOM(1)	00009140
	H = EGEOM(2)	00009150
	B1 = EGEOM(3)	00009160
	B2 = B - B1	00009170
	A2 = 1.00/(B*H)	00009180
	DC 5 [=1+NF2	00009190
	DC 5 J=1,NF3	00009200
	$5 G(I_{+}J) = 0_{+}D0$	00009210
	DG 10 I=1,NF	00009220
	II = NF + I	00009230

C-25

.

I2 = NF2 + I	00000340
$G(T,T) = -H \pm A2$	CUUUAZAU
	00009250
$G(1_111) = H \neq A2$	00009260
G(11,1) = -BZ + AZ	00009270
$\frac{6111}{11} = -81742$	00009280
$10 \ 6(11,12) = B \neq A2$	00009290
KETURN END	00009300
CNU	00004310

SUBROUTINE MIRAN (A.MA.NA.B.MB.NB.D)	
C COMPUTE $D(J_*J) = \Delta(M_*N) * B(M_*T) * B(N_*L)$	00009320
C A = SQUARE MATRIX TO BE TRANSFORMED	00009330
C MA = MAXIMUM (EDRIDAN_DIMENSIONED) STOR	00009340
C NA = ACTIAL SIZE OF A	UF A. 00009350
f = TPANSEDDWATTON MATOTY	00009360
C = 0 = 0	00009370
C D - SWUARE IRANSFURMED MATRIX.	00009380
C MO = MAXIMUM (FURTRAN-DIMENSIONED) SIZE	OF D. 00009390
C NB = ACTUAL SIZE OF D.	00009400
INTEGER MA, NA, MB, NB	00009410
LUUBLE PRECISION A(MA,1), B(MA,1), D(MB,1	) 00009420
INIEGER I, J, M	00009430
DUUBLE PRECISION C, STOR(6,9)	00009440
DU 50 I=1.NA	00009450
DU 50 J=1,NB	00009460
C = 0.00	00009470
DO 45 M=1,NA	00009480
45 C = C + A(I,M) * B(M,J)	00009490
$50 \text{ STOR}(1, J) \neq C$	00009500
DC 100 I=1,NB	00009510
DO 100 J=1,NB	00009520
C = 0.D0	00009520
DO 95 M=1,NA	00009550
95 C = C + STOR(M, J) * B(M, I)	00003340
100 D(I,J) = C	00009550
RETURN	00009300 00000530
END	00009570

SUBROUTINE ROTQ(NF,ELNC,Q,KODE)

00009590

C-26

•

٥

C		KODE=0 ROTATE DISPLACEMENTS TO ELEMENT FROM NODAL.	00009600
C		KCDE=1 ROTATE FORCES TO NODAL FROM ELEMENT.	00009610
С		NF = DOF PER NODE.	00009620
С		ELNO(J,I) = NODE NUMBERS FOR ELEMENT 1.	00009630
C		Q(I) = ELEMENT NODAL DISPLACEMENT AT ELEMENT DOF I.	00009640
C		GCOS(J,I) = DIRECTION COSINES FOR COORDINATE SYSTEM AT NODE I.	00009650
С		COORD(J,I) = COORDINATES OF NODE I.	00009660
		COMMON/COMCOS/GCOS/COMCOR/COORD	00009670
		INTEGER NF,ELNO(1),KODE	00009680
		DCUBLE PRECISION GCOS(3,3,1),COORD(3,1),Q(1)	00009690
		INTEGER N1,N2,N3,N1,N,I,J,M,IO	00009700
		DOUBLE PRECISION V21(3),V31(3),VY(3),VZ(3)	00009710
		DOUBLE PRECISION C,REG(3,3),RNG(3,3),REN(3,3),QPART(3)	00009720
С		COMPUTE MATRIX REG TO RCTATE DISPLACEMENTS ELEMENT FROM GLOBAL	00009730
		N1 = ELNC(1)	00009740
		N2 = ELNO(2)	00009750
		N3 = ELNO(3)	00009760
		DO 10 I=1,3	00009770
		V21(I) = CCORD(I,N2) - CCORD(I,N1)	00009780
	10	V31(I) = COORD(I,N3) - CCORD(I,N1)	00009790
		CALL VCROSS(V21+V31,VZ)	00009800
		CALL VNORM(V21,V21)	00009810
		CALL VNORM(VZ,VZ)	00009820
		CALL VCROSS(VZ,V21,VY)	00009830
		DU 20 J=1,NF	00009840
		$REG(1_{\mathbf{j}}) = V2L(\mathbf{j})$	00009850
	20	$REG(2_{+}J) = VY(J)$	00009860
	20	$REG(3_{1}\mathbf{J}) = VZ(\mathbf{J})$	00009870
~		UU 500 N≠1,3	00009880
ι		COMPUTE MATRIX RNG RUTATE NUDE N DISPLACEMENTS NODAL FROM GLOBAL	00009890
		NI = ELNU(N)	00009900
•			00009910
		DU 30 J=1, NF	00009920
~	30	RNG(1+J) = GUS(1+J+NI)	00009930
6		CUMPUTE MATRIX KEN KUTATE DISPLACEMENTS ELEMENT FROM NODAL	00009940
			00009950
		UU DU J=I+NF	00009960
		L = U.VU DD (E M-1 NE	00009970
	1 =	DE 40 MELINE C - C - PECLE NIERCLA NIE	00009980
	4 5	し = し + KEG(I)MJ#KNG(J)M)	00009990

· )

ļ

C-27

.

-

ł

,

50	$REN(I_{J}) = C$	00010000
C	RCTATE Q(N) PARTITION USING MATRIX REN FOR NODE N	00010000
	10 = NF*(N-1)	00010010
	DO 110 I=1.NF	00010020
110	GPART(1) = O(10+1)	00010030
		00010040
		00010050
		00010060
	IF(KODE-EQ-1)GO TO 116	00010070
	00 115 M≖l,NF	00010080
115	$C = C + REN(I_{+}M) + CPART(M)$	00010000
	GC TO 120	00010090
116	CC 118 M=1.NF	00010100
118	$C = C + RENIM_{A}TAAOOAOTAAA$	00010110
120	O(10411 + C)	00010120
500		00010130
500		00010140
	RETURN	00010150
	END	00010160
		55010100

	SUBROUTINE ROTK (NF,ELNG,K)	00010170
	RCTATE ELEMENT STIFFNESS TO NODAL FROM ELEMENT.	00010180
	NF = DOF PER NODE.	00010100
	ELNO(J+I) = NODE NUMBERS FOR FLEMENT T	00010190
	K = FIFMENT STEENESS MATOTY	00010200
	GENELLY STITUTEDS MARKA.	00010210
	CORDELLA - DERECTION COSTNER FOR COURDINATE SYSTEM AT NODE I.	00010220
	CONNENT CONCOLORDINATES OF NUDE I.	00010230
	LUMMUN/LUMCUS/GLUS/COMCOR/COORD	00010240
	INTEGER NF,ELNO(1)	00010250
	DEUBLE PRECISION GCOS(3,3,1),COORD(3,1),K(9,9)	00010260
	INTEGER N1+N2+N3+NI+N+I+J+M+I0+J0+IP+JP	00010270
	DOUBLE PRECISION V21(3).V31(3).VV(3).V/(3)	00010270
	DOUBLE PRECISION C.REG(3.3), RNG(3.3), REN(3.3), NOADTIS 31	00010280
	COMPUTE MATRIX REG TO ROTATE DISOLACENTATE OF THE ACTIVE STATE	00010290
	NI - ELNOLIN	00010300
		00010310
	NZ = CLNU(Z)	00010320
	N3 = ELNU(3)	00010330
	DU 10 1=1,3	00010340
	v21(1) = COORD(1, N2) - COORD(1, N1)	00010350
10	V31(I) = CCORD(I,N3) - CCORD(I,N1)	00010340
	CALL VCROSS(V21.V31.V7)	00010300
		00010370

C-28

с с с с с с с

С

				CALL VNDRM (V21-V21)	00010380
					00010380
					00010390
					00010400
				PEC(1, 1) = V(2)(1)	00010410
				NEG(143) - VZ1(3) DEC12 () - VV())	00010420
			20	REGIZIJJ - VILJJ DEC12 IV - VILJJ	00010430
			20	NC 500 M-1 2	00010440
		r		- UU DUV NELAD - Compute Matoly duc dotate node a dichactricate nodal cook a odia	00010450
		U		CUMPUTE MATRIX RNG RUTATE NUUE N DISPLALEMENTS NUDAL FRUM GLUBAL	00010460
				NI = ELNU(N)	00010470
•					00010480
			20	$\frac{\partial U}{\partial t} = \frac{\partial U}{\partial t} + $	00010490
		~	30	RNG(1,J) = GCUS(1,J,NI)	00010500
		¢		CUMPUTE MATRIX KEN KUTATE DISPLACEMENTS ELEMENT FROM NODAL	00010510
					00010520
					00010530
				$\mathbf{U} = 0 \cdot \mathbf{D} 0$	00010540
					00010550
	0		45	$C = C + REG(I_{\bullet}M) \neq RNG(J_{\bullet}M)$	00010560
	)í		50	$REN(\mathbf{I}_{\bullet}\mathbf{J}) = \mathbf{C}$	00010570
	- 29	~			00010580
		L		RUTATE K(IP+N) PARTITION USING MATRIX REN FOR NODE N	00010590
				$10 = NF \neq (IP - I)$	00010600
				JO = NF * (N-1)	00010610
				EC 110 I=1+NF	00010620
				DC 110 J=1+NF	00010630
			110	KPART(I,J) = K(IO+I,JO+J)	00010640
				UC 120 I=1,NF	00010650
				CC 120 J=1,NF	00010660
				C = 0.00	00010670
				DC 115 M=1,NF	00010680
			115	C = C + KPART(I,M) + REN(M,J)	00010690
			120	$K\{10+1, J0+J\} = C$	00010700
			200	CONTINUE	00010710
		_		DO 300 JP=1+3	00010720
		C		RUTATE K(N, JP) PARTITION USING MATRIX REN FOR NODE N	00010730
				$IO = NF \neq (N-1)$	00010740
				JO = NF*(JP-1)	00010750
	•			DC 210 I=1,NF	00010760
				00 210 J=1,NF	00010770

}

.

•

215 220 300 500	$ \begin{array}{l} KPART(I,J) &= K(I0+I,J0+J) \\ D0 & 220 \ I=1,NF \\ D0 & 220 \ J=1,NF \\ C &= 0,D0 \\ D0 & 215 \ M=1,NF \\ C &= C + REN(M,I) * KPART(M,J) \\ K(I0+I,J0+J) &= C \\ CONTINUE \\ CONTINUE \\ RETURN \\ END \end{array} $	00010780 00010790 00010800 00010810 00010820 00010830 00010840 00010850 00010860 00010870 00010880
10	SUBROUTINE FORCE (NEL,NN,NF,ELNO,DF,QQ,PP) COMPUTE INTERNAL FORCES PP CORRESPONDING TO DISPLACEMENTS QQ. NEL = NUMBER OF ELEMENTS. NN,NF = SYSTEM DOF,DOF PER NODE. ELNO(J,I) = NODE NUMBERS FOR ELEMENT I. DF = FINITE-DIFFERENCE SPACING USED IN COMPUTING FORCES. QQ(I),PP(I) = CUMULATIVE DISPLACEMENT,INTERNAL FORCE AT DOF I. EGEOM(J,I) = GEOMETRIC PROPERTIES FOR ELEMENT I. T(I) = THICKNESS OF ELEMENT I. NS = NUMBER OF STRAIN COMPONENTS. COMMON/COMEG/EGEOM/COMT/T/COMNS/NS INTEGER NEL,NN,NF,ELNO(3,1),NS COUBLE PRECISION Q(I),PP(I),OF,EGEOM(3,1),T(I) INTEGER I,II,M,NF2,NF3 DOUBLE PRECISION A(3,6),G(6,9),ENERGY EQUIVALENCE(A(I),G(I)) NF2 = NF*2 NF3 = NF*3 DC 10 I=1,NN PP(I) = 0.DO DC 100 II=1,NEL V = .5DO*T(II)*EGEOM(I,II),Q,D) CALL EFILL(NF,ELOC(1,II),Q,D) CALL EFILL(NF,EGOM(1,II),Q,D) CALL EFILL(NF,DEO) DC 50 I=1,NS	C0010890 C0010900 00010910 00010920 00010930 00010940 00010950 00010960 00010970 00010980 00010990 00010990 00011000 00011020 00011020 00011050 00011050 00011060 C0011070 00011080 00011090 00011100 00011110 C0011120 00011130

i.

•

C-30

. --

	· · · · · · · · · · · · · · · · · · ·	)
	DO 35 M-1 NC	000111/0
	DO 33 M-19N3	00011160
-	22 EAM/ * EVAM) C = CO(1)	00011100
	C = EU(1)	00011180
	$E(1) \neq C = Dr$	00011190
	OA = CNERGT(11)E)	00011200
	EN17 - G T DE 1:0 - ENEDEVITE EN	00011210
	UD = ENERGY(11)E)	00011220
1	CALL A E T LL INE (D A L)	00011230
	CALL AFILLINFYUYAYIJ Do 70 I-1 Aco	00011240
	$\int \frac{1}{1 + 1 + NFZ}$	00011250
	$C = U_{\bullet} D U$	00011260
	$\frac{1}{2} = \frac{1}{2} = \frac{1}$	00011270
C	0 DC(1) = C  0 DC(1) = C	00011280
	CA14 CETTAINE ECEOM(1) (T) C)	00011290
	DALE OFILEINFECCUMII/1///	00011310
	C = 0.00	00011310
	00 75 M-1.NE2	00011320
-	00 /0 M-1/NFZ 15 C = C + DD/M\±C/M_T\	00011360
0 5	D = C + FD(H) + O(H) F(H)	00011340
$\overline{1}$ $\overline{1}$	0 CALL DETILING. 51 NO(1.11),9,001	00011350
μ μ	DETIIDN	00011300
	END	00011370
		00011380
	SUBROUTINE PFORCE(PFACT, PREF, NN, P)	00011390
C	COMPUTE APPLIED NODAL FORCES P.	00011400
C	PFACT(J) = NODAL LOAD FACTOR FOR REFERENCE VECTOR J.	00011410
C	PREF(J,I) = NODAL LOAD AT DOF I FOR REFERENCE VECTOR J.	00011420
C	NN = TOTAL SYSTEM DOF.	00011430
C	P(I) = APPLIED CUMULATIVE LOAD AT DOF I.	00011440
	INTEGER NN	00011450
	DOUBLE PRECISION PFACT(1), PREF(2,1), P(1)	00011460
	INTEGER I	00011470
	CO 1CO I=1,NN	00011480
10	00 P(I) = PFACT(1)*PREF(1,I) + PFACT(2)*PREF(2,I)	00011490
	RETURN	00011500
	END	00011510

	SUBROUTINE EFORCE(IPRESS, PR, PRREF, GQ, NEL, NN, NF, ELNC, PP)	00011520
C	COMPUTE NODAL FORCES PP DUE TO ELEMENT PRESSURES PR.	00011530
6	IPRESS = NONCONSERVATIVE CODE = 0,1 FOR NO PRESSURE, PRESSURE.	00011540
C	PR = ELEMENT PRESSURE INTENSITY FACTOR.	00011550
C	PRREF(I) = INTENSITY ON ELEMENT I FOR PRESSURE REFERENCE VECTOR.	00011560
C	CO(I) = CUMULATIVE NODAL DISPLACEMENT AT DOF I.	00011570
C	NEL = NUMBER OF ELEMENTS.	00011580
Ç	NN+NF = TOTAL SYSTEM DCF+DOF PER NODE.	00011590
C	ELND(J,I) = NODE NUMBERS FOR ELEMENT I.	00011600
C	PP(I) = COMPUTED NONCONSERVATIVE PRESSURE NODAL FORCE AT DOF I.	00011610
C	EGEOM(J,I) = GEOMETRY FOR ELEMENT I (BASE,HEIGHT,PART BASE).	00011620
	CCMMON/COMEG/EGEOM	00011630
	INTEGER IPRESS, NEL, NN, NF, ELNO(3,1)	00011640
	DOUBLE PRECISION PR, PRREF(1), QO(1), PP(1), EGEOM(3,1)	00011650
	INTEGER I,II	00011660
	DOUBLE PRECISION C, V21(3), V31(3), Q(9), P(9)	00011670
	DC 10 I=1,NN	00011680
10	PP(I) = 0.00	00011690
	IF(IPRESS.EQ.O)RETURN	00011700
	DO 100 II=1,NEL	00011710
	CALL QFILL(NF,ELNO(1,II),QQ,Q)	00011720
	DO 20 I=1,3	00011730
	V21(I) = Q(3+I) - Q(I)	00011740
20	V31(I) = Q(6+I) - Q(1)	00011750
	V21(1) = V21(1) + EGEOM(1, II)	00011760
	V31(1) = V31(1) + EGEOM(3+II)	00011770
	V31(2) = V31(2) + EGEOM(2,II)	00011780
	CALL VCRUSS(V21,V31,P)	00011790
	C = PR*PRREF(II)/6.DO	00011800
	DU 50 I=1,3	00011810
	P(1) = P(1) * C	00011820
	P(3+1) = P(1)	00011830
50	P(6+I) = P(I)	00011840
100	CALL PFILL(NF,ELNO(1,II),P,PP)	00011850
	PETURN	00011860
	END	00011870
	SHERDENTINE ERCONDING NA KED BD D COOL	
~		00011880

.

COMPUTE ERROR NORM USING CUMULATIVE FORCES PP AND RESIDUALS P. 00011890

℃<del>-</del>32

С

C	UC = OUTPUT FILE UNIT NUMBER.	00011900
С	NN = FORCE-DISPLACEMENT-CONSTRAINT SPECIFICATION FOR DOF I.	00011910
C	KFD(1) = FORCE-DISPLACEMENT-CONSTRAINT SPECIFICATION FOR DOF 1.	. 00011920
C	PP(I) = CUMULATIVE NODAL FORCE AT DOF I.	00011930
C	P(I) = RESIDUAL (UNBALANCED) NODAL FORCE AT DUF I.	00011940
C	ERR = COMPUTED RESIDUAL-FORCE ERROR NORM.	00011950
	INTEGER UD,NN,KFD(1)	00011960
	DOUBLE PRECISION PP(1),P(1),ERR	00011970
	INTEGER I,J	00011980
	DOUBLE PRECISION C1.C2	00011990
2	1 FORMAT(1H , 'ERROR NORM = ',E12.5)	C0012000
	ERR = 0.00	00012010
	C1 = 0.00	00012020
	C2 = 0.D0	00012030
	DO 5 I=1,NN	00012040
	J = KFD(I)	00012050
	IF(J.LT.0.OR.J.EQ.I)GO TO 5	00012060
	P(J) = P(J) + P(I)	00012070
	P(I) = 0.00	00012080
	5 CONTINUE	00012090
	DC  10  I=1, NN	00012100
	IF(KFD(1) + EQ + 1)C1 = C1 + DABS(P(1))	00012110
]	D C2 = C2 + DABS(PP(1))	00012120
	IF(C2.GT.0.D0)ERR = C1/C2	00012130
	WRITE(U0,201)ERR	00012140
	RETURN	00012150
	END	00012160

,

1

· •

·

	SUBROUTINE STRAIN(NEL,ELNO,EGEOM,NF,CQ,EET)	00012170
C	COMPUTE STRAINS EET FROM GLOBAL DISPLACEMENTS QQ.	00012180
C	NEL = NUMBER OF ELEMENTS.	00012190
C	ELNO(J,I) = NODE NUMBERS FOR ELEMENT I.	00012200
C	EGEOM(J,I) = GEOMETRIC PROPERTIES FOR ELEMENT I.	00012210
C	NF = DOF PER NODE.	00012220
С	<pre>CC(I) = CUMULATIVE NODAL DISPLACEMENT AT DOF I.</pre>	00012230
C	EET(J,I) = COMPUTED CUMULATIVE STRAINS (XX,YY,XY) FOR ELEMENT I.	00012240
	INTEGER NEL, ELNO(3,1), NF	00012250
	DOUBLE PRECISION EGEOM(3,1),QQ(1),EET(3,1)	00012260
	INTEGER II	00012270

C--33

ł

	10	CCUBLE PRECISION Q(9),D(6) DO 10 II=1,NEL CALL QFILL(NF,ELNO(1,II),QQ,Q) CALL DFILL(NF,EGEOM(1,II),Q,D) CALL EFILL(NF,D,EET(1,II)) RETURN END	00012280 00012290 00012300 00012310 00012320 00012330 00012340
		DOUBLE PRECISION FUNCTION ENERGY(II, ET)	00012350
C		EVALUATE ENERGY DENSITY FOR ELEMENT II AT STRAINS ET (MCONEY).	00012360
L C		LI = ELEMENT NUMBER.	00012370
ل د		EI(1) = CUMULATIVE STRAINS (XX, YY, XY) FOR ELEMENT.	00012380
с С		IMAT(1) = MATERIAL NUMBER FOR ELEMENT I.	00012390
6		CONMON/COMMAT/INAT/CONFUNCTION DOL DOL	00012400
		TUMMUN/CUMMAI/IMAI/COMENU/CC1,CC2	00012410
		INTEGER IL, MAILI Double descision still sciles essent	00012420
		INTECED 1	00012430
		INTEVER I Double dreetstow et community to	00012440
		T = TMAT(TT)	00012450
		$f = c_1 f r$	00012460
		$C_2 = C_2(1)$	00012470
-		$\Delta = 2_{0} 0 0 * (FT(1) + FT(2))$	00012480
		$B = 4_00 + ET(1) + ET(2) - ET(3) + + 2$	00012490
		$D = 1_0 D + \Delta + B$	00012500
		$II = (\Delta + (\Delta + B) - B) / O$	00012510
		12 = 11 + B*(A+B)/D	00012520
		$ENERGY = C1 \times 11 + C2 \times 12$	00012530
		RETURN	00012540
		END	00012550
			00012200
		SUBROUTINE EVAL(II,ORD,N,FO,DF,ESTOR)	00012570
С		EVALUATE STRAIN ENERGY OF ELEMENT II AS FUNCTION OF STRAINS TO	00012580
C		ESTABLISH A COMPLETE INTERPOLATING POLYNOMIAL OF ORDER ORD L.E.	3.00012590
C		II = ELEMENT NUMBER.	00012600
C		ORD = MAXIMUM TENSOR ORDER TO BE USED FOR DIFFERENCE EXPANSION.	00012610
C		N = DIMENSION OF TENSORS.	00012620
L د		FULIJ = CURRENT VALUE OF INDEPENDENT VARIABLE I.	00012630

i

C-34

.

.

RUPRODUCIBILITY OF THE ORIGINAL PAGE IS POUR-

( С С

	DF(I) = FINITE DIFFERENCE IN INDEPENDENT VARIABLE I.	00012640
	ESTOR(I) = STORAGE VECTOR FOR ENERGY EVALUATIONS I.	00012650
	ENERGY(II,F) GIVES ENERGY FUR ELEMENT II AT VARIABLE STATE F.	00012660
	INTEGER II+ORD+N	00012670
	DOUBLE PRECISION FO(1), DF(1), ESTCR(1), ENERGY	00012680
	INTEGER I, J, K, M, IE	00012690
	COUBLE PRECISION F(3)	00012700
	IE = 1	00012710
	ESTOR(IE) = ENERGY(II,FO)	00012720
	IF(ORD.LT.1)GO TO 100	00012730
	DC 10 1=1,N	00012740
	IE = IE+1	00012750
	DO 5 M=1+N	00012760
5	F(M) = FO(M)	00012770
	F(I) = F(I) + DF(I)	00012780
10	ESTOR(IE) = ENERGY(II,F)	00012790
	IF(ORD+LT-2)GO TO 100	00012800
	DC 20 I=1,N	00012810
	DO 20 J=1,I	00012820
	IE = IE+1	00012830
	DC 15 M=1,N	00012840
15	F(M) = FO(M)	00012850
	F(I) = F(I) + DF(I)	00012860
	F(J) = F(J) + OF(J)	00012870
20	ESTOR(IE) = ENERGY(II,F)	00012880
	IF(ORD-LT-3)GO TO 100	00012890
	DC 30 I=1,N	00012900
	DO 30 J=1,I	00012910
	DC 30 K=1,J	00012920
	$\mathbf{IE} = \mathbf{IE} + \mathbf{I}_{\mathbf{v}}$	00012930
	DC 25 M=1,N	00012940
25	F(M) = FO(M)	00012950
	F(I) = F(I) + DF(I)	00012960
	F(J) = F(J) + DF(J)	00012970
	F(K) = F(K) + DF(K)	00012980
30	ESTOR(IE) = ENERGY(II,F)	00012990
100	RETURN	00013000

ł

.

.

•••

		SHADONT INE HIZEODMAN DE ESTOR HETORY	
	ſ	FROM STRAIN ENERGY TENSORE USION USING (MANDALITE POINTE COTO)	00013020
	r r	N = DIMENSION OF TENSORS USING VUADRALLU PUINTS ESTUR.	00013030
	č	n - DIMENSION OF TENSORS. FFTI) = FINTTE DIEREDENCE IN INDEDENDENT MARIARNE I	00013040
	č	ESTORIEL = STORAGE VECTOR FOR ENERGY EVALUATION /	00013050
	č	USTOR(1) = STORAGE VECTOR FOR EACH EVALUATION 1.	00013060
	U	INTEGER N	00013070
		DOUBLE PRECISION DE(1) ESTOP(1) USTOP(1)	00013080
		INTEGER 1.1.11.12.10CTL.N1	00013090
		DCUBLE PRECISION C.CI.CI.CI.DI.DI	00013100
		N1 = 1 + N	00013110
		$T_1 = 1$	00013120
		$I_2 = N_1$	00013130
	C	FORM OTH-ORDER TENSOR	00013140
	-	C = ESTOR(1)	00013140
		USTOR(1) = C	00013170
	C	FORM 1ST-ORDER TENSOR	00013180
		DO 100 I=1.N	00013100
		DI = DF(I)	00013300
ò		11 = 11 + 1	00013210
ώ		$CI \neq ESTOR(11)$	00013220
6		LOCII = N1 + I + (I+1)/2	00013230
	C	FORM X TYPE TERM	00013240
		USTOR(11) = (- 1.500*C + 2.00*CI500*FSTOR((0CTI))/DI	00013250
	C	FORM 2ND-ORDER TENSOR	00013260
		00 100 J=1,I	00013270
		$\theta J = DI \neq DF(J)$	00013280
		12 = 12+1	00013290
		CIJ = ESTOR(12)	00013300
		IF(J.LT.I)GO TO 79	00013310
	C	FORM XX TYPE TERM	00013320
		USTOR(12) = (C - 2.D0*C1 + CIJ)/DJ	00013330
		GD TO 100	00013340
	79	CJ = ESTOR(1+J)	00013350
	Ç	FURM YX TYPE TERM	00013360
		USTOR(12) = (C - CI - CJ + CIJ)/DJ	00013370
	100	CONTINUE	00013380
		RETURN	00013390
	"	END	00013400

	SUBROUTINE U3FORM(N+DF+ESTOR+USTOR)	00013410
C	FORM STRAIN ENERGY TENSORS USTOR USING CUBIC POINTS ESTOR.	00013420
C	N = DIMENSION OF TENSORS.	00013430
C	DF(I) = FINITE DIFFERENCE IN INDEPENDENT VARIABLE I.	00013440
C.	ESTOR(I) = STORAGE VECTOR FOR ENERGY EVALUATIONS I.	00013450
С	USTOR(I) = STORAGE VECTOR FOR TENSOR COMPONENT I.	00013460
	INTEGER N	00013470
	DOUBLE PRECISION DF(1), ESTOR(1), USTOR(1)	00013480
	INTEGER I.J.K.II.12.13.LOCII.LOCJJ.LOCIII.LOCIIJ.LOCIJJ.LOCIK.	00013490
	1LOCJK,N1,N2	00013500
	DCUBLE PRECISION C,CI,CJ,CK,CIJ,CIK,CJK,CIJK,CI16,C13,DI,DJ,DK	00013510
	N1 = 1 + N	00013520
	N2 = N1 + N + (N+1)/2	00013530
	C116 = 11.00/6.00	00013540
	C13 = 1.00/3.00	00013550
	II = 1	00013560
	I2 = N1	00013570
	I3 = N2	00013580
C	FORM OTH-ORDER TENSOR	00013590
	C = ESTOR(1)	00013600
	USTOR(1) = C	00013610
C	FORM 1ST-ORDER TENSOR	00013620
	DC 100 I=1.N	00013630
	DI = DF(1)	00013640
	I1 = I1+1	00013650
	$CI \neq ESTOR(11)$	00013660
	LOCII = N1 + I * (I+1)/2	00013670
	LOCIII = N2 + I*(I+1)*(I+2)/6	00013680
C	FORM X TYPE TERM	00013690
	USTOR(I1) = (- C116*C + C13*ESTOR(LOCIII) + 3.DO*CI	00013700
	1- 1.5DO#ESTOR(LOCII))/DI	00013710
C	FORM 2ND-ORDER TENSOR	00013720
	DG 100 J=1.I	00013730
	$DJ = DI \neq DF(J)$	00013740
	12 = 12 + 1	00013750
	CIJ = ESTOR(12)	00013760
	CJ = ESTOR(1+J)	00013770
_	IF(J.LT.I)GO TO 79	00013780
C	FORM XX TYPE TERM	00013790

ì

ų.

C-37

}

	USTOR(12) = (2.DO*C - ESTOR(LOCIII) - 5.DO*CI + 4.DO*CIJ)/DJ	00013800
	GC TO 81	00013810
	79 LOCJJ = N1 + $J*(J+1)/2$	00013820
	LDCIJJ = N2 + (I-1)*I*(I+1)/6 + (J+1)*J/2	00013830
	LCCIIJ = N2 + (I-1)*I*(I+4)/6 + J	00013840
С	FORM YX TYPE TERM	00013850
	USTOR(12) = (2.00*C - 2.500*CJ + .500*ESTOR(LOCJJ)	00013860
	15DO*ESTOR(LOCIIJ)5DO*ESTOR(LOCIJJ) + .5DO*ESTOR(LOCII)	00013870
	2 - 2.5D0+CI + 3.D0+CIJ)/DJ	00013880
C	FORM 3RD-ORDER TENSOR	00013890
	81 DC 100 K=1,J	00013900
	CK = DJ+DF(K)	00013910
	13 = 13 + 1	00013920
	CIJK = ESTOR(13)	00013930
	LOCIK = N1 + (I-1) + I/2 + K	00013940
	CIK = ESTOR(LOCIK)	00013950
	CK = ESTOR(1+K)	00013960
	IF(J.LT.I)GD TO 89	00013970
С	FORM XXX TYPE TERM	00013980
	$IF(K \cdot EQ \cdot J) \cup STOR(I3) = (-C + CIJK + 3 \cdot DO * CI - 3 \cdot DO * CIJ) / DK$	00013990
С	FORM YYX TYPE TERM	00014000
	$IF(K_LT_J)USTOR(I3) = (-C+CK+CIJK-CIJ + 2.DO+CI - 2.DO+CIK)/DK$	00014010
	GO TC 100	00014020
	89  LOCJK = N1 + (J-1) + J/2 + K	00014030
	CJK = ESTOR(LOCJK)	00014040
С	FORM YXX TYPE TERM	00014050
_	$[F(K \cdot EQ \cdot J)USTOR(I3) = (-C + 2 \cdot DO * CJ - CJK + CIJK + CI - 2 \cdot DO * CIJ)/DK$	00014060
С	FORM ZYX TYPE TERM	00014070
	$IF(K_{\bullet}LT_{\bullet}J)USTOR(T3) = (-C + CK+CI+CJ + CIJK - CIJ+CIK+CJK)/DK$	00014080
	100 CONTINUE	00014090
	RETURN	00014100
	END	00014110
	SUBROUTINE UFILL(II, ORD, ET, USTOR)	00014120
C	FILL USTOR OF ORDER ORD FOR ELEMENT II.	00014130
Ĉ	1I = ELEMENT NUMBER.	00014140
L	URD = MAXIMUM TENSOR ORDER TO BE USED FOR DIFFERENCE EXPANSION.	00014150
Č	EILLI = CURRENT VALUE OF STRAIN COMPONENT I.	00014160
C	USIUNII) = STORAGE VECTOR FOR TENSOR COMPONENT I.	00014170
С С	NS = NUMBER OF STRAIN COMPONENTS.	00014180
L	UPE = FINITE DIFFERENCE SIZE FOR STRAIN VARIABLES.	00014190

•

00014200
00014210
00014220
00014230
00014240
00014250
00014260
00014270
00014280
00014290
00014300
00014310

i.

	SUBROUTINE CFORM(Q,EGEOM,CMAT)	00014320
С	FORM 3X9 PRESSURE LOAD MATRIX CMAT.	00014330
C	Q(I) = CUMULATIVE NODAL DISPLACEMENT AT DOF I.	00014340
C	$EGEOM(J_{+}I) = GEOMETRIC PROPERTIES FOR ELEMENT I.$	00014350
C	CMAT = UNSYMMETRIC STIFFNESS PARTITION DUE TO PRESSURE LOADS	• 00014360
	DCUBLE PRECISION Q(1),EGEOM(1),CMAT(3,9)	00014370
	INTEGER I	00014380
	DOUBLE PRECISION CX2+CY2+CZ2+CX3+CY3+CZ3	00014390
	CX2 = EGEOM(1) + Q(4) - Q(1)	00014400
	CY2 = Q(5) - Q(2)	00014410
	CZ2 = Q(6) - Q(3)	00014420
	CX3 = EGEOM(3) + Q(7) - Q(1)	00014430
	CY3 = EGEDM(2) + Q(8) - Q(2)	00014440
	CZ3 = Q(9) - Q(3)	00014450
	DU 10 I=1,3	00014460
	CMAT(I,I) = 0,00	00014470
	$CMAT(\mathbf{I},\mathbf{I}+3) = 0_{\bullet}D0$	00014480
	10  CMAT(I,I+6) = 0.00	00014490
	CMAT(1,2) = CZ2-CZ3	00014500
	CMAT(1,3) = -CY2+CY3	00014510
	CMAT(1,5) = CZ3	00014520
	CMAT(1,6) = -CY3	00014530
	CMAT(1,8) = -CZ2	00014540
	CMAT(1,9) = CY2	00014550
	CMAT(2,1) = -CZ2+CZ3	00014560
	CMAT(2,3) = CX2-CX3	00014570

· .

.

C-39

}

,

		CMAT(2,4) = -CZ3	00014580
		CMAT(2,6) = CX3	00014590
		CMAT(2,7) = CZ2	00014600
		CMAT(2,9) = -CX2	00014610
		CMAT(3,1) = CY2-CY3	00014620
		CMAT(3,2) = -CX2+CX3	00014630
		CMAT(3,4) = CY3	00014640
		CMAT(3,5) = -CX3	00014650
		CMAT(3,7) = -CY2	00014660
		CMAT(3+8) = CX2	00014670
		RETURN	00014680
		END	00014690
		SUBROUTINE GENER8(II,K)	00014700
	C	FORM STIFFNESS MATRIX K FOR ELEMENT II IN NODAL COORDINATES.	00014710
	C	II.K = ELEMENT NUMBER, ELEMENT STIFFNESS MATRIX.	00014720
	ç	NS,NF = NUMBER OF STRAIN COMPONENTS,DOF PER NODE.	00014730
_	C	ELNO(J,I) = NODE NUMBERS FOR ELEMENT I.	00014740
ဂု	C	EGEOM(J,I) = GEOMETRIC PROPERTIES FOR ELEMENT I.	00014750
4	C	I(I) = THICKNESS OF ELEMENT I.	00014760
0	C	CCII) = CUMULATIVE NODAL DISPLACEMENT AT DOF I.	00014770
	C	IPRESS = NONCONSERVATIVE CODE = 0,1 FOR NO PRESSURE, PRESSURE.	00014780
	C	PR = ELEMENT PRESSURE INTENSITY FACTOR.	00014 <b>790</b>
	G	PRREF(I) = INTENSITY ON ELEMENT I FOR PRESSURE REFERENCE VECTOR.	00014800
		COMMEN/COMNS/NS/COMNF/NF/COMEL/ELNO/COMEG/EGEOM/COMT/T/COMQQ/QQ	00014810
		COMMON/COMIPR/IPRESS/COMPR/PR/COMPRR/PRREF	00014820
		INTEGER II, NS, NF, ELNO(3, 1), IPRESS	00014830
		DOUBLE PRECISION K(9,9),EGEOM(3,1),T(1),GQ(1),PR,PRREF(1)	00014840
		INTEGER 1, J, IU, NF2, NF3, II	00014850
		DCUBLE PRECISION V.E(3).S(3).DO(3.3).D(6).Q(9).A(3.6).	00014860
		1H(6.6),G(6,9),USTOR(10)	00014870
		DCUBLE PRECISION C,P6,CMAT(3,9)	00014880
		ECUIVALENCE(A(1),G(1))	00014890
		<pre>V = .5D0+T(II)+EGE0M(1,II)+EGE0M(2,II)</pre>	00014900
		$NF2 = NF \neq 2$	00014910
		NF3 = NF*3	00014920
		CALL QFILL(NF,ELNO(1,II),QQ,Q)	00014930
		CALL DFILL(NF,EGEOM(1,II),Q,D)	00014940
		CALL EFILL(NF,D,E)	00014950

.

.

.

٠

••

	CALL UFILL(II,2,E,USTOR)	00014960
	IU = 1	00014970
	DO 10 I=1,NS	00014980
	IU = IU + 1	00014990
	10 S(I) = USTOR(IU) * V	00015000
	DO 20 I=1,NS	00015010
	CO 20 J=1,I	00015020
	IU = IU + I	00015030
	DO(I+J) = USTOR(IU)*V	00015040
	$20 DO(J_{1}) = DO(I_{1})$	00015050
	CALL AFILL(NF,D,A,L)	00015060
	CALL MTRAN(DO,3,NS,A,6,NF2,H)	00015070
	DO 50 I=1,NF	00015080
·	II = NF + I	00015090
	$H(I_{*}I) = H(I_{*}I) + S(I)$	00015100
	H(I1,I1) = H(I1,I1) + S(2)	00015110
	$H(I_{+}II) = H(I_{+}II) + S(3)$	00015120
	50 H(11,1) = H(11,1) + S(3)	00015130
-	CALL GFILL(NF,EGEOM(1,II),G)	00015140
C C	CALL MTRAN(H,6,NF2,G,9,NF3,K)	00015150
41	IF(IPRESS-EQ-0)GD TO 101	00015160
F	CALL CFORM(Q,EGEOM(1,11),CMAT)	00015170
	P6 = PR*PRREF(II)/6.D0	00015180
	DO ICO I=1,NF	00015190
	DO 100 J±L,NF3	00015200
	$C = P6 + CMAT(T_{+}J)$	00015210
	K(I,J) = K(I,J) - C	00015220
	K(1+3,J) = K(1+3,J) - C	00015230
	100  K(1+6,J) = K(1+6,J) - C	00015240
	LUI LALL ROTK(NF,ELNO(1,II),K)	00015250
	REIUKN	00015260
	ENU	00015270

. )

i .

	SUBROUTINE USUM1(ORD,USTOR,N,Q,P)	00015280
С	FOR ORD=2 COMPUTE $P(1) = USTOR(1, j) * O(j)$ .	00015290
. <b>C</b>	FOR ORD=3 COMPUTE P(I) = USTOR(I,J,K)+Q(J)+Q(K).	00015300
¢	ORD = TENSCR ORDER TO BE USED FOR FINITE-DIFFERENCE EXPANSIONS.	00015310
C	USTOR = TENSOR STORAGE ARRAY.	00015320
C	N = TENSOR DIMENSION.	00015330

. }

	C	Q = TENSOR VECTOR ARGUMENT.	00015340
	Ľ	P = COMPUTED SUMMED VECTOR.	00015350
		INTEGER ORD, N	00015360
		DUUBLE PRECISION USTOR(1), P(1), Q(1)	00015370
		INTEGER 1, J, K, TU	00015380
		DUUBLE PRECISION QI,QJ,QK,QIJ,C	00015390
	100		00015400
	100	P(1) = 0.00	00015410
		10 ± 0	00015420
		IF(URU-GI-2)GO TO 201	00015430
			00015440
		U1 = U(1)	00015450
		DU 200 J=1,I	00015460
			00015470
		c = 0SIUR(10)	00015480
		$P(1) = P(1) + C \neq Q(J)$	00015490
	200	$IF(J \bullet NE \bullet I)P(J) = P(J) + C * QI$	00015500
	200	CUNTINUE	00015510
0	201	KETURN	00015520
. <u>1</u>	201		00015530
42		QI = Q(I)	00015540
			00015550
		QJ = Q(J)	00015560
	•	LIJ = UI∓UJ DD 200 v = 1 L	00015570
		$\frac{1}{2} \int \frac{1}{2} \int \frac{1}$	00015580
		WK = W(K)	00015590
		10 = 10 + 1	00015600
		U = USIUR(10)	00015610
		F(I) = F(I) + C = Q = Q = Q = Q = Q = Q = Q = Q = Q =	00015620
		$\frac{17(K \cdot E \psi \cdot J) (U - E I)}{2}$	00015630
		P(I) = P(I) + C = QJ = QK	00015640
		ICUTUTI ICALENTO POR	00015650
	01	$\frac{1}{1}$	00015660
		$P(J) = P(J) + 2 \cdot D0 \neq 0 \neq 0 \neq 0$	00015670
	71	$\frac{1}{2} \frac{1}{2} \frac{1}$	00015680
	1	F(K) = F(K) + (*Q)	00015690
		$\frac{1}{10} \frac{1}{10} \frac$	00015700
		$P(J) = P(J) + 2 \cdot D0 * C * Q[ * QK$	00015710
	300	P(K) = P(K) + C = Q(J)	00015720
	500		00015730

.

•

)

.

RETURN END

00015740 00015750

	SUBROUTINE USUM21(USTOR,N,Q1,Q2,P)	00015760
C	COMPUTE P(I) = USTOR(I,J,K)*Q1(J)*Q2(K).	00015770
C	USTOR = TENSOR STORAGE ARRAY.	00015780
C	N = TENSOR DIMENSION.	00015790
C	Q1,Q2 = TENSOR VECTOR ARGUMENTS.	00015800
С	P = COMPUTED SUMMED VECTOR.	00015810
	INTEGER N	00015820
	DOUBLE PRECISION USTOR(1),Q1(1),Q2(1),P(1)	00015830
	INTEGER I, J, K, IU	00015840
	DCUBLE PRECISION C	00015850
	DC 100 I=1,N	00015860
100	P(I) = 0.D0	00015870
	IU = 0	00015880
	DO 200 I=1.N	00015890
	00 200 J=1,I	00015900
	DO 200 K=1,J	00015910
	IU = IU+1	00015920
	C = USTOR(IU)	00015930
	$P(I) = P(I) + C \neq Q1(J) \neq Q2(K)$	00015940
	IF(K.EQ.J)GO TO 161	00015950
	P(I) = P(I) + C = Q1(K) = Q2(J)	00015960
	GO TO 171	00015970
161	IF(J.EQ.I)GO TO 2CO	00015980
	P(J) = P(J) + C*(Q1(I)*Q2(J)+Q1(J)*Q2(I))	00015990
	GO TO 200	00016000
171	$P(K) = P(K) + C \neq Q1(1) \neq Q2(J)$	00016010
	IF(J.EQ.I)GO TO 200	00016020
	P(J) = P(J) + C*(Q1(I)*Q2(K)+Q1(K)*Q2(I))	00016030
	P(K) = P(K) + C*Q1(J)*Q2(I)	00016040
200	CONTINUE	00016050
	RETURN	00016060
	END	00016070

ì

٦

.

SUBROUTINE PICOMP(NN,QQSTAR,QQDOT,PRO,PRI,PP) - COMPUTE 2ND ORDER FUNDAMENTAL LOAD TERM PP USING FUNDAMENTAL 00016080

C 00016090

c		
r r	UISPLAUEMENTS TREFERENCE VALUES OQSTAR AND RATES GODOT).	00016100
с С	NN = IUIAL SYSTEM DOF.	00016110
	UNSTARLIF = LURRENT CUMULATIVE NODAL DISPLACEMENT AT NODE 1.	00016120
с С	GUUUT(I) = NUDAL DISPLACEMENT RATE AT NODE I.	00016130
	PRU, PRI = PRESSURE FACTORS AT START, END OF LOAD STEP.	00016140
L C	PPILI = COMPUTED PSUEDO FORCE TERM AT DOF I.	00016150
L c	URD = MAXIMUM TENSOR ORDER TO BE USED FOR DIFFERENCE EXPANSION.	00016160
L C	NEL,NS,NF = NUMBER OF ELEMENTS,STRAIN COMPONENTS,DOF PER NODE.	00016170
L C	ELNU(J+L) = NODE NUMBERS FOR ELEMENT I.	00016180
ւ Հ	EGEUMIJ, I) = GEOMETRIC PROPERTIES FOR ELEMENT I.	00016190
С С	T(I) = THICKNESS OF ELEMENT I.	00016200
L C	IPRESS = NONCONSERVATIVE CODE = 0.1 FOR NO PRESSURE, PRESSURE.	00016210
L	PRREFIT) = PRESSURE INTENSITY ON ELEMENT I FOR PRESSURE VECTOR.	00016220
	CUMMON/COMORD/ORD/COMNEL/NEL/COMNS/NS/COMNF/NF	00016230
	CUMMON/COMEL/ELNO/COMEG/EGEOM/CONT/T	00016240
	COMMON/COMIPR/IPRESS/COMPRR/PRREF	00016250
	INTEGER NN, ORD, NEL, NS, NF, ELNO(3, 1), IPRESS	00016260
	DCUBLE PRECISION QQSTAR(1),QQDOT(1),PR0,PR1,PP(1),EGEOM(3,1),T(1)	+00016270
	1PRREF(1)	00016280
	INTEGER NF2,NF3,IU2,IU3,I,II,M	00016290
	DCUBLE PRECISION V,C,QSTAR(9),QDOT(9),DSTAR(6),DDOT(6),ESTAR(3),	00016300
	1EDUT(3),ASTAR(3,6),A1DCT(3,6),STCRA(3),STORB(3),STORM(6),G(6,9),	00016310
,	2P(9)+USTAR(20)	00016320
	DEUBLE PRECISION CU,C1,VSTAR(6),VDCT(6),V1(3),V2(3),V3(3)	00016330
	NF2 = NF + 2	00016340
	$NF3 = NF \neq 3$	00016350
	IU2 = 2 + NS	00016360
	IU3 = IU2 + NS + (NS + 1)/2	00016370
	DO 5 I=1,NN	00016380
	5 PP(1) = 0.00	00016390
	DO 1000 1I=1+NEL	00016400
~	V = .5D0*T(II)*EGECM(1,II)*EGEOM(2,II)	00016410
Ĺ	FORM FUNDAMENTAL REFERENCE QUANTITIES	00016420
	CALL QFILL (NF,ELNO(1,11),QQSTAR,QSTAR)	00016430
	CALL DFILL(NF,EGEOM(1,II),OSTAR,DSTAR)	00016440
	CALL AFILL (NF, DSTAR, ASTAR, 1)	00016450
	CALL EFILL (NF, DSTAR, ESTAR)	00016460
C	FURM STRESS-STRAIN TENSORS	00016470
~	CALL UFILLIII, ORD, ESTAR, USTAR)	00016480
Ċ	FORM FUNDAMENTAL RATE QUANTITIES	00016490

)

•

•

C-44

. .

CALL QFILL(NF,ELNO(1,11),QQDOT,QDOT) CALL DFILL(NF,EGEOM(1,11),QDOT,DDOT) 00016500 00016510 CALL AFILL(NF,CDOT,A1DUT,0) EC 30 I=1,NS 00016520 00016530 C = 0.0000016540 DO 25 M=1,NF2 00016550 25 C = C + ASTAR(I,M)\*DDUT(M) 00016560 30 ECOT(I) = C00016570 FORM STORM = DO PORTION OF DISPLACEMENT DERIVATIVE PSUEDO FURCES 00016580 C-00016590 CALL USUMI(2, USTAR(IU2), NS, EDOT, STORA) DC 110 I=1.NF2 00016600 00016610 C = 0.0000016620 DO 105 M=1,NS 105 C = C + STORA(M)\*ALDOT(M+I) 110 STORM(I) = 2.DO\*C DO 114 I=1.NS 00016630 00016640 00016650 C = 0.0000016660 00016670 DC 112 M=1,NF2 112 C = C + AlDOT(I,M) \* DDOT(M)00016680 00016690 114 STORB(I) = C00016700 CALL USUM1(2,USTAR(IU2),NS,STORB,STORA) 00016710 DC 120 [=1.NF2 00016720 C = 0.0000016730 DO 115 M=1.NS 115 C = C + STORA(M)\*ASTAR(M+I) 00016740 120 STORM(I) = STORM(I) + C IF(ORD.LT.3)GO TO 501 00016750 00016760 ACD DI PORTION OF DISPLACEMENT DERIVATIVE PSUEDO FORCES TO STORM 00016770 C CALL USUM1(3+USTAR(1U3)+NS+EDOT+STORA) 00016780 00016790 DO 210 I=L,NF2 C = 0.0000016800 00016810 205 C = C + STORA(M) \* ASTAR(M,I)210 STORM(I) = STORM(I) + C DO 205 M=1+NS 00016820 00016830 COMPUTE ELEMENT FORCES P(I) = VOLUME INTEGRAL OF G(M+I)\*STORM(M) 00016840 C. 501 CALL GFILL(NF,EGEOM(1,11),G) 00016850 00016860 DG 510 I=1,NF3 C = 0.00 D0 505 M=1.NF2 00016870 00016880 00016890 505 C = C + G(M,1)\*STORM(H)

	510	P(I) = C * V	00016900
	~	IF(IPRESS.EQ.0)GO TO 1000	00016910
	L	ACD NONCONSERVATIVE PRESSURE FORCES TO P	00016910
		VSTAR(1) = EGEOM(1, II) + QSTAR(4) - QSTAR(1)	00016920
		VSTAR(2) = QSTAR(5) - QSTAR(2)	00016950
		VSTAR(3) = QSTAR(6) - QSTAR(3)	00016940
		VSTAR(4) = EGEOM(3, 11) + QSTAR(7) - QSTAR(1)	00016950
		VSTAR(5) = EGEOM(2,11) + QSTAR(8) - QSTAR(2)	00016980
		VSTAR(6) = QSTAR(9) - QSTAR(3)	00016970
		VDOT(1) = QDOT(4) - QDOT(1)	00016980
		VCOT(2) = QDOT(5) - QDCT(2)	00010990
		VDOT(3) = QDOT(6) - QDOT(3)	00017000
		VDOT(4) = QDOT(7) - QDOT(1)	00017010
		VDCT(5) = QDCT(8) - QCCT(2)	00017020
		VCOT(6) = QDOT(9) - QDCT(3)	00017030
		CO = PRO * PRREF(II)/3.DO	00017040
		C1 = (PR1-PR0) * PRREF(II)/3.D0	00017050
		CALL VCROSS(VDOT, VSTAR(4), V1)	00017060
_		CALL VCROSS(VSTAR, VDOT(4), V2)	00017070
3		CALL VCROSS(VDOT, VDOT(4), V3)	00017080
-		80 610 1=1,3	00017090
1		$C = C1 \neq (V1(1) + V2(1)) + C0 \neq V3(1)$	00017100
		P(I) = P(I) - C	00017110
		P(1+3) = P(1+3) - C	00017120
	610	P(1+6) = P(1+6) - C	00017130
	C	ACD ELEMENT FORCES P TO SYSTEM FORCES OF	00017140
	1000	CALL PFILL(NF,ELNC(1.II).P.PP)	00017150
		RETURN	00017160
		END	00017170
			00017180
		SURPOUTINE DATECTIONATED AND WERE ADD	
	r	COMPLETE IST AND DATE DATE DATE TO PRO, PRI, LSIGN, RL, RRL, RQ, RRQ)	00017190
	c	LEAD DAPAMETER DATES OF DALLAS FUR LEAD DIRECTION VECTOR RP.	00017200
	ř	KNAT - SYSTEM ACCOUTAN CETTER DISPLACEMENT RATES = RO, RRO.	00017210
	č	PI = TEMPORARY EDDER STORAGE MEGTOR	00017220
	с Г	NN = TOTAL SYSTEM NOT	00017230
	č	HN - TUTHL STSTEM DUF. KED(1) - EDDCK-DICOLACENENT CONSTRANT -	00017240
	č	REALLY - PESTONAL LEAD (LOND CHORNERS INT SPECIFICATION FOR DOF I.	00017250
	ř	PRO-DRI - RESEDUAL LUAD (LUAD STEP) AT DOF I.	00017260
	<b>v</b>	THOTTAL - PRESSURE FALTURS AT START, END OF LOAD STEP.	00017270

.

C-46

•

С	LSIGN = ++- IF LOAD PARAMETER IS INCREASING.DECREASING.		00017280
C	RL,RRL = CCMPUTED LOAD PARAMETER 1ST, 2ND ORDER RATES.		00017290
C	RQ(I), RRQ(I) = COMPUTED 1ST, 2ND ORDER DISPLACEMENT RATE	AT DOF I.	00017300
C	GG(I) = CURRENT CUMULATIVE DISPLACEMENT AT DOF I.		00017310
	COMMON/COMPQ/QQ		00017320
	INTEGER NN, KFD(1)		00017330
	DOUBLE PRECISION RP(1), PRO, PR1, LSIGN, RL, RRL, RG(1), RRC(1)	.00(1).	00017340
	1KMAT(NN, NN), P1(1)	•	00017350
	INTEGER I		00017360
	DOUBLE PRECISION RSIGN		00017370
	CALL SOLVE(KMAT, NN, KFD, RP, RQ)		00017380
	CALL PICOMP(NN,QQ,RQ,PRO,PR1,P1)		00017390
	RL = 0.00		00017400
	RRL = 0.00		00017410
	DC 50 I=1,NN		00017420
	IF(KFD(I)+LT+O)GO TO 49		00017430
	$RL = RL + RP(I) \neq RG(I)$		00017440
	RRL = RRL + P1(I) + RQ(I)		00017450
	GC TC 50		00017460
	49 P1(I) = 0.00		00017470
	50 CONTINUE		00017480
	CALL SOLVE(KMAT,NN,KFD,P1,RRQ)		00017490
	DO 55 I=1,NN		00017500
	IF(KFD(I).GT.O)RRL = RRL + RP(I)*RRQ(I)		00017510
	55 CONTINUE		00017520
	RSIGN = 1.DO		00017530
	$IF(RL \cdot LT \cdot O \cdot DO)RSIGN = -1 \cdot DO$		00017540
	RL = RSIGN/RL		00017550
	RRL = RSIGN*RL**2*RRL*.5D0		00017560
	DC 60 I=1,NN		00017570
	$60 \ RRQ(I) = RRL + RQ(I) - RL + RRQ(I)$		00017580
	$RL = LSIGN \neq DSQRT(RL)$		00017590
	DU BO I=1,NN		00017600
	80  RG(1) = RL + RG(1)		00017610
	RETURN		00017620
	ENU		00017630

SUBROUTINE STEP(LSIGN,RL,RRL,NN,RQ,RRQ,JUMPR,MJUMP,NJUMP,DSLOPE, 00017640 1PATH,LAMS) 00017650

.

C-47

.

)

	C	COMPUTE PATH = PATH DISTANCE, LAMS = LOAD STEP SIZE.	00017660
	۲ د	LSIGN = +,- IF LOAD PARAMETER IS INCREASING, DECREASING.	00017670
	C.	RL,RRL = 1ST,2ND ORDER LOAD PARAMETER RATES.	00017680
	C	NN = TOTAL SYSTEM DOF.	00017690
	C	RQ(I), RRQ(I) = 1ST, 2ND ORDER DIAPLACEMENT RATES AT DOF I.	00017700
	L	JUMPR = FRACTION OF LOAD INCREMENT PRECEEDING LIMIT POINT	00017710
	C	AT WHICH LIMIT IS TO BE TRAVERSED.	00017720
	Ç	MJUMP = MAXIMUM NUMBER OF INCREMENT DIVISIONS TO PERFORM WHEN	00017730
	C	NEARING A LIMIT POINT.	00017740
	C	NJUMP = CURRENT NUMBER OF INCREMENT DIVISIONS TO PERFORM WHEN	00017750
	C	NEARING A LIMIT POINT.	00017760
	C	DSLOPE = MAXIMUM SLUPE RATIO (CHANGE/AVERAGE) DURING LOAD STEP.	00017770
	C	PATH = COMPUTED PATH STEP SIZE TO BE TAKEN.	00017780
	C	LAMS = INPUT MAXIMUM, COMPUTED ACTUAL LOAD STEP SIZE.	00017790
		INTEGER NN, MJUMP, NJUMP	00017800
		DOUBLE PRECISION LSIGN, RL, RRL, RQ(1), RRQ(1), JUMPR, DSLOPE, PATH, LAMS	00017810
		INTEGER N, I	00017820
		DOUBLE PRECISION SLIM, LAMLIM, PREC, DSLOP, CR, CRR, C	CC017830
0		IF(LSIGN*RL.LE.O.DO)STOP 601	00017840
Ì	<u>_</u>	IF(RL*RRL.GE.O.DO)GO TO 21	00017850
40	C	POSSIBLE LIMIT POINT	00017860
		SLIM = -RL/RRL	00017870
	~	LAMLIM = DABS(SLIM*RL + .5DO*SLIM**2*RRL)	00017880
	C	CHECK CLOSENESS OF LIMIT POINT	00017890
		IF(LAMLIM.LT.JUMPR)GO TO 11	00017900
	-	IF(LAMS.LT.LAMLIM/MJUMP)GO TO 21	00017910
	Ĺ	LIMIT IS CLOSE. TAKE FRACTIONAL STEP JUMP	00017920
•		NJUMP = NJUMP-1	00017930
		IF(NJUMP.LT.2)NJUMP = 2	00017940
	-	IF(LAMS.LT.LAMLIM)GO TO 7	00017950
	L	STEP IS LARGER THAN LIMIT. JUMP TOWARD LIMIT VALUE.	00017960
		LAMS = LAMLIM/NJUMP	00017970
	~	GU TO 10	00017980
	C	STEP IS SMALLER THAN LIMIT. JUMP TOWARD STEP VALUE.	00017990
		/ N = NJUMP*LAMS/LAMLIM + 1	00018000
		IF(N.EQ.I)NJUMP = NJUMP+1	00018010
	_	LAMS = LAMS/N	00018020
	1	U LAMS = LSIGN+LAMS	00018030
		PATH = (-RL+LSIGN*DSQRT(RL**2+2.DO*LAMS*RRL))/RRL	00018040
		RETURN	00018050

}

-

.

C		LIMIT IS VERY CLOSE. TRAVERSE THE LIMIT POINT	00018060
	11	NJUMP = MJUMP+1	00018070
		LSIGN = -LSIGN	00010010
		LAMS = 0.00	00010000
		$PATH = -2.00 \pm RI / RRI$	00010090
		RETURN	00018100
c		LINIT IS NOT CLOSE CHECK SLODE CHANGE FOR ALLOWADLE STER	00018110
C	21	N TUND - N TUNDAD	00018120
	21	AJUME = MJUMEFI CD = A DA	00018130
		UK = 0.00	00018140
		$LKK \neq 0.00$	00018150
		UC 50 I=1,NN	00018160
		CR = CR + DABS(RQ(I))	00018170
	50	CRR = CRR + DABS(RRQ(I))	00018180
		LAMS = LSIGN+LAMS	00018190
	51	PREC = DABS{2.DO*LAMS*RRL/RL**2}	00018200
		IF(PREC.LT.1.D-8)PATH = LAMS/RL	00018210
		IF(PREC.GE.1.D-8)PATH = (-RL+LSIGN*DSQRT(RL**2+2.DO*LAMS*RRL))/RRL	00018220
		DSLOP = PATH*RRL	00018230
		DSLOP = DSLOP/(RL+.5DO*DSLOP)	00018260
		C = PATH + CRR	00010240
		C = C/(CR+.5D0+C)	00010250
		TEIDABS(DSLOP) LE DSLOPE AND C LE DSLODENDETDON	00018260
		IAMS = .500±1AMS	00018270
		TE(DARC(LANC)   E   D_2)CTOD (00)	00018280
		TO TO EL	00018290
			00018300
			00018310

SUBROUTINE EIGENI(NN,NCODE,QQDELT,SCRIT,GQPOST,LAM,PRO,PRI,PP) 00018320 С FORM POSTBUCKLING LOAD TERM PP USING FUNDAMENTAL DISPLACEMENTS 00018330 (REFERENCE VALUES COSTAR AND CRITICAL INCREMENT OODELT) AND C 00018340 POSTBUCKLING DISPLACEMENT EIGENVECTOR COPOST. С 00018350 С NN = TOTAL SYSTEM DOF. 00018360 NCODE = 0,1 FOR LINEAR, NONLINEAR EIGEN SOLUTION. С 00018370 С CQDELT(I) = ESTIMATED INCR. DISPLACEMENT AT DUF 1 TO CRITICAL PT. 00018380 С SCRIT = ESTIMATED INCREMENTAL PATH VALUE TO CRITICAL POINT. 00018390 QQPOST(1) = ESTIMATED BUCKLING DISPLACEMENT AT DOF 1. С 00018400 LAM = ESTIMATED INCREMENTAL LOAD PARAMETER VALUE TO CRITICAL PT. С 00018410 С PRO, PR1 = PRESSURE FACTORS AT START, END OF LCAD STEP. 00018420 PP(I) = COMPUTED RIGHT-HAND-SIDE FOR INVERSE POWER ITERATION. C 00018430

C-49

)

C	ORD = MAXIMUM TENSOR ORDER TO BE USED FOR DIFFERENCE EXPANSION.	00018440
C	NEL,NS,NF = NUMBER OF ELEMENTS,STRAIN COMPONENTS,DOF PER NODE.	00018450
С	ELNO(J,I) = NODE NUMBERS FOR ELEMENT I.	00018460
C	EGEOM(J,I) = GEOMETRIC PROPERTIES FOR ELEMENT I.	00018470
C	T(I) = THICKNESS OF ELEMENT I.	00018480
С	COSTAR(I) = CURRENT CUMULATIVE DISPLACEMENT AT DOF I.	00018490
C	IPRESS = NONCONSERVATIVE CODE = 0,1 FOR NO PRESSURE, PRESSURE.	00018500
C	PRREF(I) = PRESSURE INTENSITY ON ELEMENT I FOR PRESSURE VECTOR.	00018510
	COMMEN/COMORD/ORD/COMNEL/NEL/COMNS/NS/COMNF/NF	00018520
	COMMON/COMEL/ELNO/COMEG/EGEOM/COMT/T/COMQQ/QQSTAR	00018530
	COMMEN/COMIPR/IPRESS/COMPRR/PRREF	00018540
	INTEGER NN,NCODE,ORD,NEL,NS,NF,ELNC(3,1),1PRESS	00018550
	DOUBLE PRECISION QQDELT(1),SCRIT,QQPOST(1),LAM,PRO,PR1,PP(1)	00018560
	DCUBLE PRECISION EGEOM(3,1),T(1),QQSTAR(1),PRREF(1)	00018570
	INTEGER NF2,NF3,IU2,IU3,I,II,M	00018580
	DOUBLE PRECISION V,C,QSTAR(9),QDELT(9),CPOST(9),DSTAR(6),DDELT(6	,00018590
	1DPOST(6),ESTAR(3),EDELT(3),EPOST(3),ASTAR(3,6),A1DELT(3,6),	00018600
	2A1POST(3,6),STORA(3),STORB(3),STORM(6),G(6,9),P(9),USTAR(20),CA	00018610
	DOUBLE PRECISION CO,C1,VSTAR(6),VDELT(6),VPOST(6),V1(3),V2(3),	00018620
	1V3(3),V4(3)	00018630
	NF2 = NF*2	00018640
	NF3 = NF*3	00018650
	IU2 = 2 + NS	00018660
	103 = 102 + NS*(NS+1)/2	00018670
	DC 5 1=1,NN	00018680
	5 PP(1) = 0.00	00018690
	CO 1000 II=1,NEL	00018 <b>70</b> 0
_	<pre>V = .5D0*T(II)*EGEOM(1,II)*EGEOM(2,II)</pre>	00018710
C	FORM FUNDAMENTAL REFERENCE QUANTITIES	00018720
	CALL QFILL(NF,ELNO(1,II),QQSTAR,QSTAR)	00018730
	CALL DFILL(NF,EGEOM(1,II),GSTAR,CSTAR)	00018740
	CALL AFILL(NF, DSTAR, ASTAR, 1)	00018750
_	CALL EFILL(NF,DSTAR,ESTAR)	00018760
C	FORM STRESS-STRAIN TENSORS	00018770
-	CALL UFILL(II, ORD, ESTAR, USTAR)	00018780
C	FORM FUNDAMENTAL CRITICAL INCREMENT QUANTITIES	00018790
	LALL WFILL(NF,ELNG(1,II),QQDELT,QDELT)	00018800
	LALL DFILL(NF,EGEOM(1,II),QDELT,DDELT)	00018810
	CALL AFILL(NF,DDELT,A1DELT,0)	00018820
	DU 20 I=1+NS	00018830

C−50
			C = 0.00	00018840
			DO 15 M=1,NF2	00018850
			CA = ASTAR(I,M)	00018860
			IF(NCODE.GT.O)CA = CA + .5DO*SCRIT*A1DELT(1.M)	00018870
		15	$C = C + CA \neq DDELT(M)$	00018880
		20	EDELT(I) = C	00018890
	C		FORM POSTBUCKLING EIGEN QUANTITIES /	00018900
			CALL QFILL(NF,ELNO(I,II),QQPOST,QPOST)	00018910
			CALL DFILL(NF,EGEOM(1,II),QPOST,DPOST)	00018920
			CALL AFILL(NF, DPOST, A1POST, 0)	00018930
			DC 30 I=1,NS	00018940
			C = 0.00	00018950
			DO 25 M=1,NF2	00018960
			CA = ASTAR(I,M)	00018970
			$IF(NCODE_GT_O)CA = CA + SCRIT*AIDELT(I_M)$	00018980
		25	$C = C + CA \neq DPOST(M)$	00018990
	_	30	EPOST(I) = C	00019000
	C		FORM STORM = DO PORTION OF DISPLACEMENT DERIVATIVE PSUEDO FORCES	00019010
			DC 104 I=1,NS	00019020
C C			C = 0.00	00019030
ப்			DC 102 M=1,NF2	00019040
		102	$C = C + AIDELI(I_{\bullet}M) * DPLSI(M)$	00019050
		104	SIUKB(1) = C	00019060
			LALL USUMITZ+USTAKTIUZ),NS,STURB,STURA)	00019070
			DU 110 1=1,NF2	00019080
			$U = U_{\bullet} U $	00019090
		105	UU = 100  M = 1  (N)	00019100
		110	C = C + SIUKA(M) + ASIAK(M+1)	00019110
		110	STURMALE - C Care Distantes (1112) as Ednet Studat	00019120
•			DD 120 1-1.NE2	00010140
			C = 0.00	00019140
			DO 115 M=1-NS	00019150
		115	C = C + STORA(M) * A OFT T(M, T)	00019180
		120	C = C + C + C + C	00019170
			CALL USUM1 (2.11STAR (1112) - NS. EDELT. STORA)	00019180
			DO 130 I=1.NE2	00019190
			C = 0.00	00019200
			DO 125 M=1.NS	00019210
		125	$C = C + STORA(M) * Alpost(M \cdot I)$	00019230

	130	STORM(I) = STORM(I) + C	00019240	
		IF(ORD+LT-3)GO TO 501	00019250	
С		ACD D1 PORTION OF DISPLACEMENT DERIVATIVE PSUEDO FORCES TO STORM	00019260	
		CALL USUM21(USTAR(IU3),NS,EPOST,EDELT,STORA)	00019270	
		DC 210 I=1,NF2	00019280	
		C = 0.D0	00019290	
		DO 205 M=1,NS	00019300	
	205	C = C + STORA(M) * ASTAR(M, I)	00019310	
	210	STORM(I) = STORM(I) + C	00019320	
		IF(NCODE.EQ.0)GO TO 501	00019330	
		CO 220 I=1,NF2	00019340	
		C = 0.D0	00019350	
		DO 215 M=1,NS	00019360	
	215	$C = C + STORA(M) \neq A1DELT(M, I)$	00019370	
	220	STORM(I) = STORM(I) + SCRIT C	00019380	
		CALL USUMI (3, USTAR (1U3), NS, EDELT, STORA)	00019390	
		DU 230 I=1,NF2	00019400	
		$\mathbf{C} = 0 \cdot \mathbf{D}0$	00019410	
		DC 225 M=1,NS	00019420	
	225	$C = C + STORA(M) \neq Alpost(M, I)$	00019430	
	230	STORM(I) = STORM(I) + .5DO * SCRIT * C	00019440	
С		COMPUTE ELEMENT FORCES P(I) = VOLUME INTEGRAL OF G(M, 1)*STORM(M)	00019450	
	501	LALL GFILL(NF,EGEOM(1,II),G)	00019460	
		DO 510 I=1,NF3	00019470	
		$\mathbf{C} = 0 \cdot \mathbf{D} 0$	00019480	
		DD 505 M=1+NF2	00019490	
	505	$C = C + G(M_{\gamma}I) \neq STORM(M)$	00019500	
	510	P(1) = (xy)	00019510	
~	•	IF(IPRESS.EQ.0)GO TO 1000	00019520	
C		ACU NUNCONSERVATIVE PRESSURE FORCES TO P	00019530	
		VSTAR(1) = EGEOM(1, II) + QSTAR(4) - QSTAR(1)	00019540	
		VSIAR(2) = QSTAR(5) - QSTAR(2)	00019550	
		VSTAR(3) = QSTAR(6) - QSTAR(3)	00019560	
		VSTAR(4) = EGEOM(3+II) + QSTAR(7) - QSTAR(1)	00019570	
		VSTAR(5) = EGEOM(2, II) + QSTAR(8) - QSTAR(2)	00019580	
		VSTAR(6) = QSTAR(9) - QSTAR(3)	00019590	
		VUELT(1) = QUELT(4) - QDELT(1)	00019600	
		VUELT(Z) = QUELT(5) - QUELT(2)	00019610	
		VUELI(3) = QUELT(6) - QUELT(3)	00019620	
		VUELT(4) = QDELT(7) - QDELT(1)	00019630	

		VCELT(5) = QDELT(8) - QDELT(2)	00019640
		VDELT(6) = QDELT(9) - QDELT(3)	00019650
		VPOST(1) = QPOST(4) - QPOST(1)	00019660
		VPOST(2) = QPOST(5) - QPOST(2)	00019670
		VPOST(3) = QPOST(6) - QPOST(3)	00019680
		VPOST(4) = QPOST(7) - QPOST(1)	00019690
		VPOST(5) = QPOST(8) - QPOST(2)	00019700
		VPOST(6) = QPOST(9) - QPOST(3)	00019710
		CALL VCROSS(VSTAR, VPOST(4), V1)	00019720
		CALL VCROSS(VPOST,VSTAR(4)+V2)	00019730
		CALL VCROSS(VDELT, VPOST(4), V3)	00019740
		CALL VCROSS(VPOST, VDELT(4), V4)	00019750
		$CO = LAM \neq (PR1-PRO) \neq PRREF(II)/6.DO$	00019760
		C1 = PRO*PRREF(11)/6.CO	00019770
		$IF(NCODE GT \cdot O)CI = CI + SCRIT CO$	COO19780
		DO 610 1=1,3	00019790
		C = CO*(V1(I)+V2(I)) + C1*(V3(I)+V4(I))	00019800
		P(I) = P(I) - C	00019810
_		P(I+3) = P(I+3) - C	00019820
C T	610	P(I+6) = P(I+6) - C	00019830
ហ	C	ACD ELEMENT FORCES P TO SYSTEM FORCES PP	00019840
	1000	CALL PFILL(NF,ELNO(1,II),P,PP)	00019850
		KETURN	00019860
		END	00019870

	SUBROUTINE EIGEN(UU,KMAT,P1,CO,NN,KFD,PRO,PR1,RL,RRL,RQ,RRQ,	00019880
	IMIL,MI2,ERR1,ERR2,EIGL,Q1,N)	00019890
ú	SOLVE FOR EIGEN LOAD EIGI, AND VECTOR Q1 WITH MAX. INDEX VALUE N.	00019900
C	UD = OUTPUT UNIT FILE.	00019910
C	KMAT = SYSTEM JACOBIAN STIFFNESS MATRIX.	00019920
C	P1 = TEMPORARY FORCE STORAGE VECTOR.	00019930
С	QO ≠ TEMPORARY DISPLACEMENT STORAGE VECTOR.	00019940
C	NN = TOTAL SYSTEM DOF.	00019950
С	<pre>KFD(I) = FORCE-DISPLACEMENT-CONSTRAINT SPECIFICATION FOR DOF 1.</pre>	00019960
C	PRO,PR1 = PRESSURE FACTORS AT START,END OF LCAD STEP.	00019970
C	RL,RRL = IST,2ND CRDER LOAD PARAMETER RATES.	00019980
C	RG(I),RRQ(I) = 1ST,2ND ORDER DISPLACEMENT RATE AT DOF I.	00019990
C	MIL,MI2 = MAXIMUM ITERATIONS FOR LINEAR, LINEAR+NONLINEAR SOLUTION.	.00020000
C	ERRI,ERR2 = MAXIMUM ERROR FOR LINEAR,LINEAR+NONLINEAR SOLUTION.	00020010

 $\begin{array}{ccc} C & c \\ \frac{1}{53} & C \\ \frac{1}{53} & 10 \end{array}$ 

•

		•		
		C	EIGL = COMPUTED EIGENVALUE (CRITICAL PATH VALUE).	00020020
		C	Q1(I) = ITH COMPONENT OF EIGENVECTOR (BUCKLING DISPLACEMENT).	00020030
		Ç	N = DOF OF LARGEST Q1 COMPONENT.	00020040
			COMMON/COMNF/NF	00020050
			INTEGER UD,NN,KFD(1),MI1,MI2,N	00020060
			DOUBLE PRECISION KMAT(NN,NN),P1(1),Q0(1),PR0,PR1,	00020070
•			<pre>lRL,RRL,RQ(1),RRQ(1),ERR1,ERR2,EIG1,Q1(1)</pre>	00020080
			INTEGER I, NI	00020090
			DOUBLE PRECISION EIGO,MAX1,DL,C	00020100
		20	)1 FORMATC1H ,'EIGENVALUE = ',D15.8,5X,'DOF OF LARGEST COMPONENT OF	E00020110
			1IGEN VECTOR = ',15)	00020120
			NI = 0	00020130
			NCODE = 0	00020140
			$EIG1 = 0 \cdot D0$	00020150
			00 5 I=1+NN	00020160
			5 QI(I) = I	00020170
		1	.1 EIGO = EIG1	00020180
			DL = RL	00020190
			IF(NCODE.GT.O)DL = DL + .5DO*EIGO*RRL	00020200
•	ò		DO 20 I=1,NN	00020210
	ហ		QO(I) = QI(I)	00020220
	4		Q1(I) = RQ(I)	00020230
			$IF(NCODE_GT_0)Q1(I) = Q1(I) + .5D0 \neq EIG0 \neq RRQ(I)$	00020240
		2	20 CONTINUE	00020250
			CALL EIGENI(NN,NCODE,Q1,EIG0,00,DL,PR0,PR1,P1)	00020260
			NOD = NN/NF	00020270
			CALL OUTPQ(6,NOD,NF,P1,Q0)	00020280
			DC 50 I=1,NN	00020290
			IF(KFD(1),LT,O)P1(1) = 0,D0	00020300
		5	O CONTINUE	00020310
			CALL SOLVE(KMAT,NN,KFD,P1,Q1)	00020320
			N = 0	00020330
			MAX1 = 0.D0	00020340
			D0 70 I≠1,NN	00020350
			IF(DABS(Q1(I)).LE.DABS(MAX1))GO TO 70	00020360
			$N = \mathbf{I}$	00020370
			MAX1 = Ql(I)	00020380
		7	O CONTINUE	00020390
			MAX1 = 1.DO/MAX1	00020400
			EIGI = -MAXI	00020410

• 1

	LF(KFD(N).NE.N)N = KFD(N)	00020420
	DC 1CO I=1,NN	00020430
100	$Q1(I) = Q1(I) \neq MAXI$	00020440
	WRITE(U0+201)EIG1+N	00020450
	NI = NI + 1	00020460
	C = DABS((EIGI-EIGO)/EIGI)	00020470
	IF(NCODE.GT.O.AND.(NI.GE.MI2.OR.C.LE.ERR2))RETURN	00020480
	IF(NCUDE.EG.O.AND.(NI.GE.MII.OR.C.LE.ERRI))NCODE = 1	00020490
	GO TO 11	00020500
	END	00020510

i

00020790

SUBROUTINE POST2(NN,QQCRIT,QQDCT,GQPOST,LCRIT,LDOT,PRO,PR1,PP,	00020520
1KODE)	00020530
FORM 2ND ORDER POSTBUCKLING LOAD TERM PP (=P21,P22 FCR KODE=1,2).	00020540
NN = TOTAL SYSTEM DOF.	00020550
QQCRIT(I) = PREDICTED TOTAL DISPLACEMENT OF DOF I AT CRITICAL PT.	00020560
CCDOT(1) = PREDICTED DISPLACEMENT RATE OF DOF 1 AT CRITICAL PT.	00020570
GOPOST(I) = ITH DISPLACEMENT OF CRITICAL BUCKLING EIGENVECTOR.	00020580
LCRIT = PREDICTED LOAD PARAMETER VALUE AT CRITICAL POINT.	00020590
LDOT = PREDICTED LOAD PARAMETER RATE AT CRITICAL POINT.	00020600
PRO, PRI = PRESSURE FACTORS AT START OF LOAD STEP, CRITICAL POINT.	00020610
PP(I) = COMPUTED PSUEDG-FORCE TERM AT DOF I.	00020620
KCDE = CODE FOR DUAL USE OF SUBROUTINE.	00020630
ORD = MAXIMUM TENSOR ORDER TO BE USED FOR DIFFERENCE EXPANSION.	00020640
NEL,NS,NF = NUMBER OF ELEMENTS,STRAIN COMPONENTS,DOF PER NODE.	00020650
ELNO(J,I) = NODE NUMBERS FOR ELEMENT I.	00020660
EGEOM(J,I) = GEOMETRIC PROPERTIES FOR ELEMENT I.	00020670
T(I) = THICKNESS OF ELEMENT I.	00020680
IPRESS = NONCONSERVATIVE CODE = 0,1 FOR NO PRESSURE, PRESSURE.	00020690
PRREF(I) = PRESSURE INTENSITY ON ELEMENT I FOR PRESSURE VECTOR.	00020700
COMMON/COMORD/ORD/COMNEL/NEL/COMNS/NS/COMNF/NF	00020710
COMMON/COMEL/ELNO/COMEG/EGEOM/COMT/T	00020720
COMMON/COMIPR/IPRESS/COMPRR/PRREF	00020730
INTEGER NN,KODE,ORD,NEL,NS,NF,ELNC(3,1),IPRESS	00020740
DCUBLE PRECISION GGCRIT(1), GQDOT(1), GQPOST(1), LCRIT, LDOT, PRO, PRI,	00020750
1PP(1),EGEOM(3,1),T(1),PRREF(1)	00020760
INTEGER NF2,NF3,IU2,IU3,I,II,M	00020770
DOUBLE PRECISION V,C,QCRIT(9),QDCT(9),QPOST(9),DCRIT(6),DDOT(6),	00020780

UUUBLE PRECISION V;C;QCRIT(9);QDCT(9);QPOST(9);DCRIT(6);DDOT(6) 1DPOST(6);ECRIT(3);EDOT(3);EPOST(3);ACRIT(3;6);A1DOT(3;6);

C-55

ն Շ Շ

C C C C C C

C C

١

	1ALPOST(3,6), STORA(3), STORB(3), STORM(6), G(6,9), P(9), UCRIT(20)	00020800
	ULUBLE PRECISION CO,C1,VCRIT(6),VDCT(6),VPOST(6),V1(3),V2(3),	00020810
		00020820
	NF2 = NF72	00020830
	NF3 = NF73	00020840
	$102 \neq 2 + NS$	00020850
	103 = 102 + NS*(NS+1)/2	00020860
	UU 5 [=1,NN	00020870
5	PP(1) = 0.00	00020880
	DU 1000 II=1,NEL	00020890
<u> </u>	V = .5D0 + T(II) + EGECM(1, II) + EGEOM(2, 1I)	00020900
L	FCRM FUNDAMENTAL CRITICAL QUANTITIES	00020910
	CALL QFILL(NF,ELNC(1,II),QQCRIT,QCRIT)	00020920
1901	FORMAT(1H0,9E14.7)	00020930
	WRITE(6,1901)(QCRIT(1),1=1,NF3)	00020940
	CALL DFILL(NF, EGEOM(1, II), QCRIT, DCRIT)	00020950
	WRITE(6,1901)(DCRIT(I),I=1,NF2)	00020960
	CALL AFILL(NF, DCRIT, ACRIT, 1)	00020970
1010	00 1910 1=1,3	00020980
1910	WRITE(6,1901)(ACRIT(I,J),J=1,NF2)	00020990
	CALL EFILL(NF, DCRIT, ECRIT)	00021000
~	WRITE(6,1901)(ECRIT(I),I=1,3)	00021010
<b>ل</b>	FURM STRESS-STRAIN TENSORS	00021020
~	LALL UFILL(II,ORD,ECRIT,UCRIT)	00021030
L	FURM FUNDAMENTAL RATE QUANTITIES	00021040
	CALL QFILL(NF,ELNO(1,II),QQDOT,QDOT)	00021050
	WR11E16,1901)(QDOF(I),I=1,NF3)	00021060
	LALL DFILL(NF,EGEOM(1,[I),GDCT,DDOT)	00021070
	WRIIE(6,1901)(DDOT(1),1=1,NF2)	00021080
	CALL AFILL(NF, DDOT, A1DOT, O)	00021090
1000	DU 1920 I=1,3	00021100
1920	WRIFE(6, 1901)(A1DOT(I, J), J=1, NF2)	00021110
	DO 20 1=1,NS	00021120
	C = 0.00	00021130
•		00021140
15	L = L + AURII(I,M)*DDCT(M)	00021150
20	EUU1111 = U	00021160
c	WKIJE (0+1901)(EUU((1)+1=1+3))	00021170
ι L	FURM PUSTBUCKLING EIGEN QUANTITIES	00021180
	CALL WHILEINF, ELNU(1,11), QQPOST, QPOST)	00021190

C-56

o ·

• • ••

WRITE(6,1901)(QPOST(1),1=1,NF3) 00021200 CALL DFILL(NF,EGEUM(1,II),QPOST,DPOST) 00021210 00021220 WRITE(6,1901)(DPOST(I),1=1,NF2) 00021230 CALL AFILL (NF, DPOST, AIPOST, 0) 00021240 DO 1930 I=1,3 00021250 1930 WRITE(6,1901)(A1POST(I,J),J=1,NF2) 00021260 DO 30 I=1.NS 00021270 C = 0.0000021280 UO 25 M=1+NF2 25 C = C + ACRIT(I,M) + DPOST(M)00021290 00021300 30 EPOST(I) = C00021310 WRITE(6,1901)(EPOST(1),1=1,3) FORM STORM = DO PORTION OF DISPLACEMENT DERIVATIVE PSUEDO FORCES 00021320 C 00021330 00 104 I=1.NS 00021340 C = 0.0000021350 DO 102 M=1,NF2 00021360 102 C = C + AlDOT(I,M) \* DPOST(M)00021370 104 STORB(I) = CCALL USUM1(2,UCRIT(IU2),NS,STORB,STORA) 00021380 00021390 DO 110 I=1,NF2 00021400 C = 0.0000021410 DC 105 M=1.NS 00021420 105 C = C + STORA(M) \* ACRIT(M+I)00021430 110 STORM(I) = CWRITE(6,1901)(STORM(1),I=1,NF2) CALL USUM1(2,UCRIT(IU2),NS,EDOT,STGRA) 00021440 00021450 00021460 DO 120 I=1.NF2 00021470 C = 0.0000021480 DO 115 M=1,NS 00021490 115 C = C + STORA(M)\*A1POST(M,I) 00021500 120 STORM(I) = STORM(I) + CwRITE(6,1901)(STORM(I),I=1,NF2) CALL USUM1(2,UCRIT(IU2),NS,EPOST,STORA) 00021510 00021520 00021530 DO 130 I=1.NF2 00021540 C = 0.0000021550 DC 125 M=1,NS 125 C = C + STORA(M) \* Aldot(M+I)00021560 00021570 130 STORM(I) = STORM(I) + C00021580 WRITE(6,1901)(STORM(I),I=1,NF2) 00021590 IF(ORD.LT.3)GO TO 501

		C	ACO DI PORTION OF DISPLACEMENT DERIVATIVE PSUEDO FORCES TO STORM	00021600
			CALL USUM21(UCRIT(IU3),NS,EDOT,EPOST,STORA)	00021610
			DO 210 I=1,NF2	00021620
			C = 0.00	00021630
			DC 205 M=1,NS	00021640
		205	C = C + STORA(M)*ACRIT(M,I)	00021650
		210	STORM(I) = STORM(I) + C	00021660
		C	COMPUTE ELEMENT FORCES P(I) = VOLUME INTEGRAL OF G(M,I)*STORM(M)	00021670
		501	CALL GFILL(NF,EGEOM(1,1I),G)	00021680
			DO 510 I=1,NF3	00021690
			C = 0.00	00021700
			DD 505 M=1,NF2	00021710
		505	$5 C = C + G(M_*I) + STORM(M)$	00021720
		510	P(I) = C * V	00021730
			WRITE(6,1901)(P(I),I=1,NF3)	00021740
			IF(IPRESS.EQ.0)G0 TO 1000	00021750
		C	ADD NONCONSERVATIVE PRESSURE FORCES TO P	00021760
			VOOT(1) = QOOT(4) - QOOT(1)	00021770
	_		$VCOT(2) \neq QDOT(5) - QDOT(2)$	00021780
	Ģ		VDOT(3) = QDOT(6) - QDOT(3)	00021790
	л С		VCOT(4) = QDOT(7) - QDOT(1)	00021800
			VCOT(5) = QDOT(8) - QDCT(2)	00021810
			VCOT(6) = QDOT(9) - QDOT(3)	00021820
			VPOST(1) = QPOST(4) - QPOST(1)	00021830
			VPOST(2) = QPOST(5) - QPOST(2)	00021840
			VPOST(3) = QPOST(6) - QPOST(3)	00021850
			VPOST(4) = QPOST(7) - QPOST(1)	00021860
			VPOST(5) = QPOST(8) - QPOST(2)	00021870
			VPOST(6) = QPOST(9) - QPOST(3)	00021880
•			CO = (PRO + LCRIT*(PR1-PRO))*PRREF(II)/6.DO	00021890
	,		C1 = 0.00	00021900
			CALL VCROSS(VDOT, VPOST(4), V1)	00021910
			CALL VCROSS(VPOST, VDOT(4), V2)	00021920
			IF(KODE.EQ.2)GO TO 601	00021930
			VCRIT(1) = EGEOM(1,II) + QCRIT(4) - QCRIT(1)	00021940
			VCRIT(2) = QCRIT(5) - QCRIT(2)	00021950
			V(RII(3) = QCRIT(6) - QCRIT(3)	00021960
			VCRIT(4) = EGEOM(3,II) + QCRIT(7) - QCRIT(1)	00021970
			VCRII(5) = EGEOM(2, II) + QCRIT(8) - QCRIT(2)	00021980
			VCRIT(6) = QCRIT(9) - QCRIT(3)	00021990

)

· ·

	C1 = LDOT+(PR1-PR0)+PRREF(II)/6.DO	00022000
	CALL VCROSS(VCRIT.VPOST(4).V3)	00022000
	CALL VCROSS(VPOST.VCRIT(4).V4)	00022020
601	L DO 610 I=1.3	00022020
	C = C0*(V1(T)+V2(T)) + C1*(V3(T)+V4(T))	00022050
	P(I) = P(I) - C	00022040
	P(I+3) = P(I+3) - C	00022050
610	P(1+6) = P(1+6) - C	00022000
C	ACC ELEMENT FORCES P TO SYSTEM FORCES PP	00022080
1000	CALL PFILL(NF,ELNC(1.II).P.PP)	00022000
	RETURN	00022100
	END	00022110
		00022110
	SUBROUTINE POST3(NN,QQCRIT,QQDGT1,C0POS1,00DCT2,00POS2,SPOST1,	00022120
	1LCRIT,LOUT1,LOUT2,PR0,PR1,PP)	00022130
C	FORM 3RD URDER POSTBUCKLING LOAD TERM PP.	00022140
C	NN = TOTAL SYSTEM DOF.	00022150
C	QCCRIT(I) = PREDICTED TOTAL DISPLACEMENT OF COF 1 AT CRITICAL PT.	00022160
C	QQDOT1(1) = CRITICAL 1ST ORDER DISPLACEMENT RATE AT DOF 1.	00022170
C	GQPOSI(I) = ITH EIGENVECTOR VALUE (IST ORDER POSTBUCKLING RATE).	00022180
C	QQDOT2(I) = CRITICAL 2ND ORDER DISPLACEMENT RATE AT COF I.	00022190
C	COPOS2(I) = ITH VALUE OF 2ND ORDER POSTBUCKLING DISPLACEMENT.	00022200
C	SPOST1 = 1ST ORDER FUNCAMENTAL PATH RATE.	00022210
C	LCRIT = PREDICTED LOAD PARAMETER VALUE AT CRITICAL POINT.	00022220
C	LDOT1 $=$ 1ST ORDER LOAD PARAMETER RATE AT CRITICAL POINT.	00022230
C	LUCT2 = 2ND ORDER LOAD PARAMETER RATE AT CRITICAL POINT.	00022240
С	PRO,PR1 = PRESSURE FACTORS AT START OF LOAD STEP, CRITICAL PUINT.	00022250
C	PP(I) = COMPUTED PSUEDO FORCE TERP AT DOF I.	00022260
C	ORD = MAXIMUM TENSOR ORDER TO BE USED FOR DIFFERENCE EXPANSION.	00022270
C	NEL,NS,NF = NUMBER OF ELEMENTS,STRAIN COMPONENTS,DOF PER NODE.	00022280
C	ELNO(J+I) = NODE NUMBERS FOR ELEMENT I.	00022290
C	EGEOM(J,I) = GEOMETRIC PROPERTIES FOR ELEMENT I.	00022300
C	I(I) = THICKNESS OF ELEMENT I.	00022310
C	IPRESS = NONCONSERVATIVE CODE = 0,1 FOR NO PRESSURE, PRESSURE.	00022320
C	PRREF(I) = PRESSURE INTENSITY ON ELEMENT I FOR PRESSURE VECTOR.	00022330
	COMMON/COMORD/ORD/COMNEL/NEL/COMNS/NS/COMNF/NF	00022340
	COMMON/COMEL/ELNO/COMEG/EGEOM/COMT/T	00022350
	COMMON/COMIPR/IPRESS/COMPRR/PRREF	00022360
	INTEGER NN, URD, NEL, NS, NF, ELNO(3, 1), IPRESS	00022370

•

DOUBLE PRECISION CCCRIT(1), CODOT1(1), COPOS1(1), CODOT2(1).	00022380
ICOPOS2(1), SPOST1, LCRIT, LDOT1, LDOT2, PRO, PRI, PP(1), EGEOM(3,1), T(1),	00022390
2PRREF(1)	00022400
INTEGER NF2,NF3, LU2, LU3, I, II,M	00022410
CCUBLE PRECISION V,C,QCRIT(9),QDOT1(9),QPOST1(9),QCOT2(9).	00022420
1QPOST2(9), DCRIT(6), DDOT1(6), DPOST1(6), DDOT2(6), DPOST2(6).	00022430
2ECRIT(3), EDUT1(3), EPOST1(3), EDUT2(3), EPOST2(3), ACRIT(3.6).	00022440
3A100T1(3,6),A1POS1(3,6),A100T2(3,6),A1POS2(3,6),STORA(3),STORB(3)	.00022450
4STORM(6)+G(6,9)+P(9),UCRIT(20)	00022450
DOUBLE PRECISION CO,C1,C2,VCRIT(6),VDOT1(6),VPOST1(6),VDOT2(6),	00022400
1VPOST2(6), V1(3), V2(3), V3(3), V4(3), V5(3), V6(3), V7(3), V8(3), V9(3),	00022490
2V10(3),V11(3),V12(3),V13(3)	00022400
$NF2 = NF \neq 2$	00022500
NF3 = NF*3	00022510
102 = 2 + NS	00022520
IU3 = IU2 + NS + (NS + 1)/2	00022520
DO 5 I=1,NN	00022540
5 PP(1) = 0.00	00022550
DO 1000 II=1,NEL	00022550
<pre>V = .5D0*T(II)*EGECM(1,II)*EGECM(2,II)</pre>	00022570
FORM FUNDAMENTAL CRITICAL QUANTITIES	00022580
CALL QFILL(NF,ELNO(1,II),QQCRIT,QCRIT)	00022590
CALL DFILL(NF,EGEOM(1,II),QCRIT,DCRIT)	00022600
CALL AFILL(NF, CCRIT, ACRIT, 1)	00022610
CALL EFILL(NF, DCRIT, ECRIT)	00022620
FORM STRESS-STRAIN TENSORS	00022630
CALL UFILL(II,ORD,ECRIT,UCRIT)	00022640
FCRM 1ST ORDER FUNCAMENTAL RATE QUANTITIES	00022650
CALL QFILL(NF,ELNC(1,II),QQDOT1,QDOT1)	00022660
CALL DFILL(NF,EGEOM(1,II),QDOT1,CDOT1)	00022670
CALL AFILL(NF,CDOTI,A1DOT1,0)	00022680
00 20 I=1,NS	00022690
C = 0.00	00022700
DC 15 M=1,NF2	00022710
15 C = C + ACRIT(I,M) * DDOT1(M)	00022720
20  EDUT1(I) = C	00022730
FORM 1ST ORDER POSTBUCKLING RATE QUANTITIES	00022740
CALL QFILL(NF,ELNO(1,II),QQPOS1,QPOST1)	00022750
CALL DFILL(NF, EGEOM(1, 11), QPOST1, DPOST1)	00022760
CALL AFILL(NF;DPOST1;A1POS1;0)	00022770

Ð

C-60

0

· •

•

-

С

С

С

		DO 30 I=1,NS	00022780
		C = 0.00	00022790
		DO 25 M=1,NF2	00022800
	25	C = C + ACRIT(I,M) * OPOST1(M)	00022810
	- 30	EPOST1(I) = C	00022820
C		FORM 2ND ORDER FUNDAMENTAL RATE QUANTITIES	00022830
		CALL WFILL(NF,ELNO(1,II),QQDOT2,QDOT2)	00022840
		CALL DFILL(NF,EGEOM(1,II),QD0T2,DDCT2)	00022850
		CALL AFILL(NF, DDOT2, A1DOT2, 0)	00022860
		DC 40 I=1,NS	00022870
		C = 0.00	00022880
		DC 35 M=1,NF2	00022890
	35	C = C + ACRIT(I,M) + DDOT2(M) + AlDOT1(I,M) + DDCT1(M)	00022900
	40	ECOT2(1) = C	00022910
C		FORM 2ND ORDER POSTBUCKLING RATE QUANTITIES	00022920
		CALL QFILL(NF,ELNC(1,II),QQPCS2,QPOST2)	00022930
		CALL DFILL(NF,EGEOM(1,II),QPOST2,DPOST2)	00022940
		CALL AFILL(NF,DPOST2,A1POS2,0)	00022950
		DO 50 I=1,NS	00022960
		C = 0.00	00022970
	•	DO 45 M=1.NF2	00022980
	45	C = C + ACRIT(1,M)*DPOST2(M) + A1POS1(I,M)*(DPOST1(M) + 2.DO*	00022990
	1	LSPOST1*CDOT1(M))	00023000
_	50	EPOST2(1) = C	00023010
C		FORM STORM = DO PORTION OF DISPLACEMENT DERIVATIVE PSUEDO FORCES	00023020
		DO 104 I=1.NS	00023030
		C = 0.00	00023040
		CC 102 M=1,NF2	COO23050
	102	$C = C + Alposl(I_*M)*DDCT2(M)$	00023060
	104	STORB(I) = C	00023070
		CALL USUM1(2,UCRIT(IU2),NS,STORB,STORA)	00023080
		DC 110 I=1,NF2	00023090
		C = 0.00	00023100
		DC 105 M=1,NS	00023110
	105	C = C + STORA(M) * ACRIT(M, I)	00023120
	110	STURM(I) = SPUSI1**2*C	00023130
		CALL USUMITZ + UCRIT(IUZ) + NS + EPOSTI + STORA)	00023140
			00023150
		$U = 0_{\bullet} U U$	00023160
		UU IID M=I,NS	00023170

. .

.

. ......

C-61

-

115 C = C + STURA(M)*ALDOT2(N,I)       0002318         120 STORM(I) = STORM(I) + SPOST1**2*C       0002319         CALL USUM1(2,UCRIT(IU2),NS,EDOT2,STORA)       0002320         DC 130 I=1,NF2       0002321         C = 0.D0       0002322         DO 125 M=1.NS       0002322
120       STURM(1) = STURM(1) + SPUSTI**2*C       0002319         CALL       USUM1(2,UCRIT(IU2),NS,EDOT2,STORA)       0002320         0C       130       I=1,NF2       0002321         C       = 0.D0       0002322       0002322         D0       125       M=1,NS       0002322
CALL OSOMI(2,OCRIT(102),NS,EDOT2,STORA)       0002320         OC 130 I=1,NF2       0002321         C = 0.D0       0002322         D0 125 M=1.NS       0002322
C = 0.00 D0 125 M=1.NS C = 0.00 C = 0.00
DO 125 M=1+NS
UNIZO MELINS
0002323
$125 \ \text{C} = \ \text{C} + \ \text{STURA}(\text{M}) + \text{A1POS1}(\text{M}, \text{I})$ 0002324
130  STURM(I) = STURM(I) + SPOST1 + 2 + C 0002325
0002326
C = 0.00 0002327
0002328
$132 \ U \neq U + Albuti(I,M) \neq DPOST2(M) $ $0002329$
134  SIURB(1) = 0 0002330
CO02331
DU 140 I±1,NF2 0002332
L = 0.00 0002333
DU 135 M=1,NS 0002334
$135 \ C = C + STORA(M) * ACRIT(M, I) 0002335$
140  STURM(I) = STORM(I) + SPOST1*C 0002336
CALL USUMI(2,UCRIT(IU2),NS,EDOT1,STORA) 0002337
0002338
C = 0.00 0002339
UU 145 M=1,NS 0002340
145 C = C + STORA(M) * A1POS2(M, 1) 0002341
150 SIURM(1) = STORM(1) + SPOST1 = 00023420
CALL USUMI(2,UCRIT(IU2),NS,EPOST2,STORA) 0002343
0002344
C = 0.00 00023450
00023460
$155 \ C = C + SIORA(M) * A1DOT1(M, I) 0002347($
160  STURM(1) = STURM(1) + SPOST1*C 0002348(
00023490
C = 0.00 0002350(
00023510 00023510
$102 U = U + AIPUSI(I_{+}M) * DPUST2(M)$ 00023520
104  SIURB(1) = 0 0002353(
CALL USUMI(2,UCRIT(TU2),NS,STORB,STORA) 0002354(
DU 170 I=1,NF2 00023550
C = 0.00 00023560
DU LOD M=1,NS 00023570

•

.

. -

•

.

	165	$C = C + STORA(M) \pm ACRIT(M, 1)$	00033590
	170	C = C + STORM(M) + RC(T) + C	00023500
	1,0	CALL HSUM1/2.UCDIT/FH21.NS.EDUST1.STODAN	00023590
		DO 190 I-1.NE2	00023600
		C = 0.00	00023610
		0. 175 M+1.NS	00023620
	175	$\Gamma = \Gamma + CTODV(M) * VDUCOTW + V$	00023030
	180	C - C + STURALMITALFUSZ(M)II STOPMIIN - STOPMIIN I C	00023640
	100		00023650
		DO 100 T-1.NE2	00023660
			00023670
		G - 0.00 DC 195 M-1 NS	00023680
	196		00023690
	100	C = C + SIUKA(M) + AIPUSI(M, I)	00023700
	190	STURMALL # STURMALL # 6	00023710
r		IFTURDALTADIOU IU DUL APD Di Dontion of disdiacement depinative deurog codser to stody.	00023720
L		ALL DI PORTION OF DISPLACEMENT DERIVATIVE PSUEDU FURLES TO STURM	00023730
		UALE USUM211UUK1111US/#MS#CPUSI1#EUUF2#STURA/	00023740
		C = 0.00	00023750
			00023760
	205	C = C + STODA(M) + ACDIT(M I)	00023770
	205	C = C + SIUKA(M) + ACKII(M)II	00023780
	210		00023790
		UALE USUMITS FUCKITTIUS / FNS FEDUTI + STUKA)	00023800
		102201=1002	00023810
		U = U.UU DD 316 M-1 MC	00023820
	216	UU ZIJ M=I INS	00023830
	213	L = L + STUKA(M)*ALPUSI(M;1) STORM(1) = STORM(1) + SROST)++2+0	00023840
	220	STURMIT = STURMITT + SPUSIL $\mp 2 \mp C$	00023850
		LALL USUEZITUUKITTUSI,NS,EPUSII,EDUTI,STURA)	00023860
		DU 230 I=I,NF2	00023870
		$U = U \cdot U U$	00023880
	225	DU ZZS M=1,NS	00023890
	225	C = C + SIUKA(M) + AIUU(I(M, I))	00023900
	230	SIURM(L) = SIURM(L) + 2.00*SPUSIL**2*C	00023910
		CALL USUM2I(UCRIT(1U3),NS,EDOTI,EPOST2,STORA)	00023920
		UU 240 I=1+NF2	00023930
			00023940
	o = -	UU 235 M=I,NS	00023950
	235	C = C + STURA(M) = ACRIT(M, I)	00023960
	240	STORM(I) = STORM(I) + SPOST1+C	00023970

•

5

C-63

		CALL USUMI(3,UCRIT(IU3),NS,EPOST1,STORA)	00023980
		DO 250 I=1,NF2	00023990
		C = 0.00	00024000
		DC 245 M=1,NS	00024010
	245	$C = C + STORA(M) \neq Aldorl(M, I)$	00024020
	250	STORM(I) = STORM(I) + SPOST1+C	00024020
		CALL USUM21(UCRIT(IU3),NS,EDOT1,EPOST1,STORA)	00024040
		DO 260 I=1,NF2	00024040
		C = 0.00	00024050
		DO 255 M=1.NS	00024000
	255	$C = C + STORA(M) * A1POS1(M \cdot 1)$	00024070
	260	STORM(I) = STORM(I) + 2.00 + SPOST1 + C	00024080
		CALL USUM21(UCRIT(IU3).NS.EPOSTI.EPOST2.STORA)	00024040
		DC 270 I=1,NF2	00024100
		C = 0.00	00024110
		DO 265 M=1.NS	00024120
	265	C = C + STORA(M) * ACRIT(M,T)	00024150
	270	STORM(I) = STORM(I) + C	00024140
		CALL USUMI (3. UCRIT(IU3).NS. EPOSTI.STORAL	00024150
		DO 280 I=1.NF2	00024100
		C = 0.00	00024170
		DC 275 M=1.NS	00024180
	275	$C = C + STORA(M) \neq \Delta POST(M, T)$	00024190
	280	STORM(T) = STORM(T) + C	00024200
C.		COMPUTE FLEMENT EDROES P(T) = VOLUME INTEGRAL OF $C(M, T)$ + STORM(M)	00024210 .
Ť	501	CALL GETTI (NE-EGETMIL, TI)-C)	00024220
		DE 510 1=1-NE3	00024230
		C = 0.00	00024240
		D( 505 M#1 NE2	00024250
	505	$C = C + C(M, 1) \times STOPM(M)$	00024260
	510	$P(1) = C \pm V$	00024270
	210	JELIPPESS ED ALCA TO LOAD	00024280
r		ADD NONCONSERVATIVE DRESSURE EDuced to p	00024290
C		$\frac{1}{1}$	00024300
		VCRIT(2) = 0CRIT(2) + QCRIT(2) + QCRIT(2)	00024310
		VCP[T(2) = OCP[T(2)] = OCP[T(2)]	00024320
		YUNIYIJJ - WUNIYIDJ - WUNIYIJJ WCQITIA) - CCCDWIJ III A GCDITIJI - GCGITIJI	00024330
		$V_{\text{CRITIC}} = \mathbb{E}(\mathbb{E}(\mathbb{D} \setminus \mathbb{D}) + \mathbb{E}(\mathbb{E}(\mathbb{D} \setminus \mathbb{D})) + \mathbb{E}(\mathbb{E}(\mathbb{E}(\mathbb{D} \setminus \mathbb{D})) + \mathbb{E}(\mathbb{E}(\mathbb{E}(\mathbb{D} \setminus \mathbb{D}))) + \mathbb{E}(\mathbb{E}(\mathbb{E}(\mathbb{D} \setminus \mathbb{D})) + \mathbb{E}(\mathbb{E}(\mathbb{E}(\mathbb{D} \setminus \mathbb{D}))) + \mathbb{E}(\mathbb{E}(\mathbb{E}(\mathbb{D} \setminus \mathbb{D}))) + \mathbb{E}(\mathbb{E}(\mathbb{E}(\mathbb{D} \setminus \mathbb{D})) + \mathbb{E}(\mathbb{E}(\mathbb{E}(\mathbb{D} \setminus \mathbb{D}))) + \mathbb{E}(\mathbb{E}(\mathbb{E}(\mathbb{D} \setminus \mathbb{D}))) + \mathbb{E}(\mathbb{E}(\mathbb{E}(\mathbb{D} \setminus \mathbb{D})) + \mathbb{E}(\mathbb{E}(\mathbb{E}(\mathbb{D} \setminus \mathbb{D}))) + \mathbb{E}(\mathbb{E}(\mathbb{E}(\mathbb{D} \setminus \mathbb{D}))) + \mathbb{E}(\mathbb{E}(\mathbb{E}(\mathbb{D} \setminus \mathbb{D})) + \mathbb{E}(\mathbb{E}(\mathbb{E}(\mathbb{D} \setminus \mathbb{D}))) + \mathbb{E}(\mathbb{E}(\mathbb{E}(\mathbb{D} \setminus \mathbb{D}))) + \mathbb{E}(\mathbb{E}(\mathbb{E}(\mathbb{E}(\mathbb{D} \setminus \mathbb{D}))) + \mathbb{E}(\mathbb{E}(\mathbb{E}(\mathbb{E}(\mathbb{E}(\mathbb{D} \setminus \mathbb{D}))) + \mathbb{E}(\mathbb{E}(\mathbb{E}(\mathbb{E}(\mathbb{E}(\mathbb{D} \setminus \mathbb{D}))) + \mathbb{E}(\mathbb{E}(\mathbb{E}(\mathbb{E}(\mathbb{E}(\mathbb{D} \setminus \mathbb{D}))) + \mathbb{E}(\mathbb{E}(\mathbb{E}(\mathbb{E}(\mathbb{E}(\mathbb{E}(\mathbb{D} \setminus \mathbb{D}))) + \mathbb{E}(\mathbb{E}(\mathbb{E}(\mathbb{E}(\mathbb{E}(\mathbb{E}(\mathbb{E}(\mathbb{E}($	00024340
		$\frac{1}{2} = \frac{1}{2} = \frac{1}$	00024350
		$V_{0} = V_{0} = V_{0$	00024360
		$\mathcal{A}_{\text{COLTTIT}} = \mathcal{A}_{\text{COLTTT}} + \mathcal{A}_{\text{COLTTT}}$	00024370

C-64

2

•

$v_{COT1}(2) = 000T1(5) - 000T1(2)$	00024380
$v_{COT1}(3) = 00011(6) - 00011(3)$	00024390
$v_{00}r_{1}(4) = 000r_{1}(7) - 000r_{1}(1)$	00024400
$v_{COT}(5) = 00011(8) - 00011(2)$	00024410
$v_{00}(1/6) = 000(1/6) - 000(1/3)$	00024420
VPOSTI(1) = OPOSTI(4) - OPOSTI(1)	00024430
VPOST1(2) = OPOST1(5) - OPOST1(2)	00024440
VPOSTI(3) = OPOSTI(6) - OPOSTI(3)	00024450
vPOST1(4) = OPOST1(7) - OPOST1(1)	00024460
VPOST1(5) = OPOST1(8) - OPOST1(2)	00024470
VPOST1(6) = OPOST1(9) - OPOST1(3)	00024480
VCOT2(1) = QDOT1(4) - QDOT1(1)	00024490
VDOT2(2) = QDOT1(5) - QDOT1(2)	00024500
VCOT2(3) = QDOT1(6) - QDOT1(3)	00024510
VDOT2(4) = QDOT1(7) - QDOT1(1)	00024520
VD0T2(5) = QD0T1(8) - QD0T1(2)	00024530
VCOT2(6) = QDOT1(9) - QDOT1(3)	00024540
VPOST2(1) = QPOST2(4) - QPOST2(1)	00024550
VPOST2(2) = QPOST2(5) - QPOST2(2)	00024560
VPOST2(3) = QPOST2(6) - QPOST2(3)	00024570
VPOST2(4) = QPOST2(7) - QPOST2(1)	00024580
vpost2(5) = upost2(8) - upost2(2)	00024590
VPOST2(6) = QPOST2(9) - QPOST2(3)	00024600
CO = (PRO + LCRIT*(PRI+PRO))*PRREF(11)/6.DO	00024610
C1 = LDOT1 + (PR1 - PR0) + PRREF(II)/6.00	00024620
C2 = LDOT2*(PR1-PR0)*PRREF(II)/6.DO	00024630
CALL VCROSS(VDOT2,VPOST1(4),V1)	00024640
CALL VCROSS(VPOST1,VDOT2(4),V2)	00024650
CALL VCROSS(VDOT1,VPOST1(4),V3)	00024660
CALL VCROSS(VPOST1+VDOT1(4)+V4)	00024670
CALL VCROSS(VCRIT, VPOST1(4), V5)	00024680
CALL VCROSS(VPOST1,VCRIT(4),V6)	00024690
CALL VCROSS(VDOT1, VPOST2(4), V7)	00024700
CALL VCROSS(VPOST2,VDOT1(4),V8)	00024710
CALL VCROSS(VCRIT, VPOST2(4), V9)	00024720
CALL VCROSS(VPCST2,VCRIT(4),V10)	00024730
CALL VCROSS(VPOSTI,VPOSTI(4),V11)	00024740
CALL VCROSS(VPOST2,VPOST1(4),V12)	00024750
CALL VCROSS(VPOST1, VPOST2(4), V13)	00024760
$00 \ 610 \ I=1,3$	00024770

-

C-65

1

C = C0*(V1(1)+V2(1)+V7(1)+V8(1)+V12(1)+V13(1)) + C1*(2.D0*V3(1))	00024780
<pre>1+2.D0#V4(1)+V9(1)+V10(1)+2.D0#V11(1)) + C2#(V5(1)+V6(1))</pre>	00024790
P(I) = P(I) - C	00024800
P(I+3) = P(I+3) - C	00024810
610 P(I+6) = P(I+6) - C	00024820
C ADD ELEMENT FORCES P TO SYSTEM FORCES PP	00024830
1000 CALL PFILL(NF,ELNO(1,11),P,PP)	00024840
RETURN	00024850
END	00024860

)

.

	SUBROUTINE PRATES(KMAT+P1+Q1+P2+Q2+NN+IPOST+KFD+SCRIT+	00024870
	199CRIT,99D0T1,99D0T2,LCRIT,LD0T1,LD0T2,PR0,PR1,	00024880
	2CQPOS1,LPOST1,CQPOS2,LPOST2,QQPOS3)	00024890
C	COMPUTE POSTBUCKLING RATES FOR LCAD PARAMETER AND DISPLACEMENTS.	00024900
C	KMAT = SYSTEM JACOBIAN STIFFNESS MATRIX.	00024910
C	P1,Q1,P2,Q2 = TEMPORARY STORAGE VECTORS.	00024920
C	NN = TOTAL SYSTEM DOF.	00024930
C	IPOST = DOF OF LARGEST EIGENVECTOR COMPONENT.	00024940
C	<b>KFD(1) = FORCE-DISPLACEMENT-CONSTRAINT SPECIFICATION FOR DOF 1.</b>	00024950
C	SCRIT = PREDICTED CRITICAL VALUE OF FUNDAMENTAL PATH.	00024960
C	QCCRIT(I) = PREDICTED CRITICAL VALUE OF DISPLACEMENT AT DOF 1.	00024970
C	CQDOTI(I), QQDOT2(I) = 1ST, 2ND ORDER CRITICAL FUNDAMENTAL	00024980
C	DISPLACEMENT RATE AT DOF I.	00024990
С	LCRIT = PREDICTED CRITICAL VALUE OF LOAD PARAMETER.	00025000
С	LCOT1,LOOT2 = 1ST,2ND ORDER CRITICAL LOAD PARAMETER RATES.	00025010
C	PRO, PR1 = PRESSURE FACTORS AT START OF LOAD STEP, CRITICAL POINT.	00025020
¢.	LPOST1, LPOST2 = COMPUTED 1ST, 2ND ORDER POSTBUCKLING LOAD RATES.	00025030
C	QUPOSI(I), QQPOS2(I), QQPOS3(I) = 1ST, 2ND, 3RD CRDER POSTBUCKLING	00025040
C	DISPLACEMENT RATES AT DOF I.	00025050
C	NF = DOF PER NODE.	00025060
	COMMON/COMNF/NF	00025070
	INTEGER NN, IPOST, KFD(1)	00025080
	DOUBLE PRECISION KMAT(NN,NN),P1(1),Q1(1),P2(1),Q2(1),SCRIT.	00025090
	1QQCRIT(1),QQDOT1(1),QQDOT2(1),LCRIT,LDOT1,LDCT2,PR0,PR1,	00025100
	2CCPOS1(1), LPOST1, CCPOS2(1), LPOST2, CQPOS3(1)	00025110
	INTEGER I	00025120
	DOUBLE PRECISION CN,CD,SPOST1,SPOST2	00025130
C	SOLVE FOR 1ST ORDER DISPLACEMENTS	00025140
	DO 10 I=1, NN	00025150

C-66

· •	
10  P1(1) = 0.00	00025140
10 + 1(1) = 0.00	00025180
CALL SDIVELYMATING KED D1 CODOS11	00025170
SHEL SULVERNMATENREPTINGPEDTT STOD - NN/ME	00025180
NLU - NN/NF CALL DUTDOLA NOD NE DI CODOCIN	00025190
CALL DUIPHIOPHUDPHEPHIPHUPUSI)	00025200
C SULVE FUR IST URDER LUAUS AND ZNU URDER DISPLACEMENTS	00025210
CALL PUSIZINNAGEURITAEUUTIAEUPUSIAEURITAEUUTIAPROAPRIATA	00025220
CALL PUSIZINN,QULKII,QUPUSI,QUPUSI,LURII,LUUII,PRU,PRI,PZ,Z)	00025230
CALL UUIPQ(6,NUD,NF,PI,P2)	00025240
LN = 0.D0	00025250
CU = 0.00	00025260
DC 110 I=1,NN	00025270
CN = CN + QQPOS1(I) * P2(I)	00025280
$110 CC = CD + GQPOS1(1) \neq P1(1)$	00025290
SPOSTI =5DO*CN/CD	00025300
LPOSTI = LOOTI*SPUSTI	00025310
0C 120 I=1,NN	00025320
IF(KFD(I).GT.0)GC TU 120	00025330
P1(1) = 0.00	00025340
$\rho = P2(I) = 0.00$	00025350
δ 120 CENTINUE	00025360
<pre>CALL SOLVE(KMAT,NN,KFD,PI,QI)</pre>	00025370
CALL SOLVE(KMAT,NN,KFD,P2,Q2)	00025380
CALL OUTPQ(6,NOD,NF,P1,P2)	00025390
CALL OUTPQ(6,NOD,NF,Q1,Q2)	00025400
DO 130 I=1,NN	00025410
130  QQPOS2(I) = -(2.00  SPOST1  Q1(I) +  Q2(I))	00025420
CALL OUTPO(6,NOD,NF,QQDOT2,QQPOS2)	00025430
C SOLVE FOR 2ND ORDER LOADS AND 3RD ORDER DISPLACEMENTS	00025440
CALL POST3(NN,QQCRIT,QCDOT1,QQPOS1,QQDOT2,QQPOS2,SPOST1,	00025450
ILCRIT, LDOT1, LDOT2, PRO, PR1, P2)	00025460
CALL OUTPQ(6,NOD,NF,QQCRIT,P2)	00025470
CALL UUTPQ(6,NOD,NF,QQUOT1,QQPOS1)	00025480
CN = 0.D0	00025490
DO 210 I=1.NN	00025500
210  CN = CN + COPOS1(I) + P2(I)	00025510
SPOST2 = -CN/CD	00025520
LPOST2 = LOOT1*SPOST2 + 100T2*SPOST1**2	00025530
00 220 I=1.NN	00025540
	00025550

	22(	P2(1) = 0.00 Continue	00025560
		CALL SOLVE (KMAT, NN, KFD, P2, Q2)	00025580
		CALL OUTPQI6,NOD,NF,P2,Q2)	00025590
		DO 230 I=1,NN	00025600
	230	) CQPDS3(I) = -3.D0*(SPDST2*Q1(I) + Q2(I))	00025610
		CALL OUTPQ(6,NOD,NF,QQPOS3,QQPOS3)	00025620
		RETURN	00025630
		DEBUG INIT(SPOST1,LPOST1,SPOST2,LPCST2,CN,CD)	00025640
		END	00025650
		DOUBLE PRECISION FUNCTION VONTIVI,V21	00025440
	C	COMPUTE VECTOR DOT PRODUCT VOOT = V1 DOT V2-	00025670
	-	DOUBLE PRECISION VI(1).V2(1)	00025670
		VDOT = V1(1)*V2(1) + V1(2)*V2(2) + V1(3)*V2(3)	00025690
		RETURN	00025700
		END	00025710
~			
ע הע	~	SUBROUTINE VCROSS(V1+V2+VV)	00025720
D	L	$LUMPUTE \ VECTOR \ CROSS \ PRODUCT \ V1 \ X \ V2 = VV.$	00025730
		DUUBLE PRECISION VI(1),V2(1),VV(1)	00025740
		VV(1) = V1(2) = V1(3) = V1(3) = V1(3)	00025750
		VV(2) = V(3) = V(1) =	00025760
		$\Delta A (2) = A T (T) + A C (S) + A T (S) + A C (T)$	00025770
			00025780
			00025790
		DOUBLE PRECISION FUNCTION VLENTH(V)	00025800
	C	COMPUTE VLENTH = LENGTH OF VECTOR V.	00025810
		COUBLE PRECISION V(1)	00025820
		VLENTH = DSQRT(V(1)**2 + V(2)**2 + V(3)**2)	00025830
		RETURN	00025840
		END	00025850
		SUBROUTINE VNORM(V.VV)	00025860
	C	COMPUTE NORMALIZED UNIT VECTOR VV EROM GIVEN VECTOR V.	00025870

С-68 С

.

.

COUBLE PRECISION V(1),VV(1),C	00025880
C = DSQRT(V(1)**2 + V(2)**2 + V(3)**2)	00025890
IF(C.LE.0.DO)C = 1.DO	00025900
VV(1) = V(1)/C	00025910
VV(2) = V(2)/C	00025920
VV(3) = V(3)/C	00025930
RETURN	00025940
END	00025950

1

		000050/0
~	SUDRUUTINE MERGE(NMA);ELNU;RFU;NEL;NN;NF;	00025960
L -	GENERALE AND MERGE ELEMENTAL MAIRICES INTO GLUBAL MAIRIX KMA	00025970
C	ASSUMES ELEMENTS HAVE 3 NOUES, MATRICES MAY BE UNSYMMETRIC.	00025980
Ç	KMAT = GLOBAL (TOTAL SYSTEM) MATRIX, WHICH MAY BE UNSYMMETRI	C. 00025990
C	ELNO(J,I) = NODE NUMBERS FOR ELEMENT I.	00026000
C	<pre>KFD(I) = FORCE-DISPLACEMENT-CONSTRAINT SPECIFICATION FOR DOF</pre>	I. 00026010
C	<pre>KFD(I) = +,- = SPECIFIED FORCE,DISPLACEMENT.</pre>	00026020
C	KFD(I) = +-I INDICATES INDEPENDENT DOF.	00026030
C	<pre>kfd(I) = ++J INDICATES DEPENDENT FREEDOM WHOSE DISPLACEMENT</pre>	IS 00026040
C	CONSTRAINED TO EQUAL DISPLACEMENT AT J.	00026050
C	NEL = NUMBER OF ELEMENTS.	00026060
C	NN = TOTAL SYSTEM DOF = NUMBER OF NODES TIMES NF.	00026070
C	NF = DOF PER NODE.	00026080
С	GENER8 ROUTINE GENERATES ELEMENTAL MATRIX.	00026090
	INTEGER ELNO(3,1), KFD(1), NEL, NN, NF	00026100
	DOUBLE PRECISION KMAT(NN,NN)	00026110
	INTEGER IE, I, J, II, JJ, IO, JO, IIO, JJO, IP, JP, IN, JN, ILOC, IILOC	00026120
	DOUBLE PRECISION K(9,9)	00026130
С	ZERO OUT GLOBAL MATRIX	00026140
	DO 5 II=1,NN	00026150
	DO 5 JJ=1.NN	00026160
	5  KMAT(II.JJ) = 0.00	00026170
	DO 100 IE=1.NEL	00026180
	CALL GENERBLIE.K)	00026190
С	IP.JP ARE PARTITION ROW.COLUMN NUMBERS.	00026200
-	DO 100 TP=1.3	00026210
	$IN = EINO(IP \cdot IE)$	00026220
	$IQ = NF \neq IP - NF$	00026230
	$110 = NE \pm 1N + NE$	00026240
	DO 100 JP=1.3	00026250
		000000290

C-69

.

1

		JN = ELNO(JP, IE)	00026260
		$JO = NF \neq JP - NF$	00026270
		JJO = NF * JN + NF	00026280
	C	MERGE PARTITION INTO GLOBAL MATRIX	00026290
		DC 50 I=1.NF	00026300
		DO 50 J=1.NF	00026310
	50	KMAT(110+1.JJ0+J) = KMAT(110+1.JJ0+J) + K(10+1.J0+J)	00026320
	100	CONTINUE	00026320
	C	ADD CONSTRAINED ROWS AND COLUMNS TO INDEPENDENT ROWS AND COLUMNS	00026340
	-	DO 200 ILOC=1.NN	00020340
		[1] OC = KED(1) OC)	00020330
		$\frac{1}{1} = \frac{1}{1} = \frac{1}$	00026380
		I = I = I = I = I = I = I = I = I = I =	00026370
		DD 100 T-1 NN	00026380
	100	UU LOU IFIANN - 	00026390
	190	$KMAI(I_{F}I_{L}UU) = KMAI(I_{F}I_{U}U) + KMAI(I_{F}I_{U}U)$	00026400
		UU 190 J=1,NN	00026410
	190	<pre>KMAT(IILOC,J) = KMAT(IILOC,J) + KMAT(ILOC,J)</pre>	00026420
	200	CONTINUE	00026430
		RETURN	00026440
ç	•	END	00026450

.

•

.

C-70

<sub>\$2</sub>

	SUBROUTINE DECOMP(KMAT,NN,KFD,IDET,DET)	00026460
C	DECOMPOSE GLOBAL STIFFNESS MATRIX KMAT, AND COMPUTE DETERMINANT.	00026470
C	KMAT = GLOBAL (TOTAL SYSTEM) MATRIX, WHICH MAY BE UNSYMMETRIC.	00026480
C	NN = TOTAL SYSTEM OOF.	00026490
C	<pre>KFD(1) = FORCE-DISPLACEMENT-CONSTRAINT SPECIFICATION FOR DOF I.</pre>	00026500
С	<pre>KFD(I) = +,- = SPECIFIED FORCE,DISPLACEMENT.</pre>	00026510
C	KFD(1) = +-1 INDICATES INDEPENDENT DOF.	00026520`
C	<pre>kfd(I) = +-J INDICATES DEPENDENT FREEDOM WHOSE DISPLACEMENT IS</pre>	00026530
C	CONSTRAINED TO EQUAL DISPLACEMENT AT J.	00026540
Ċ	IDET,DET = SIGN,LOGARITHM (BASE 10) OF DETERMINANT.	00026550
	INTEGER NN,KFD(1),IDET	00026560
	DCUBLE PRECISION KMAT(NN,NN),DET	00026570
	INTEGER NM1, II, JJ1, JJ2, I, J	00026580
	DCUBLE PRECISION D,C	00026590
	IF(NN.EQ.1)RETURN	00026600
	NM1 = NN-1	00026610
	DC 200 II=1,NM1	00026620
	IF(KFD(II).NE.II)GO TO 200	00026630

· .

		0 = 1.00/KMAT(II,II)	00026640
		JJ1 = II+1	00026650
С		DISTRIBUTE IITH ROW OF UPPER TRIANGULAR MATRIX	00026660
		CE 90 J=JJI.NN	00026670
		C = D * K M A T (1 L, J)	00026680
		IF(C.E0.0.D0)G0 T0 90	00026690
C		DISTRIBUTE KMAT(II,J) TO JTH COLUMN	00026700
		DC 50 I=JJ1,J	00026710
	50	KMAT(I,J) = KMAT(I,J) - C * KMAT(I,II)	00026720
	90	CONTINUE	00026730
С		DISTRIBUTE IITH COLUMN OF LOWER TRIANGULAR MATRIX	00026740
		JJ2 = II+2	00026750
		IF(JJ2.GT.NN)GO TO 200	00026760
		DC 190 J=JJ2,NN	00026770
		$C = D \neq K MAT(J, II)$	00026780
		IF(C.EQ.0.D0)G0 T0 190	00026790
С		DISTRIBUTE KMAT(J,II) TO JTH ROW	00026800
		JM1 = J-1	00026810
		DO 150 I=JJ1+JM1	00026820
	150	$KMAT(J,I) = KMAT(J,I) - C \neq KMAT(II,I)$	00026830
	190	CONTINUE	00026840
	200	CONTINUE	00026850
С		COMPUTE DETERMINANT	00026860
		ICET = 1	00026870
		DET = 0.00	00026880
		DO 500 II=1,NN	00026890
		IF(KFD(II).NE.II)GO TO 500	00026900
		$D = KMA_{i}T(II_{i}II)$	00026910
		IF(D.LT.0.DO)IDET = -IDET	00026920
		DET = DET + DLOGIO(DABS(D))	00026930
	500	CONTINUE	00026940
1	201	FORMAT(1H0, JACOBIAN DETERMINANT SIGN,LOGARITHM =1,15,015.7)	00026950
		WRITE(6,1201)IDET,DET	00026960
		RETURN	00026970
		END	00026980

SUBROUTINE SOLVE(KMAT, NN, KFD, P,Q) 00026990 FCRWARD-BACK SUBSTITUTE TO SOLVE FOR FORCES P AND DISPLACEMENTS Q.00027000 KMAT = DECOMPOSED SYSTEM JACOBIAN MATRIX (MAY BE UNSYMMETRIC). 00027010 С C

	С	NN = TOTAL SYSTEM DDE.	00037030
	Ċ	KFO(I) = FORCE-DISPLACEMENT-CONSTRAINT SPECIFICATION FOR DOG I	00027020
	С	$KFD(I) = +_{s} = SPECIFIED FORCE_DISPLACEMENT_$	00027050
	C	KFD(I) = +-I INDICATES INDEPENDENT DDE.	00027040
	C	KFD(I) = +-J INDICATES DEPENDENT EREFDOM WHOSE DISPLACEMENT IS	00027050
	C	CONSTRAINED TO EQUAL DISPLACEMENT AT	00027080
	C	P(I),Q(I) = FORCE,DISPLACEMENT AT COF I.	00027090
	C	EACH DOF HAS EITHER SPECIFIED FORCE OR DISPLACEMENT.	00027080
	С	INPUT $P = SPECIFIED FORCES AND DISPLACEMENTS, D = GARBAGE$	00027090
	C	CUTPUT P = SPECIFIED AND COMPUTED FORCES, $\alpha = \text{Specifies AND}$	00027100
	С	COMPUTED DISPLACEMENTS.	00027110
		INTEGER NN, KFD(1)	00027120
		DOUBLE PRECISION KMAT(NN.NN).P(1).C(1)	00027150
		INTEGER I, J, II, IDUM	00027150
		DOUBLE PRECISION C	00027150
	C	CORRELATE DEPENDENT-INDEPENDENT DUES. ZERO OUT O	00027130
		DO 5 I±1,NN	00027190
		J = IABS(KFD(I))	00027100
		-IF(J.EQ.I)GO TO 5	00027190
Ģ		IF(KFD(J),GT,O)P(J) = P(J) + P(I)	00027200
7		P(1) = 0.00	00027270
N		5 Q(I) = 0.00	00027220
	Ċ	FORWARD SUBSTITUTION	00027240
		DO 100 I=1,NN	00027250
Ċ,		IF(KFD(1).NE.I)GO TO 31	00027260
		$C = \{Q(I) + P(I)\} / KMAT(I,I)$	00027270
		Q(I) = C	00027280
		GC TO 41	00027290
		31 C = P(1)	00027300
		P(I) = -Q(I)	00027310
		C(1) = C	00027320
		41 IF(I-EQ-NN)GO TO 100	00027330
		II = I + I	00027340
		DC 50 J=11,NN	00027350
		50 Q(J) = Q(J) - KMAT(J,I) C	00027360
		100 CONTINUE	00027370
	C	BACKWARD SUBSTITUTION	00027380
		I = NN+1	00027390
		DD 200 IDUM=1,NN	00027400
		$I = I - I \qquad \bigcirc \qquad \qquad$	00027410

)

•

.

C-72

<u>\_\_\_\_</u>

- • ·

		C = 0.00	00027420
		IF(KFD(I).NE.I)GO TO 131	00027430
		IF(I.EQ.NN)GO TO 200	00027440
		II = I + I	00027450
		DC 120 J=I1, NN	00027460
	120	C = C + KMAT(I,J) + Q(J)	00027470
		Q(I) = Q(I) - C/KMAT(I,I)	00027480
		GC TO 200	00027490
	131	DG 140 J=I.NN	00027500
	140	$C = C + KMAT(I_{*}J)*Q(J)$	00027510
		P(I) = P(I) + C	00027520
	200	CCNTINUE	00027530
· C		SET DEPENDENT DISPLACEMENTS, AND ADJUST INDEPENDENT FORCES	00027540
		DC 205 I=1+NN	00027550
		J = IABS(KFD(I))	00027560
		IF(J.EQ.I)GO TO 205	00027570
		Q(1) = Q(J)	00027580
		P(J) = P(J) - P(I)	00027590
	205	CONTINUE	00027600
		RETURN	00027610
		ENC	00027620

START 5 5 6 TORUS MESH-12 (RR#10, R=2, T=.05) MOONEY MATERIAL, C1=80, C2#20 R. VOS 3 2 1 5 1.0-6 1.0-3 1.0-6 2.01 0.1 -10.0 1.0 3.05 1 2 80.0 20.0 1 8.0 0.0 0.0 1 1 2 1 8.0 2.0 0.0 L 3 1 8.06814830.0 .517631257 1 4 1 8.06814832.0 .517631257 1 1 8.26794920.0 1.0 5 1 6 1 8.26794922.0 1.0 Ł 7 1 8.58578640.0 1.41421356 1 8 1 8.58578642.0 1.41421356 1

, .							
	_						
	9	1 9.0 0.0	1.73205081	1			
	10	1 9.0 2.0	1.73205081	1			
	11	1 9.48236870.0	1.93185165	1			
	12	1 9.48236872.0	1.93185165	1			
	1.4		2.0	L 1			
	16		2+0	1			
	16	110.51763130.0	1.02105145	1			
	17		1+7320502103	1			
	18	111.0 2.0	1.73205081	4			
	19	111-41421360-0	1.41421356	1			
	20	111.41421362.0	1.41421356	1			
ę.	21	111.73205080.0	1.0	1			
	22	111.73205082.0	1.0	1			
	23	111.93185170.0	.517631257	1			
	24	111.93185172.0	.517631257	1	,		
	25	112.0 0.0	0.0	1			
	26	112.0 2.0	0.0	1			
· •	,	а Э — Э	1 7				
	2	2 2					
74	2	2 J	3 6				
	4	2 5	63				
	5	2 6	57				
	6		8 6.				
	7	2. 8	7 10		н. Н		
	-8	2 9	10 7				
	9	2 10	9 11				
	10	2 11	12 10				
	11	2 12	11 14				
	12	2 13	14 11				
	13	2 14	13 15				
	14	2 15	16 14				
	15	2 16	15 18				
	16	2 17	18 15				
	17	2 18	17 19				
	18	2 19	20 18				
	19	2 20	19 22				
	20	2 21	22 19				
	51	2 22	21 23			na na marina ang ang ang ang ang ang ang ang ang a	

		22 23 24	2 2 2			23 24 25	24 23 26	22 26 23									
		1 9 17 25 8 16	3 2 2 2 2 2 2 2 2			2 3 11 19 2 10 18	3 2 2 2 2 2 2 2 2 2			25 5 13 21 4 12 20	3 2 2 2 2 2 2 2 2			26 7 15 23 6 14 22	3 2 2 2 2 2 2 2		
• •		24 2 6 10 14 18 22	2 1 1 1 1 1	1 5 9 13 17 21	1 1 1 1	26 6 10 14 18 22	2 3 3 3 3 3	5 9 13 17 21	3 3 3 3 3	4 8 12 16 20 24	1 1 1 1 1	3 7 11 15 19 23	1 1 1 1 1 1	4 8 12 16 20 24	3 3 3 3 3 3	3 7 11 15 19 23	3 3 3 3 3 3
	C-75	26	1 2	25	1												
		$ \begin{array}{c} 1 \\ 1 \\ 1 \\ 0 \\ 5 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 17 \\ 1 \\ 0 \\ 21 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0$				2 1.0 6 1.0 10 1.0 14 1.0 18 1.0 22 1.0				3 1.0 7 1.0 11 1.0 15 1.0 19 1.0 23 1.0			1 1 2 2			L.0 L.0 L.0 L.0 L.0	
			8				0. 1 2 3 4 4 5	1 5 0 0 0 0 5 0					· ·				